UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS

PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE CASOS EN EL CENTRO DE ATENCIÓN TERAPÉUTICO”

CASO: UNIDAD DE ATENCIÓN INTEGRAL A LA FAMILIA DEL GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO

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

Postulante: Dennis Genova Villasanti Yanarico

Tutor Metodológico: Lic. Ing. Helen Fanny Suntura Escobar

Tutor Especialista: Lic. Ing. Yolanda Escobar Mancilla

Tutor Revisor: Lic. Ing. Edwin Mamani Viscarra

EL ALTO – BOLIVIA

2024

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Dennis Genova Villasanti Yanarico, estudiante con C.I. 9886618 LP mediante la presente declaro de manera pública que la propuesta del TRABAJO DE GRADO titulada **"SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE CASOS EN EL CENTRO DE ATENCIÓN TERAPÉUTICO" CASO: UNIDAD DE ATENCIÓN INTEGRAL A LA FAMILIA DEL GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO** es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados.

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

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

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

El Alto, noviembre de 2024

Univ. Dennis Genova Villasanti Yanarico
C.I. 9886618 LP
Denisita222yanarico@gmail.com

DEDICATORIA

Este proyecto de grado está dedicado:

A Dios, por ser el inspirador y darme fuerza para continuar este proceso y obtener uno de los anhelos más deseados.

A mis padres quienes con su amor han sabido guiarme, gracias a ustedes lo he logrado llegar hasta aquí y convertirnos en lo que soy.

Ha todas estas personas que nos ha apoyado y han hecho posible que pueda cumplir mis objetivos.

AGRADECIMIENTO

Gracias a Dios por permitirme tener a mi familia, gracias a ellos por apoyarme en todo en especial en mi etapa universitaria, agradezco a la universidad por formar parte de ella.

Agradezco también a mi tutor metodológico, Lic. Ing. Helen Fanny Suntura Escobar, tutor especialista, Lic. Ing. Yolanda Escobar Mancilla, Lic. Ing. Edwin Mamani Viscarra por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento, así como también tenerme paciencia para guiarme en el desarrollo de mi proyecto.

Mi agradecimiento al Gobierno Autónomo Municipal de El Alto por permitirme que realice mi proyecto.

RESUMEN

En la actualidad las organizaciones han comprendido la importancia de implementar algún sistema como medio de apoyo al acceso rápido a recursos y servicios de educación especial y de salud a los ciudadanos de El Alto, el cual les sirva como soporte para la toma de decisiones en el Centro de Atención Terapéutico del Gobierno Autónomo Municipal de El Alto.

El Centro de Atención Terapéutico", realiza los registros y seguimientos en el caso de las víctimas de forma semiautomatizada (Excel), lo cual no permite obtener informes oportunos. La falta de un sistema de información web para el control de casos en el centro de atención terapéutico caso: unidad de atención integral a la familia del gobierno autónomo municipal de el alto repercute en la perdida de información de los pacientes y búsquedas tediosas en cuanto al seguimiento de cada víctima.

Se desarrollará un sistema de información web para el control de casos en el centro de atención terapéutico (CAT), con el fin de mejorar el control y seguimiento de los casos de las victimas proporcionando información completa y actualizada que faciliten la toma de decisiones en la unidad.

Para desarrollar el Sistema Web se aplicará la metodología UWE, para la fase de diseño y también la UML (Lenguaje de Modelado Unificado). En el proceso de evaluación de calidad de software, también se tomará en cuenta la ISO 25000, la estimación de costo del software se la realizará con COCOMO II basado en puntos de función.

Palabras Claves: Casos, Terapias, Seguimiento, Control. Sesiones

ABSTRACT

Organizations have understood the importance of implementing a system as a means of supporting rapid access to special education and health resources and services for the citizens of El Alto, which serves as support for decision-making in the Therapeutic Care Center of the Municipal Autonomous Government of El Alto.

The Therapeutic Care Center carries out records and follow-ups in the case of victims in a semi-automated manner (Excel), which does not allow timely reports to be obtained. The lack of a web information system for case control in the case therapeutic care center: comprehensive family care unit of the autonomous municipal government of El Alto has an impact on the loss of patient information and tedious searches regarding the follow up on each victim.

A web information system will be developed for case control in the therapeutic care center (CAT), in order to improve the control and monitoring of victims' cases by providing complete and updated information that facilitates decision-making in the unit.

To develop the Web System, the UWE methodology will be applied for the design phase and also the UML (Unified Modeling Language). In the software quality evaluation process, ISO 25000 will also be taken into account, the software cost estimate will be carried out with COCOMO II based on function points.

Keywords: Cases, Therapies, Follow-Up, Control. Sessions

CONTENIDO

1 CAPÍTULO I - MARCO PRELIMINAR 2

1.1 INTRODUCCIÓN 2

1.2 ANTECEDENTES 3

1.2.1 Antecedentes Institucionales 3

1.2.1.1 Visión 3

1.2.1.2 Misión 3

1.2.1.3 Organigrama 4

1.2.2 Antecedentes Afines al Proyecto de Grado 4

1.2.2.1 Antecedentes Internacionales 5

1.2.2.2 Antecedentes Nacionales 5

1.2.2.3 Antecedentes Locales 6

1.3 PLANTEAMIENTO DEL PROBLEMA 7

1.3.1 Problema Principal 7

1.3.2 Problemas Secundarios 7

1.3.3 Formulación del Problema 8

1.4 OBJETIVOS 8

1.4.1 Objetivo General 8

1.4.2 Objetivos Específicos 8

1.5 JUSTIFICACIÓN 8

1.5.1 Justificación Técnica 8

1.5.2 Justificación Económica 9

1.5.3 Justificación Legal 9

1.5.4 Justificación Social 9

1.6 METODOLOGÍA DE DESARROLLO 10

1.6.1 Métricas de Calidad 11
1.6.2 Estimación de Costos de Software 12
1.6.3 Seguridad ISO/IEC 25001 12
1.6.4 Pruebas de Software 13
1.6.4.1 Pruebas de Caja Blanca 13
1.6.4.2 Pruebas de Caja Negra 13
1.6.4.3 Pruebas de Estrés 14
1.6.4.4 Pruebas de Accesibilidad 14
1.6.4.5 Pruebas de Unitarias 15
1.7 HERRAMIENTAS 16
1.8 ALCANCES Y LIMITES 18
1.8.1 Alcances 18
1.8.2 Limites 19
1.9 APORTES 20
2 CAPÍTULO II - MARCO TEÓRICO 22
2.1 INTRODUCCIÓN 22
2.2 DERECHOS HUMANOS 22
2.2.1.1 Violación de los Derechos Humanos 24
2.2.1.2 Violencia Contra las Mujeres 25
2.2.1.3 Violencia Contra los Niños 27
2.2.2 Control Terapéutico 28
2.2.2.1 Tipos de Control Terapéutico 29
2.3 SISTEMA 30
2.4 SISTEMAS DE INFORMACIÓN 30

2.5 SISTEMA WEB 31

2.5.1 Web 2.0 32

2.5.2 Diferencias. 32

2.6 CONTROL Y SEGUIMIENTO 33

2.6.1 Importancia del Seguimiento 33

2.6.2 Tipos de Control 33

2.7 METODOLOGÍA BASADA EN UML(UWE) 35

2.7.1 Metodología UWE (UML-Based Web Engineering) 35

2.7.2 Características de UWE 35

2.7.3 Modelos de la Metodología UWE 36

2.7.3.1 Modelo de Requisitos “Casos de Uso” 36

2.7.3.2 Modelo de Contenido “Conceptual” 37

2.7.3.3 Modelo de Navegación 38

2.7.3.4 Modelo de Presentación 38

2.7.3.5 Modelo de Proceso 39

2.7.4 Fases de la Metodología UWE 41

2.7.4.1 Captura, Análisis y Especificación de Requisitos 41

2.7.4.2 Diseño del Sistema 42

2.7.4.3 Codificación del Software 42

2.7.4.4 Pruebas 42

2.7.4.5 La Instalación o Fase de Implementación 42

2.7.4.6 El Mantenimiento: 42

2.8 ARQUITECTURA DE SOFTWARE (MVC) 43

2.8.1 Patrones de Arquitectura de software 44

2.8.2 Patrón modelo vista controlador (MVC). 45

2.9 HERRAMIENTAS 47

2.9.1 PHP (Hipertext Pre-processor) 47

2.9.2 Apache 48

2.9.3 Gestor de Base de Datos 48

2.9.4 Bootstrap 49

2.9.5 IDE (Visual Studio Code) 49

2.9.6 Framework CodeIgniter 50

2.10 MÉTRICAS DE CALIDAD DE SOFTWARE 52

2.10.1 ISO/IEC 25010 54

2.10.1.1 Funcionalidad 55

2.10.1.2 Confiabilidad 58

2.10.1.3 Usabilidad 59

2.10.1.4 Eficiencia 60

2.10.1.5 Mantenibilidad 60

2.10.1.6 Portabilidad 61

2.11 ESTIMACIÓN DE COSTOS 62

2.11.1 Cocomo II 62

2.11.1.1 Ecuaciones del Modelo COCOMO 63

2.11.1.2 Modelos de COCOMO II 64

2.11.1.3 Ventajas de COCOMO II 66

2.12 SEGURIDAD 67

2.12.1 ISO 27001 68

2.12.2 Seguridad Física 69

2.12.3 Seguridad Lógica 69

2.12.4 Seguridad de la Base de Datos 70

2.12.5 Seguridad del Framework CodeIgniter 70

2.13 PRUEBAS DE SOFTWARE 71

2.13.1 Pruebas de Caja Blanca 71

2.13.2 Pruebas de Caja Negra 72

2.13.3 Pruebas de Estrés 72

2.13.4 Pruebas de Accesibilidad 72

2.13.5 Pruebas de Unitarias 76

3 CAPÍTULO III - MARCO APLICATIVO 79

3.1 INTRODUCCIÓN 79

3.2 DESARROLLO DE LA METODOLOGÍA 79

3.2.1 Requerimientos Funcionales 80

3.2.2 Requerimientos No Funcionales 81

3.2.3 Descripción de Actores 82

3.2.4 Diagramas de Caso de Uso General 83

3.2.5 Diagrama de Caso de Uso Específico 84

3.2.6 Diagrama de Caso de Uso: Gestión de Roles y Permisos 85

3.2.7 Diagrama de Caso de Uso: Gestión historial de documentos 92

3.2.8 Modelo Conceptual 91

3.2.9 Modelo de Presentación 92

3.2.10 Modelo de Proceso 98

4 CAPÍTULO IV - CALIDAD, SEGURIDAD Y COSTO 109

4.1 INTRODUCCIÓN 109

4.2 MÉTRICAS DE CALIDAD - ESTÁNDAR ISO/IEC 25000 109

4.2.1 Funcionalidad 109

4.2.1 Confiabilidad 114

4.2.2 Usabilidad 116
4.2.3 Eficiencia 118
4.2.4 Mantenibilidad 120
4.2.5 Portabilidad 122
4.3 ESTIMACIÓN DEL COSTO 123
4.4 COCOMO II 124
4.5 PRUEBAS DEL SISTEMA 126
4.5.1 Pruebas de Caja Blanca. 126
4.5.2 Pruebas de Caja Negra. 128
4.5.2.1 Prueba de Caja Negra – Inicio de Sesión 128
4.5.3 Pruebas de Estrés 130
4.5.4 Pruebas de Accesibilidad 130
4.5.1 Pruebas de Unitarias 132
4.6 SEGURIDAD 133
4.6.1 Identificación y Autenticación 133
4.6.2 Encriptación 134
4.6.3 Seguridad Física 134
4.6.4 Seguridad Organizativa 134
5 CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES 136
5.1 CONCLUSIONES 136
5.2 RECOMENDACIONES 137
BIBLIOGRAFÍA 138
ANEXOS 143

ÍNDICE DE FIGURAS

Figura 1 Organigrama Unidad Centro de Atención Terapéutico 4

Figura 2 Sistema Universal de Protección de DDHH 23

Figura 3 Gráfico Derechos Humanos 25

Figura 4 Porcentaje por Número de Casos 26

Figura 5 Tipos de Violencia 27

Figura 6 Denuncia por Violencia 28

Figura 7 Control Psicólogo 29

Figura 8 *Niveles del Sistema de Información* 31

Figura 9 Modelo de Casos de Uso 37

Figura 10 Modelo de Contenido 38

Figura 11 Modelo de Navegación 39

Figura 12 Modelo de Presentación 41

Figura 13 Patrones de Arquitectura 44

Figura 14 Patrón Modelo Vista Controlador (MVC) 46

Figura 15 Factores que Afectan la Calidad de Software 53

Figura 16 ISO/IEC 25010 55

Figura 17 Funciones del Axe DevTools 73

Figura 18 Funciones de Colour Contrast Analyser 74

Figura 19 Prueba de Accesibilidad WAVE 76

Figura 20 Funciones de la Pruebas Unitarias 77

Figura 21 Caso de Uso General del Sistema 83

Figura 22 Gestión de Usuarios 84

Figura 23 Gestión de Roles y Permisos 85

Figura 24 Gestión de Tipologías 86

Figura 25 Gestión de Institución 87

Figura 26 Gestión de Asignar Casos 87

Figura 27 Gestión de Terapeuta 88

Figura 28 Gestión Pacientes 89

Figura 29 Gestión de Control de Pacientes 90

Figura 30 Gestión de Sesiones 91

Figura 31 Gestión Historial de Documentos 92

Figura 32 Gestión de Seguimiento 93

Figura 33 Gestión Reportes 94

Figura 34 Gestión de Datos Estadísticos 90

Figura 35 Diseño de la Base de Datos Relacional 91

Figura 36 Modelo de Presentación – Login 92

Figura 37 Modelo de Presentación - Administrador 92

Figura 38 Modelo de Presentación - Usuarios 93

Figura 39 Modelo de Presentación - Roles y Permisos 93

Figura 40 Modelo de Presentación – Registro de Tipología 94

Figura 41 Modelo de Presentación – Registro de Institución 94

Figura 42 Modelo de Presentación – Registro de Terapeuta 94

Figura 43 Modelo de Presentación – Registro de Nuevo Caso 95

Figura 44 Modelo de Presentación – Registro de Asignar Horario 95

Figura 45 Modelo de Presentación – Registro de Casos 96

Figura 46 Modelo de Presentación – Gestión Casos 96

Figura 47 Modelo de Presentación – Registro de Pacientes 97

Figura 48 Seguimiento de Casos 97

Figura 49 Control de Terapeutas 97

Figura 50 Página Principal 98

Figura 51 Datos del Centro 98

Figura 52 Datos de la Empresa 99

Figura 53 Inicio de Sesión del Sistema 99

Figura 54 Ingreso al Sistema - Administrador 100

Figura 55 Gestión de Usuarios 100

Figura 56 Gestión de Roles 100

Figura 57 Gestión de Tipologías 101

Figura 58 Gestión de Institución 101

Figura 59 Registro de Casos 102

Figura 60 Administración de Registro de Pacientes 102

Figura 61 Administración de Registro de Sesiones Terapeuta 103

Figura 62 Registro de Nuevo Casos 103

Figura 63 Administración de Asignar Horario 104

Figura 64 Control de Casos 104

Figura 65 Control de Seguimiento del Paciente 104

Figura 66 Control de Sesión del Paciente 105

Figura 67 Gestión de Documentos 105

Figura 68 Administración de Datos estadísticos 105

Figura 69 Cantidad de Casos por Mes 106

Figura 70 Cantidad de Tipologías por Mes 106

Figura 71 Búsqueda de Estado de Pacientes 106

Figura 72 Derivación de Casos 107

Figura 73 Datos Estadísticos de Tipologías 107

Figura 74 Técnica de Camino Básico 126

Figura 75 Prueba de Caja Negra Inicio de Sesión 128

Figura 76 Caja Negra Ingreso al Sistema 129

Figura 77 Prueba de Estrés..... 130

Figura 78 Prueba de Accesibilidad..... 131

Figura 79 Resultado del Software..... 131

Figura 80 Prueba Unitaria 132

Figura 81 Identificación y Autenticación..... 133

Figura 82 Seguridad de Password..... 134

ÍNDICE DE TABLAS

Tabla 1 Índices de Usabilidad 56

Tabla 2 Tabla de Valores de Modelo Básico 65

Tabla 3 Herramientas para la Obtención de Requerimientos 79

Tabla 4 Requerimientos Funcionales 80

Tabla 5 Requerimientos No Funcionales 81

Tabla 6 Descripción de Actores 82

Tabla 7 Detalle del Caso de Uso – Gestión de Usuarios. 84

Tabla 8 Detalle del Caso de Uso – Gestión de Roles y Permisos 85

Tabla 9 Detalle del Casos de Uso – Gestión de Tipologías 86

Tabla 10 Detalle de Casos de Uso – Gestión Institución 87

Tabla 11 Detalle del Casos de Uso – Gestión de Asignar Casos 87

Tabla 12 Detalle del Casos de Uso – Gestión de Terapeuta 88

Tabla 13 Detalle de Casos de Uso - Gestión de Pacientes 89

Tabla 14 Detalle de Casos de Uso - Gestión Control 90

Tabla 15 Detalle del Casos de Uso – Gestión de Sesiones 91

Tabla 16 Detalle del Casos de Uso – Gestión Historial de Documentos 92

Tabla 17 Detalle de Casos de Uso - Gestión de Seguimiento 93

Tabla 18 Detalle de Casos de Uso - Gestión de Reportes 94

Tabla 19 Detalle de Casos de Uso - Gestión de Datos Estadísticos 90

Tabla 20 Factores de Ponderación 110

Tabla 21 Valores de Ajuste de Complejidad 111

Tabla 22 Indicadores de Facilidades de Uso 117

Tabla 23 Usabilidad del Sistema 117

Tabla 24 Escala de Valores de Eficiencia 119

Tabla 25 Evaluación de Eficiencia 119

Tabla 26 Indicadores de Mantenibilidad 121

Tabla 27 Resultados Obtenidos - Norma ISO 25000 123

Tabla 28 Prueba de Caja Negra Ingreso el Sistema 129

ÍNDICE DE ECUACIONES

(1) Fórmula de Ajuste 57

(2) Punto Función Máxima 57

(3) Formula de la Funcionalidad 58

(4) Probabilidad de Fallas 59

(5) Probabilidad de Trabajo Sin Fallas 59

(6) Función de la Confiabilidad del Sistema MT 59

(7) Función para Determinar la Usabilidad MT 60

(8) Fórmula de la Eficiencia MT 60

(9) Cálculo del Nivel de Madurez del Sistema 61

(10) Ecuación de Esfuerzo 63

(11) Ecuación Tiempo de Esfuerzo 63

(12) Ecuación Número de Personas 64

(13) Ecuación Personas Por Mes Según Modelo 65

(14) Ecuación Tiempo de Proyecto Según Modelo 65

(15) Ecuación Costo Personal 66

(16) Ecuación de Costo Total del Proyecto 66

(17) Ecuación de Ajuste 113

(18) Función de la Confiabilidad del Sistema 114

(19) Función para Determinar la Usabilidad 116

(20) Fórmula Eficiencia 120

(21) Ecuación para el Índice de Madurez del Sistema 120

(22) Fórmula Portabilidad 122

LISTADO DE ABREVIATURAS

FLDC: Factor de conversión de líneas de código entregado.

TDES: Tiempo empleado en el desarrollo.

PF: Puntos de función.

PHP: Preprocesador de Hipertexto.

UWE: Ingeniería Web basada en UML.

UML: Lenguaje de Modelado Unificado.

HTML: Lenguaje de Marcado de Hipertexto.

HTTP: Protocolo de Transferencia de Hipertexto.

KLDC: Kilo Líneas de Código.

IDE: Entorno de Desarrollo Integrado

MVC: Modelo Vista Controlador

ISO: Organización Internacional de Normalización

IEC: Comisión Electrotécnica Internacional

RNF: Requerimientos No Funcionales

FAE: Factor de Ajuste de Esfuerzo

GAMEA: Gobierno Autónomo Municipal de El Alto

PF: Punto de Función

CT: Conteo Total

CAT: Centro de Atención Terapéutico

CAPÍTULO I

MARCO PRELIMINAR

INGENIERÍA DE SISTEMAS

UNIVERSIDAD PUBLICA DE EL ALTO

1 CAPÍTULO I - MARCO PRELIMINAR

1.1 INTRODUCCIÓN

En la actualidad las organizaciones han comprendido la importancia de implementar algún sistema como medio de apoyo al acceso rápido a recursos y servicios de educación especial y de salud a los ciudadanos de El Alto, el cual les sirva como soporte para la toma de decisiones en el Centro de Atención Terapéutico del Gobierno Autónomo Municipal de El Alto.

La inteligencia artificial (IA) es un campo de la informática que se enfoca en crear sistemas que puedan realizar tareas que normalmente requieren inteligencia humana,

La Unidad de Centro de Atención Terapéutico se encarga de mantener y promover los casos de las víctimas de la Ciudad de El Alto con actividades en cuanto a la atención a niñas, niños, adolescentes y mujeres víctimas de violencia quienes reciben terapia para buscar su rehabilitación. El tratamiento se realiza es individual, familiar y tratamiento grupal, se hacen cargo de todo el trabajo psicosocial para salud mental de la víctima.

El presente proyecto de grado propone solucionar el acceso a la documentación por direcciones, como en este caso los recursos y servicios que ofrece la ciudad de El Alto está encargada de la Unidad de Atención Integral a la Familia y la inexistencia de un portal limita la posibilidad de beneficiar a la población con estas necesidades en la ciudad de El Alto y con el presente Proyecto de Grado, propone solucionar las debilidades en la inexistencia de información y desorganización, ya que solo cuenta con planillas improvisadas que son llenadas de manera manual ya que las informaciones generadas en esas planillas no son al 100% y esto repercute en la toma de decisiones.

Para desarrollar el Sistema Web se aplicará la metodología UWE, para la fase de diseño y también la UML (Lenguaje de Modelado Unificado). En el proceso de evaluación de

calidad de software, también se tomará en cuenta la ISO 25000, la estimación de costo del software se la realizará con COCOMO II basado en puntos de función.

1.2 ANTECEDENTES

1.2.1 Antecedentes Institucionales

En las ciudades de El Alto, La Paz y provincias del Departamento, Bolivia El Alto es reconocida, porque se ha especializado en la atención terapéutica de niñas y adolescentes mujeres que han sido víctimas de violencia física, psicológica y sexual, como también de adolescentes mujeres en situación de marginación con problemas de adicciones y de conducta, a quienes se restablecen sus derechos y con el desarrollo de un programa de apoyo integral - terapéutico se logra su inserción o reinserción social y laboral.

1.2.1.1 *Visión*

El Centro de Atención Terapéutica “C.A.T.” tiene la misión de brindar atención de calidad a la familia, en su conjunto para favorecer la salud mental y social por medio de compromiso y profesionalismo de quienes lo constituyen (manual de Usuarios, 2020)

1.2.1.2 *Misión*

Consolidarnos como una institución de tercer nivel que brinda atención terapéutica sistémica de calidad y que constituya a mejorar el bienestar de las personas, con una atención basada en el compromiso hacia nuestros pacientes a través del trabajo en equipo y actualización profesional de (Psicólogos Terapeutas y Trabajo Social). (unidad centro de atención terapéutico).

1.2.1.3 *Organigrama*

Figura 1

Organigrama Unidad Centro de Atención Terapéutico

Nota. Organigrama, GAMEA, 2014, Plan Estratégico de Desarrollo Institucional

1.2.2 Antecedentes Afines al Proyecto de Grado

Haciendo referencia a trabajos realizados a nivel internacional, nacional y local se puede citar:

1.2.2.1 *Antecedentes Internacionales*

- Según Rivero (2022), ”Diseño e implementación de un sistema web para mejorar el registro de casos de víctimas de violencia familiar y/o sexual en lima cercado”. Objetivo Diseñar e implementar un sistema web para mejorar el registro de casos de víctimas de violencia familiar y/o sexual. Se utilizó las herramientas cómo Lenguaje JAVA, HTML y un gestor de DB MySQL para su elaboración. Mediante pruebas de aceptación, generadas con una muestra de diez colaboradores, se recaudaron datos para evaluar la eficiencia del SW para la mejora el proceso de registro de casos (almacenamiento de casos, seguimiento de los casos y atención de caso.

- De acuerdo a Torres (2022), “Sistema de información para el seguimiento de casos de la comisaría de familia del municipio de chiquinquirá (comsaf).” Desarrollar un sistema de información que permita el registro y control de los procesos relacionados a violencias intrafamiliar, Procesos administrativos de restablecimientos de derechos (PARD) y alimentos de adulto mayor llevados a cabo en la Comisaría de Familia del municipio de Chiquinquirá. Herramientas del Sistema de Información Full Stack Tecnológico MEVN. Este tipo de desarrollo Full-Stack está enfocado en JavaScript, es decir, es el conjunto de tecnologías necesarias para el desarrollo de todas las capas de una aplicación web con JavaScript. Está compuesto por cuatro de las más importantes tecnologías de la industria: MongoDB, Express, Vue y Node.js.

1.2.2.2 Antecedentes Nacionales

A continuación, se describe:

- De acuerdo a Vaca (2022), “Capacitación en políticas multidisciplinarias de intervención con enfoque social para las familias de niñas, niños víctimas de violencia sexual en el Slim del municipio de cercado Cochabamba”. El proyecto tiene el objetivo general de Reducir

los casos de desestimación de familias afectadas con niños y niñas víctimas de violencia sexual en SLIM del municipio de Cercado. El proyecto incluye estrategias de trabajo innovadoras que permiten, por un lado, fortalecer la aplicación del sistema de protocolo de atención a las familias con niño y niñas víctimas de violencia sexual por otro lado, contribuir a mejorar la calidad de información hacia los tutores de las víctimas.

1.2.2.3 Antecedentes Locales

A continuación, se describe:

- Según Serrano (2022), “Programa contra el maltrato emocional en niños y niñas en la defensoría de la niñez y adolescencia de la zona san Antonio alto de la ciudad de La Paz”. Elaborar un programa informativo y preventivo contra el maltrato emocional para identificar de qué manera los niños y niñas que acuden a la Defensoría de la Niñez y Adolescencia afrontan el maltrato emocional y a su vez conocer la incidencia del mismo. Se ejecutaron las actividades durante seis meses de tiempo completo, ocho horas diarias, se estableció que el apoyo en el área de psicología en la Defensoría de la Niñez y Adolescencia se realizó todos los días con excepción de los días agendados para la realización del “Programa Preventivo e Informativo Contra el Maltrato Emocional”, todo lo referente estuvo enmarcado en la agenda de actividades del área de psicología.

- De acuerdo Maceda (2021),” Plataforma virtual para prevención de suicidio e intervención en momentos de crisis”. Desarrollar una plataforma virtual que brinde un espacio de apoyo psicológico y emocional online para la intervención en momentos de crisis y la prevención del suicidio. utilizando la tecnología JWT (Jason Web Token) que en esta oportunidad se guardó en localStorage del usuario permitiendo la autenticación y autorización en la base de datos se utilizó la librería Bcrypt para el resguardo de las contraseñas de los usuarios para que a través de Node.JS se pueda implementar la comprobación en el inicio de sesión de forma secreta, y que, si alguien logra acceder a la base de datos, las contraseñas se

encuentren resguardadas correctamente.

1.3 PLANTEAMIENTO DEL PROBLEMA

Actualmente el centro CAT en la ciudad de El Alto al mes generan 50 casos los cuales no todos son atendidos eficientemente debido realiza los registros y seguimientos en el caso de las víctimas de forma semiautomatizada (Excel), lo cual repercute en la pérdida de documentos, archivos, información y datos estadísticos es debido a que en el centro no realizan un adecuado registro de los casos que derivan en el centro. La falta de un sistema de información web para el control de casos en el centro de atención terapéutico caso: unidad de atención integral a la familia del gobierno autónomo municipal de el alto repercute en la perdida de información de los pacientes y búsquedas tediosas en cuanto al seguimiento de cada víctima.

1.3.1 Problema Principal

A causa de un ineficiente registro de casos en el centro de atención terapéutico caso: unidad de atención integral a la familia del gobierno autónomo municipal de el alto repercute en la perdida de información de los pacientes, victimas, personas con disfunciones familiares lo cual genera búsquedas tediosas en cuanto al seguimiento de casos de cada víctima.

1.3.2 Problemas Secundarios

Se puede mencionar los siguientes problemas específicos:

- Pérdidas en los datos del diagnóstico de cada víctima debido a la deficiente seguridad en los registros de los casos.
- Falta de coherencia en el llenado de registros de los casos de las víctimas.
- Distorsión en la percepción que los terapeutas tienen respecto a los pacientes y sus tratamientos que deberían de realizar.
- Demoras en el llenado de datos de las victimas
- Dificultades en la atención y seguimiento de los casos de las víctimas.

1.3.3 Formulación del Problema

¿De qué manera el sistema en el Control de Información de casos de registros de las víctimas en el “Centro de Atención Terapéutico” (CAT) diagnosticara a cada víctima y reflejaría los beneficios en la unidad?

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un sistema de información web para el control de casos en el centro de atención terapéutico (CAT), con el fin de mejorar el control y seguimiento de los casos de las victimas proporcionando información completa y actualizada que faciliten la toma de decisiones en la unidad.

1.4.2 Objetivos Específicos

- Desarrollar una base de datos que se acomode a los requerimientos de la unidad, mejorando el diagnóstico de cada víctima.
- Implementar un módulo de control de casos de las víctimas en la unidad, registros de cada víctima.
- Realizar reportes y datos estadísticos de la información de los casos y sus tratamientos de las víctimas.
- Aplicar un módulo de atención y seguimiento de los casos de las víctimas.
- Integrar una Interfaz de usuario amigable y bastante intuitivo con el fin de facilitar el manejo del sistema a los usuarios.

1.5 JUSTIFICACIÓN

1.5.1 Justificación Técnica

El proyecto se justifica técnicamente para el registro y seguimiento de casos de las víctimas y para su desarrollo se utilizará las siguientes herramientas y detalles tecnológicos a utilizar, como ser el servidor central de la Institución el cual proporcionara un dominio para la

implementación y para el desarrollo de la plataforma como requerimiento de hardware será cualquier dispositivo con acceso a internet[1] y como software se realizará el uso del lenguaje de programación de CodeIgniter haciendo uso de sus frameworks.

1.5.2 Justificación Económica

El presente proyecto sistema de información para el control de casos en el C.A.T. se justifica analizando los ahorros que se generan a través de la implementación del sistema de información en el caso de las víctimas, con el cual se contara con una información útil, organizado y confiable de todos los casos el uso de software libre[2], también reciben ayuda de ONGs y de la Alcaldía.

1.5.3 Justificación Legal

(CUMPLIMIENTO OBLIGATORIO DE LA NORMATIVA MUNICIPAL) de la Ley N° 482 de Gobiernos Autónomos Municipales, prescribe: “La normativa legal del Gobierno Autónomo Municipal, en su jurisdicción, emitida en el marco de sus facultades y competencias, tiene carácter obligatorio para toda persona natural o colectiva, pública o privada, nacional o extranjera; así como el pago de Tributos Municipales y el cuidado de los bienes públicos”.

1.5.4 Justificación Social

La implementación del sistema de información de casos de las víctimas en el Centro de Atención Terapéutico brindará servicios a la población de la ciudad de El Alto y será beneficiado a:

---

[1] ***Internet*** *es un espacio único de información en el que se organiza un entorno de comunicación entre individuos y grupos de personas es el medio de comunicación de masas más eficaz, ya que permite la comunicación entre muchas personas sin restricciones.*
Software libre es el software que respeta la libertad de los usuarios y la comunidad.

- Jefe de unidad, como administrar todos los casos.
- A los terapeutas como en hacer seguimiento de los pacientes.
- Los usuarios como en la mejora de su salud psicológica.

También la misma unidad, todos los procesos manuales, brindando datos actualizados para el beneficio de las actividades diarias que realizan en el área, proporcionando información rápida y oportuna a los administradores del centro, así mejorando el tiempo de atención dentro de la institución.

1.6 METODOLOGÍA DE DESARROLLO

La metodología fue utilizar un lenguaje común o por lo menos definir un metamodelo basado en el mapeo a lo largo de las diferentes etapas. En esa época UML prometía convertirse en un estándar para el modelamiento de sistemas. Por este motivo, UWE se adhirió a UML y no a otra técnica de modelado. UWE se ha adaptado a las nuevas características de los sistemas web como transacciones, personalizaciones y aplicaciones asíncronas, y por otro lado ha evolucionado para incorporar técnicas de ingeniería de software como el modelamiento orientado a aspectos y nuevos lenguajes de transformación para mejorar la calidad del diseño. (Narváez, 2020, p.50).

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML. Las fases de la metodología UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas (Evangellys, 2013, p. 5).

- Captura, análisis y especificación de requisitos
- Diseño del sistema
- Codificación del software
- Pruebas
- La Instalación o Fase de Implementación
- El Mantenimiento

1.6.1 Métricas de Calidad

Para evaluar este proyecto se aplicará las métricas de calidad según el estándar ISO/IEC 25010 El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y su característica.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las nueve características de calidad que se muestran en la siguiente figura: (Hidalgo, 2020)

- Adecuación funcional
- Eficiencia de desempeño
- Compatibilidad
- Capacidad de interacción

- Fiabilidad
- Seguridad
- Mantenibilidad
- Flexibilidad
- Protección

1.6.2 Estimación de Costos de Software

Para estimar el costo se utilizará la herramienta COCOMO II el cual permitirá realizar estimaciones en función del tamaño del software, y en un conjunto de factores de costos y de escala. COCOMO II posee tres modelos que se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información estos tres modelos son: (Machicado, 2020, p. 637).

- Modelo de composición de aplicación
- Modelo de fase de diseño previo
- Modelo de fase posterior a la arquitectura

1.6.3 Seguridad ISO/IEC 25001

Para el presente trabajo de grado se utilizará la norma 25001, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del software.

La información es un activo muy valioso que puede impulsar a destruir una empresa. Si se gestiona de forma adecuada, le permite trabajar con confianza capacidad de protección

de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. (Miss & Baena, 2019, p. 37)

Por lo tanto, el sistema para la seguridad de la información se basará en dicha norma.

1.6.4 Pruebas de Software

Existe varios métodos modernos para realizar pruebas a distintos softwares que se realizan, la prueba de caja blanca, pruebas de caja negra, pruebas de carga, entre otros. Se hará uso de los siguientes:

1.6.4.1 *Pruebas de Caja Blanca*

La prueba de caja blanca es una forma de prueba de aplicaciones que proporciona al evaluador un conocimiento completo de la aplicación que se está probando, incluido el acceso al código fuente y a los documentos de diseño. Esta visibilidad en profundidad hace posible que las pruebas de caja blanca identifiquen problemas que son invisibles para las pruebas de caja gris y negra. (Pressman, 2023, p.120)

Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- Revisen estructuras de datos internas para garantizar su validez.

1.6.4.2 Pruebas de Caja Negra

La prueba de caja negra, test funcional o prueba comportamental, es un tipo de prueba de software directa, cuya finalidad es analizar la compatibilidad entre las interfaces de cada

uno de los componentes del software. No tiene en consideración la lógica interna del sistema. (Pressman, 2023, p.120)

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a bases de datos externas.
- Errores de comportamiento o rendimiento.
- Errores de inicialización y terminación.

1.6.4.3 Pruebas de Estrés

Las pruebas de estrés se diseñan para simular situaciones de carga del mundo real. Conforme aumenta el número de usuarios simultáneos de la aplicación o el número de transacciones en línea o la cantidad de datos descargados o subidos estas pruebas de rendimiento ayudarán a responder preguntas sobre el rendimiento de la aplicación. (Pressman, 2023, p.120)

1.6.4.4 Pruebas de Accesibilidad

- **Axe (Axe DevTools)**

Esta herramienta es de utilidad para inspeccionar la estructura de la página web y verificar si la misma cumple con todas las pautas de la WCAG y otros estándares existentes ya mencionados. Para utilizarla en Google Chrome, simplemente se debe instalar la extensión desde el Chrome Web Store (link). Para ejecutarla se debe abrir la consola de desarrollador y seleccionar la pestaña “Axe DevTools”. La herramienta brinda distintas opciones de validación, es posible escanear y validar toda la página web o un elemento específico. Por ejemplo, si se selecciona la opción de validar toda una página, rápidamente se genera un reporte completo

de todos los hallazgos de accesibilidad que contiene el sitio, categorizados en distintos niveles de criticidad.

- **Colour Contrasr Analyser (CCA)**

Las pruebas que se realizan utilizando la herramienta **CCA** son fundamentales para simular situaciones de personas con baja sensibilidad al contraste de colores. Esta condición puede presentarse en personas en situación de discapacidad visual como, por ejemplo, algunas de las variantes del daltonismo. A su vez, este tipo de situación se vuelve más común en personas mayores de edad y puede verse empeorada con el pasar de los años.

- **WAVE (Website Accessibility Versatile Evaluator)**

WAVE es una suite de aplicaciones que ayuda a los diseñadores, creadores y autores de sitios web a garantizar la accesibilidad de sus contenidos. Esta suite puede ser utilizada por empresas de todos los tamaños, y también dispone de una herramienta de seguimiento e informes a nivel empresarial llamada herramienta de accesibilidad Pop Tech. WAVE tiene previsto ofrecer una API pública completa que permitirá a los programadores acceder a todas las funciones clave.

Los usuarios pueden emplear extensiones de Firefox o Chrome para utilizar el programa y comprobar fácilmente páginas individuales. WAVE incorpora comentarios directamente en el contenido del sitio web con detalles sobre posibles errores.

1.6.4.5 Pruebas de Unitarias

Las pruebas unitarias son el proceso en el que se prueba la unidad funcional de código más pequeña. Las pruebas de software ayudan a garantizar la calidad del código y son una parte integral del desarrollo de software. Una práctica recomendada en el desarrollo de software es escribir el software como unidades pequeñas y funcionales, y luego escribir una

prueba unitaria para cada unidad de código. Puede escribir primero pruebas unitarias como código. Luego, ejecute ese código de prueba de forma automática cada vez que realice cambios en el código del software.

1.7 HERRAMIENTAS

En lo que respecta a las herramientas para el desarrollo del software se usará el servidor apache, como editor de interfaces web HTML, el FRAMEWORK de codeigniter, editor de código VISUAL STUDIO CODE además del lenguaje de alto nivel PHP para operar las consultas del sistema, para la construcción de la base de datos y poder centralizar la información se usará el gestor de base de datos MySQL que nos permitirá:

- Crear y administrar la base de datos, los usuarios y los permisos.
- Diseñar y ejecutar las instrucciones SQL.
- Realizar las consultas requeridas por el usuario.

- **Apache (servidor Web HTTP)**

El servidor web Apache es el principal producto de software mantenido por la fundación Apache. Es un software de servidor web de código abierto, multiplataforma y gratuito. Como todo servidor web es responsable de atender las solicitudes de los clientes cuando quieren consultar una URL. Apache está atento a las solicitudes, que llegan mediante el protocolo HTTP y se encarga de enviar las respuestas a los clientes.(Zúñiga, 2023, p. 54)

- **Framework**

CodeIgniter es un framework de desarrollo web basado en PHP, diseñado para crear aplicaciones web de manera rápida y sencilla. Se destaca por su simplicidad, velocidad y bajo consumo de recursos, lo que lo convierte en una excelente opción para proyectos pequeños y medianos. Utiliza el patrón de arquitectura MVC (Modelo-Vista-Controlador), lo que separa

la lógica del negocio de la presentación, facilitando el mantenimiento y la escalabilidad de las aplicaciones. (Jacinto, 2023, p. 10).

- **IDE (Visual Studio Code)**

Visual Studio Code, al que conocemos también como VSCode, es un editor de código para programadores gratuito, de código abierto y multiplataforma. Está desarrollado por Microsoft, una compañía con una dilatada experiencia en la creación de IDEs (entornos de desarrollo integrados), que ha conseguido plasmar su larga tradición en el sector para ofrecer una herramienta ligera y práctica que la comunidad ha adoptado en masa(Zúñiga, 2024, p. 2).

- **Lenguaje de Programación**

PHP es un lenguaje de programación de uso general que se utiliza, sobre todo, en el entorno del desarrollo web. Este lenguaje se utiliza generalmente para desarrollar el backend de una web, el lado del servidor. Aun así, tiene numerosas utilidades en frontend. Es por esto que es uno de los principales lenguajes de programación en el mundo de la programación web.

Cualquiera puede beneficiarse del aprendizaje de PHP, pero es aún más esencial para aquellos interesados en programación web. PHP está disponible en los principales sistemas operativos como Linux, Microsoft Windows y macOS. La mayoría de los servidores web incluyendo a Apache e IIS, también soportan PHP. (Institute, 2022, p. 2).

- **Gestor de Base de Datos**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo. Dicho de otro modo, es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran

archivo, lo cual permite mayor velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. (Bartholomew, 2014, p. 4).

1.8 ALCANCES Y LIMITES

1.8.1 Alcances

El sistema de información C.A.T. permite realizar el control de los casos presentados centro el cual brinda un control de los procesos de ingreso y registro de casos que permiten centralizar de manera organizada la información, para después ser almacenada en la Base de Datos.

El C.A.T. cuenta con los siguientes roles de usuario acorde a las funciones que desarrolla cada funcionario del centro:

- **Módulo de control sistema:**
    - rol de usuario administrados
    - rol de usuario jefe de unidad
    - rol de usuario al terapeuta
- **Módulo de Registro de casos:**
    - registro de nuevos casos
    - registro de pacientes
- **Módulo seguimiento y desarrollo:**
    - Sesión de horarios
    - Seguimiento del caso
    - Concluir el caso

- **Modulo Historial:**
  - Historial en general de los casos
- **Módulo de Reportes**
  - Informe final de los casos
  - Búsqueda de casos de las carpetas

1.8.2 Limites

El sistema de información de casos del Centro de Atención Terapéutica tiene las siguientes limitaciones:

- El sistema de desarrollo tendrá medidas de seguridad y confidencialidad que será implementado para el manejo del mismo.
- El sistema no realizara informes de costos en el transcurso de seguimientos de los casos de las víctimas.
- El acceso a la información sobre los casos será restringido de acuerdo a los tipos de niveles de usuario.
- El sistema planteado se limita a los requerimientos funcionales específicos del control de los registros y cuantificar los datos incurridos en el periodo de ejecución del sistema.
- El sistema es accesible solo para la institución pertinente del centro de atención terapéutico

1.9 APORTES

El siguiente aporte del presente proyecto, será la implementación de un sistema de información para el control de casos como una herramienta de consulta que podrá brindar apoyo al encargado del centro para el seguimiento sean administrados de forma ágil y confiable.

- Como aporte para la Unidad de centro de atención terapéutico es poderles brindar el sistema de información web para que se pueda desarrollar mejor sus actividades y tener su información a simple descarga para la toma de decisiones.

CAPÍTULO II

MARCO TEÓRICO

2 CAPÍTULO II - MARCO TEÓRICO

2.1 INTRODUCCIÓN

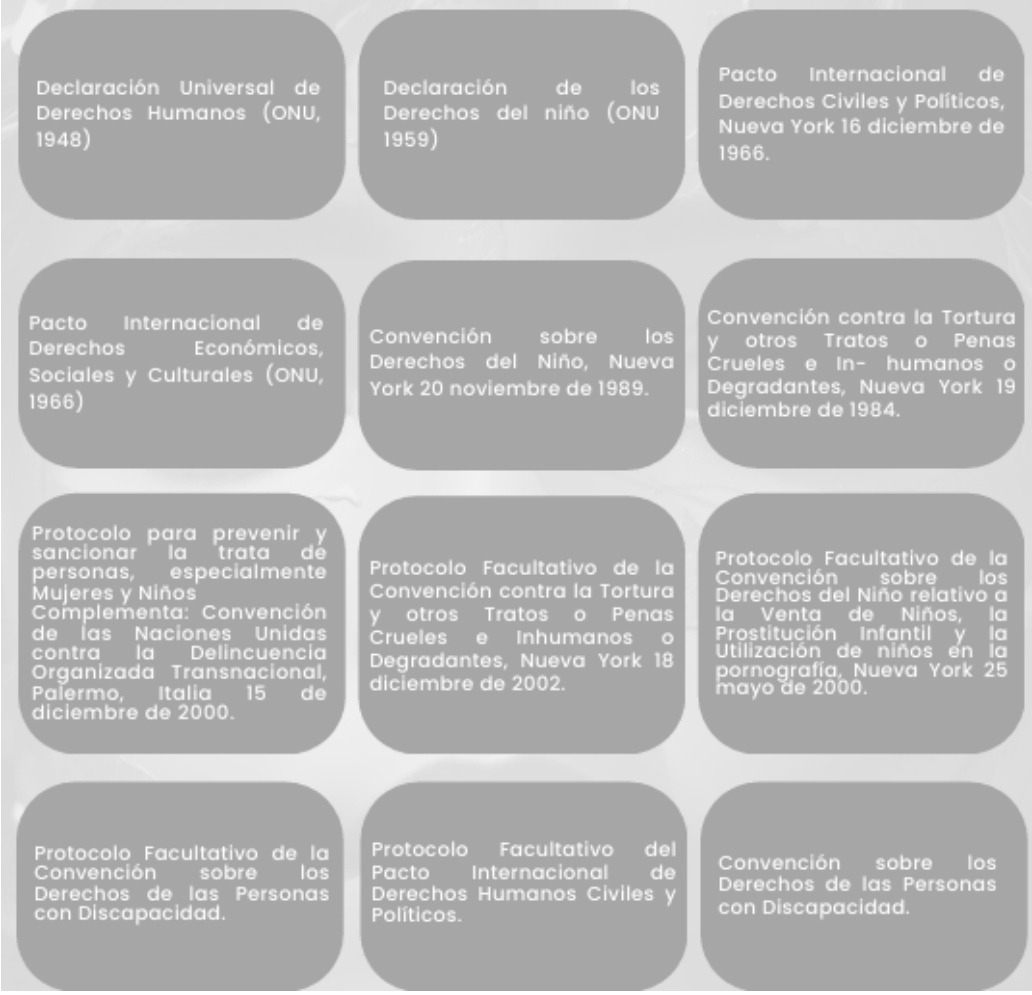
En este capítulo se abordará los fundamentos y objetivos detrás de la renovación y optimización de sitios web para el centro de atención terapéutico (CAT), en caso de las víctimas, destacando la importancia de la integración de tecnologías modernas para mejorar la experiencia del usuario y aumentar la efectividad de los registros de los casos en la unidad.

2.2 DERECHOS HUMANOS

El marco normativo de protección de los derechos de niñas, niños y adolescentes en Bolivia se fundamenta en principios esenciales del derecho constitucional, como el bloque de constitucionalidad y el control de constitucionalidad. Estos principios aseguran que las normas y disposiciones internas estén alineadas con los tratados internacionales y las convenciones sobre derechos humanos, otorgándoles una jerarquía superior en el ordenamiento jurídico nacional. Además, los principios de favorabilidad y progresividad desempeñan un rol crucial en la defensa de los derechos de esta población, al garantizar que cualquier interpretación legal o medida aplicada siempre priorice el mayor beneficio para el niño, niña o adolescente, y que los derechos reconocidos evolucionen hacia una mayor protección y disfrute de los mismos. Así, la normativa de protección no solo se articula dentro de un marco legal robusto y coherente, sino que también se desarrolla en un contexto dinámico que busca adaptarse y avanzar conforme a las necesidades cambiantes de la niñez y adolescencia. (Perez,2020, p.50).

Figura 2

Sistema Universal de Protección de DDHH



Nota. Universal de Protección de DDHH, ONU, 1948 Declaración Universal de Derechos Humanos de las Naciones Unidas

Recordando que en la Declaración Universal de Derechos Humanos las Naciones Unidas proclamaron que la infancia tiene derecho a cuidados y asistencia especiales, Convencidos de que la familia, como grupo fundamental de la sociedad y medio natural para el crecimiento y el bienestar de todos sus miembros, y en particular de los niños, debe recibir la protección y asistencia necesarias para poder asumir plenamente sus responsabilidades dentro de la comunidad, Reconociendo que el niño, para el pleno y armonioso desarrollo de su personalidad, debe crecer en el seno de la familia, en un ambiente de felicidad, amor y

comprensión. Considerando que el niño debe estar plenamente preparado para una vida independiente en sociedad y ser educado en el espíritu de los ideales proclamados en la Carta de las Naciones Unidas y en particular, en un espíritu de paz, dignidad, tolerancia, libertad, igualdad y solidaridad.

1. Protección integral contra la discriminación (Art. 24.1)
2. Registro de nacimiento e identidad (Art. 24.2)
3. Derecho a la nacionalidad (Art. 24.3):

Son el conjunto de derechos y libertades fundamentales para el disfrute de la vida humana en condiciones de plena dignidad, y se definen como intrínsecos a toda persona por el mero hecho de pertenecer al género humano. Estos derechos, establecidos en la Constitución y en las leyes, deben ser reconocidos y garantizados por el Estado. (Flores, 2024, p. 57).

2.2.1.1 Violación de los Derechos Humanos

La violación de los derechos humanos indica la vulneración o negación de los derechos de mujeres y hombres, por parte de personas servidoras públicas o autoridades que puedan ejercer poder sobre las personas.

Ante la violación de los derechos humanos, resulta vital incorporar la perspectiva de género cruzada con otras consideraciones socio económicas, físicas y culturales, para dar luz a las situaciones de vida de mujeres tales como: aquellas pertenecientes a grupos étnicos, mujeres migrantes, niñas, adultas mayores, con discapacidad y todas aquellas que por alguna condición adicional se encuentran en una posición de mayor vulnerabilidad.

Figura 3

Gráfico Derechos Humanos

Nota. Violaciones a derechos humanos de población, SISDHES-CODHES,2020, Analisis de los derechos humanos.

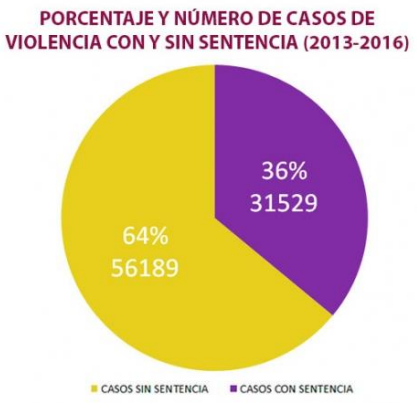
2.2.1.2 Violencia Contra las Mujeres

La Ley General de Acceso de las Mujeres a una Vida Libre de Violencia (2007), en su artículo 5, fracción IV, define la violencia contra las mujeres como:

Esta ley describe que las mujeres (por el hecho de ser mujeres) sufren violencia de diversos tipos, como: violencia psicológica; violencia física; violencia patrimonial; violencia económica; violencia sexual y cualquier otra forma análoga que lesione o dañe la dignidad, integridad o libertad de las mujeres. Las modalidades de la violencia refieren el espacio o ámbito en que tiene lugar la acción y se identifican: la violencia laboral y docente; violencia en la comunidad; v violencia institucional y violencia feminicida. (Pachas, 2019, p. 15)

Figura 4

Porcentaje por Número de Casos



Nota. Número de casos de violencia, Ministerio Publico, 2013, Coordinadora de la mujer.

La violencia contra las mujeres tiene su origen en las relaciones de desigualdad de poder, resultante de las estructuras culturales e históricas del patriarcado; es vital reconocerla y erradicarla ya que es una de las expresiones más extremas de violación a los derechos humanos de las mujeres. La violencia contra las mujeres no distingue etnia, clase, religión o edad, y se manifiesta en varias acciones como humillaciones, persecución, prohibiciones, aislamiento, control o cualquier otra acción que impida que las mujeres gocen de sus derechos y libertades. (Pachas, 2019, p. 15)

Figura 5

Tipos de Violencia

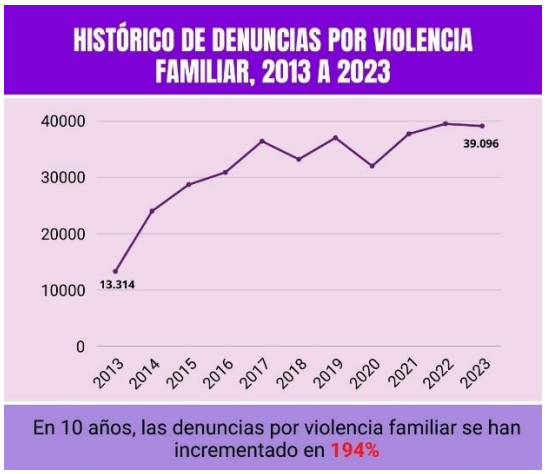
Nota. Instituto Nacional Estadísticas, Ministerio Publico, 2013, Características de la Violencia Contra la Mujer.

2.2.1.3 Violencia Contra los Niños

El Estado Plurinacional de Bolivia ratificó la CDN el 8 de marzo de 1990. Esta Convención garantiza a los niños los derechos inherentes a los seres humanos y tiene como objetivo proteger a los niños como miembros vulnerables de la sociedad. Se complementa con tres Protocolos facultativos, a saber: la Convención relativa a la participación de niños en los conflictos armados (2000), la Convención sobre la venta de niños, la prostitución infantil y la utilización de niños en la pornografía (2000), y la Convención relativa al establecimiento de un procedimiento de comunicaciones (2011). Todos estos textos fueron ratificados por el Estado Plurinacional de Bolivia en 2004, 2003 y 2013 respectivamente. (Pachas, 2019, p. 15)

Figura 6

Denuncia por Violencia



Nota. Histórico de Denuncias por Violencia, Ministerio Publico, 2013, Características de la Violencia Contra los Niños.

A nivel nacional, la Constitución boliviana consagra los derechos del niño prohibiendo todas las formas de violencia cometida contra ellos y el trabajo forzoso. También consagra el derecho a la educación, el derecho al acceso a la cultura y al deporte, el derecho a vivir en un ambiente sano, protegido y equilibrado, así como el derecho a la autodeterminación de las personas (Constitución de Bolivia, 2009). Además, la aprobación en 1999 del Código Niña, Niño y Adolescente permitió avanzar considerablemente en la aplicación de la Convención Internacional de los Derechos del Niño.

2.2.2 Control Terapéutico

Un tratamiento terapéutico implica, entre otras cosas, escuchar con atención lo que el paciente o victima tiene que decir y buscar qué aspectos personales, sociales, familiares, etc., son responsables del problema. También supone informar al paciente sobre cómo puede resolver los problemas planteados y emplear técnicas psicológicas específicas tales como, por ejemplo, el entrenamiento en respiración o relajación, la resolución de problemas

interpersonales, el cuestionamiento de creencias erróneas, el entrenamiento en habilidades sociales, etc. El empleo de fármacos no es parte del tratamiento psicológico, aunque pueden combinarse ambos tipos de terapia cuando se considera oportuno. (Pachas, 2019, p. 15)

¿Cuándo es conveniente o necesario un tratamiento terapéutico?

Un tratamiento terapéutico o psicológico debe aplicarse cuando se tiene un problema que desborda a la persona, es decir, que le dificulta o impide vivir de la forma deseada o que le produce gran malestar y sufrimiento.

Figura 7

Control Psicólogo

Nota. Terapia psicológica en niños, Ministerio Publico, 2016, Características de terapia psicológica en niños.

2.2.2.1 Tipos de Control Terapéutico

Los principales tratamientos psicológicos existentes en la actualidad son:

- ✓ La terapia cognitivo-conductual (basada en el modo de aprender nuevas formas de pensar, actuar y sentir).
- ✓ Las psicoterapias psicoanalíticas y dinámicas (centradas en el estudio introspectivo del ser humano).
- ✓ las terapias de corte existencial-humanista (basadas en gran parte en la relación terapeuta-paciente).

- ✓ Las terapias sistémicas (que consideran los problemas de una persona como la expresión de que algo funciona mal en su sistema familiar o de pareja, lo que implica hacer cambios en dicho sistema).

2.3 SISTEMA

La definición de sistemas web son aquellos sistemas alojados en internet o una intranet, con la peculiaridad que brindan funcionalidades potentes y más específicas a diferencia de las páginas web. Las ventajas de usar un sistema web es que provee de facilidad para actualizar las aplicaciones constantemente sin la necesidad de instalar un nuevo software, a su vez, puede ser codificado en lenguajes de programación para navegadores como JavaScript, finalmente, este sistema también nos permite ejecutarlo en múltiples plataformas.(Pachas, 2019, p. 15)

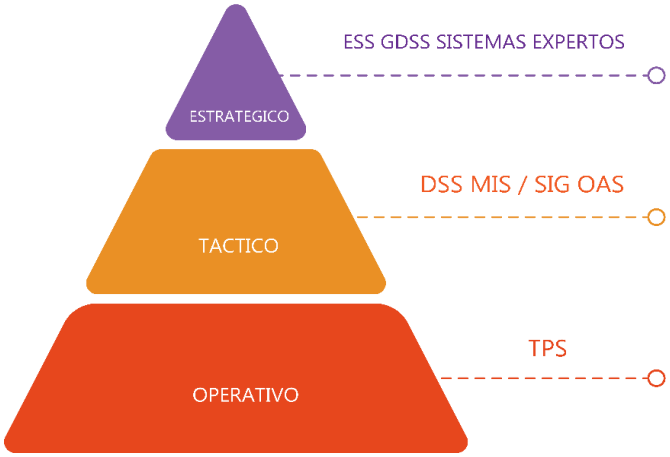
2.4 SISTEMAS DE INFORMACIÓN

Debido a la complejidad de los procesos de manejo de la información y los diversos grados o niveles en los que, según los problemas, es posible estructurar datos y procesos, es necesaria la existencia de diferentes categorías de sistemas de información (SI) que puedan cubrir toda la información que la organización necesita.

Los sistemas de información se dividen en cuatro niveles: estratégico, administrativo, de conocimiento y operativo.

Figura 8

Niveles del Sistema de Información



Nota. Niveles del sistema de información, Benítez, 2012, sistemas de información.

2.5 SISTEMA WEB

Según R. Pressman (2002), un sistema web o WebApp, es una aplicación que no está ligada a una plataforma específica, sino que está desarrollada para ser implementada en un servidor web con características y capacidades mayores a una página web.

Las características de una WebApp son las siguientes:

- Inmediatez. Las aplicaciones Web ofrecen una inmediatez que no se encuentra en otros tipos de software. Es decir, el tiempo que se tarda en comercializar un sitio Web completo puede ser cuestión de días o semanas. Los desarrolladores deberán utilizar los métodos de planificación, análisis, diseño, implementación y pruebas que se hayan adaptado a planificaciones apretadas en tiempo para el desarrollo de WebApps.
- Seguridad. Dado que las WebApps son accesibles a través de la red, es difícil, restringir el acceso a los usuarios finales puede ser complicado. Para salvaguardar información sensible y asegurar la transmisión de datos deberán

implementarse fuertes medidas de seguridad tanto en la infraestructura que apoya una WebApp como dentro de la misma aplicación.

- Estética. La apariencia y la interacción son aspectos cruciales del atractivo de una WebApp. Cuando una aplicación se diseña para comercializar productos o ideas, la estética juega un papel muy importante para el éxito del diseño técnico.

2.5.1 Web 2.0

Web 2.0, por otra parte, es un concepto que permite nombrar a las aplicaciones de Internet que facilitan la interacción de los usuarios. Los sitios que forman parte de la Web 2.0 posibilitan que los internautas compartan información y desarrollen tareas de forma mancomunada. Entre los ejemplos de Web 2.0 pueden nombrarse a las redes sociales (como **Facebook**), los portales de alojamiento de videos (**YouTube**) e imágenes (**Flickr**) y los servicios wikis ((Choque, 2019, p.86),).

2.5.2 Diferencias.

La diferencia entre web 1.0 y 2.0 se puede definir al contemplar los cambios en la forma de consumir contenidos y de relacionarnos, la proliferación de actividad en las redes sociales, y principalmente el paso a una nueva forma de acercarnos a la generación de contenidos.

Web 1.0 Este modelo no permitía la interacción entre usuarios, comentarios, citas, ni aportar ningún tipo de contenido; esta función era destinada por completo al administrador del sitio.

web 2.0 Con el paso del tiempo los avances en tecnología permitieron que aumentara el tamaño del ancho de banda, la consolidación de los dispositivos y funcionalidades inalámbricos, y la capacidad de conexión a la red. Así, se ha podido llegar a un nuevo punto en el modelo de creación y distribución del contenido, borrando las brechas técnicas del pasado y ofreciendo herramientas al alcance de un público más amplio. En síntesis, todos

estamos en posibilidad de consumir, pero también de generar y compartir contenidos. Esta es una transformación de la dinámica unidireccional de la web 1.0 a una versión con rasgos muy participativos para el usuario. (Choque, 2019, p.86),

2.6 CONTROL Y SEGUIMIENTO

El control y seguimiento son procesos en la gestión de calidad, que buscan garantizar la efectividad y eficiencia de los servicios ofrecidos. La implementación de sistemas de control permite a las organizaciones ver el desempeño y realizar ajustes necesarios para mejorar la satisfacción del cliente y la operatividad.

2.6.1 Importancia del Seguimiento

El seguimiento de servicios implica la evaluación continua del desempeño y la calidad de los servicios ofrecidos. Parasuraman, Zeithaml y Berry (1985) desarrollaron el modelo, que mide la brecha entre las expectativas del cliente y la percepción del servicio recibido, resaltando la importancia del seguimiento para cerrar estas brechas.

2.6.2 Tipos de Control

El control administrativo se despliega en diversos tipos adaptados a las diferentes etapas y aspectos de la gestión empresarial.

Control Preventivo: su enfoque radica en prever posibles problemas y desviaciones y se implementa antes de que ocurran las actividades planificadas. **Por ejemplo**, establecer procedimientos y políticas para evitar conflictos y asegurar que las operaciones se adhieran a las normativas internas.

Control Concurrente: se lleva a cabo durante el desarrollo de la actividad para identificar posibles desviaciones en tiempo real y corregirlas de inmediato. Un **ejemplo** sería la supervisión constante de un proyecto para asegurar que se adhiera al cronograma y al presupuesto establecido.

Control Retroalimentativo o Posterior: se realiza después de completar la actividad. Se analizan los resultados y se aprende de la experiencia. **Por ejemplo**, la revisión de los informes financieros al final del trimestre para evaluar el rendimiento y tomar decisiones para el próximo periodo.

Control Estratégico: está orientado a asegurar que la organización alcanza sus objetivos a largo plazo y se mantiene en línea con su visión y estrategia. **Ejemplo:** revisión y ajuste de la estrategia comercial como respuesta a los cambios que experimenta el mercado o la competencia.

Control Táctico o de Gestión: se centra en supervisar y evaluar el desempeño de los departamentos y equipos intermedios para garantizar que contribuyan eficazmente a los objetivos organizacionales. Por **ejemplo**, hacer una evaluación del rendimiento de un departamento de ventas para asegurar el cumplimiento de cuotas y metas establecidas.

Control Financiero: dirigido a garantizar la eficiencia y eficacia en la gestión financiera, controlando los presupuestos, costos y flujos de efectivo. Un **ejemplo** es la revisión periódica que evalúa la salud financiera para poder tomar decisiones basadas en la situación económica.

Control de Calidad: su objetivo es garantizar la calidad de los productos o servicios ofrecidos por la empresa. **Un ejemplo** son las inspecciones de calidad que se realizan durante la producción para asegurar que los productos cumplen con los estándares establecidos.

Control de Recursos Humanos: se ocupa de supervisar el desempeño de los empleados y de garantizar un ambiente laboral positivo. Un **ejemplo** sería la evaluación del rendimiento anual, la capacitación continua y los programas de retroalimentación para mejorar el desempeño y la satisfacción de los empleados.

2.7 METODOLOGÍA BASADA EN UML(UWE)

La metodología que se emplea para este proyecto es UWE (Ingeniería Web Basada en UML) por las características que presenta. UWE es una metodología basada en el proceso unificado y UML (Lenguaje Unificado de Modelado) para el desarrollo de aplicaciones web, cubre todo el ciclo de vida de las aplicaciones web (Escobar , 2015, p. 2).

2.7.1 Metodología UWE (UML-Based Web Engineering)

Es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario(Sanz, 2022, p. 9).

2.7.2 Características de UWE

La metodología UWE define vistas especiales representadas gráficamente por diagramas en UML, tales como el modelo de navegación y el modelo de presentación. Los diagramas se pueden adaptar como mecanismos de extensión basados en estereotipos que proporciona UML.

Los diagramas se pueden adaptar como mecanismos de extensión basados en estereotipos que proporciona UML. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son los que finalmente se utilizan en las vistas especiales para el modelado de aplicaciones web. De esta manera se obtiene una notación UML adecuada para un dominio especifico a la que se conoce como “perfil UML”

Un perfil UML consiste en una jerarquía de estereotipos y un conjunto de restricciones. Los estereotipos son utilizados para preparar instancias de las clases. Las ventajas de utilizar los perfiles de UML es que casi todas las herramientas CASE de UML los reconocen. Los modelos deben ser fácilmente adaptables al cambio en cualquier etapa del desarrollo. (Narváez, p. 23).

2.7.3 Modelos de la Metodología UWE

El modelo que propone UWE está compuesto por sub modelos:

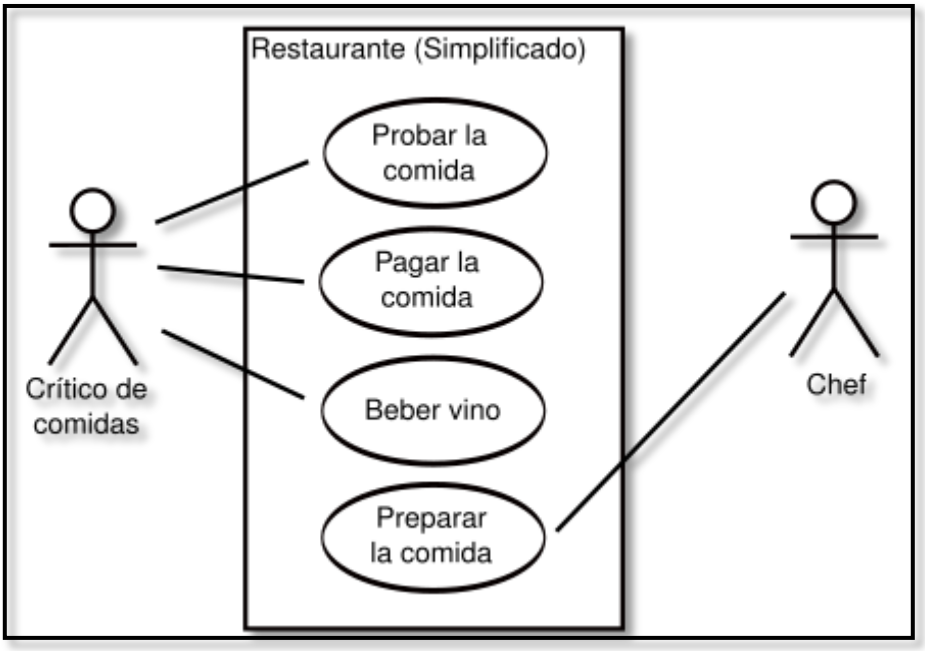
- Modelo de Requisitos “caso de uso”
- Modelo de contenido “conceptual”
- Modelo de navegación
- Modelo de presentación
- Modelo de proceso

2.7.3.1 Modelo de Requisitos “Casos de Uso”

El modelo de caso de uso identifica los requisitos del sistema en términos de la funcionalidad que debe existir para conseguir los objetivos establecidos por el usuario o para resolver un problema identificado por el usuario. Los casos de uso describen los comportamientos principales identificados en los requisitos y describen el valor que los resultados dan a los usuarios; no describen cómo funciona internamente el sistema. Los actores son los usuarios del sistema y representan los diferentes roles que la gente y otros sistemas representan cuando interactúan con el sistema. (Model, 2021, p. 2)

Figura 9

Modelo de Casos de Uso



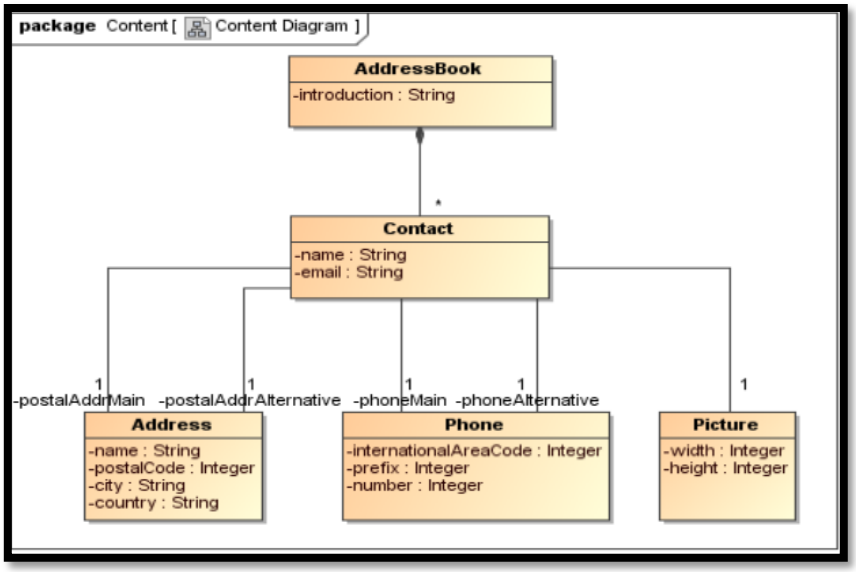
Nota. El modelo de caso de uso identifica los requisitos del sistema Web, Guerrero, 2014, según UWE-UML

2.7.3.2 Modelo de Contenido "Conceptual"

UWE apunta a construir un modelo conceptual de una aplicación Web, procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web. La construcción de este modelo lógico-conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web. (Choque, 2019, p.86),

Figura 10

Modelo de Contenido



Nota. Modelo de contenido en Ingeniería de la Web, Guerrero, 2014, según UWE-UML.

2.7.3.3 Modelo de Navegación

Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. El primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionarán. (Pachas, 2019, p. 15)

2.7.3.4 Modelo de Presentación

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario. Los elementos de modelado son:

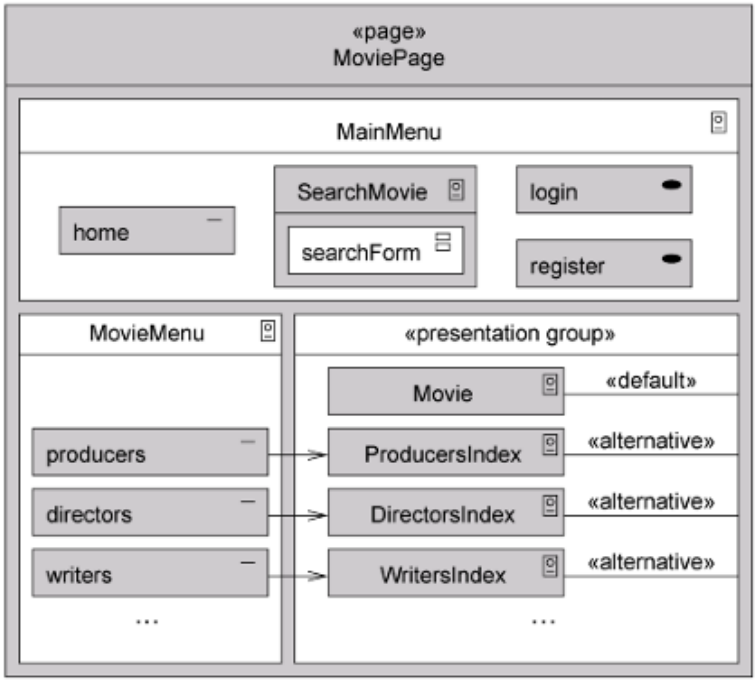
- **Clases de Navegación,** que se denotan con (0), representan los nodos navegables de la estructura de hipertexto.
- **Links de Navegación,** que muestran el vínculo directo entre las clases de navegación.

- **Caminos de Navegación Alternativos,** los cuales son visualizados con el estereotipo <<menu>> ().

Primitivas de Acceso, *las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación(<<index>> o <<guided tour>>) o para seleccionar ítems (<<query>>).*

Figura 11

Modelo de Navegación



Nota. Modelo de navegación en Ingeniería de la Web, Guerrero, 2014, según UWE-UML.

2.7.3.5 Modelo de Proceso

Representa los aspectos dinámicos de la aplicación Web y especifica funcionalidad cómo transacciones y workflows de actividades. Se modela mediante un diagrama de actividades de UML, y es resultado de refinar el diagrama de actividades modelado durante la especificación de requerimientos. Muestra el flujo de la ejecución representado por nodos de actividad conectados, los nodos de control que proveen constructores de flujo de control como

decisiones y sincronización y nodos de objetos que representan el flujo de datos.(Rotta, 2016, p. 10)

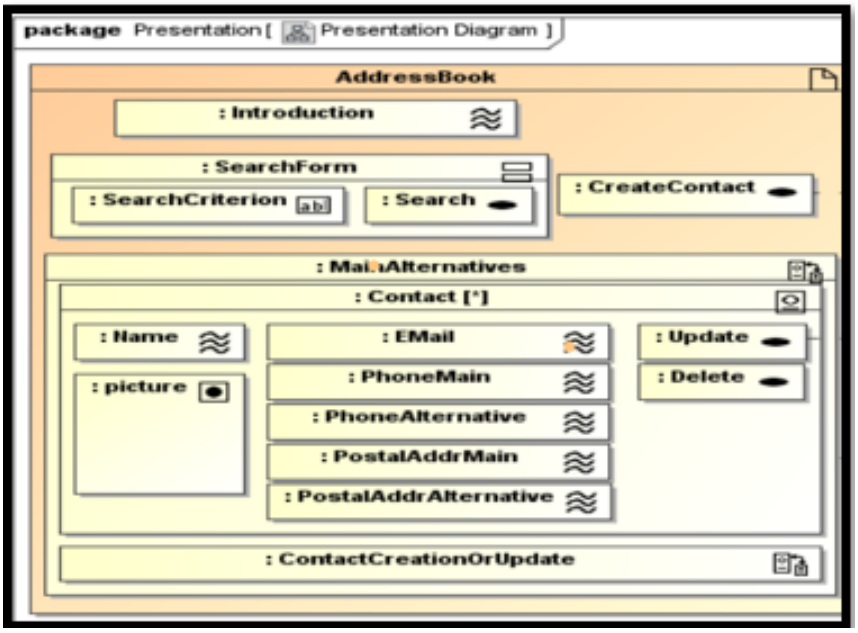
Clases de Presentación, las cuales se basan directamente en los nodos del modelo de navegación, cada clase está compuesta por elementos de UI tales como, texto (<<text>>), vinculo (<<anchor>>), botón (<<button>>), imagen (<<image>>), formulario (<<form>>), y colección de vínculos (<<anchored collection>>)

Páginas Web (<<page>>), que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.

Grupo de Presentación (<<presentation group>>), el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación.

Figura 12

Modelo de Presentación



Nota. Modelo de presentación en Ingeniería de la Web, Guerrero, 2014, según UWE-UML.

2.7.4 Fases de la Metodología UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

2.7.4.1 Captura, Análisis y Especificación de Requisitos

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

2.7.4.2 Diseño del Sistema

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

2.7.4.3 Codificación del Software

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

2.7.4.4 Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

2.7.4.5 La Instalación o Fase de Implementación

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

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

2.7.4.6 El Mantenimiento:

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

2.8 ARQUITECTURA DE SOFTWARE (MVC)

La arquitectura nos permite planificar a priori nuestro desarrollo y elegir el mejor conjunto de herramientas para llevar a cabo nuestros proyectos, es por tanto un paso crítico antes siquiera de pasar a programar ya que determinará en gran medida el ritmo del desarrollo e incluso los factores económicos y humanos durante el proceso. Por tanto, a la hora de elegir un patrón de arquitectura siempre es necesario pensar en una serie de cuestiones que determinan el uso final que vamos a darle a nuestro software:

Coste - ¿Cuánto estamos dispuestos a invertir en el desarrollo y mantenimiento de nuestro sistema? Como hemos visto hay ciertos patrones más complejos, que requieren más infraestructura y cuyo desarrollo puede ser más irregular, por tanto, hemos de saber cuánto estamos dispuestos a invertir primero en el desarrollo de nuestra aplicación.

Tiempo de desarrollo - Igualmente, y muy relacionado con lo anterior, debemos de preguntarnos cuanto tiempo disponemos para desarrollar el producto, y cómo de cerca se encontraría la fecha de entrega o de salida al mercado.

Número de usuarios - Sin duda uno de los ítems críticos a la hora de desarrollar el producto es preguntarnos qué tipo de producto es y cuantos usuarios soporta ¿Funciona a través de web? ¿Es stand-alone? ¿Debe de soportar cargas elevadas por diseño?, estas preguntas pueden declinarnos a elegir patrones más o menos distribuidos, pasando, por ejemplo, de uno menos distribuido como el de capas al más distribuido o broker.

Nivel de aislamiento - Otro factor importante a tener en cuenta es si nuestro producto funciona de forma aislada al resto de productos del usuario o si debe de integrarse o permitir integraciones de terceros. Algunas arquitecturas, como la de capas, son más cerradas y podrían dificultar estas integraciones si lo escogemos sobre otras.

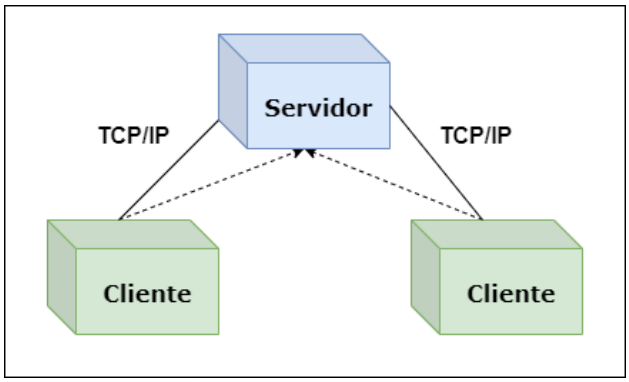
En ocasiones, y cuando tenemos estos hechos bien planteados y razonados, elegir un patrón de arquitectura también puede ser una cuestión de familiaridad, comodidad o simple preferencia, por eso es aconsejable probarlos, para intentar también familiarizarse con ellos y con el diferente flujo de trabajo que proponen.

2.8.1 Patrones de Arquitectura de software

El patrón cliente servidor es muy usado sobre todo en el diseño de webs y servicios online, y se basa en el concepto de la existencia de un servidor (que proporciona el servicio) y una serie de clientes, que piden al servidor y reciben una respuesta del mismo.

Figura 13

Patrones de Arquitectura



Nota. El patrón cliente servidor proporciona el servicio, Serrano, 2014, MVC.

Ventajas

- ✓ **Centralización -** Todos los recursos disponibles se hayan centralizados en un único punto, lo que hace más sencillo su administración y más difícil para un cliente el uso de acciones dañinas.
- ✓ **Escalabilidad -** Al funcionar de manera independiente es más sencillo mejorar cada pieza de forma separada o añadir nuevos nodos a la red creada.

- ✓ **Mantenimiento simplificado -** Al funcionar de manera independiente y con separación clara de responsabilidades, es más sencillo mantener cada una de las piezas e incluso poder trasladar con sencillez el servidor a nuevo hardware/software si fuera necesario.

Desventajas

- ✓ **Disponibilidad -** Al depender de un servidor para satisfacer las peticiones de los clientes se requiere que este esté activo y disponible en cada momento, una caída del servidor o incluso una congestión debido a la cantidad de peticiones de clientes resulta en una pérdida de funcionalidad absoluta del servicio.
- ✓ **Requisitos -** Debido a que debe satisfacer un gran número de peticiones, el software y el hardware del servidor son determinantes a la hora de usar este patrón, y usar alguno deficiente impactará negativamente en el funcionamiento de nuestro sistema.
- ✓ **Distribución -** El cliente no posee físicamente el producto ni tiene acceso a los recursos utilizados, cualquier caída del servidor implicará que el cliente no pueda acceder a su trabajo en curso.

2.8.2 Patrón modelo vista controlador (MVC).

Este famoso patrón, también conocido como patrón MVC, divide una aplicación interactiva en tres partes diferenciadas:

Modelo: Contiene la funcionalidad central y los datos.

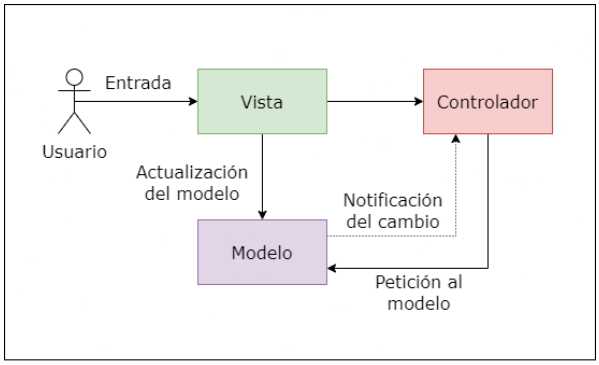
Vista: Muestra la información al usuario, siempre es posible definir una o más vistas para una misma aplicación.

Controlador: Maneja la entrada del usuario. Esto se hace para separar las representaciones internas de la información de las formas en que se presenta y se acepta la información del usuario. De esta manera se desacopla los componentes y permite una reutilización eficiente del código.

Este patrón es muy popular en el desarrollo de aplicaciones web, tanto en el caso de la parte trasera (back-end) y la frontal (front-end), siendo el patrón base de muchos frameworks conocidos como son el caso de Angular y en algún lenguaje como Java con el framework Spring.

Figura 14

Patrón Modelo Vista Controlador (MVC)



Nota. El patrón cliente servidor proporciona el servicio, Serrano, 2014, MVC.

Ventajas

Fácil colaboración - Al separar fuertemente responsabilidades es posible desarrollar muchas nuevas características sin necesidad muchas veces de tocar todas las piezas implicadas, de forma que la colaboración entre diferentes desarrolladores se amplifica notablemente.

Aplicaciones multi-vista - Al aislar las vistas del resto de la lógica de la aplicación es posible presentar las mismas funcionalidades en diferentes vistas que pueden dirigirse incluso a diversos dispositivos distintos.

Desventajas

Complejidad - Es un patrón complejo que requiere que los desarrolladores implicados tengan muy claros los conceptos de responsabilidad asignados a cada una de las tres partes implicadas.

Lento en ocasiones - En comparación a otros patrones desarrollar para este puede implicar tocar muchas y diversas piezas y tener que seguir un flujo de trabajo cerrado que a veces dificulta meter cambios que para cualquier otro desarrollo basado en otro patrón serían nimios.

2.9 HERRAMIENTAS

2.9.1 PHP (Hipertext Pre-processor)

PHP, acrónimo de Hypertext Preprocessor, es un lenguaje de programación orientado a objetos, de código abierto muy popular para el desarrollo web que puede ajustarse en HTML. Es considerado como uno de los lenguajes más potentes y populares para el desarrollo web y de aplicaciones. (González, 2006, pág. 50)

Las páginas de PHP contienen HTML con código incrustado que cumple una función y está encerrado entre las etiquetas especiales de comienzo y final.

Lo que distingue a PHP es su extrema simplicidad para programadores, desarrolladores o ingenieros en un nivel principiante, pero al mismo tiempo ofrece una gran variedad de características avanzadas para los profesionales en niveles más desarrollados.

Funciones del lenguaje PHP

- Programación del lado del servidor: ideal para páginas webs de prueba antes de la publicación en la web.

- Programación a través de la línea de comandos: los scripts PHP pueden crearse sin necesidad de un navegador o servidor, sólo se requiere de un analizador PHP. Los scripts son ideales para las tareas regulares de una web.
- Aplicaciones de escritorio: a pesar de ser una función que se utiliza con menor frecuencia, el lenguaje PHP también es adecuado para el desarrollo de aplicaciones de escritorio gracias a su nivel de complejidad.

2.9.2 Apache

Apache es un servidor web que se encarga de almacenar, procesar y servir las páginas web a los usuarios de las mismas. Se distribuye bajo una licencia de código abierto, lo que quiere decir que es gratuito y fácilmente adaptable, personalizable y reutilizable.

Creado por el desarrollador de software estadounidense Robert McCool en 1995, es considerado una de las tecnologías que impulsó el crecimiento inicial de Internet.

Desde entonces, es ofrecido por la Apache Software Foundation, una organización sin fines de lucro que se encarga de supervisar la comunidad de usuarios que lo desarrollan y mantienen.

2.9.3 Gestor de Base de Datos

Es un sistema de gestión de bases de datos relacionales (RDBMS) gratuito y de código abierto. Fue creado por los desarrolladores originales de MySQL por la preocupación de que MySQL pasara a ser comercializado después de que Oracle lo adquiriera en 2009.

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo. Dicho de otro modo, es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran

archivo, lo cual permite mayor velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. (Bartholomew, 2014, p. 4).

2.9.4 Bootstrap

Bootstrap es un framework que permite a los desarrolladores web construir páginas web responsives de una forma más rápida y sencilla. En este sentido, proporciona un conjunto de componentes y plantillas CSS, HTML y JavaScript que cualquiera puede utilizar o modificar de manera gratuita.

Su origen lo encontramos en las oficinas de Twitter, donde Mark Otto y Jacob Thornton lanzaron la primera versión en agosto de 2011 bajo el nombre Twitter Blueprint. Inicialmente, esta fue una herramienta interna que permitía que el diseño de esta popular red social fuera coherente y consistente. Sin embargo, meses más tarde, Twitter liberó el código y el framework en GitHub y lo renombró como Bootstrap.

Actualmente, Bootstrap se encuentra en su quinta versión y se ha convertido en uno de los *frameworks* de desarrollo más populares. De hecho, según estadísticas de W3Techs, el 19% de las páginas web lo utilizan.

2.9.5 IDE (Visual Studio Code)

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Es un editor de código fuente súper rápido y liviano que se puede usar para ver, editar, ejecutar y depurar código fuente para aplicaciones. Es utilizado principalmente por desarrolladores front-end. Visual Studio Code es un producto completamente diferente en comparación con Visual Studio.

2.9.6 Framework CodeIgniter

CodeIgniter es un framework de desarrollo web basado en PHP, diseñado para crear aplicaciones web de manera rápida y sencilla. Se destaca por su simplicidad, velocidad y bajo consumo de recursos, lo que lo convierte en una excelente opción para proyectos pequeños y medianos. Utiliza el patrón de arquitectura MVC (Modelo-Vista-Controlador), lo que separa la lógica del negocio de la presentación, facilitando el mantenimiento y la escalabilidad de las aplicaciones (Alvarez, 2000, pág. 45).

Funciones principales de CodeIgniter:

- **MVC (Modelo-Vista-Controlador):** Organiza el código en tres componentes principales:

✓ Modelo: Gestiona los datos y la lógica de negocio.

✓ Vista: Maneja la presentación de la interfaz de usuario.

✓ Controlador: Actúa como intermediario entre el Modelo y la Vista.

- **Rutas de URL amigables:** Ofrece control completo sobre las rutas de la aplicación, permitiendo URLs amigables y optimizadas para SEO.
- **Soporte para bases de datos:** Incluye una capa de abstracción de base de datos que simplifica la interacción con diferentes bases de datos como MySQL, PostgreSQL, y SQLite.
- **Validación de formularios:** Facilita la validación de datos de entrada, lo que garantiza que los formularios sean seguros y los datos sean procesados correctamente.
- **Seguridad:** Proporciona características integradas como la prevención de ataques CSRF (Cross-Site Request Forgery) y XSS (Cross-Site Scripting).

- **Bibliotecas y ayudas:** Ofrece un amplio conjunto de bibliotecas y ayudas (helpers) para tareas comunes, como manejo de sesiones, carga de archivos, envío de correos electrónicos, etc.
- **Soporte para migraciones de bases de datos:** Permite controlar la estructura de la base de datos a través de migraciones, lo que facilita la actualización y el mantenimiento de bases de datos en distintos entornos.
- **Cacheo:** Mejora el rendimiento de la aplicación mediante el almacenamiento en caché de resultados de consultas y páginas completas.
- **Módulos reutilizables:** Facilita la creación de módulos o “paquetes” que pueden reutilizarse en diferentes partes de la aplicación o en diferentes proyectos.

Características clave de CodeIgniter:

- ✓ **Ligero y rápido:** Uno de los frameworks más rápidos de PHP, con un núcleo pequeño y pocas dependencias externas.
- ✓ **Fácil de configurar:** Su configuración es sencilla y permite poner en marcha un proyecto rápidamente sin una curva de aprendizaje elevada.
- ✓ **Documentación clara y completa:** CodeIgniter es conocido por su excelente documentación, lo que facilita el aprendizaje y la resolución de problemas.
- ✓ **Soporte para URL amigables:** Las URLs generadas por CodeIgniter son legibles y adecuadas para SEO.
- ✓ **Extensible:** Puedes extender su funcionalidad mediante bibliotecas personalizadas o utilizando bibliotecas externas.

- ✓ **Gran comunidad:** Al ser un framework popular, cuenta con una gran comunidad de desarrolladores, lo que facilita encontrar soporte y recursos.

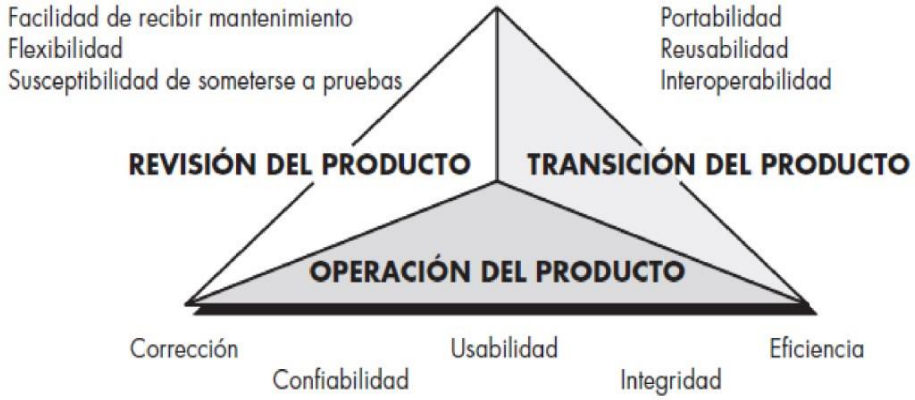
2.10 MÉTRICAS DE CALIDAD DE SOFTWARE

David Garvin, de Harvard Business School, sugiere que “la calidad es un concepto complejo y multifacético” que puede describirse desde cinco diferentes puntos de vista.El punto de vista trascendental dice que la calidad es algo que se reconoce de inmediato, pero que no es posible definir explícitamente. El punto de vista del usuario concibe la calidad en términos de las metas específicas del usuario final. Si un producto las satisface, tiene calidad. El punto de vista del fabricante la define en términos de las especificaciones originales del producto. Si éste las cumple, tiene calidad. Según Pressman (2010), el punto de vista del producto sugiere que la calidad tiene que ver con las características inherentes (funciones y características) de un producto. Por último, el punto de vista basado en el valor la mide de acuerdo con lo que un cliente está dispuesto a pagar por un producto, en realidad, la calidad incluye todo esto y más

Los factores de la calidad de McCall, Richards y Walters proponen una clasificación útil de los factores que influyen en la calidad del software. y se centran en tres aspectos importantes del producto de software: sus características operativas, su capacidad de modificación y su adaptabilidad a nuevos entornos.

Figura 15

Factores que Afectan la Calidad de Software



Nota. Factores que afectan la calidad de software, Pressman, 2010, Ingeniería del software

- Corrección. Nivel en el que un programa cumple y satisface las especificaciones alcanzando los objetivos establecidos por el cliente.
- Confiabilidad. Grado en el que se espera que un programa cumpla con su función y con la precisión requerida.
- Eficiencia. Cantidad de recursos de cómputo y de código requeridos por un programa para llevar a cabo su función.
- Integridad. Grado en el que es posible controlar el acceso no autorizado al software o a los datos.
- Usabilidad. Esfuerzo que se requiere para aprender, operar, preparar las entradas e interpretar las salidas de un programa.
- Facilidad de recibir mantenimiento. Esfuerzo requerido para detectar y corregir errores en un programa (ésta es una definición muy limitada).
- Flexibilidad. Esfuerzo necesario para modificar un programa en funcionamiento.

- Susceptibilidad de someterse a pruebas. Esfuerzo que se requiere para probar un programa a fin de garantizar que realiza la función que se pretende.
- Portabilidad. Esfuerzo que se necesita para transferir el programa de un entorno de hardware o software a otro.
- Reusabilidad. Grado en el que un programa o partes de él, pueden ser utilizados nuevamente en otras aplicaciones (se relaciona con el empaque y el alcance de las funciones que lleva a cabo el programa).
- Interoperabilidad. Esfuerzo requerido para acoplar un sistema con otro.

2.10.1 ISO/IEC 25010

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subccaracterísticas.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las nueve características de calidad que se muestran en la siguiente figura.

Figura 16

ISO/IEC 25010

CALIDAD DEL PRODUCTO SOFTWARE								
ADECUACIÓN FUNCIONAL	EFICIENCIA DE DESEMPEÑO	COMPATIBILIDAD	CAPACIDAD DE INTERACCIÓN	FIABILIDAD	SEGURIDAD	MANTENIBILIDAD	FLEXIBILIDAD	PROTECCIÓN
COMPLETITUD FUNCIONAL CORRECCIÓN FUNCIONAL PERTINENCIA FUNCIONAL iso25000.com	COMPORTAMIENTO TEMPORAL UTILIZACIÓN DE RECURSOS CAPACIDAD	COEXISTENCIA INTEROPERABILIDAD	RECONOCIBILIDAD DE ADECUACIÓN APRENDIZABILIDAD OPERABILIDAD PROTECCIÓN FRENTE A ERRORES DE USUARIO INVOLUCRACIÓN DEL USUARIO INCLUSIVIDAD ASISTENCIA AL USUARIO AUTO-DESCRIPTIVIDAD	AUSENCIA DE FALLOS DISPONIBILIDAD TOLERANCIA A FALLOS RECUPERABILIDAD	CONFIDENCIALIDAD INTEGRIDAD NO-REPUDIO RESPONSABILIDAD AUTENTICIDAD RESISTENCIA	MODULARIDAD REUSABILIDAD ANALIZABILIDAD CAPACIDAD DE SER MODIFICADO CAPACIDAD DE SER PROBADO	ADAPTABILIDAD ESCALABILIDAD INSTALABILIDAD REEMPLAZABILIDAD	RESTRICCIÓN OPERATIVA IDENTIFICACIÓN DE RIESGOS PROTECCIÓN ANTE FALLOS ADVERTENCIA DE PELIGRO INTEGRACIÓN SEGURA

Nota. Funcionalidad, rendimiento, seguridad, mantenibilidad 2011, Según ISO/IEC 25010)

2.10.1.1 Funcionalidad

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas de los usuarios cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Completitud funcional. Grado en el que el conjunto de funcionalidades del producto cubre todas las tareas y los objetivos de usuario especificados.
- Corrección funcional. Capacidad del producto o sistema para proveer resultados exactos cuando es usado por los usuarios especificados.
- Pertinencia funcional. Capacidad del producto software para proporcionar un conjunto de funciones que facilitan la consecución de tareas y objetivos de usuario especificados.

Se llevará a cabo una exhaustiva evaluación para determinar si el software cumple con los requisitos funcionales establecidos. Este proceso garantiza que el programa posea las

características esenciales para satisfacer las demandas y expectativas del usuario, asegurando su plena funcionalidad.

A continuación, en la tabla se muestra la suma total de los puntos de función sin ajustar, para obtener los valores ajustados se aplican factores de complejidad específicos que se detallan a continuación en la siguiente tabla. Estos factores consideran diversas características del sistema.

Tabla 1

Índices de Usabilidad

IMPORTANCIA	0%	20%	40%	60%	80%	100%
ESCALA	NO INFLUENCIA	INCIDENTAL	MODERADO	MEDIO	SIGNIFICATIVO	ESENCIAL
FACTOR	0	1	2	3	4	5
¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
¿Se requiere 0 comunicación de datos?					X	
¿Existe funciones de procesos distribuidos?					X	
¿Es crítico el rendimiento?				X		
¿El sistema web será ejecutado el SO Actual?						X
¿Se requiere una entrada interactiva para el sistema?						X
¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?				X		
¿Se actualiza los archivos de forma interactiva?					X	
¿Son complejas las entradas, salidas, los archivos o las peticiones?					X	

¿Es complejo el procesamiento interno del sistema?	X	
¿Se ha diseñado el código para ser reutilizado?		X
¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?	X	

Fórmula para ajustar puntos de función, que considera factores como la complejidad técnica, el rendimiento y la usabilidad.

(1) Fórmula de Ajuste

$$PF = CT\left[0.65 + 0.01 * \sum fi\right] \qquad (1)$$

Donde:

$\sum(fi)$: Sumatoria de los valores de los factores de ajuste.

PF : puntos de función

CT : conteo total (unidades)

Con el límite superior establecido en la suma de los factores de ajuste de acuerdo a los parámetros propuestos, se realiza el cálculo del valor siguiente en la secuencia, este proceso es fundamental para mantener la coherencia en la evaluación de la complejidad del sistema.

(2) Punto Función Máxima

$$PFmax = CT\left[0.65 + (0.01 * \sum fi)\right] \qquad (2)$$

Donde:

$\sum(fi)$: Sumatoria de los valores de los factores de ajuste.

CT : Conteo Total

PFmax : Punto función máxima.

Con estos factores podemos encontrar el valor de la funcionalidad.

(3) Formula de la Funcionalidad

$$F = \frac{PF}{PF\ max} * 100 \qquad (3)$$

Donde:

F : Funcionalidad

PF : Punto función.

$PF\ max$: Punto función máxima.

2.10.1.2 Confiabilidad

La confiabilidad de un sistema se refiere a su capacidad para mantener un nivel de funcionamiento adecuado durante un período de prueba establecido que implica considerar aspectos como la madurez del sistema. Aplicaremos las fórmulas correspondientes, teniendo en cuenta las probabilidades de fallas y éxitos debido a varios factores.

- Comportamiento temporal. Grado en que un producto o realiza sus funciones de forma que el tiempo de respuesta y la ratio de rendimiento cumple los requisitos especificados.
- Utilización de recursos. Grado en que la cantidad y tipos de recursos utilizados por el producto al llevar a cabo su función bajo condiciones determinadas no exceden lo especificado.
- Capacidad. Grado en que el producto cumple los requisitos relativos a límites máximos para un parámetro (ítems almacenados, usuarios concurrentes, ancho de banda de comunicaciones).

Considerando que se tiene en cuenta que:

(4) Probabilidad de Fallas

$$P(T =< t) \quad => PF \qquad (4)$$

Donde:

P : Probabilidad

PF : Probabilidad de fallas

t : Tiempo de prueba del sistema.

(5) Probabilidad de Trabajo Sin Fallas

$$P(T > t) = 1 - F(t) => PSF \qquad (5)$$

Se considera la siguiente función al calcular la confiabilidad del sistema.

(6) Función de la Confiabilidad del Sistema MT

$$F(t) = f * e^{-\mu * t} \qquad (6)$$

Donde:

f = Funcionalidad del sistema

μ = Probabilidad de error del sistema

t = Tiempo de prueba del sistema

$F(t)$ = Confiabilidad

PSF = Probabilidad sin fallas

2.10.1.3 Usabilidad

Al evaluar la usabilidad de un sistema, es importante considerar el factor humano, asegurando que el software cumpla con los requisitos y expectativas del usuario, por lo tanto, se lleva a cabo una evaluación mediante encuestas dirigidas a los usuarios del sistema.

Estas encuestas permiten recopilar información valiosa sobre la experiencia del usuario, su satisfacción y la facilidad de uso percibida que se utiliza luego para calcular la

usabilidad del sistema, aplicando una fórmula diseñada específicamente para medir la eficacia y la experiencia del usuario en la interacción con el software.

(7) Función para Determinar la Usabilidad MT

$$FU = (\frac{\sum valor}{n} * 100)/5 \qquad (7)$$

Donde:

FU	=	Función usabilidad
$\sum \text{valor}$	=	Sumatoria de los valores de la usabilidad del sistema
n	=	Cantidad de valores de la usabilidad del sistema

2.10.1.4 Eficiencia

Capacidad del producto software para que el usuario interactúe mediante su interfaz intercambiando información para completar determinadas tareas.

Para determinar la eficiencia, se emplea la fórmula siguiente.

(8) Fórmula Eficiencia

$$E = \frac{\sum x_i}{n} * \frac{100}{n} \qquad (8)$$

Donde:

$\sum x_i$	=	Sumatoria de los valores de eficiencia
n	=	Número de preguntas
E	=	Eficiencia

2.10.1.5 Mantenibilidad

Esta métrica evalúa la cantidad de trabajo requerido para realizar cambios en el sistema, ya sea para corregir errores o agregar nuevas funcionalidades. El estándar IEEE94

recomienda utilizar el índice de madurez del sistema como indicador de su estabilidad. Por lo tanto, la ecuación para este índice es la siguiente:

(9) Cálculo del Nivel de Madurez del Sistema

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt} \qquad (9)$$

Donde:

Mt	=	Número de módulos total de la versión actual
Fa	=	Número de módulos de la versión actual que se añadieron.
Fc	=	Número de módulos de la versión actual que se cambiaron.
Fd	=	Número de módulos de la versión anterior que se eliminaron en la versión actual
IMS	=	Nivel de madurez del sistema

2.10.1.6 Portabilidad

La capacidad de un software para ser transferido de un entorno a otro incluye varios aspectos importantes:

Adaptabilidad

Habilidad del software para ajustarse a diferentes entornos sin requerir modificaciones adicionales.

Facilidad de Instalación

Esfuerzo requerido para instalar el software en un entorno específico.

Conformidad

Verificación de que el software cumple con estándares o convenciones de portabilidad.

Capacidad de sustitución

Facilidad y esfuerzo necesarios para sustituir el software con otro producto que tenga funciones similares.

El sistema, desarrollado con CodeIgniter, puede implementarse en cualquier servidor con Apache y las herramientas necesarias.

Como la tecnología web es ejecuta fácilmente en cualquier dispositivo con conexión a Internet y un navegador.

2.11 ESTIMACIÓN DE COSTOS

El objetivo de la estimación de costo de software consiste en estimar el tamaño, el esfuerzo, la complejidad y el costo del proyecto de software para poder encontrar la mejor decisión de desarrollo y asegurar que el gasto se encuentre de acuerdo con lo presupuestado. Por todo lo anterior, la estimación de costos de software es importante para la planificación, programación, presupuesto y establecimiento del precio indicado al desarrollo del software (Alarcón, 2023, p. 14).

2.11.1 Cocomo II

Como se conoce, una de las tareas de mayor importancia en la planificación de proyectos de software es la estimación, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos.

Este trabajo describe un modelo de estimación, propuesto por Barry Boehm, llamado COCOMO II. Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica las economías

y deseconomías de escala producidas a medida que un proyecto de software incrementa su tamaño.(Gómez, 2024, p. 1)

Cocomo II posee tres modelos y se adaptan a las necesidades de los diferentes sectores, como al tipo de cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son.(*COCOMO II*, 2016b, p. 5)

2.11.1.1 Ecuaciones del Modelo COCOMO

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses/persona (PM) y representa los meses de trabajo de una persona full-time, requeridos para desarrollar el proyecto (Bustamante, 2021, p. 5).

Las ecuaciones que se utilizan en los tres modelos son:

(10) Ecuación de Esfuerzo

$$E = a * KLDC^{b} * FAE \qquad (10)$$

Donde

$\boldsymbol{E}$ = Es el esfuerzo requerido por el proyecto en persona-mes.

$\boldsymbol{KLDC}$ = Es la cantidad de líneas de código en miles

$\boldsymbol{FAE}$ = Factores de atributos de esfuerzo.

a = Coeficiente Cocomo II

b: = Coeficiente Cocomo II

(11) Ecuación Tiempo de Esfuerzo

$$Tdev = c(E)^{d} \qquad (11)$$

$\boldsymbol{E}$ = Es el esfuerzo requerido por el proyecto, en persona-mes.

$\boldsymbol{Tdev}$ = Es el esfuerzo requerido por el proyecto, en meses.

$\boldsymbol{c\ y\ d}$ = Constantes con valores definidos en una tabla

(12) Ecuación Número de Personas

$$P = \frac{E}{Tdev} \qquad (12)$$

Dónde:

$\boldsymbol{E}$ = Es el esfuerzo requerido por el proyecto, en persona-mes.

$\boldsymbol{Tdev}$ = Es el esfuerzo requerido por el proyecto, en meses.

$\boldsymbol{P}$ = Número de personas

A la vez, cada SubModelo también se divide en modos que representan el tipo de proyecto, y puede ser:

Modo Orgánico: Un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unas pocas líneas.

Modo Semilibre o Semiencajado: Corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

Modo Rígido o Empotrado: El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia.

2.11.1.2 Modelos de COCOMO II

- **Modelo Básico**

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes.

Tabla 2

Tabla de Valores de Modelo Básico

Modo	a	b	c	d
Orgánico	2.40	1.05	2.5	0.38
Semi acoplado	3.00	1.12	2.5	0.35
Empotrado	3.60	1.20	2.5	0.32

Estos valores son para las fórmulas:

(13) Ecuación Personas Por Mes Según Modelo

$$E = a * KLDC^{b} * FAE \qquad (13)$$

Donde:

a = Coeficiente Cocomo II

b = Coeficiente Cocomo II

$KLDC$ = Líneas de código

FAE = Factor de esfuerzo

(14) Ecuación Tiempo de Proyecto Según Modelo

$$Tdev = c(E)^{d} \qquad (14)$$

Donde:

c = Coeficiente Cocomo II

d = Coeficiente Cocomo II

E = Esfuerzo

$Tdev$ = Tiempo del proyecto

(15) Ecuación Costo Personal

$$P = \frac{E}{Tdev} \qquad (15)$$

Donde:

E = Esfuerzo

$Tdev$ = Tiempo requerido

P = Costo personal

(16) Ecuación de Costo Total del Proyecto

$$Csof = Smes * P * Tdev \qquad (16)$$

Donde:

$Smes$ = Sueldo mes

P = Número de personas

$Tdev$ = Tiempo requerido

$Csof$ = Costo total del proyecto

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno

2.11.1.3 Ventajas de COCOMO II

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura. (Alban&Galarza, 2017, p. 10)

Las ventajas del modelo de estimación de costos COCOMO II son:

- Es fácil de realizar y de interpretar
- Tiene pocas variables
- Se acerca a la realidad en la mayoría de los casos

Desventajas del Modelo Cocomo II

- No saca resultados fiables en proyectos demasiado pequeños
- La elección de las variables es muy subjetiva depende de la persona que realiza el estudio.

2.12 SEGURIDAD

La información es un activo valioso que puede hacer crecer o destruir a su organización. Cuando se gestiona apropiadamente, le permite operar con confianza. La gestión de seguridad de la información le da la libertad de crecer, innovar y ampliar su base de clientes con el conocimiento que toda su información confidencial permanecerá de esa manera.

Seguridad es la capacidad de dar respuesta efectiva a riesgos, amenazas o vulnerabilidades y estar preparados para prevenirlos, contenerlos y enfrentarlos. El término, identifica a una situación de ausencia o disminución de riesgos para un determinado entorno social y natural. (entiéndase riesgo como el daño o pérdida que pudiera causar un particular evento, encontrándose en función de la intensidad de la amenaza y del grado de vulnerabilidad que experimente la sociedad en el momento determinado; esta vulnerabilidad podrá disminuir si el Estado o la comunidad poseen determinadas capacidades para enfrentarla).

2.12.1 ISO 27001

La norma ISO 27001 es un estándar internacional que establece un marco para la gestión para la seguridad de la información en las organizaciones. Según la Organización Internacional de Normalización (ISO), esta norma "define requisitos para establecer, implementar, mantener y mejorar continuamente un sistema de gestión de la seguridad de la información Dentro del contexto de la organización" (ISO, 2013, p. 1).

Su objetivo principal es salvaguardar la confidencialidad, integridad y disponibilidad de la información a través de la aplicación de controles y procesos de gestión de riesgos.

Según Calder & Watkins (2015), una de las características fundamental de la ISO

27001 es su enfoque en el ciclo de mejora continua PDCA (Planificar, Hacer, Verificar, Actuar). Este ciclo permite a las organizaciones evaluar y mejorar continuamente su sistema de gestión de seguridad de la información (SGSI) para adaptarse a los cambios en los requisitos y a nuevas amenazas. El objetivo esencial de la norma ISO 27001 es proporcionar un marco de gestión para establecer, implementar, operar, monitorear, revisar, mantener y mejorar un SGSI dentro de una organización, centrado en preservar tres aspectos clave de la seguridad de la información: confidencialidad, integridad y disponibilidad.

- Confidencialidad: La ISO 27001 establece controles y medidas para asegurar que la información no sea accedida, divulgada o revelada a personas, entidades o procesos no autorizados. "La confidencialidad se refiere a la protección de la información confidencial contra su divulgación no autorizada" (Calder & Watkins, 2015, p. 18).
- Integridad: La norma busca garantizar que la información no sea modificada, alterada o destruida de manera no autorizada, ya sea de forma accidental o

deliberada. "La integridad se refiere a la protección de la exactitud y completitud de la información y los métodos de procesamiento" (Diesterer, 2013, p. 96).

- Disponibilidad: La ISO 27001 resalta la importancia de garantizar que la información y los sistemas asociados estén accesibles para las personas, procesos y entidades autorizadas cuando sea necesario. "La disponibilidad se refiere a la protección contra la negación de acceso a la información o recursos por parte de usuarios autorizados" (Calder & Watkins, 2015, p. 19).

Al preservar estos tres pilares esenciales de la seguridad de la información, la ISO 27001 permite a las organizaciones proteger sus activos de información críticos, mantener la confianza de los clientes y socios comerciales, cumplir con las normativas legales, y reducir los riesgos asociados con posibles brechas de seguridad.

2.12.2 Seguridad Física

La seguridad física se refiere a la identificación y análisis de las amenazas y riesgos que enfrentan o pueden llegar a enfrentar instalaciones, bienes y procesos a fin de implementar planes y sistemas tendientes a prevenir, dificultar o limitar los resultados de las posibles acciones dañinas contra la seguridad de ellas y ellos.

Seguridad física podemos entender todas aquellas mecanismos - generalmente de prevención y detección, destinados a proteger físicamente cualquier recurso del sistema; estos recursos son desde un simple teclado hasta una cinta de backup con toda la información que hay en el sistema, pasando por la propia CPU de la máquina.

2.12.3 Seguridad Lógica

La seguridad lógica se define como un conjunto de procesos destinados a garantizar la seguridad en el uso de los sistemas y los programas destinados a gestionar los datos y procesos de cualquier empresa. Del mismo modo, también hace referencia al acceso

autorizado y ordenado de los usuarios a la información almacenada por la compañía. (Osmosis, 2024, p. 10).

Por lo tanto, la seguridad lógica, en términos de ciberseguridad, engloba todas las medidas de administración necesarias con el objetivo de reducir al máximo los riesgos asociados a las actividades cotidianas realizadas usando tecnología de la información.

2.12.4 Seguridad de la Base de Datos

La seguridad de las bases de datos son los procesos, herramientas y controles que aseguran y protegen las bases de datos contra amenazas accidentales e intencionadas. El objetivo de la seguridad de las bases de datos es proteger los datos confidenciales y mantener la confidencialidad, disponibilidad e integridad de la base de datos. (Osmosis, 2024, p. 10).

- **Confidencialidad** significa que los datos, objetos y recursos están protegidos contra la visualización y otros accesos no autorizados.
- **Integridad** significa que los datos están protegidos de cambios no autorizados para garantizar que son fiables y correctos.
- **Disponibilidad** significa que los usuarios autorizados tienen acceso a los sistemas y recursos que necesitan.

2.12.5 Seguridad del Framework CodeIgniter

CodeIgniter es un framework de PHP que ofrece herramientas para hacer que las aplicaciones sean seguras, pero es responsabilidad del desarrollador usarlas correctamente.

Para mejorar la seguridad de las aplicaciones web PHP con CodeIgniter, se pueden implementar otras medidas, como:

- Usar HTTPS y SSL

- Realizar copias de seguridad y actualizaciones con regularidad
- Implementar la protección CSRF
- Ubicar las carpetas del sistema y de la aplicación por encima de la raíz web

Restricciones en las URL

CodeIgniter limita los caracteres que se pueden usar en las cadenas de URL para evitar que se pasen datos maliciosos a la aplicación. Las URL solo pueden contener texto alfanumérico, tilde, signo de porcentaje, punto, dos puntos, guion bajo y guion.

Función xss_clean()

Esta función filtra posibles ataques XSS, pero solo elimina los atributos de las etiquetas HTML que se cerraron correctamente.

2.13 PRUEBAS DE SOFTWARE

Las pruebas de software son el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe hacer. Entre los beneficios de unas buenas pruebas se incluyen la prevención de errores y la mejora del rendimiento.

2.13.1 Pruebas de Caja Blanca

El objetivo de las pruebas de caja blanca es realizar pruebas que cubran la estructura interna de un sistema, por estructura interna nos referimos a código, arquitectura y flujos de trabajo. En las pruebas de caja blanca, el código es visible para los testers, por lo que también se denominan pruebas de caja transparente o pruebas de caja abierta.

La profundidad de las pruebas de caja blanca se puede medir a través de la cobertura estructural. La cobertura estructural es la medida que se utiliza para saber si todas las rutas dentro de un módulo se han ejecutado al menos una vez.

2.13.2 Pruebas de Caja Negra

Las pruebas de caja negra, una forma de prueba que se realiza sin conocimiento de los componentes internos de un sistema, se pueden realizar para evaluar la funcionalidad, la seguridad, el rendimiento y otros aspectos de una aplicación. Análisis dinámico de código es un ejemplo de pruebas automatizadas de seguridad de caja negra. Los evaluadores de caja negra definen casos de prueba e interactúan con el software como lo haría un usuario para validar que hace lo que debería, como debería (Zaptest, 2024, p. 7).

2.13.3 Pruebas de Estrés

Una prueba de estrés (stress) consiste en probar los límites que un sistema puede soportar. En este tipo de pruebas se suele enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación.

Por otro lado, las pruebas de estrés (stress testing) son realizadas sobrecargando un sistema más allá de sus especificaciones, para verificar cómo y cuándo fallará. Dentro de informática podemos colocar una gran carga en la base de datos, entradas (peticiones) continuas al sistema o almacenar información más allá de la capacidad de memoria del sistema. (Micro Focus, 2022, p. 7).

2.13.4 Pruebas de Accesibilidad

- **Axe (Axe DevTools)**

Axe DevTools es un conjunto de herramientas para probar la accesibilidad de sitios web y aplicaciones móviles. Su objetivo es ayudar a los equipos de desarrollo a: Identificar problemas de accesibilidad de manera más rápida, Reducir riesgos y costos, Confiar en los resultados.

Axe DevTools incluye las siguientes características:

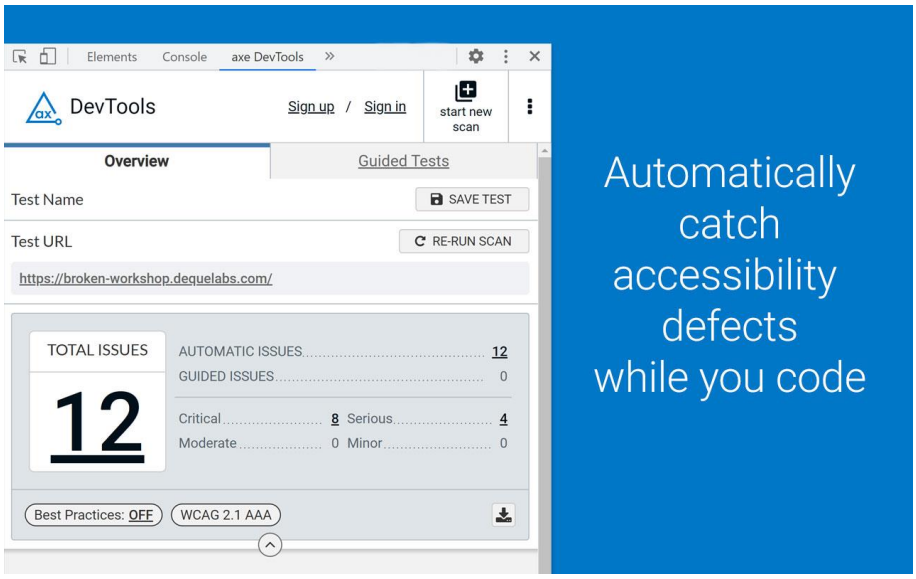
- Extensiones para navegadores de Axe, disponibles para Chrome, Firefox y Edge

- Axe Auditor, una herramienta de prueba de accesibilidad manual paso a paso
- Axe Monitor, que analiza, monitorea e informa sobre el estado de accesibilidad de un sitio

Axe DevTools está impulsado por axe-core, el motor de prueba de accesibilidad desarrollado por Deque.

Figura 17

Funciones del Axe DevTools



- **Colour Contrasr Analyser (CCA)**

El Colour Contrast Analyser (CCA) es una herramienta gratuita que permite comprobar el contraste entre dos colores, como el texto y el fondo de una página web. Se utiliza para optimizar el contenido para personas con baja visión o ceguera al color.

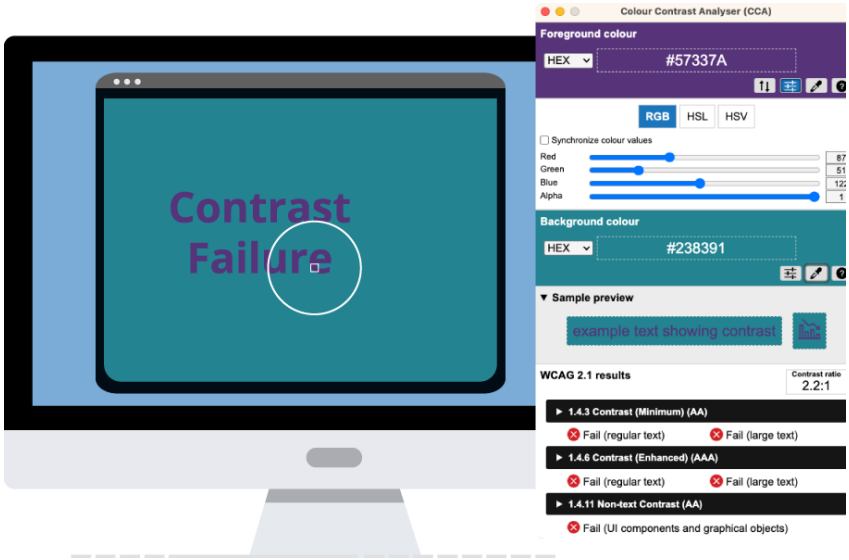
Algunas de las características del CCA son:

- Calcula la ratio de contraste utilizando el algoritmo de las WCAG 2.1
- Muestra si el contraste de color cumple o no con los criterios
- Simula condiciones visuales como la ceguera al color dicromático y las cataratas
- Incluye indicadores de cumplimiento para WCAG 2.1

- Cuenta con modo oscuro
- Permite seleccionar los colores de varias formas
- Soporta la transparencia alfa en los colores de primer plano

Figura 18

Funciones de Colour Contrast Analyser



- **Pruebas de Accesibilidad con WAVE Accessibility Evaluation Tool**

Las herramientas de evaluación de accesibilidad juegan un papel muy importante al proporcionar a los testers, desarrolladores y diseñadores web una forma efectiva y rápida de verificar la accesibilidad de sus aplicaciones.

Una de las herramientas más populares y utilizadas en este campo es WAVE (Web Accessibility Evaluation Tool), desarrollada y mantenida por la organización WebAIM.

Herramientas WAVE

WAVE es una extensión de navegador que proporciona informes detallados sobre la accesibilidad de un sitio web.

Esta herramienta evalúa diversos aspectos de la accesibilidad web, como la estructura semántica del contenido, el contraste de color, la navegación con teclado, la disponibilidad de

alternativas textuales para elementos multimedia, entre otros. Además, proporciona recomendaciones y sugerencias para corregir los problemas de accesibilidad detectados.

La herramienta puede comparar el contenido digital con las pautas de accesibilidad establecidas en la WCAG (Web Content Accessibility Guidelines). La WCAG es un conjunto de estándares y recomendaciones internacionales desarrollados por el World Wide Web Consortium (W3C) para garantizar la accesibilidad web.

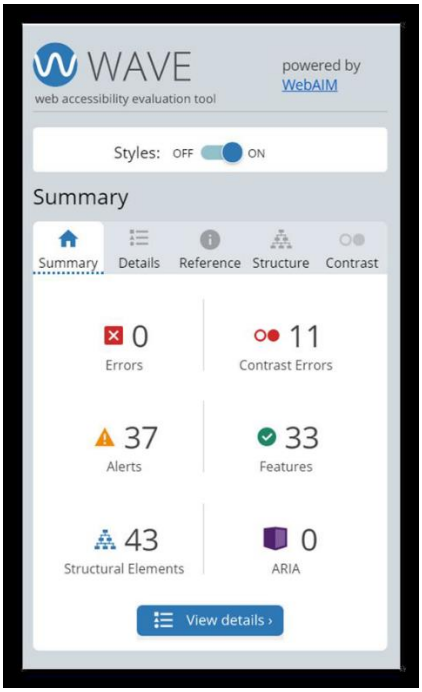
La WCAG se divide en diferentes niveles de conformidad: A (el nivel más básico), AA (nivel intermedio) y AAA (nivel más alto). Estas pautas establecen una serie de criterios y recomendaciones específicas para que el contenido digital sea accesible para una amplia gama de usuarios, incluyendo situaciones de discapacidad visuales, auditivas, cognitivas, motoras, entre otras.

Actualmente WAVE es compatible con las pautas de accesibilidad web WCAG 2.1. Esta pauta se basa en la versión WCAG 2.0 y amplía sus criterios de accesibilidad para abordar más casos y situaciones.

WAVE identifica una variedad de problemas e inconvenientes de accesibilidad dentro de una aplicación o sitio web. Estos pueden ser , por ejemplo, falta de textos alternativos en imágenes, contraste deficiente entre color de texto y fondo, etiquetas ausentes en elementos interactivos, estructura de encabezados incorrecta, enlaces ambiguos o poco descriptivos. A su vez, los informes de WAVE proporcionan todo el detalle específico sobre cada problema, su importancia e impacto en la accesibilidad y sugerencias de cómo corregirlos.

Figura 19

Prueba de Accesibilidad WAVE



2.13.5 Pruebas de Unitarias

Las pruebas unitarias son un recurso viable para estos efectos, al aplicarse con el propósito de identificar errores, vulnerabilidades o funciones defectuosas. A continuación, explicamos a detalle qué son, sus características y las ventajas de su implementación.

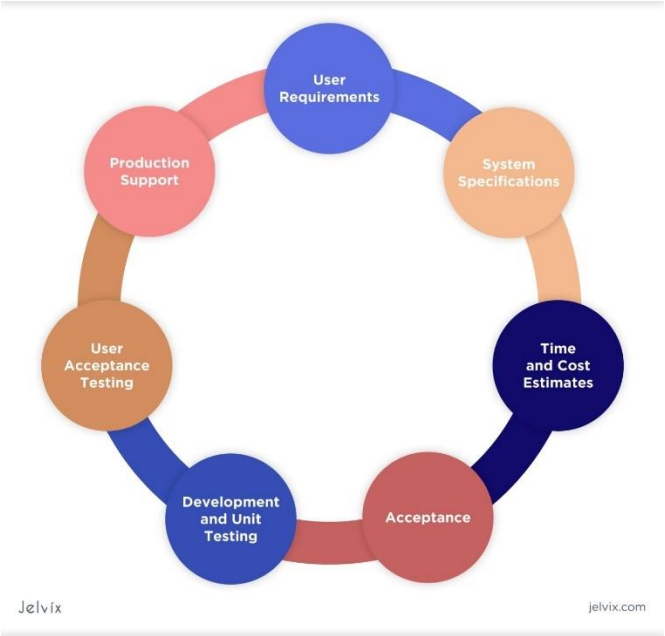
Las pruebas de software son un recurso fundamental para los desarrolladores, quienes recurren a ellas para asegurar la calidad, funcionamiento y rendimiento de un producto informático.

De acuerdo con un informe del Comité Internacional de Certificaciones de Pruebas de Software (ISTQB, por sus siglas en inglés), la detección de bugs, el correcto desempeño de un sistema y el aumento de la confianza con los clientes son los principales beneficios que ofrecen las actividades de testing para una organización.

El 28% de los encuestados aseguró realizar una inspección de código fuente antes de ejecutar una prueba de software, lo cual implica la revisión del lenguaje de programación mediante prácticas específicas.

Figura 20

Funciones de la Pruebas Unitarias



CAPÍTULO III

MARCO APLICATIVO

3 CAPÍTULO III - MARCO APLICATIVO

3.1 INTRODUCCIÓN

En este capítulo se hará un análisis de los requerimientos funcionales y no funcionales, se aplicará la metodología UWE, con todas las fases y diagramas correspondientes, con el fin de identificar los requerimientos que se precisan en el sistema, también se identificarán los principales actores.

3.2 DESARROLLO DE LA METODOLOGÍA

En la aplicación de la metodología obtendremos los requerimientos funcionales y no funcionales, se debe obtener la información de forma clara y precisa, a través de un análisis profundo, requisitos que necesita el sistema, esto con el fin de tener el correcto desarrollo de este capítulo, aplicando las herramientas descritas.

Tabla 3

Herramientas para la Obtención de Requerimientos

HERRAMIENTA	CARACTERÍSTICAS
Observación	Se hizo la observación, en las oficinas de la Unidad correspondiente, viendo los pasos a seguir durante el registro.
Documentación	Se revisaron los documentos que se manejan dentro del proceso, el tipo de presentación, las normas que se debe seguir y los formatos para la realización de los reportes.
Entrevistas	Se realizó entrevistas con el personal involucrado dentro del proceso, quienes son: Personal administrativo Personal técnico

3.2.1 Requerimientos Funcionales

Se describen los requisitos mínimos que necesita el sistema para los diferentes procesos, a continuación, en la siguiente tabla se da los detalles de los requerimientos funcionales.

Tabla 4

Requerimientos Funcionales

ROL	DESCRIPCIÓN	FUNCIÓN
F-1	Acceder al sistema	Accede al sistema 3 tipos de usuarios (Administrador, responsable jefe, Terapeuta), mediante su usuario y contraseña
F-2	Gestionar Usuarios	Permite registrar, editar, listar, cambio de contraseña, dar de baja a usuarios
F-3	Administración del software	Permite al usuario con el perfil de Administrador o director de la Unidad, tener acceso total a los módulos y operaciones de la gestión de la información: creación, listado, actualización, inactivación de los recursos del sistema.
F-4	Gestionar los casos	Permite registrar, editar, listar todos los propietarios y sus mascotas en la unidad correspondiente.
F-5	Gestionar registros de las victimas	Permite registrar, editar, listar a las víctimas, que se realizará en la unidad.
F-6	Gestionar registro de los casos	Permite registrar los casos, en qué lugar de la urbe alteña, descripción y otros.
F-7	Administrar roles y permisos	Administra los roles y permisos que se otorga a los usuarios.
F-8	Generar reportes de usuarios.	Permite generar un reporte pdf de los usuarios de acuerdo al rol especificado.

F-9	Gestionar datos estadísticos	Permite generar un reporte de los datos estadísticos de acuerdo a lo especificado.
F-10	Gestión de sesiones	Permite generar las sesiones de los pacientes de acuerdo a lo especificado.
F-11	Gestión de Terapeutas	Permite registrar los casos, de sus pacientes
F-12	Gestión de Documentos	Permite generar todos los documentos de los pacientes de acuerdo a lo especificado.

3.2.2 Requerimientos No Funcionales

Los Requerimientos no funcionales (RFN), se describen cualidades y características del sistema que el usuario puede visualizar. A continuación, se muestra la siguiente tabla:

Tabla 5

Requerimientos No Funcionales

ROL	FUNCIÓN
RNF-1	El Software debe ser compatible en todos los navegadores, debe funcionar correctamente en: Google Chrome, Mozilla Firefox, IE/Edge y otros.
RNF-2	La interfaz del sistema debe ser intuitiva y fácil de usar, asegurando una experiencia óptima para el usuario y facilitando su interacción de manera eficiente y sin complicaciones.
RNF-3	El sistema debe incluir un menú principal con opciones desplegables o submenús para facilitar la navegación del usuario.
RNF-4	El mantenimiento y soporte periódico del sistema garantizan un rendimiento óptimo, prolongando su vida útil y evitando posibles fallos, lo que mejora su eficiencia y funcionalidad general.
RNF-5	El sistema deberá responde a todas las solicitudes del usuario, con toda disponibilidad.
RNF-6	El sistema debe proteger las contraseñas de los usuarios mediante un proceso de encriptación, asegurando que los datos confidenciales

	permanezcan seguros.
RNF-7	La sesión del usuario estará activa durante un tiempo específico, después del cual se cerrará automáticamente para garantizar mayor seguridad y protección de la información.
RNF-8	El sistema debe ser escalable a agregar nuevos módulos y funciones a futuro.
RNF-9	Los cambios realizados en la base de datos deben estar actualizados para todos los usuarios según los permisos definidos
RNF-10	Conexión a Internet
RNF-11	El sistema debe incluir manuales de usuario bien estructurados.
RNF-12	La sesión de usuario tendrá un tiempo de actividad definido.
RNF-13	Solo el administrador de acceso a datos podrá modificar los permisos de acceso al sistema.
RNF-14	El sistema necesita un menú lateral para facilitar navegación.
RNF-15	Conexión a Internet

3.2.3 Descripción de Actores

De acuerdo a los principales actores que interactuaran con el sistema, se debe identificar a estos, de acuerdo a los roles y permisos que se registren en este proyecto de Sistema de Información Web, para la Unidad del Centro de Atención Terapéutico

Tabla 6

Descripción de Actores

DESCRIPCIÓN	RESPONSABILIDADES
Administrador, responsable del Área del Centro de Atención Terapéutico	Administra a los usuarios en el sistema web, asignando roles y permisos adecuados. Además, gestiona las configuraciones del sistema para garantizar su correcto funcionamiento y optimización.

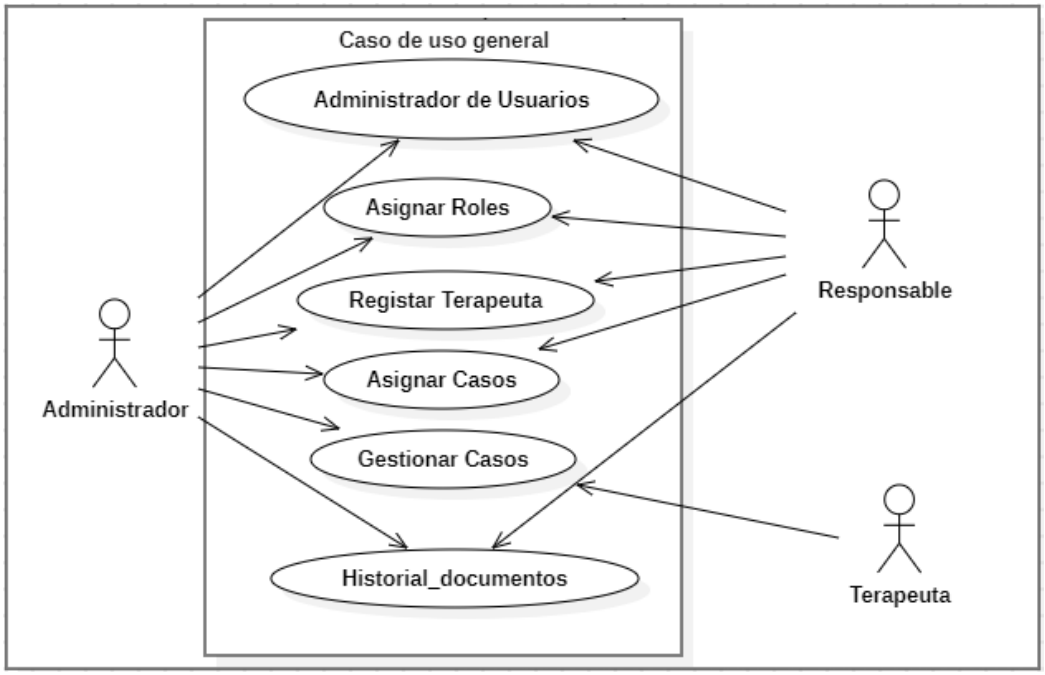
Personal jefe del centro	Es responsable de registrar los casos, asegurando un seguimiento adecuado para cada caso.
Terapeuta	Responsable de registrar y mantener un seguimiento actualizado de los casos de las víctimas, garantizando su salud y cumplimiento de los requisitos.

3.2.4 Diagramas de Caso de Uso General

El diagrama de caso de uso general del sistema es una parte esencial del análisis y diseño que proporciona una visión global de las interacciones entre usuarios y sistema, simplificando la comprensión de las funciones, esta representación visual estructurada de las funciones disponibles no solo facilitara la comunicación entre los desarrolladores y las partes interesadas, sino que también guía la planificación y el diseño detallado.

Figura 21

Caso de Uso General del Sistema



3.2.5 Diagrama de Caso de Uso Específico

Diagrama de Caso de Uso: Administración de Usuarios

Figura 22

Gestión de Usuarios

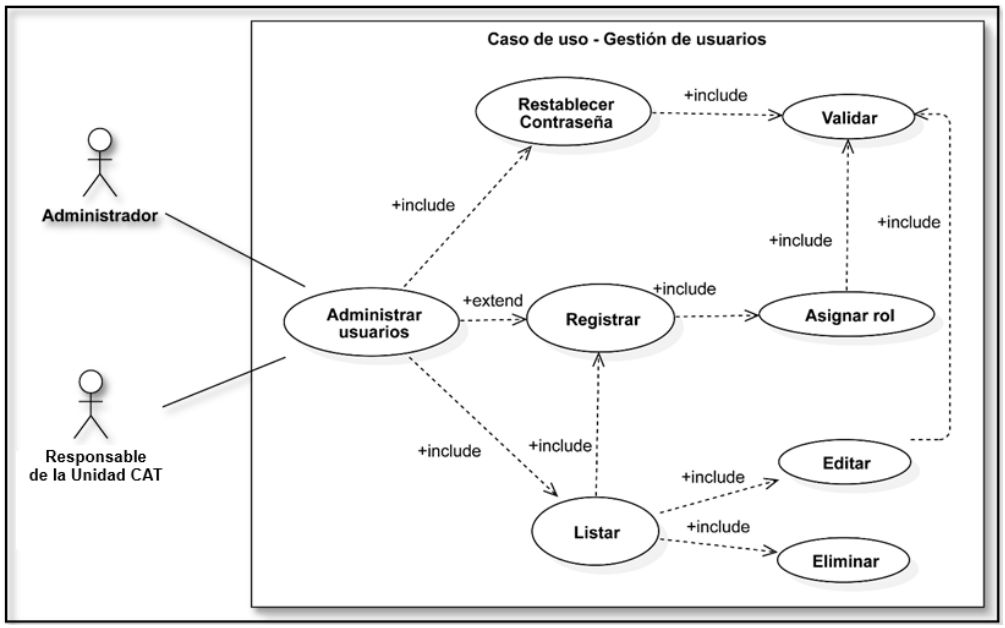


Tabla 7

Detalle del Caso de Uso – Gestión de Usuarios.

CASO DE USO	GESTIÓN DE USUARIOS
IDENTIFICADOR	CU – 1
TIPO	Primario
ACTORES	Administrador, responsable de la Unidad
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al panel de administración.
DESCRIPCIÓN	El proceso comienza cuando el usuario selecciona la opción de usuarios, visualiza la lista de usuarios y realiza acciones como crear, modificar y/o activar/desactivar a los usuarios, seguidamente el sistema debe verificar la información ingresada.

3.2.6 Diagrama de Caso de Uso: Gestión de Roles y Permisos

Figura 23

Gestión de Roles y Permisos

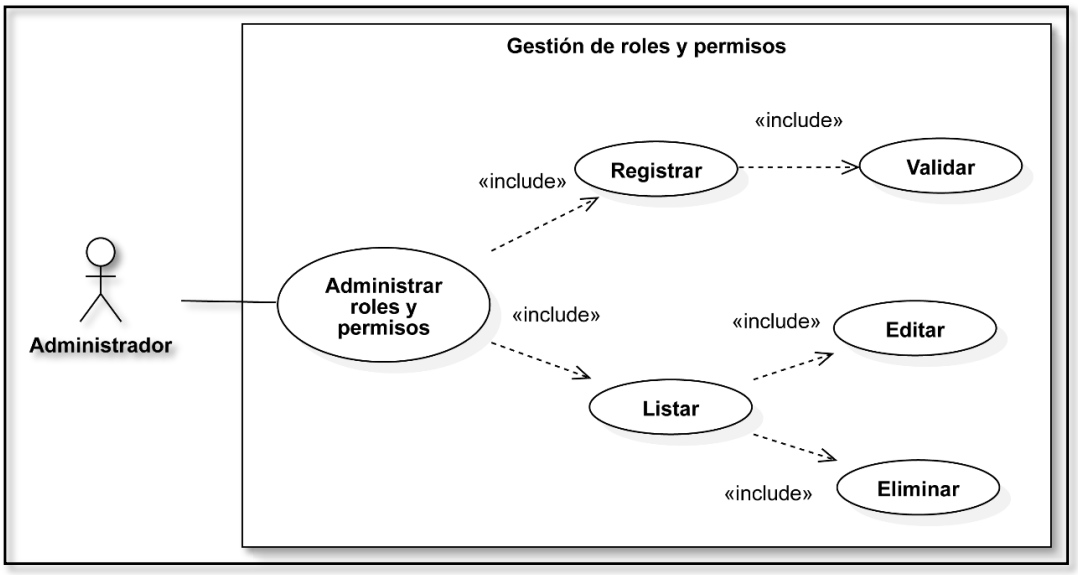


Tabla 8

Detalle del Caso de Uso – Gestión de Roles y Permisos

CASO DE USO	GESTIÓN DE ROLES Y PERMISOS
IDENTIFICADOR	CU – 2
TIPO	Primario
ACTORES	Administrador, responsable de la Unidad
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al panel de administración.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona la opción de roles y permisos, visualiza la lista de roles y ejecuta acciones como crear nuevos roles, asignando los permisos correspondientes, modificar los roles existentes, seguidamente el sistema verifica la información ingresada.

Figura 24

Gestión de Tipologías

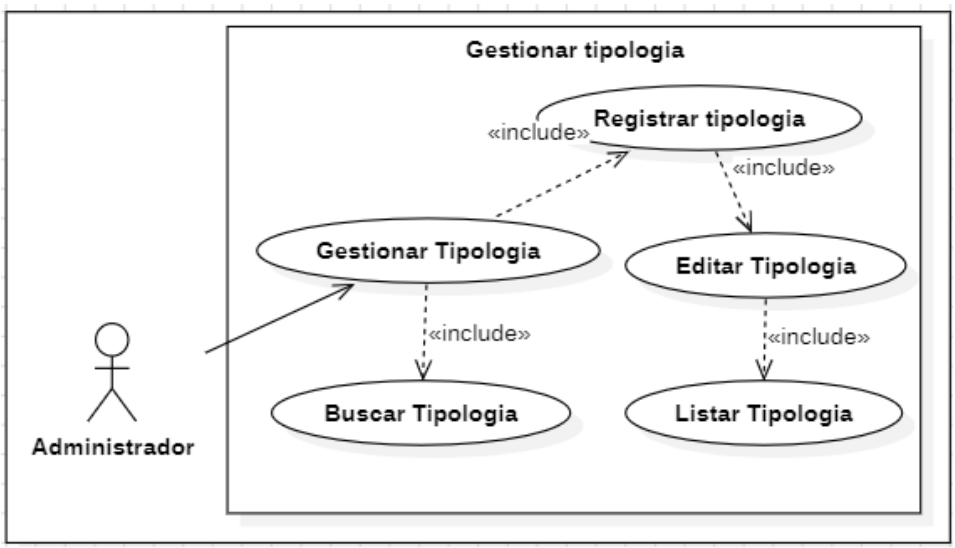


Tabla 9

Detalle del Casos de Uso – Gestión de Tipologías

CASO DE USO	GESTIÓN DE TIPOLOGÍA
IDENTIFICADOR	CU – 3
TIPO	Primario
ACTORES	Responsable del Administrador
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como administrador.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona, visualiza un formulario para la respectiva, se enlista este proceso.

Figura 25

Gestión de Institución

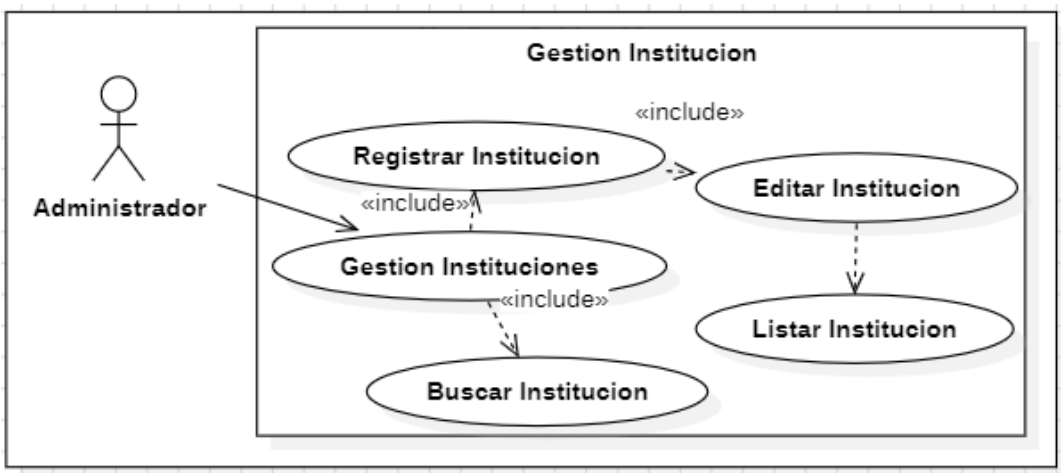


Tabla 10

Detalle de Casos de Uso – Gestión Institución

CASO DE USO	GESTIÓN DE INSTITUCIÓN
IDENTIFICADOR	CU – 4
TIPO	Primario
ACTORES	Responsable del Administrador
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como administrador.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona, visualiza un formulario para la respectiva, se enlista este proceso.

Figura 26

Gestión de Asignar Casos

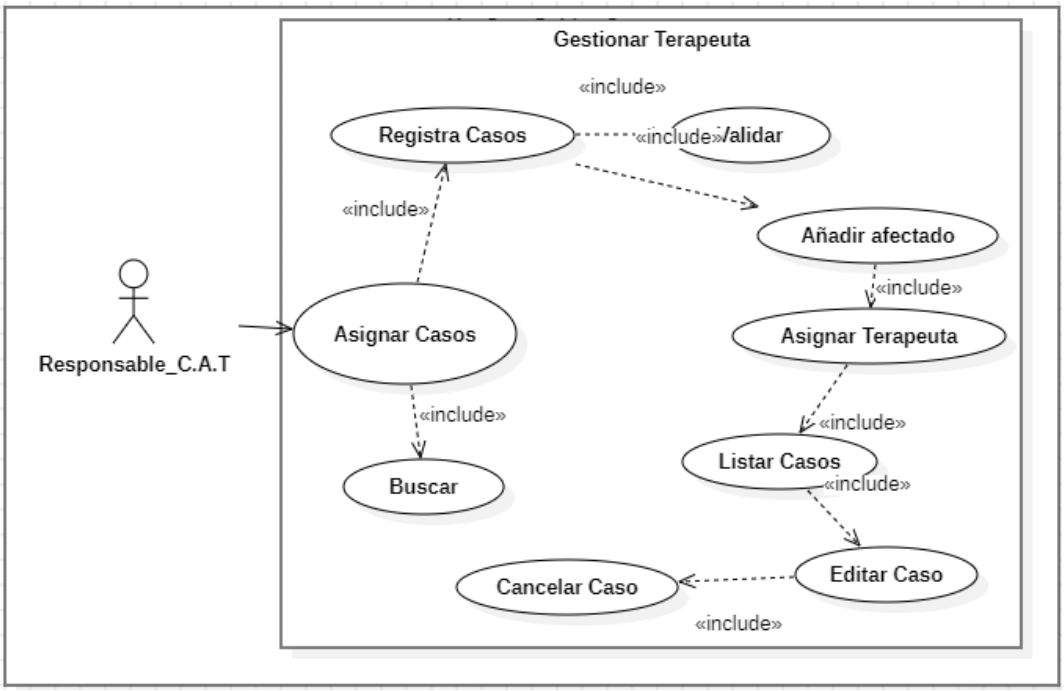


Tabla 11

Detalle del Casos de Uso – Gestión de Asignar Casos

CASO DE USO	GESTIÓN DE ASIGNAR CASOS
IDENTIFICADOR	CU – 5
TIPO	Primario
ACTORES	Administrador, responsable del centro de atención terapéutico
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al panel de administración.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona la opción asignar casos, visualiza un formulario para la respectiva, este se enlista la cual este proceso y ejecuta acciones como crear, modificar y eliminar.

Diagrama de Caso de Uso: Gestionar Terapeuta

Figura 27

Gestión de Terapeuta

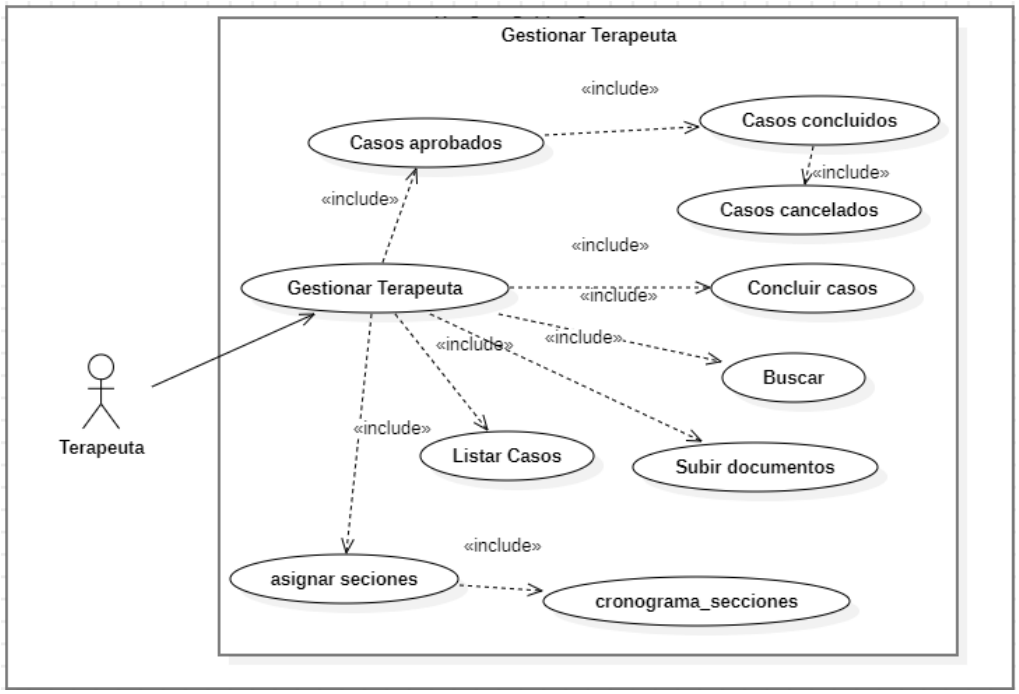


Tabla 12

Detalle del Casos de Uso – Gestión de Terapeuta

CASO DE USO	GESTIÓN DE TERAPEUTA
IDENTIFICADOR	CU – 6
TIPO	Primario
ACTORES	Personal operativo del centro
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al panel de administración.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona la opción de gestionar terapeutas, visualiza la lista de pacientes y ejecuta acciones como crear nuevas sesiones, modificar seguidamente el sistema verifica la información ingresada.

Figura 28

Gestión Pacientes

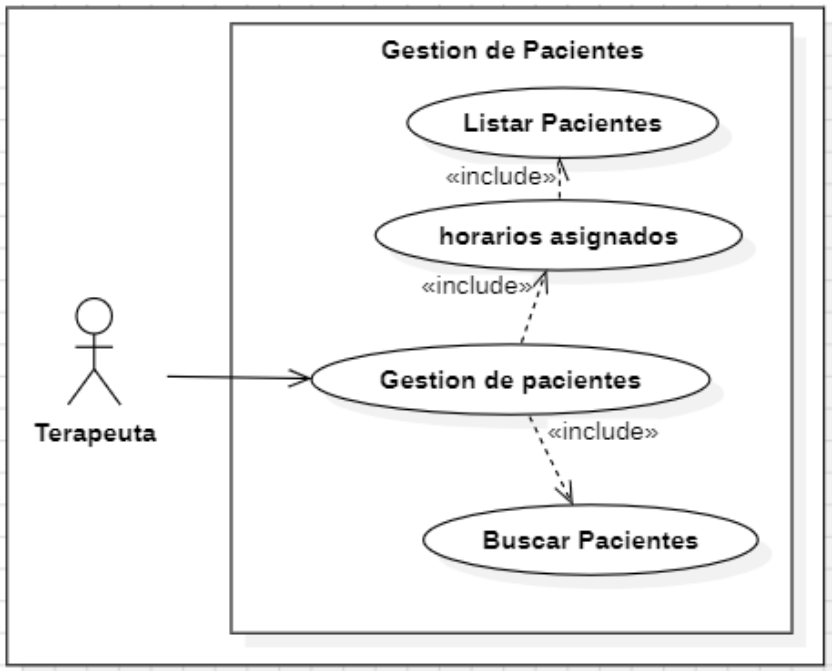


Tabla 13

Detalle de Casos de Uso - Gestión de Pacientes

CASO DE USO	GESTIÓN DE PACIENTE
IDENTIFICADOR	CU – 7
TIPO	Primario
ACTORES	Responsable del Terapeuta
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como terapeuta.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona, visualiza un formulario para la respectiva, se enlista este proceso.

Figura 29

Gestión de Control de Pacientes

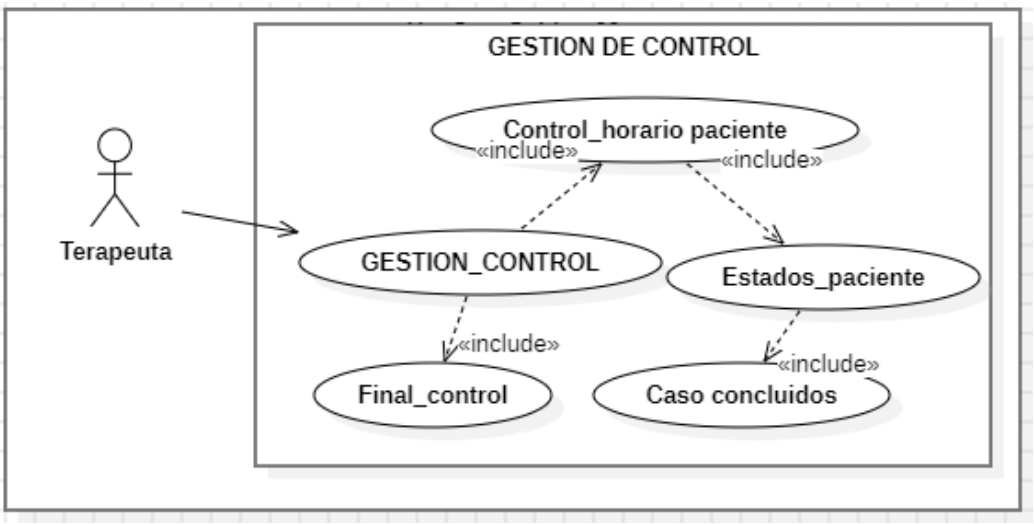


Tabla 14

Detalle de Casos de Uso - Gestión Control

CASO DE USO	GESTIÓN DE CONTROL DE PACIENTES
IDENTIFICADOR	CU – 8
TIPO	Primario
ACTORES	Responsable del Terapeuta
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como terapeuta.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona, visualiza un formulario para la respectiva, se enlista este proceso.

Figura 30

Gestión de Sesiones

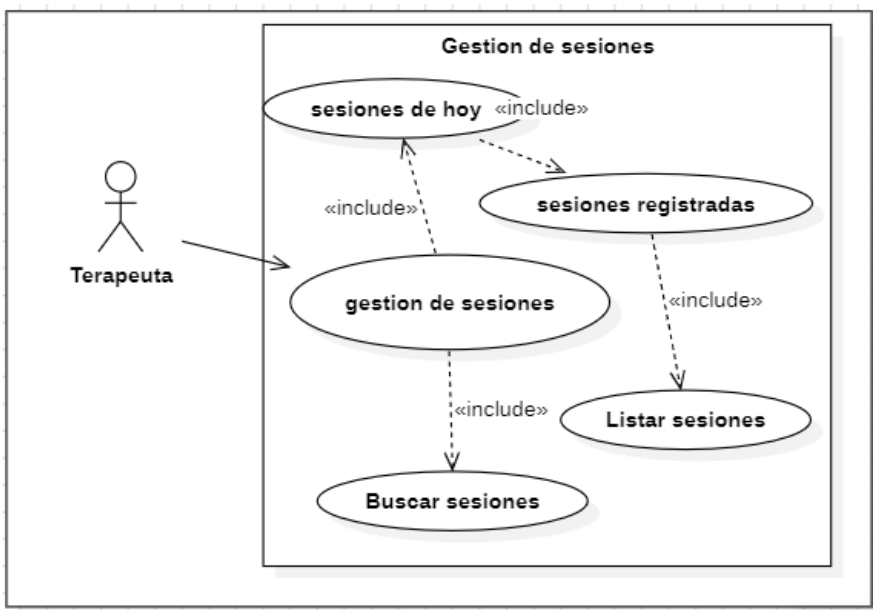


Tabla 15

Detalle del Casos de Uso – Gestión de Sesiones

CASO DE USO	GESTIÓN DE SESIONES
IDENTIFICADOR	CU – 9
TIPO	Primario
ACTORES	Responsable del terapeuta
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como terapéutico.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona la opción sesiones, visualiza un formulario para la respectiva, se enlista este proceso.

3.2.7 Diagrama de Caso de Uso: Gestión historial de documentos

Figura 31

Gestión Historial de Documentos

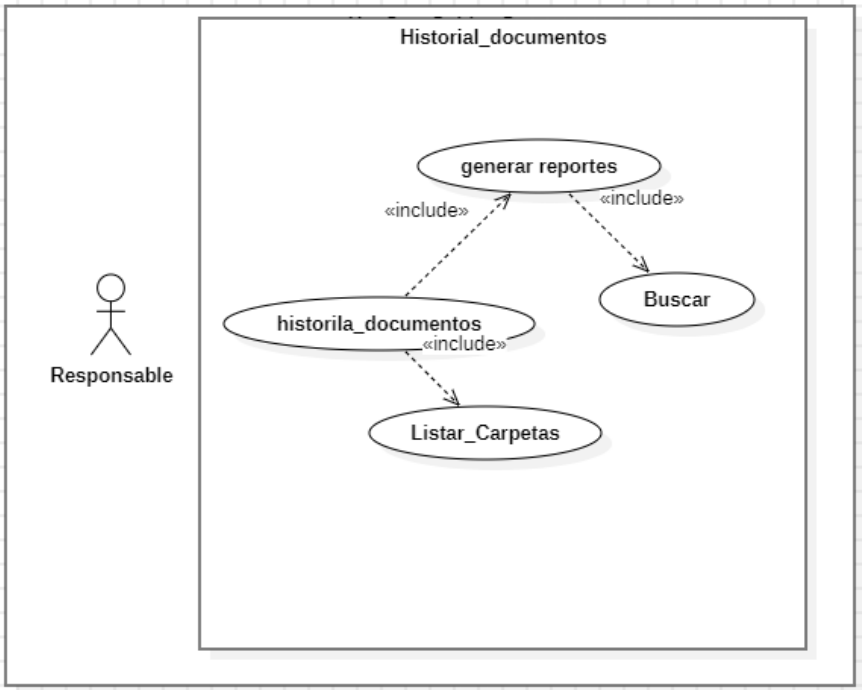


Tabla 16

Detalle del Casos de Uso – Gestión Historial de Documentos

CASO DE USO	GESTIÓN HISTORIAL DE DOCUMENTOS
IDENTIFICADOR	CU – 10
TIPO	Primario
ACTORES	Responsable del terapeuta
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al panel de administración.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona la opción documentos, visualiza un formulario para la respectiva, se enlista este proceso.

Figura 32

Gestión de Seguimiento

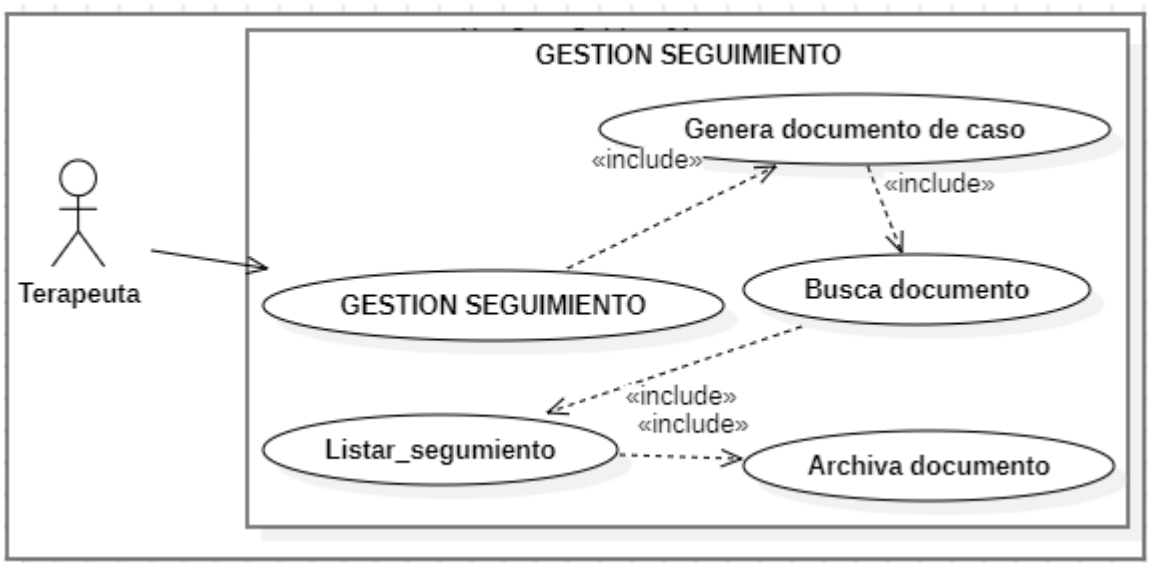


Tabla 17

Detalle de Casos de Uso - Gestión de Seguimiento

CASO DE USO	GESTIÓN DE SEGUIMIENTO
IDENTIFICADOR	CU – 11
TIPO	Primario
ACTORES	Responsable del Terapeuta
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como terapeuta.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona, visualiza un formulario para la respectiva, se enlista este proceso.

Figura 33

Gestión Reportes

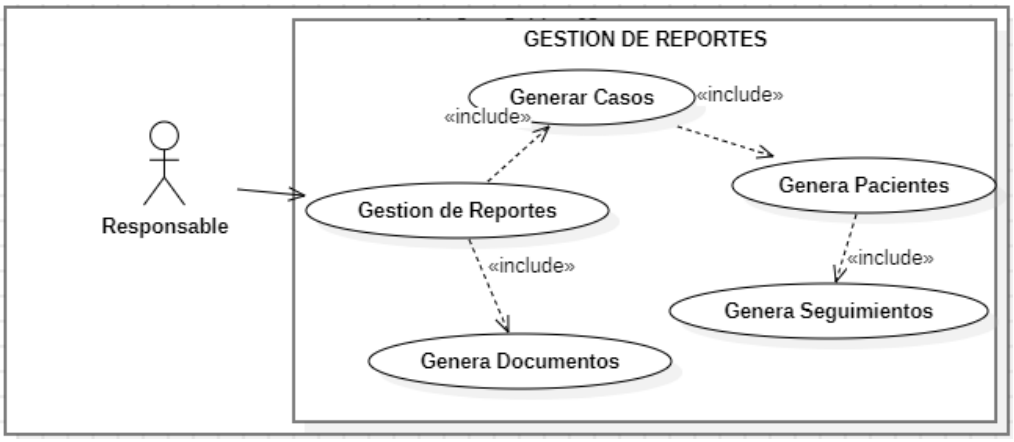


Tabla 18

Detalle de Casos de Uso - Gestión de Reportes

CASO DE USO	GESTIÓN DE REPORTES
IDENTIFICADOR	CU – 12
TIPO	Primario
ACTORES	Responsable del responsable
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como terapeuta.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona, visualiza el reporte para la respectiva, se enlista este proceso.

Figura 34

Gestión de Datos Estadísticos

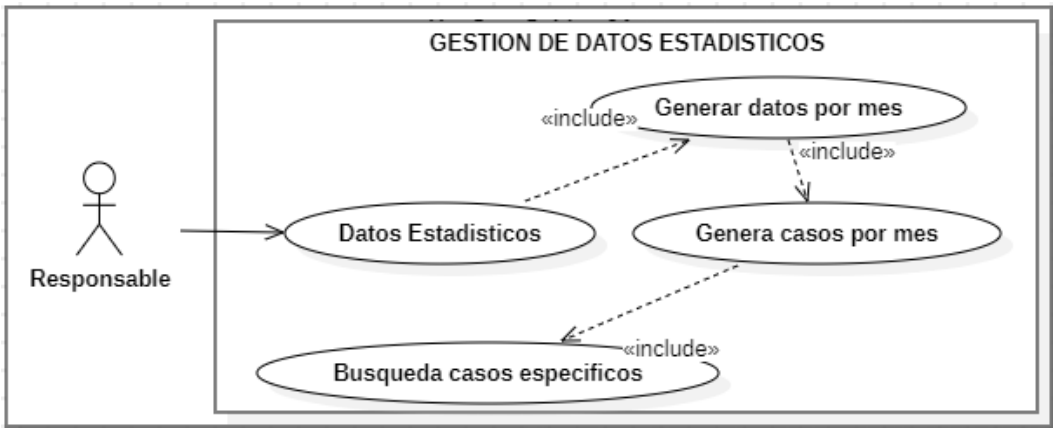


Tabla 19

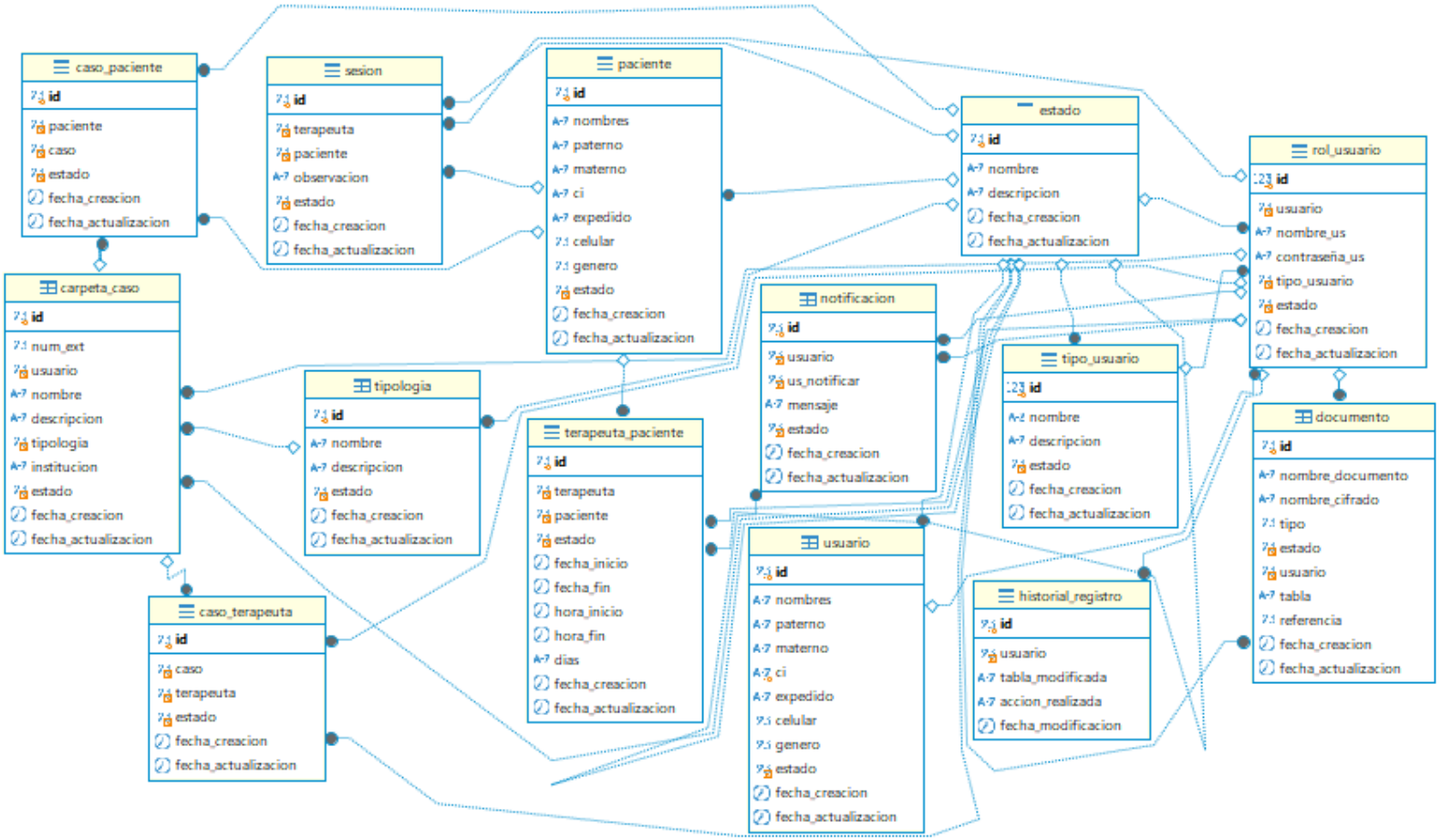
Detalle de Casos de Uso - Gestión de Datos Estadísticos

CASO DE USO	GESTIÓN DE DATOS ESTADÍSTICOS
IDENTIFICADOR	CU – 13
TIPO	Primario
ACTORES	Responsable del responsable
REQUISITOS PREVIOS	El usuario debe registrarse, autenticarse e ingresar al sistema como terapeuta.
DESCRIPCIÓN	El proceso inicia cuando el usuario selecciona, visualiza los datos estadísticos para la respectiva, se enlista este proceso.

3.2.8 Modelo Conceptual

Figura 35

Diseño de la Base de Datos Relacional



3.2.9 Modelo de Presentación

En esta fase se presenta el diseño del sistema en Balsamiq mostrando la peticiones y módulos del sistema.

Figura 36

Modelo de Presentación – Login

Figura 37

Modelo de Presentación - Administrador

Figura 38

Modelo de Presentación - Usuarios

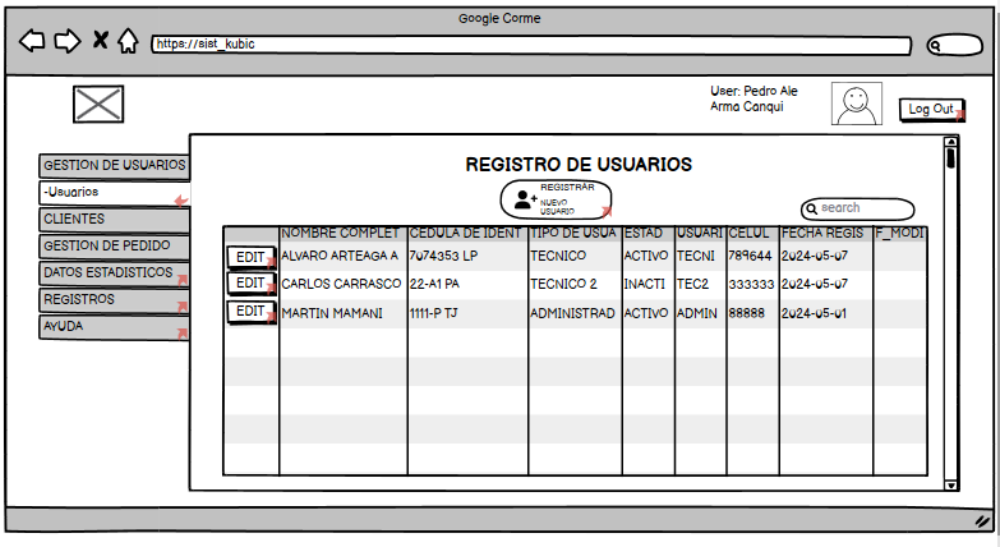


Figura 39

Modelo de Presentación - Roles y Permisos

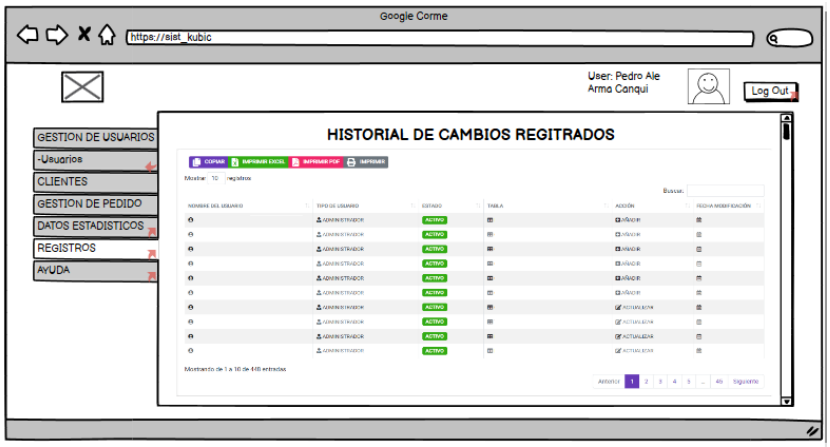


Figura 40

Modelo de Presentación – Registro de Tipología

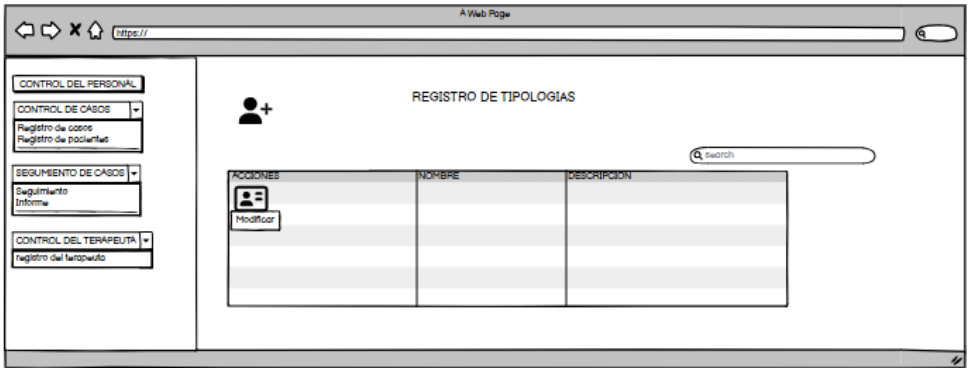


Figura 41

Modelo de Presentación – Registro de Institución

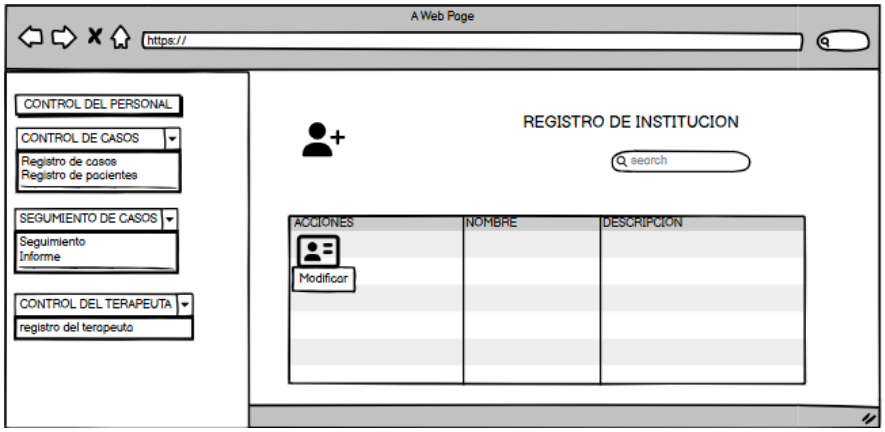


Figura 42

Modelo de Presentación – Registro de Terapeuta

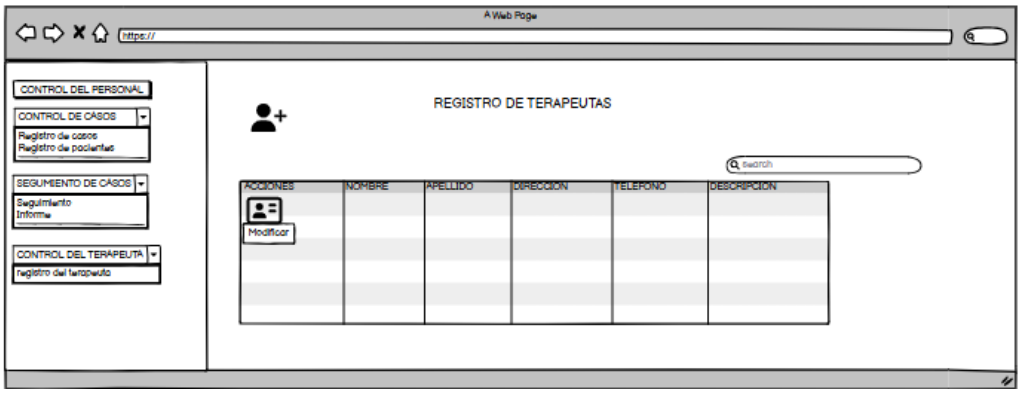


Figura 43

Modelo de Presentación – Registro de Nuevo Caso

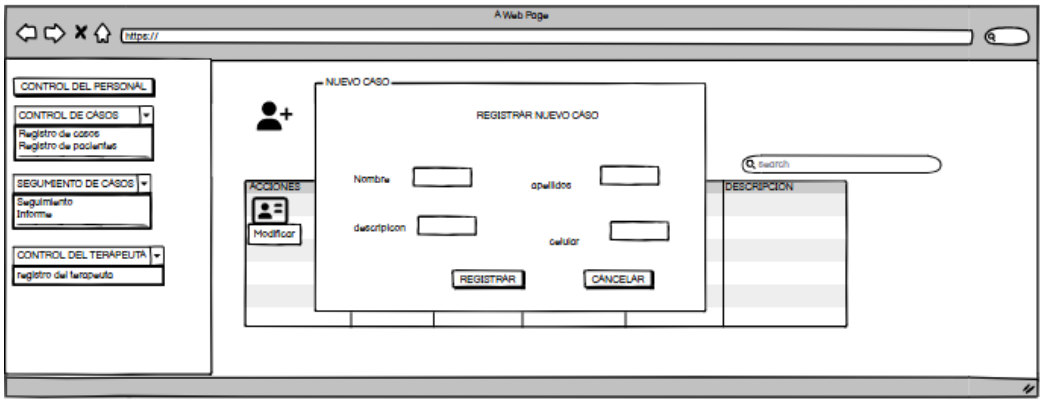


Figura 44

Modelo de Presentación – Registro de Asignar Horario

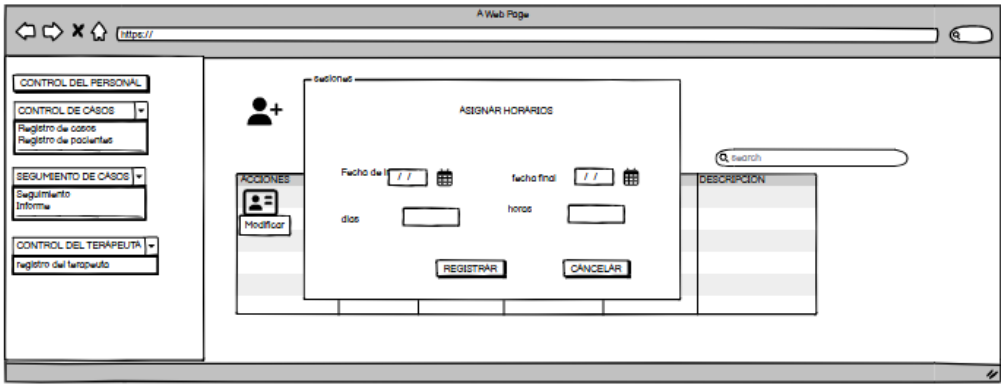


Figura 45

Modelo de Presentación – Registro de Casos

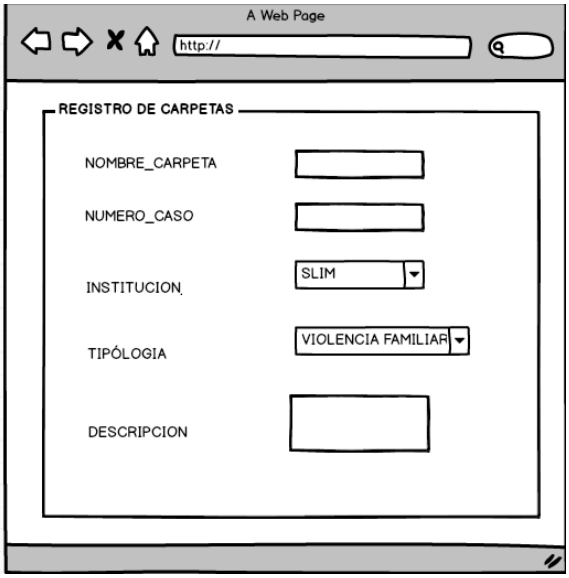


Figura 46

Modelo de Presentación – Gestión Casos

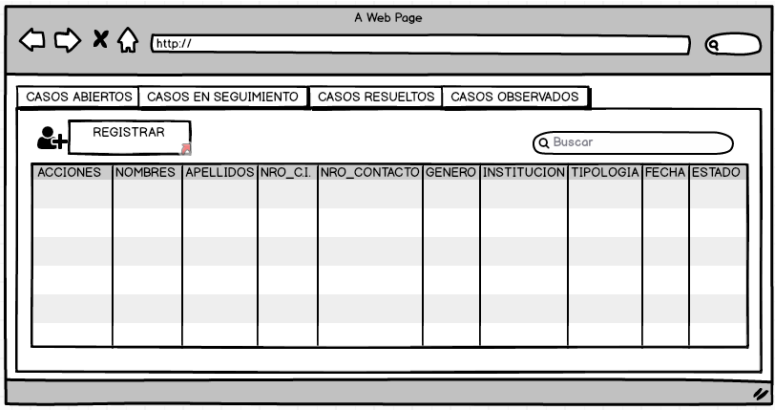


Figura 47

Modelo de Presentación – Registro de Pacientes

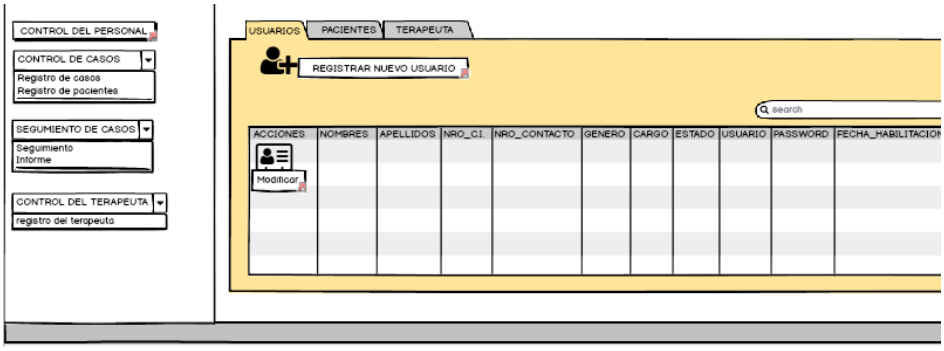


Figura 48

Seguimiento de Casos

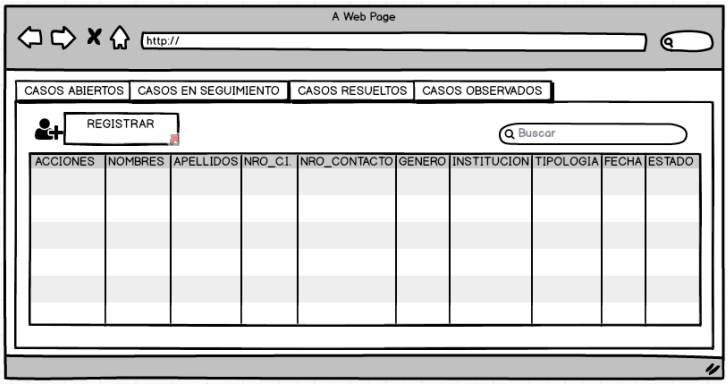
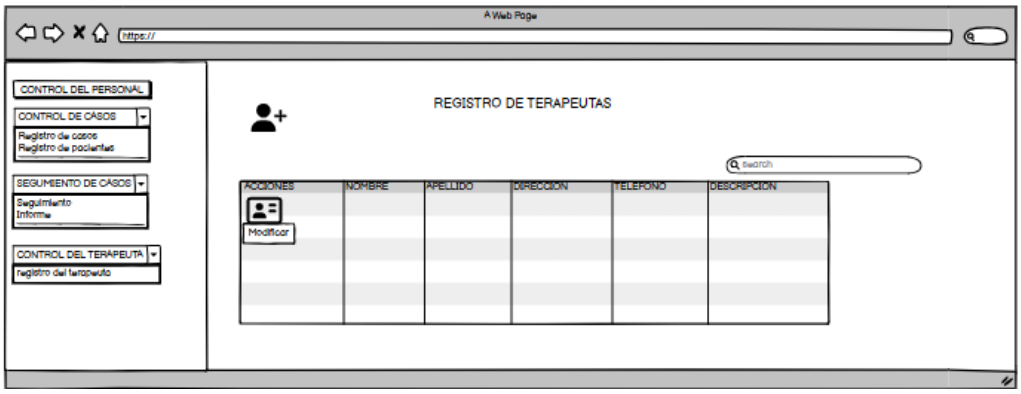


Figura 49

Control de Terapeutas



3.2.10 Modelo de Proceso

Figura 50

Página Principal

Figura 51

Datos del Centro

Figura 52

Datos de la Empresa

Figura 53

Inicio de Sesión del Sistema

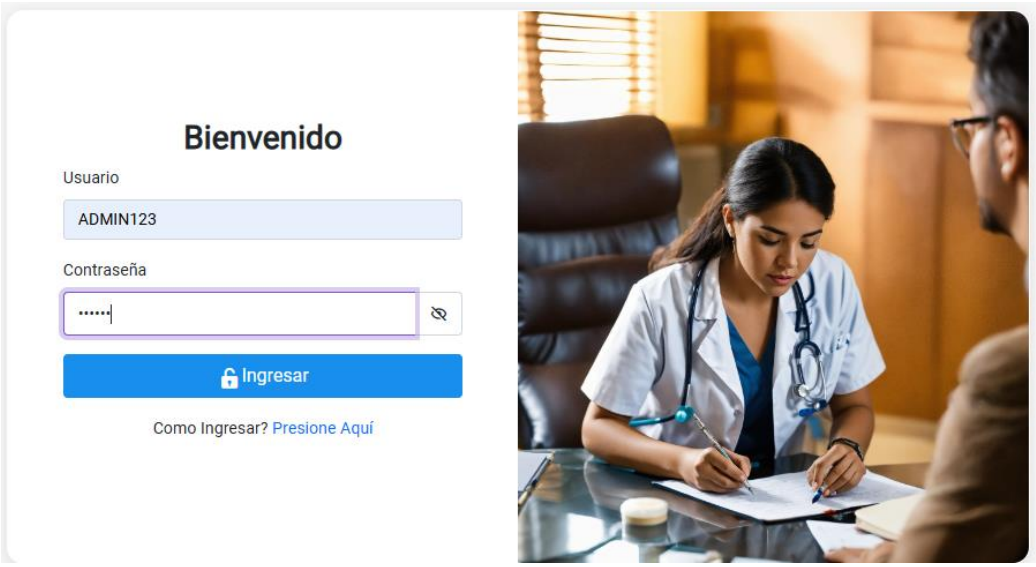


Figura 54

Ingreso al Sistema - Administrador

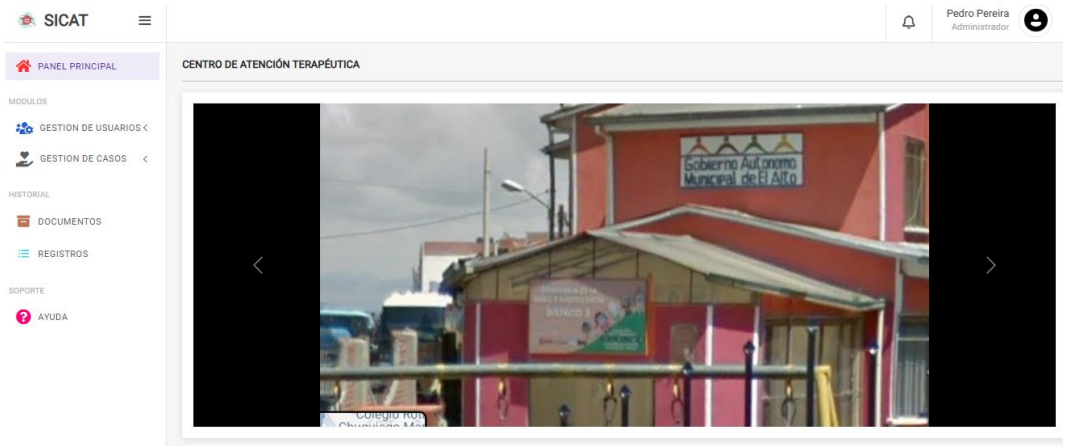


Figura 55

Gestión de Usuarios

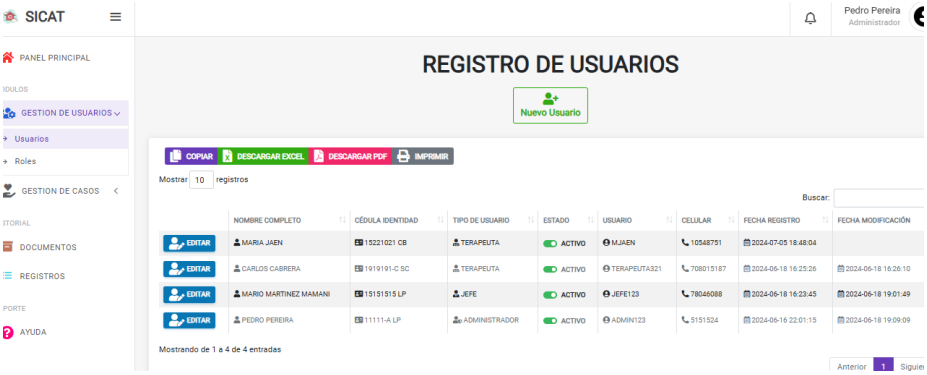


Figura 56

Gestión de Roles

Figura 57

Gestión de Tipologías

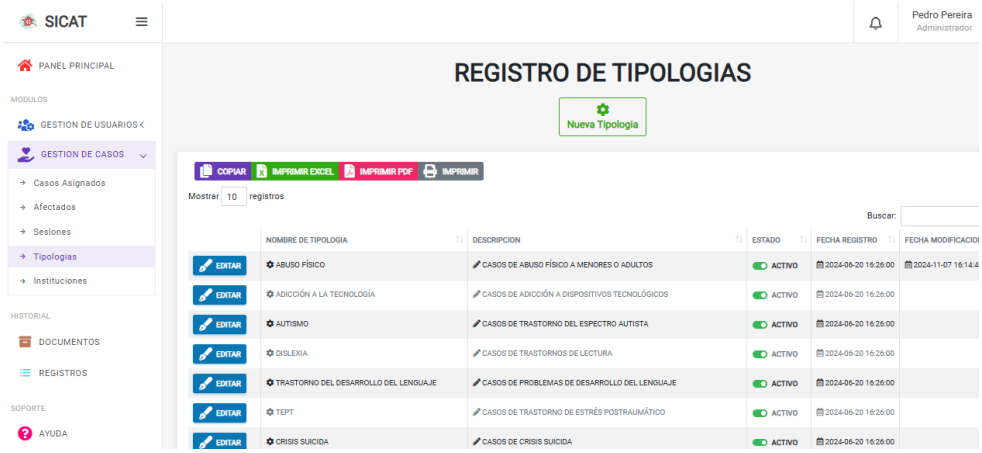


Figura 58

Gestión de Institución

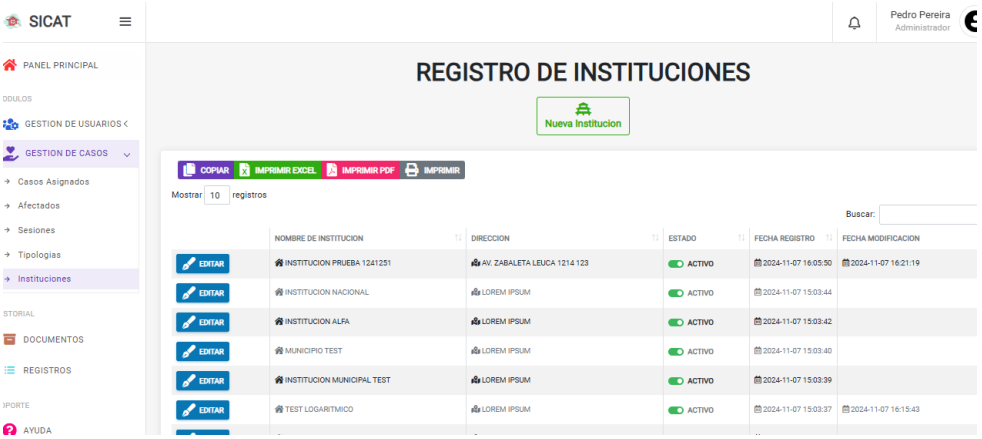


Figura 59

Registro de Casos

Figura 60

Administración de Registro de Pacientes

Figura 61

Administración de Registro de Sesiones Terapeuta

Figura 62

Registro de Nuevo Casos

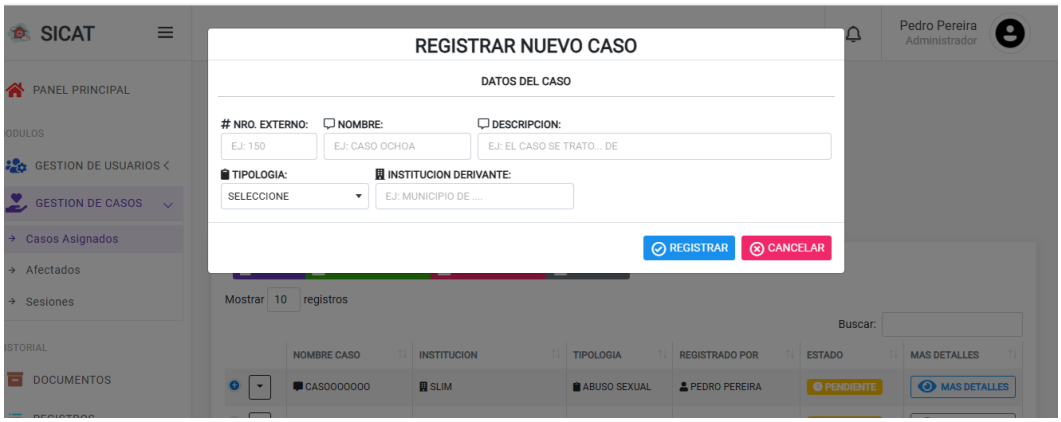


Figura 63

Administración de Asignar Horario

Figura 64

Control de Casos

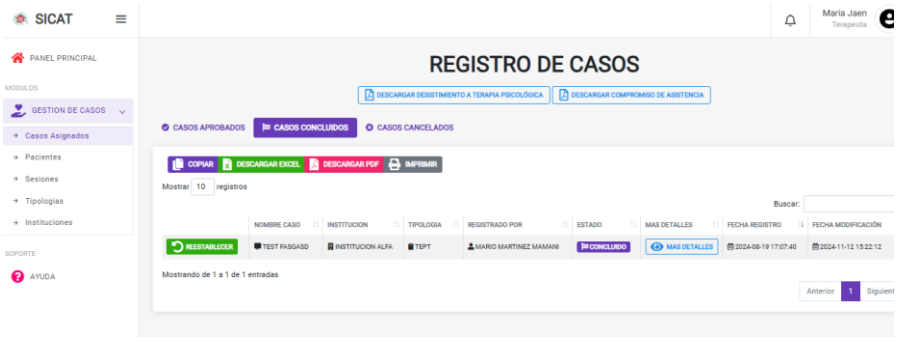


Figura 65

Control de Seguimiento del Paciente

Figura 66

Control de Sesión del Paciente

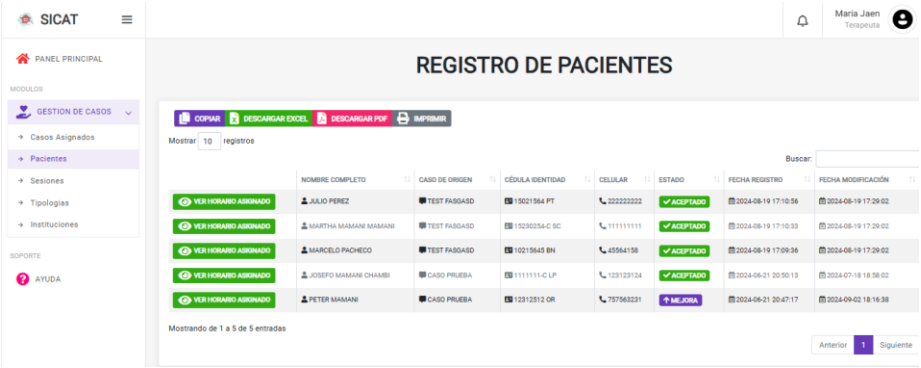


Figura 67

Gestión de Documentos

Figura 68

Administración de Datos estadísticos

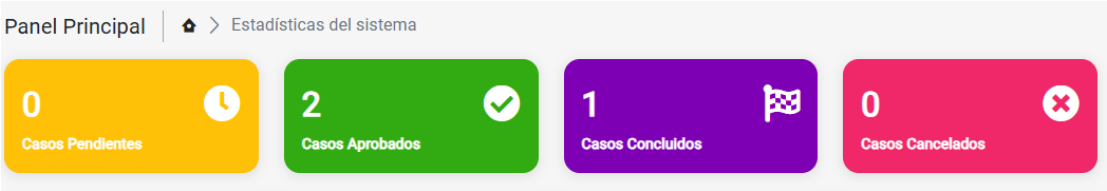


Figura 69

Cantidad de Casos por Mes

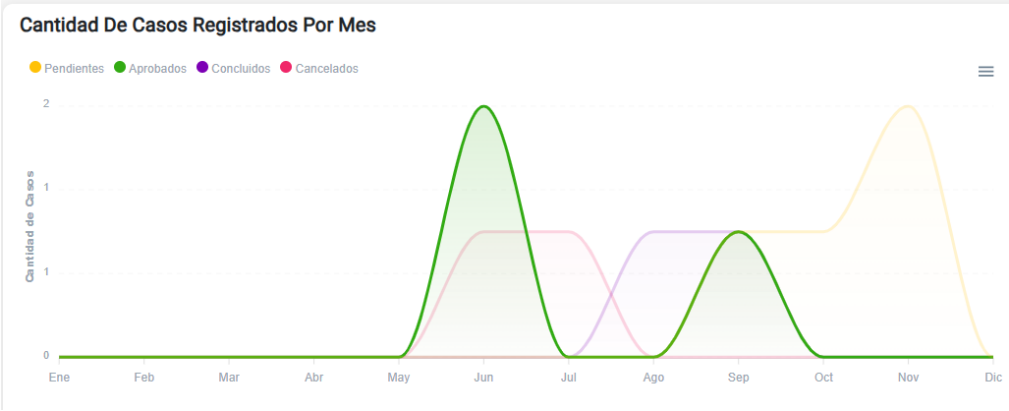


Figura 70

Cantidad de Tipologías por Mes

Figura 71

Búsqueda de Estado de Pacientes

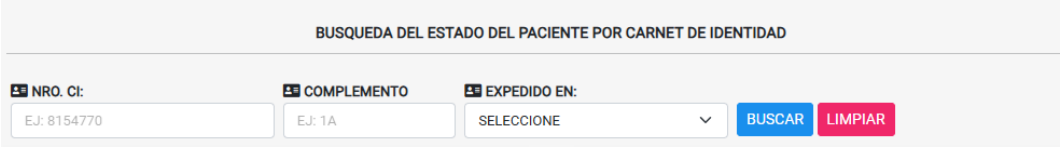


Figura 72

Derivación de Casos

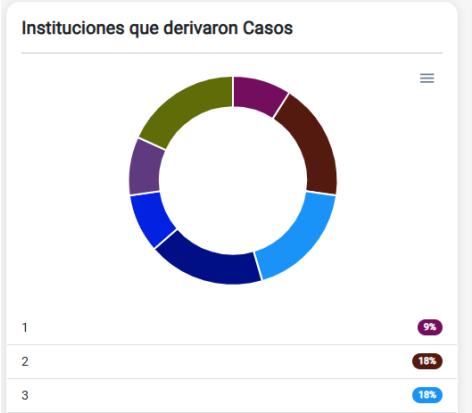
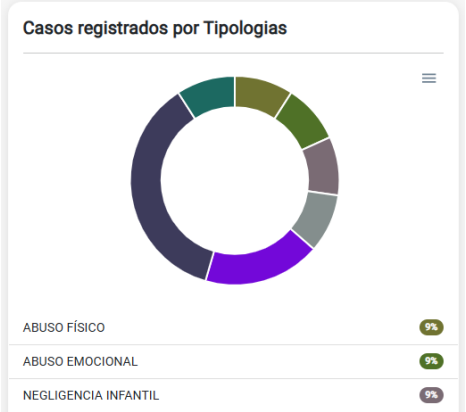


Figura 73

Datos Estadísticos de Tipologías



CAPÍTULO IV
CALIDAD, SEGURIDAD Y COSTOS

INGENIERÍA DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO

4 CAPÍTULO IV - CALIDAD, SEGURIDAD Y COSTO

4.1 INTRODUCCIÓN

En este punto, se realizará la medición de la calidad del software aplicando las métricas de la ISO 25000, que nos dice un componente de calidad de software puede ser descrito en seis características, las cuales son: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenimiento y Portabilidad así mismo aplicamos el COCOMO II con los costos.

4.2 MÉTRICAS DE CALIDAD - ESTÁNDAR ISO/IEC 25000

Para el presente Proyecto de Grado, se han implementado métricas de calidad que permiten evaluar diversos parámetros del sistema. Estas métricas son fundamentales para especificar de manera ordenada las características y atributos del software. Entre los diversos modelos y criterios de calidad para productos de software, uno de los más destacados es el ISO 25000. Este estándar ofrece un marco integral para evaluar la calidad del software y se utiliza para medir varios aspectos, incluyendo la funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad del sistema, de acuerdo al siguiente detalle:

4.2.1 Funcionalidad

La funcionalidad se enfoca en las capacidades del software y en el proceso de desarrollo asociado. Por esta razón, es fundamental evaluar la funcionalidad a través del análisis de puntos de función. Este método permite medir el tamaño y la complejidad del sistema en función de las características que ofrece al usuario, estableciendo un criterio estándar para su evaluación.

Estos aspectos incluyen:

- Cantidad de entradas de usuario
- Cantidad de salidas de usuario

- Cantidad de petición de usuario
- Cantidad de archivos
- Cantidad de interfaces externas

Para calcular el valor total, se emplea una tabla de factores de ponderación que ayuda a asignar el valor adecuado a cada uno de estos elementos.

Tabla 20

Factores de Ponderación

	Parámetro de Medición	Cuenta	Factor medio	Total
1	Numero de entrada del usuario	10	3	30
2	Número de Salidas del Usuario	10	4	40
3	Número de Petición del Usuario	10	5	55
4	Número de archivos	8	5	40
5	Número de interfaces externas	1	5	6
CT	Cuenta Total			171

Factores de ajuste para la complejidad basados en las siguientes interrogantes, en base a los siguientes parámetros:

0 - No influencia: El factor no tiene impacto en la complejidad del proyecto.

1 - Incidencial: Impacto menor y ocasional en la complejidad.

2 - Moderado: Impacto moderado que requiere atención.

3 - Medio: Impacto medio con influencia considerable.

4 - Significativo: Impacto significativo y frecuente en la complejidad.

5 - Esencial: Impacto esencial y crítico en la complejidad del proyecto.

Después de contar los parámetros de función, se procede al cálculo de los puntos de función sin ajustar en la tabla siguiente, este proceso implica sumar las diversas métricas recolectadas, proporcionando una medida cuantitativa que refleja la funcionalidad total del sistema antes de considerar cualquier ajuste de complejidad.

En la tabla se muestra la suma total de los puntos de función sin ajustar, para obtener los valores ajustados se aplican factores de complejidad específicos que se detallan a continuación en la tabla 20, estos factores consideran diversas características del sistema.

Ajustar los puntos de función iniciales con estos factores permite una evaluación más precisa y realista de la carga de trabajo y los recursos necesarios para el desarrollo del software, este método asegura que todas las particularidades y desafíos del proyecto se reflejen adecuadamente en la estimación final.

Tabla 21

Valores de Ajuste de Complejidad

	Factor 2	**0**	**1**	**3**	**4**	**5**	**Fi**
1	¿Requiere el sistema copias de seguridad y recuperación?				X		4
2	¿Se requiere comunicación de datos?				X		4
3	¿Existen funciones de procesos distribuido?				X		4
4	¿Es crítico el rendimiento?				X		4

5	¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?			X		4
6	¿Requiere el sistema entrada de datos interactiva?				X	5
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas operaciones?			X		4
8	¿Se actualizan los archivos maestros de forma interactiva?		X			3
9	¿Son complejas las entradas, salidas, archivos o peticiones?			X		4
10	¿Es complejo el procesamiento interno?			X		4
11	¿Se diseña el código para ser reutilizable?			X		4
12	¿Están incluidas en el diseño la conversión y la instalación?			X		4
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones?			X		4
14	¿Se ha diseñado para facilitar cambios y ser fácilmente utilizada por el usuario?				X	4
FACTOR DE COMPLEJIDAD		FCT				56

Fórmula para ajustar puntos de función, que considera factores como la complejidad técnica, el rendimiento y la usabilidad.

(17) Ecuación de Ajuste

$$PF = conteo\ total * \ [0.65 + 0.01 * \sum fi] \qquad (17)$$

Donde:

$\sum(fi)$: Sumatoria de los valores de los factores de ajuste.

$Conteo\ total$: Número de parámetros del punto función

PF : Ajuste

Se obtiene la sumatoria de ponderación de los índices de usabilidad.

$\sum(f_i)$ = 61

$$PF = 171 * [0.65 + 0.01 * 56]$$

$$PF = 206$$

Con el límite superior establecido en la suma de los factores de ajuste máximo $\sum(fi\ max)$ = 61, se realiza el cálculo del valor siguiente en la secuencia, este proceso es fundamental para mantener la coherencia en la evaluación de la complejidad del sistema, hallamos el punto función máximo.

$$PFmax = Cuenta\ Total\left[0.65 + (0.01 * \sum fi\ max)\right]$$

$$PFmax = 171 * [0.65 + (0.01 * 70)]$$

$$PF\ max = 230$$

La medición de la funcionalidad se basa en comparar el valor máximo posible con el valor obtenido del punto de función ajustado, esta relación proporciona información crucial sobre la eficiencia y la complejidad del sistema evaluado.

$$Funcionalidad = \frac{PF}{PFmax} * 100$$

$$Funcionalidad = \frac{206}{230} * 100$$

$$Funcionalidad = 0.89 * 100$$

$$\boxed{\boldsymbol{Funcionalidad = 89}\ \%}$$

Entonces, la eficacia del sistema alcanza el 89 %, indicando su alta capacidad operativa con un riesgo mínimo.

4.2.1 Confiabilidad

La confiabilidad de un sistema se refiere a su capacidad para mantener un nivel de funcionamiento adecuado durante un período de prueba establecido que implica considerar aspectos como la madurez del sistema. Aplicaremos las fórmulas correspondientes, teniendo en cuenta las probabilidades de fallas y éxitos debido a varios factores.

Considerando que se tiene en cuenta que:

Probabilidad de fallas

$$F(t) = P(T =< t)$$

Probabilidad de trabajo sin fallas

$$P(T > t) = 1 - F(t)$$

Se considera la siguiente función al calcular la confiabilidad del sistema.

(18) Función de la Confiabilidad del Sistema

$$(t) = f * e^{-\mu * t} \qquad (18)$$

Donde:

f	=	Funcionalidad del sistema
μ	=	Probabilidad de error del sistema
t	=	Tiempo de prueba del sistema
(t)	=	Confiabilidad

Durante un periodo de prueba de 20 días, se registra una falla por cada 12 ejecuciones del sistema.

$$F(\mathrm{t}) = 0.89 * \mathrm{e}^{-\frac{1}{10}*18}$$

$$F(\mathrm{t}) = 0.89 * 0.135$$

$$F(t) = 0.120$$

$$F(\mathrm{t}) = 0.120 * 100$$

$$\boxed{\boldsymbol{F}(\mathbf{t}) = \mathbf{12}\ \%}$$

Este valor representa la probabilidad de fallas del sistema en un periodo de tiempo, por lo que la siguiente formula representa la probabilidad de que el sistema sea confiable.

Sustituyendo:

$$\mathrm{Confiabilidad} = \mathrm{P}(\mathrm{T} \leq \mathrm{t}) = 1 - \mathrm{F}(\mathrm{t})$$

$$\boxed{\boldsymbol{Confiabilidad} = \mathbf{P}(\mathbf{T} \leq \mathbf{t}) = 0.120}$$

$$\mathrm{Confiabilidad} = \mathrm{P}(\mathrm{T} \leq \mathrm{t}) = 1 - 0.120$$

$$\mathrm{Confiabilidad} = \mathrm{P}(\mathrm{T} \leq \mathrm{t}) = 0.89 * 100$$

$$\boxed{\boldsymbol{Confiabilidad} = \mathbf{P}(\mathbf{T} \leq \mathbf{t}) = \mathbf{89}\ \%}$$

Por lo tanto, para este software desarrollado, la confiabilidad se sitúa en un sólido 87.5%, verificada a lo largo de un periodo de prueba de 20 días, este resultado confirma la consistencia y fiabilidad del sistema en condiciones operativas durante este período de evaluación establecido, cabe mencionar que en todo proceso se tiene un margen de error, que en lo recomendable sea mínimo.

4.2.2 Usabilidad

Al evaluar la usabilidad de un sistema, es importante considerar el factor humano, asegurando que el software cumpla con los requisitos y expectativas del usuario, por lo tanto, se lleva a cabo una evaluación mediante encuestas dirigidas a los usuarios del sistema.

Estas encuestas permiten recopilar información valiosa sobre la experiencia del usuario, su satisfacción y la facilidad de uso percibida que se utiliza luego para calcular la usabilidad del sistema, aplicando una fórmula diseñada específicamente para medir la eficacia y la experiencia del usuario en la interacción con el software.

(19) Función para Determinar la Usabilidad

$$FU = \left(\frac{\sum valor}{n} * 100\right)/5 \qquad (19)$$

Donde:

$\sum \text{valor}$ = Sumatoria de los valores de la usabilidad del sistema

n = Cantidad de valores de la usabilidad del sistema

FU = Usabilidad

Tabla 22

Indicadores de Facilidades de Uso

ESCALA	VALOR
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

A continuación, en la siguiente tabla se muestra:

Tabla 23

Usabilidad del Sistema

PREGUNTA	VALOR
¿Puede Utilizar con facilidad el sistema?	5
¿Puede Controlar operaciones que el sistema solicita?	5
¿Las Respuestas del sistema son complicadas?	4
¿El Sistema permitió la retroalimentación de información?	5
¿El sistema cuenta con interface agradable a la vista?	4
¿La respuesta del sistema es satisfactoria?	4
¿Le parece complicada las funciones del sistema?	5
¿Se hace difícil o dificultoso aprender a manejar el sistema?	5
¿Los resultados que proporciona el sistema facilitan el trabajo?	4

¿Durante el uso del sistema se produjo errores?	4
TOTAL	**45**

A partir de los datos recabados a través del cuestionario, llevamos a cabo un análisis exhaustivo para evaluar la usabilidad del sistema. Este proceso implica calcular diversos indicadores y métricas que reflejen la facilidad con la que los usuarios interactúan con el sistema.

$$FU=(\frac{45}{10} * \quad 100)/5$$

$$FU=(0.46 * \quad 100)/5$$

$$\boxed{\textbf{Usabilidad} = \textbf{FU} = \textbf{94\%}}$$

Con esta información, podemos afirmar que la facilidad de uso del sistema alcanza un porcentaje del 94%, este dato proporciona una medida concreta de la experiencia del usuario y su capacidad para interactuar efectivamente con el sistema.

4.2.3 Eficiencia

La eficacia evalúa si el sistema utiliza eficientemente sus recursos. Para determinar su nivel, se considera una escala específica como referencia.

Tabla 24

Escala de Valores de Eficiencia

ESCALA	VALOR
Excelente	5
Bueno	4
Aceptable	3
Deficiente	2
Pésimo	1

Tabla 25

Evaluación de Eficiencia

PREGUNTA	PORCENTAJE
¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	4
¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad de movimientos?	5
¿Los datos de salida están presentados de modo que se entienden de inmediato?	5
¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	5
¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	5
TOTAL	24

Para determinar la eficacia, se emplea la fórmula siguiente.

(20) Fórmula Eficiencia

$$E = \frac{\sum x_i}{n} * \frac{100}{n} \qquad (20)$$

Donde:

$\sum x_i$	=	Sumatoria de los valores de eficiencia
n	=	Número de preguntas
E	=	Eficiencia

Sustituyendo se tiene lo siguiente:

$$E = \frac{24}{5} * \frac{100}{5}$$

$$\boldsymbol{Eficiencia = E = 96\,\%}$$

Con estos datos, determinamos que la eficiencia del sistema se sitúa en el 96%, lo que indica que los recursos del sistema se están utilizando de manera efectiva y óptima.

4.2.4 Mantenibilidad

Esta métrica evalúa la cantidad de trabajo requerido para realizar cambios en el sistema, ya sea para corregir errores o agregar nuevas funcionalidades. El estándar IEEE94 recomienda utilizar el índice de madurez del sistema como indicador de su estabilidad. Por lo tanto, la ecuación para este índice es la siguiente:

(21) Ecuación para el Índice de Madurez del Sistema

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt} \qquad (21)$$

Donde:

Mt = Número de módulos total de la versión actual

Fa = Número de módulos de la versión actual que se añadieron.

Fc = Número de módulos de la versión actual que se cambiaron.

Fd = Número de módulos que se eliminaron en la versión anterior a la actual.

IMS = Madurez del sistema

Tabla 26

Indicadores de Mantenibilidad

DESCRIPCIÓN	VALOR
Mt	6
Fa	0
Fc	2
Fd	0

Sustituyendo en la ecuación se tiene:

$$IMS = \frac{[8 - (0 + 1 + 0)]}{8}$$

$$IMS = 0.875 * 100\%$$

$$\boxed{\boldsymbol{IMS = 87.5\%}}$$

En conclusión, el índice de mantenibilidad del sistema desarrollado alcanza el 87.5% indicando que el sistema tiene una alta capacidad para ser modificado, corregido y mejorado, facilitando su adaptación a nuevas necesidades y requisitos a lo largo del tiempo.

4.2.5 Portabilidad

La capacidad de un software para ser transferido de un entorno a otro incluye varios aspectos importantes:

Adaptabilidad

Habilidad del software para ajustarse a diferentes entornos sin requerir modificaciones adicionales.

Facilidad de Instalación

Esfuerzo requerido para instalar el software en un entorno específico.

Conformidad

Verificación de que el software cumple con estándares o convenciones de portabilidad.

Capacidad de sustitución

Facilidad y esfuerzo necesarios para sustituir el software con otro producto que tenga funciones similares.

El sistema, desarrollado con Laravel, puede implementarse en cualquier servidor con Apache y las herramientas necesarias.

Como la tecnología web es ejecuta fácilmente en cualquier dispositivo con conexión a Internet y un navegador.

(22) Fórmula Portabilidad

$$P = 1 - (\frac{NAS}{NIS}) \qquad (22)$$

Sustituyendo la fórmula

$$Portabilidad = 1 - (\frac{1}{12})$$

$$Portabilidad = 0.917 * 100\%$$

$$\boxed{Portabilidad = \mathbf{91,7}\%}$$

Después de un análisis exhaustivo de cada parámetro según los criterios establecidos por la norma ISO 25000 se ha obtenido resultados que reflejan el rendimiento global del sistema, el cálculo detallado de cada aspecto relevante para asegurar que el sistema cumple con los estándares de calidad y funcionalidad esperados, proporcionando así una visión clara y detallada de su desempeño general.

Tabla 27

Resultados Obtenidos - Norma ISO 25000

CRITERIOS EVALUADOS	RESULTADOS
Funcionalidad	89 %
Confiabilidad	89 %
Usabilidad	94 %
Eficiencia	92%
Mantenibilidad	87,5 %
Portabilidad	91,7 %
TOTAL	**90.53%**

4.3 ESTIMACIÓN DEL COSTO

Por medio del análisis costo-beneficio del este proyecto, se realiza la determinación del valor de la inversión para el desarrollo e implementación del sistema web, haciendo uso y combinación de los métodos de estimación de esfuerzo, tiempo y costo, ya que son los más utilizados en el desarrollo de proyectos informáticos. Cabe mencionar que el análisis fue realizado mediante el COCOMO II básico.

4.4 COCOMO II

En la siguiente tabla se detalla el número de líneas de código desarrolladas según el lenguaje de programación utilizado.

Tabla 73

Determinación del LCD

Lenguaje de programación	Líneas de código
CSS	300
JavaScript	3000
PHP	1500
TOTAL	4800

A continuación, aplicamos las fórmulas de esfuerzo, tiempo calendario y personal requerido para lo cual utilizaremos las siguientes formulas:

Esfuerzo Aplicado en Personas

$$EA = 2{,}4 * (KLDC)^{1{,}05}$$

Tiempo de Desarrollo

$$PD = 2{,}5 * (ED)^{0.38}$$

Donde:

ED: Esfuerzo aplicado en personas por mes.

PD: Tiempo de desarrollo en mes.

KLDC = Número estimado de líneas de código distribuidas.

LDC = 4800

Ya que la suma de las líneas de código son un total de 5800 LDC, procedemos a realizar la siguiente operación para determinar el valor de KLDC número estimado de líneas de código.

KLDC =4800/1000

KLDC= 4.8

Los modos de COCOMO II son el orgánico, semiacoplado y el empotrado. Este proyecto se adecua a las características del modo orgánico. Por lo tanto, realizamos el desarrollo de las fórmulas mencionadas anteriormente en base a las especificaciones de la siguiente tabla:

Tabla 74

Tabla de Valores de COCOMO II

MODO	A	B	C	D
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,2	2,5	0,32

- Esfuerzo aplicado

ED = $2{,}4 * (4800)^{1{,}05}$ = 17,60 [Personas/mes] - Tiempo de desarrollo

TD = $2{,}5 * (17{,}60)^{0{,}38}$ = 7.43 [Meses]

Se necesitan dos programadores para el desarrollo del proyecto. Tomando en cuenta el salario mínimo de trabajo en Bolivia que es de 2500 bolivianos/mes se determina el costo del software mediante la siguiente formula:

Costo del Software

$CS = \# P * SP * \#$

$$CS = 2 * 2500 * 7 = 35000$$

Por lo tanto, el costo total del software es de 35000 bolivianos.

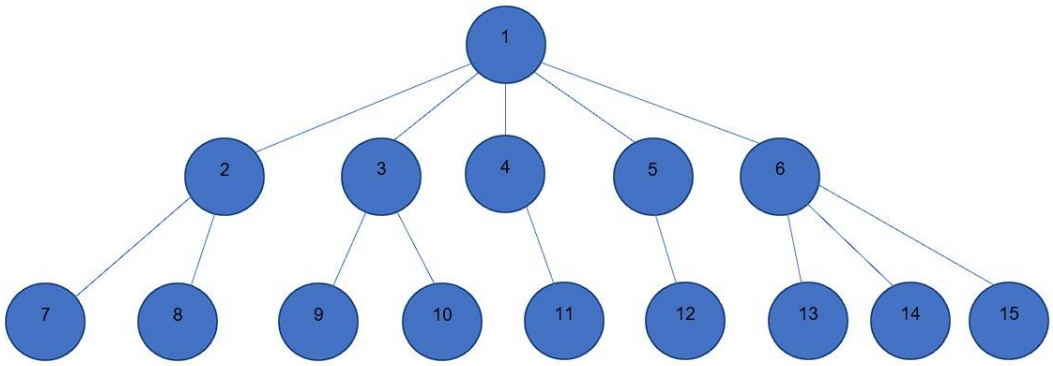
4.5 PRUEBAS DEL SISTEMA

4.5.1 Pruebas de Caja Blanca.

Las pruebas de caja blanca se diferencian de las pruebas de caja negra, que se centran en los resultados visibles desde el exterior, ya que exploran las funciones internas de un módulo de software. Su propósito es asegurar que el código opere de manera correcta y satisfaga los requisitos especificados, analizando cada línea, condición y bucle para identificar errores lógicos, fallos en la estructura o posibles fugas de memoria.

Figura 74

Técnica de Camino Básico



Inicio de sesión (1)

Módulo de casos (2)

Módulo de sesiones (3)

Módulo de pacientes (4)

Módulo de reportes (5)

Administración de usuarios (6)

Registro de pacientes (7)

Administración de casos (8)

Horario sesiones (9)

Datos (10)

Reporte (11)

Usuarios (12)

Roles (13)

Fin ciclo sistema (14)

Fin del sistema (15)

Después de construir el grafo, determinaremos su complejidad ciclomática utilizando la fórmula siguiente:

Complejidad Ciclomática

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

Donde:

A = 14 (Aristas)

N = 15 (nodos)

Por lo tanto, la complejidad ciclomática V(G):

V(G) = 14 – 15 + 2 = 1

Para identificar los caminos linealmente independientes que necesitan ser probados, es necesario considerar la estructura del sistema, ya que existen múltiples caminos posibles que un usuario puede seguir.

4.5.2 Pruebas de Caja Negra.

En resumen, una prueba de caja negra se asemeja a un sistema enigmático. Solo tenemos acceso a las entradas y salidas, sin conocer los detalles internos del funcionamiento. Imagina una caja cerrada con mecanismos ocultos que transforman lo que entra en lo que sale. Para la historia de usuario “Gestión de Aperturas”, que especifica que el Superadministrador debe poder crear aperturas, que son instancias organizativas dentro del sistema, y que la interfaz debe ser intuitiva y ofrecer una visualización clara de la estructura organizativa, diseñaremos una prueba de caja negra

4.5.2.1 Prueba de Caja Negra – Inicio de Sesión.

Figura 75

Prueba de Caja Negra Inicio de Sesión

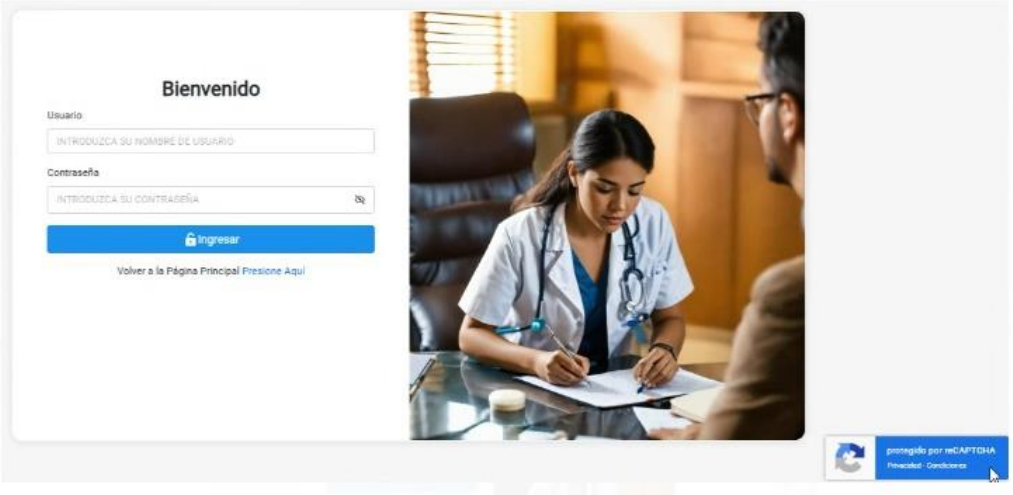


Tabla 75.

Prueba de Caja Negra Inicio de Sesión

ENTRADAS		SALIDAS	RESULTADOS
Usuario	Contraseña	"Ingrese usuario y contraseña"	El sistema valida que no ingresen datos en blanco
Administrador	admin	"Bienvenido al sistema concede sistema"	acceso al ingreso de datos correctos

Figura 76

Caja Negra Ingreso al Sistema

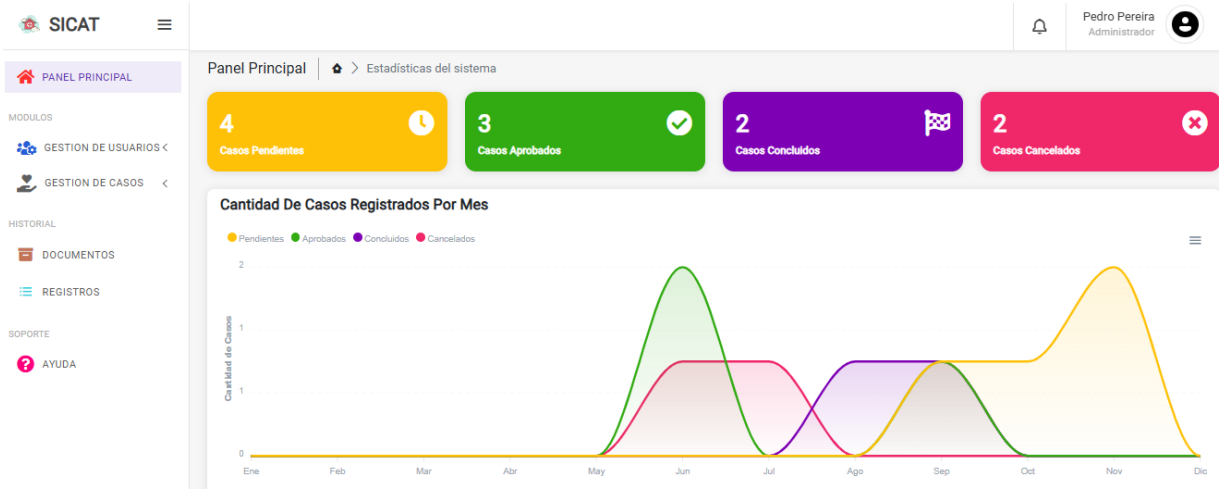


Tabla 28

Prueba de Caja Negra Ingreso el Sistema

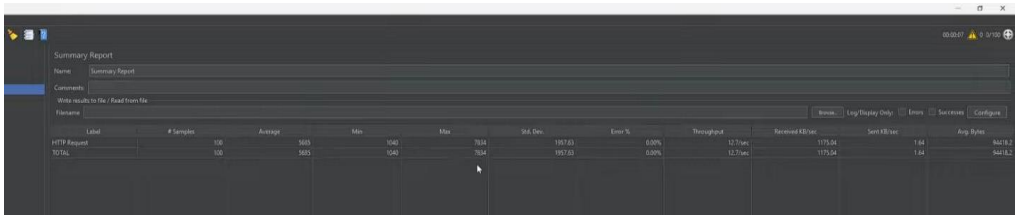
ENTRADA		RESULTADO
Usuario	admin	El sistema verifica que no queden campos vacíos y al introducir datos correctos, concede el acceso.
Contraseña	*****	
Captcha	✓	

4.5.3 Pruebas de Estrés

Las pruebas de estrés se emplean para evaluar el rendimiento del software en condiciones extremas. Al exponer el sistema a una carga de trabajo mucho mayor de lo habitual, estas pruebas identifican sus límites y puntos de falla, asegurando que pueda manejar picos de demanda sin comprometer su estabilidad y funcionalidad. En este punto, se realizará una prueba de estrés en PDF del sistema web.

Figura 77

Prueba de Estrés



Respecto a las pruebas de estrés se utilizó la herramienta Jmeter donde se observa que si en un minuto entran 100 usuarios la página tardaría en iniciar 5.6 segundos como máximo con un margen de error de 0%, ya si ingresaran 500 usuarios al mismo tiempo tendría un margen de error del 15% y una carga de 15 segundos como máximo. Estos resultados indican que el sistema tiene un buen rendimiento bajo condiciones de estrés, manteniendo tiempos de respuesta bajos y una eficiente tasa de procesamiento. Esto sugiere que el sistema puede soportar una carga considerable sin comprometer la calidad del servicio, lo cual es crucial para garantizar una experiencia de usuario satisfactoria en escenarios de alta demanda.

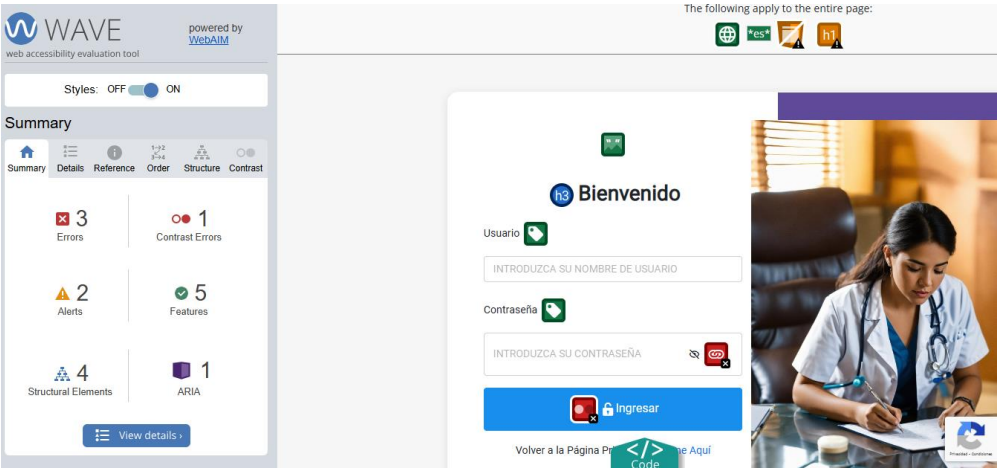
4.5.4 Pruebas de Accesibilidad

En resumen, las pruebas de accesibilidad son una práctica fundamental para optimizar las aplicaciones web y móviles, asegurando que sean utilizables por una amplia audiencia,

incluyendo personas que estén en situación de discapacidad, ya sea visual, auditiva u otras situaciones físicas o cognitivas.

Figura 78

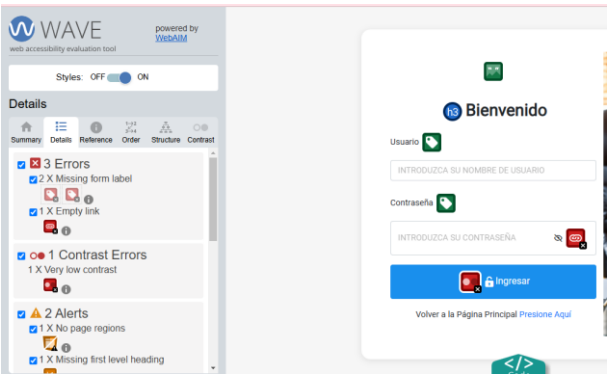
Prueba de Accesibilidad



Según la Prueba de Accesibilidad WAVE se puedo evidenciar los resultados del software con los siguientes detalles: 3 Errores, 1 contraste de errores, 1 falta el encabezado del primer nivel y 1 texto alternativo, lo cual está permitido en el rango de lo que se puede tener fallas en el sistema este margen de errores permite la funcionalidad del sistema con normalidad.

Figura 79

Resultado del Software



4.5.1 Pruebas de Unitarias

Una prueba unitaria de software —también conocida como *unit testing*— es el instrumento utilizado para validar un fragmento de código fuente. Los desarrolladores aíslan una línea del lenguaje codificado para saber si el sistema está operando correctamente en una función, proceso o actividad específica.

La palabra unidad alude a un componente individual del sistema que, a su vez, es desglosado por el programa de *testing* para obtener información detallada sobre el funcionamiento y los comportamientos que lo definen.

Según la Prueba Unitaria PHPUnit se puedo evidenciar los resultados del software con los siguientes detalles:

Figura 80

Prueba Unitaria

```
/Applications/XAMPP/xamppfiles/htdocs/curso/65. unit testing/tutorial  master ±  ./vendor/bin/phpunit tests
PHPUnit 9.2.2 by Sebastian Bergmann and contributors.

..                                                                  2 / 2 (100%)

Time: 00:00.008, Memory: 6.00 MB

OK (2 tests, 2 assertions)
/Applications/XAMPP/xamppfiles/htdocs/curso/65. unit testing/tutorial  master ±  ./vendor/bin/phpunit tests
PHPUnit 9.2.2 by Sebastian Bergmann and contributors.

..E                                                                 3 / 3 (100%)

Time: 00:00.009, Memory: 6.00 MB

There was 1 error:

1) OperationsTest::testSumWithNotNumericValues
A non-numeric value encountered

/Applications/XAMPP/xamppfiles/htdocs/curso/65. unit testing/tutorial/src/Operations.php:10
/Applications/XAMPP/xamppfiles/htdocs/curso/65. unit testing/tutorial/tests/OperationsTest.php:24

ERRORS!
Tests: 3, Assertions: 2, Errors: 1.
✗  /Applications/XAMPP/xamppfiles/htdocs/curso/65. unit testing/tutorial  master ±
```

Según la Prueba Unitaria se puede ver que el sistema se acepta con la medida de errores que presenta ya q no son muy graves para el sistema y nos permite trabajar con normalidad.

4.6 SEGURIDAD

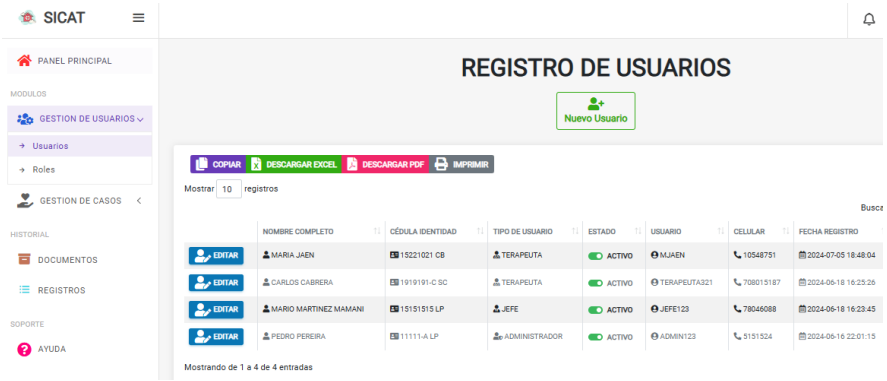
La ISO 27001 es una norma internacional que ofrece un marco sólido para la gestión de la seguridad de la información. Según Alan Calder (2009), esta norma “proporciona una estructura para establecer, implementar, operar, supervisar, revisar, mantener y mejorar un Sistema de Gestión de Seguridad de la Información (SGSI)”. Reconocida globalmente, esta norma establece requisitos detallados que permiten a las organizaciones gestionar de manera estructurada la seguridad de su información. Además de ayudar a identificar y reducir los riesgos de seguridad, la norma ofrece metodologías para evaluar continuamente la efectividad de los controles de seguridad en vigor. Al seguir la ISO 27001, las organizaciones pueden garantizar la confidencialidad, integridad y disponibilidad de sus datos, aspectos cruciales para mantener la confianza de las partes interesadas y cumplir con los requisitos legales y regulatorios.

4.6.1 Identificación y Autenticación

Los usuarios cuentan con usuario y contraseña estrictamente controlada para acceder al sistema.

Figura 81

Identificación y Autenticación

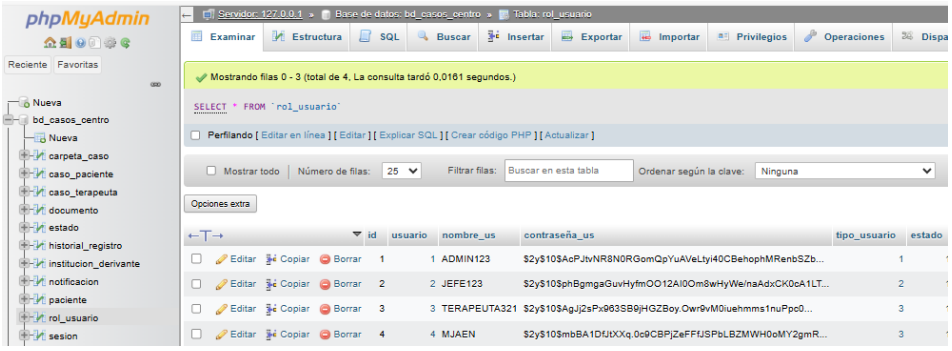


4.6.2 Encriptación

Para la seguridad del password estará encriptado con el algoritmo que es una encriptación de alta seguridad para el ingreso al sistema.

Figura 82

Seguridad de Password



4.6.3 Seguridad Física

Los Back-up de la base de datos deberán ser protegidas en áreas seguras, que solo permita el acceso a personal autorizado, deben ser almacenados por fechas.

4.6.4 Seguridad Organizativa

Manejar el Back-up de acuerdo a la fecha que se realizó para mayor organización.

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

INGENIERÍA DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO

5 CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Después de culminar con el Proyecto de Grado denominado “SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE CASOS EN EL CENTRO DE ATENCIÓN TERAPÉUTICO” de acuerdo a los objetivos establecidos, cumpliendo con los requerimientos.

A continuación, a partir de los objetivos planteados, se llegaron a las siguientes conclusiones:

- Se diseño una página web con interfaz amigable al usuario, combinando colores de la empresa donde muestra toda la información correspondiente.
- Se crearon los perfiles para todos los niveles de acceso y permisos dependiendo del rol a la cual pertenece los usuarios, aplicando la seguridad correspondiente para los accesos.
- Se centralizo toda la información de la institución, en un gestor de base de datos de licencia gratuita.
- Se implemento el sistema, con un panel de administración de amigable navegación y acceso a la información de acuerdo al rol establecido.
- El acceso a la información de registros se optimizo a través del acceso a la información en tiempo real y en cualquier momento, con la conexión a internet correspondiente.
- Se genera los reportes estadísticos necesarios para la institución para mejorar la toma de decisiones a futuro.

5.2 RECOMENDACIONES

Ya concluido el presente Proyecto de Grado, se proponen las siguientes recomendaciones a tomar en cuenta:

- Realizar el cambio continuo de contraseñas de accesos para todos los usuarios del sistema.
- Realizar las copias de seguridad de manera periódica, para garantizar el respaldo de la información en caso de pérdida de datos del sistema.
- El mantenimiento periódico del software es muy recomendable al menos en un periodo de cada seis meses.
- El acceso a internet es de vital importancia para la conexión y acceso a la plataforma, por lo que se recomienda tener el servicio estable y de alta disponibilidad.
- Para la actualización de nuevos módulos o nuevas tecnologías, se debe tomar en cuenta el manual de usuario y el manual técnico del Software.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Alarcón, J. (2023). *Cocomo y punto de fusion.*

Alvarez, M. A. (2000). *Manual de CodeIgniter.*

Barish, R. A., & Arnold , T. (12 de Septiembre de 2022). *Mordeduras MSD*. Obtenido de https://www.msdmanuals.com/es/hogar/traumatismos-y-envenenamientos/mordeduras-y-picaduras/mordeduras-de-animales

Bartholomew, D. (2014). *libro de MariaDB.* Packt Publishing.

Brinck. (2002). *Usability for the Web: Designing Web Sites that Work.* Obtenido de https://www.uxlumen.com/modelos-de-navegacion/

Bustamante, P. (2021). *Estimación de esfuerzo de trabajo y planificación de proyectos de desarrollo con restricciones de recursos.* Obtenido de repositorio UCHILE: https://repositorio.uchile.cl/

Calero , W. (16 de Octubre de 2010). *Ingeniería de software*. Obtenido de Ingeniería de software: https://ingenieraupoliana.blogspot.com/2010/10/cocomo.html

Clutton-Brock. (1999). *Una Historia Natural de los Mamíferos Domesticados. Prensa de la Universidad de Cambridge.*

Conocimiento, D. N. (2022). *Guía para la elaboración de instrumentos de Evaluación de Residentes.* Argentina : Ministerio de Salud.

Flores, A. (20 de Febrero de 2024). *Ecosistemas*. Obtenido de https://ecosistemas.win/que-animales-silvestres-y-domesticos/

Galvan, H. (24 de Septiembre de 2014). *SlideShare.* Obtenido de SlideShare: https://es.slideshare.net/slideshow/cocomo-39479541/39479541

Gardey, J. (2021). Definicion.de. Obtenido de https://definicion.de/web-2-0/.

Gómez, J. (8 de Diciembre de 2013). *Estimación de Costes con COCOMO 81 (II)*. Obtenido de Estimación de Costes con COCOMO 81 (II): https://www.laboratorioti.com/2013/04/22/estimacion-de-costes-con-cocomo-81-ii/

Gonzales, E. (2021). Desarrollo Web con PHP y MySQL.

González, E. (2006). https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193.

GRANADA, A. C. (2024). *City Wok Granada*. Obtenido de https://www.asadorcitywokgranada.com/fructosa-que-es-y-cuales-son-sus-beneficios/#:~:text=Beneficios%20de%20la%20fructosa,-Se%20encuentra%20f%C3%A1cilmente&text=Se%20tolera%20mejor%20por%20los,l%C3%ADquidos%20que%20el%20az%C3%BAcar%20com%C3%BAn

Jaimes, J., & Mitzi , N. (2014). *Modelo de estimación del proyecto Cocomo.*

John E. Hall, M. E. (2021). *Libro de texto de fisiología médica.* Canadá.

Lucena, P. (Junio de 2023). *Universidad CESUMA*. Obtenido de Universidad CESUMA: https://www.cesuma.mx/blog/que-son-los-sistemas-de-informacion-y-por-que-son-necesarios.html#abh_posts

Mamani, R. (2012). *sistemas contables*. Obtenido de https://normasapa.pro/como-instalar-normas-apa-septima-edicion-en-word/.

Maniscalchi, M. T. (diciembre de 2010). *scielo*. Obtenido de scielo: https://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1315-25562010000200009

Marco, S. B. (2016). *Test de Graham*. Obtenido de FAPap: https://fapap.es/articulo/395/test-de-

graham#:~:text=El%20test%20de%20Graham%20es,esta%20manera%2C%20hacer%20el%20diagn%C3%B3stico.

Marrugo, Y. (23 de Junio de 2010). *Modelo Cocomo*. Obtenido de https://es.slideshare.net/jedaro/modelo-cocomo-4593779

Mendoza, L. E. (18 de Marzo de 2022). *Goconqr.* Obtenido de Goconqr: https://www.goconqr.com/flashcard/35942991/metodologia-extreme-programming-xp#google_vignette

Miss, F. G., & Baena, R. (05 de 06 de 2019). *ENUMED.NET*. Obtenido de ENUMED: http://www.eumed.net/rev/ce/2019/norma-iso-eic.html

Modelo Cocomo. (19 de Julio de 2008). Obtenido de https://acevedodelacru.wordpress.com/cocomo-ii/

Narváez, A. (2020). *metodologia UWE.*

Osmosis Latina. (15 de Abril de 2024). *Osmosis Latina*. Obtenido de Osmosis Latina: https://www.osmosislatina.com/jmeter/basico.htm

Parry, C. M. (2008). *Resistencia a los antimicrobianos en Salmonella: Una Perspectiva Global Enfermedades infecciosas clínicas.*

Platzi. (9 de septiembre de 2018). Obtenido de https://platzi.com/blog/pruebas-esenciales-para-evaluar-el-rendimiento-de-software/

Pressman, R. (2023). *Ingeniería de software.*

Risso, A., Ruiz Yanzi, M. V., & Volij, C. (12 de Diciembre de 2022). *evidencia.org*. Obtenido de evidencia.org: https://www.evidencia.org/index.php/Evidencia/article/view/7049

salud. (21 de julio de 2022). *Freshis*. Obtenido de Freshis: https://freshis.com/blog/cuantas-frutas-hay-que-comer-al-dia

Smith, J. S. (1994). *Virus de la Rabia en Microbiología Veterinaria.*

Tenter, A. M. (2000). *Toxoplasma gondii: de los Animales a los Humanos." Revista Internacional de Parasitología.*

Thompson, R. C. (2019). *Zoonoses: Biology and Emerging Issues.*

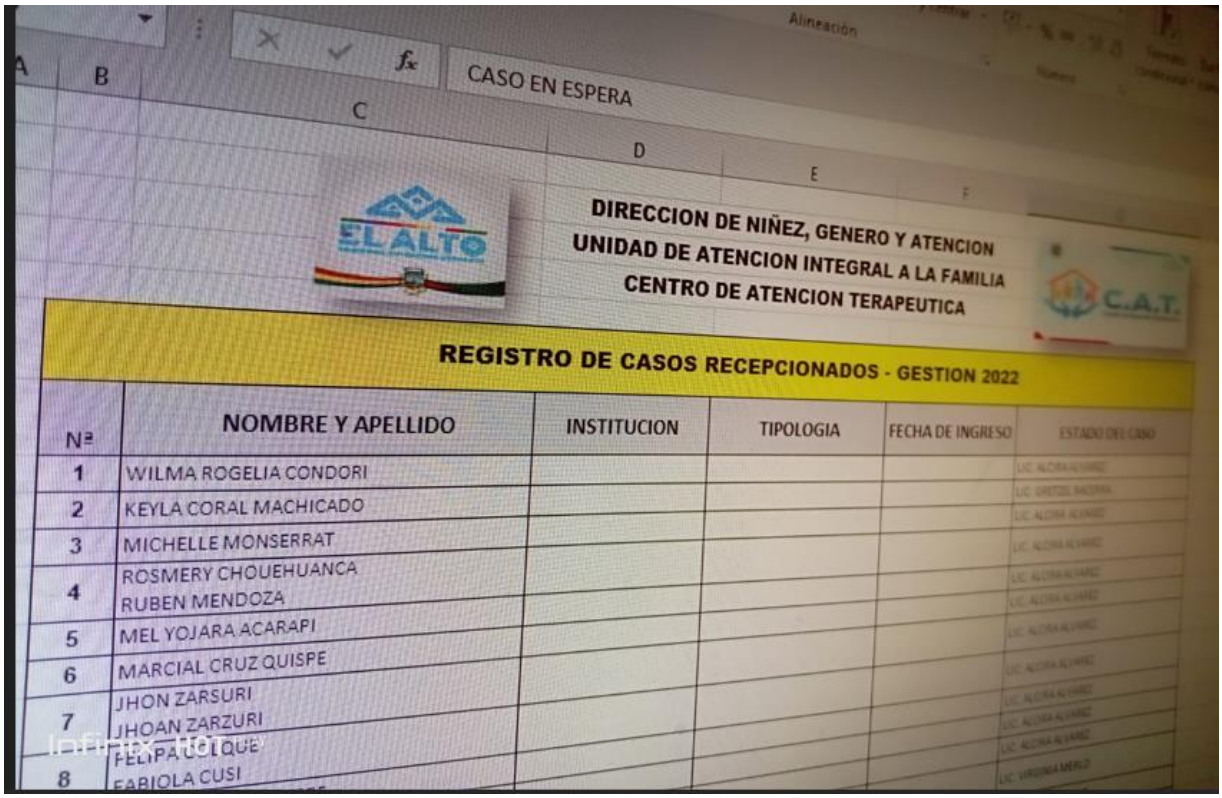
Vargas, M. S. (25 de Junio de 2017). *Ingenieria basada en modelos.* Obtenido de Ingenieria basada en modelos: http://marcelosalasvargas.blogspot.com/2017

Wormser, G. P. (2006). *Guías de práctica de la sociedad de Enfermedades Infecciosas de Estados Unidos para el Tratamiento de la Enfermedad de Lyme.*

Zaptest. (07 de 03 de 2024). *Copyright 2024*. Obtenido de https://www.zaptest.com/es/pruebas-de-caja-negra-que-son-tipos-procesos-enfoques-herramientas-y-mucho-mas

ANEXOS

ANEXO A. Registro de casos recepcionados



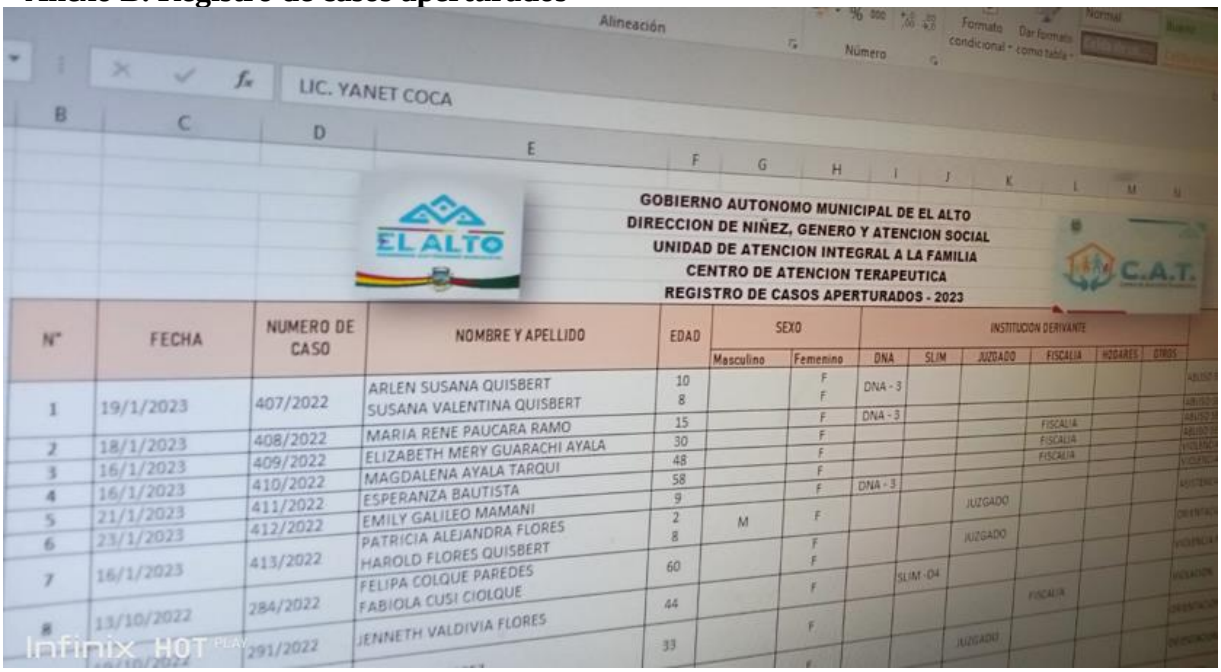
CASO EN ESPERA

DIRECCION DE NIÑEZ, GENERO Y ATENCION
UNIDAD DE ATENCION INTEGRAL A LA FAMILIA
CENTRO DE ATENCION TERAPEUTICA

REGISTRO DE CASOS RECEPCIONADOS - GESTION 2022

Nº	NOMBRE Y APELLIDO	INSTITUCION	TIPOLOGIA	FECHA DE INGRESO	ESTADO DEL CASO
1	WILMA ROGELIA CONDORI				
2	KEYLA CORAL MACHICADO				
3	MICHELLE MONSERRAT				
4	ROSMERY CHOUEHUANCA RUBEN MENDOZA				
5	MEL YOJARA ACARAPI				
6	MARCIAL CRUZ QUISPE				
7	JHON ZARSURI JHOAN ZARZURI				
8	FELIPA LICLQUE FABIOLA CUSI				

Anexo B. Registro de casos aperturados

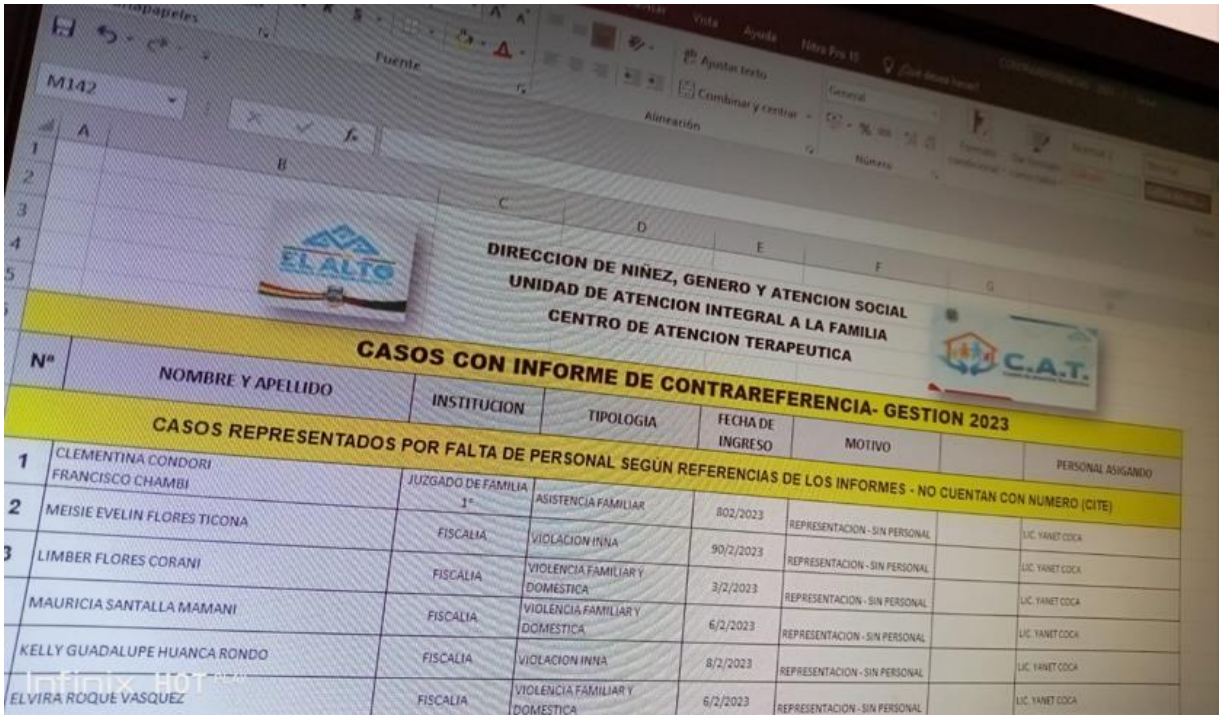


LIC. YANET COCA

GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO
DIRECCION DE NIÑEZ, GENERO Y ATENCION SOCIAL
UNIDAD DE ATENCION INTEGRAL A LA FAMILIA
CENTRO DE ATENCION TERAPEUTICA
REGISTRO DE CASOS APERTURADOS - 2023

N°	FECHA	NUMERO DE CASO	NOMBRE Y APELLIDO	EDAD	SEXO		INSTITUCION DERIVANTE						
					Masculino	Femenino	DNA	SLIM	JUZGADO	FISCALIA	HOGARES	OTROS	
1	19/1/2023	407/2022	ARLEN SUSANA QUISBERT SUSANA VALENTINA QUISBERT	10 8		F F	DNA - 3 DNA - 3						
2	18/1/2023	408/2022	MARIA RENE PAUCARA RAMO	15		F				FISCALIA			
3	16/1/2023	409/2022	ELIZABETH MERY GUARACHI AYALA	30		F				FISCALIA			
4	16/1/2023	410/2022	MAGDALENA AYALA TARQUI	48		F				FISCALIA			
5	21/1/2023	411/2022	ESPERANZA BAUTISTA	58		F	DNA - 3						
6	23/1/2023	412/2022	EMILY GALILEO MAMANI	9		F			JUZGADO				
7	16/1/2023	413/2022	PATRICIA ALEJANDRA FLORES HAROLD FLORES QUISBERT	2 8	M	F F			JUZGADO				
8	13/10/2022	284/2022	FELIPA COLQUE PAREDES FABIOLA CUSI CIOLQUE	60 44		F F		SLIM - D4		FISCALIA			
		291/2022	JENNETH VALDIVIA FLORES	33		F			JUZGADO				

Anexo C. casos de informe

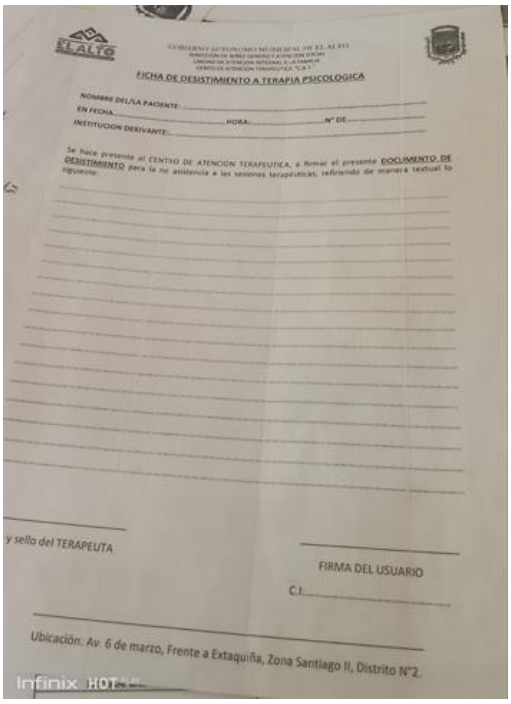


DIRECCION DE NIÑEZ, GENERO Y ATENCION SOCIAL
UNIDAD DE ATENCION INTEGRAL A LA FAMILIA
CENTRO DE ATENCION TERAPEUTICA

CASOS CON INFORME DE CONTRAREFERENCIA- GESTION 2023

Nº	NOMBRE Y APELLIDO	INSTITUCION	TIPOLOGIA	FECHA DE INGRESO	MOTIVO		PERSONAL ASIGANDO
CASOS REPRESENTADOS POR FALTA DE PERSONAL SEGÚN REFERENCIAS DE LOS INFORMES - NO CUENTAN CON NUMERO (CITE)							
1	CLEMENTINA CONDORI FRANCISCO CHAMBI	JUZGADO DE FAMILIA 1°	ASISTENCIA FAMILIAR	802/2023	REPRESENTACION - SIN PERSONAL		LIC. YANET COCA
2	MEISIE EVELIN FLORES TICONA	FISCALIA	VIOLACION INNA	90/2/2023	REPRESENTACION - SIN PERSONAL		LIC. YANET COCA
3	LIMBER FLORES CORANI	FISCALIA	VIOLENCIA FAMILIAR Y DOMESTICA	3/2/2023	REPRESENTACION - SIN PERSONAL		LIC. YANET COCA
	MAURICIA SANTALLA MAMANI	FISCALIA	VIOLENCIA FAMILIAR Y DOMESTICA	6/2/2023	REPRESENTACION - SIN PERSONAL		LIC. YANET COCA
	KELLY GUADALUPE HUANCA RONDO	FISCALIA	VIOLACION INNA	8/2/2023	REPRESENTACION - SIN PERSONAL		LIC. YANET COCA
	ELVIRA ROQUE VASQUEZ	FISCALIA	VIOLENCIA FAMILIAR Y DOMESTICA	6/2/2023	REPRESENTACION - SIN PERSONAL		LIC. YANET COCA

Anexo D: Registro de desistimiento



FICHA DE DESISTIMIENTO A TERAPIA PSICOLOGICA

NOMBRE DEL/LA PACIENTE: _____

EN FECHA_____ HORA:_____ N° DE_____

INSTITUCION DERIVANTE:_____

Se hace presente al CENTRO DE ATENCION TERAPEUTICA, a firmar el presente DOCUMENTO DE DESISTIMIENTO para la no asistencia a las sesiones terapéuticas, refiriendo de manera textual lo siguiente:

_____

y sello del TERAPEUTA

FIRMA DEL USUARIO

C.I._____

Ubicación: Av. 6 de marzo, Frente a Extaquiña, Zona Santiago II, Distrito N°2.

MANUAL DE

USUARIO

SISTEMA PARA EL REGISTRO DE CASOS TERAPEÚTICOS

“SICAT.” G.A.M.E.A.

Versión V1

UNIDAD DE SISTEMAS

1 INTRODUCCIÓN.

El sistema SICAT "Sistema para el Registro de Casos Terapeúticos" es un sistema que permite tener un registro sobre los casos que se presentan bajo el cargo del Centro de Atención Terapéutica del Gobierno Autónomo Municipal de El Alto.

2 OBJETIVOS.

Instruir al usuario en el uso correcto del sistema y la solución de los problemas que puedan suceder en la cada operación.

3 REQUERIMIENTOS.

3.1 CONOCIMIENTOS BÁSICOS.

Manejo de Navegadores de Internet.

3.2 REQUERIMIENTOS DE HARDWARE.

Computadora de escritorio o personal.

Conexión a internet.

3.3 REQUERIMIENTOS DE SOFTWARE.

Sistema Operativo: Windows (7 o superior)

Navegadores: Google Chrome (3.5 o superior), Opera (3.5 o superior), Microsoft Edge (3.5 o superior), Mozilla Firefox (3.5 o superior), o navegadores de distribuciones Linux.

3.4 REQUERIMIENTOS DE RED.

Conexión a la intranet de la Alcaldía de El Alto.

4 GENERALIDADES.

4.1 INGRESO AL SISTEMA.

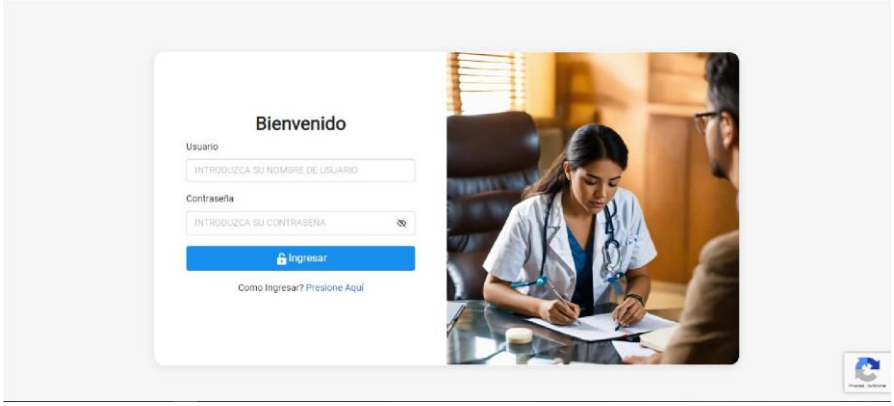
En la url de su navegador ingrese el siguiente enlace: , inmediatamente le aparecerá una ventana como la mostrada en la Figura 1. Figura 1.

Dirección URL para el acceso del sistema SiCat

4.2 INICIO DE SESIÓN.

En la ventana de la Figura 2 es donde el usuario deberá ingresar sus datos de autenticación para ingresar al sistema. Figura 2.

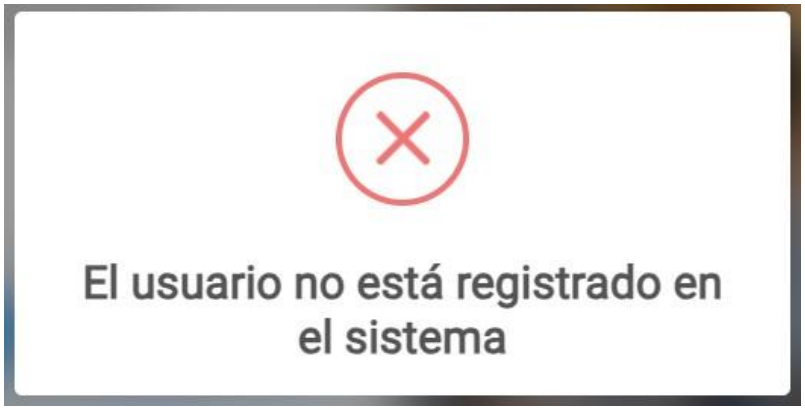
Login (Inicio de sesión) del Sistema SiCat.



Si los datos ingresados son los correctos, se le permitirá el acceso al sistema como se puede observar en la Figura 5 (cabe recalcar que esta ventana principal es diferente para cada usuario) y si no son los datos correctos le aparecerá el mensaje de la Figura 3.

Figura 3.

Usuario no Autenticado



4.3 CERRAR SESIÓN.

Para cerrar la sesión nos dirigimos a la parte superior derecha de la ventana para desplegar las opciones y elegimos Cerrar Sesión.

Tal y como se puede observar en la Figura 4. Figura 4.

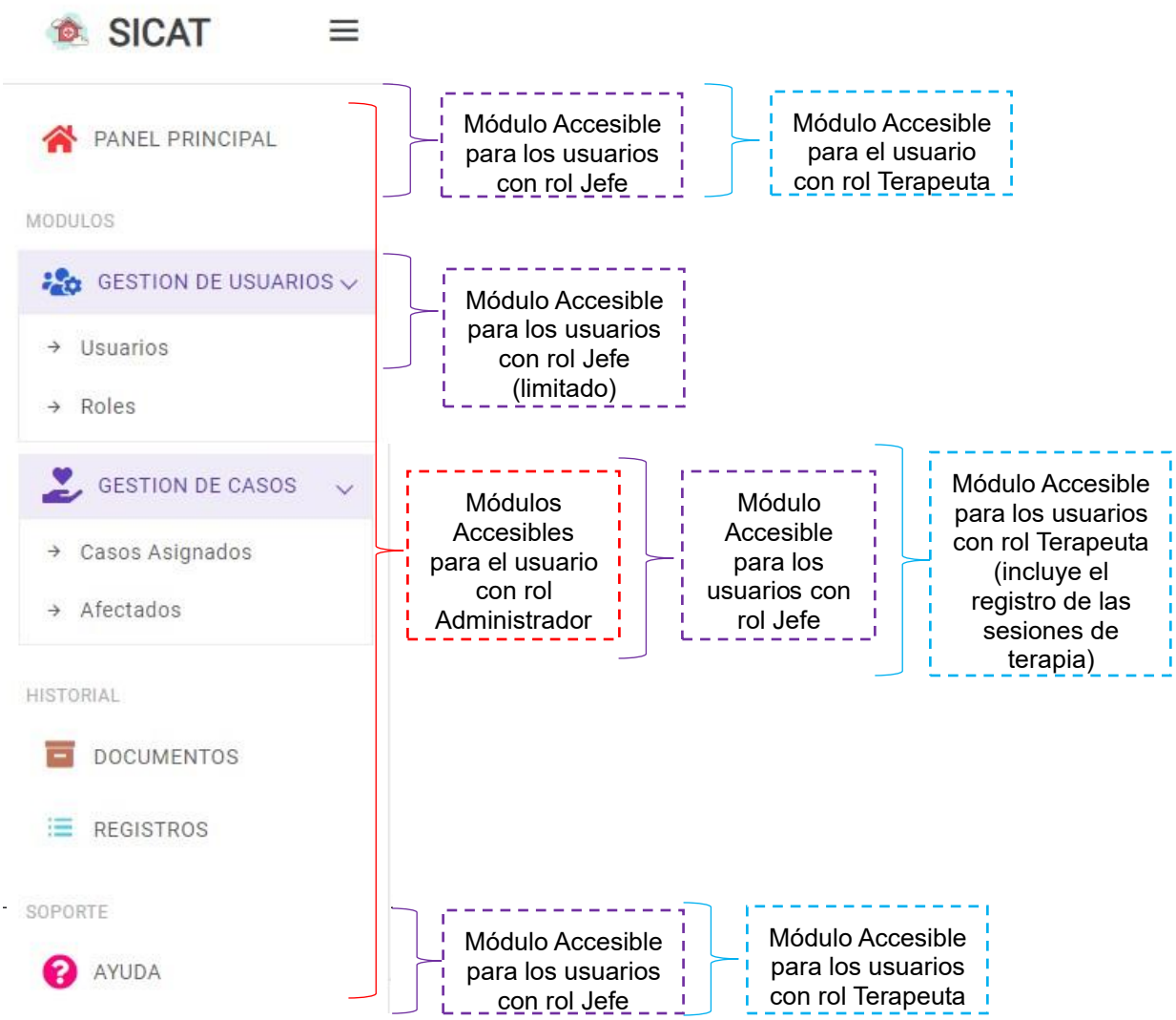
Opciones del menú de usuario: Cerrar Sesión

5 MENÚ LATERAL IZQUIERDO.

En esta sección se encontrará el menú de vistas disponibles, para los usuarios.

Las opciones están limitadas según el rol del usuario, como se puede ver en la Figura 5.

Módulos disponibles para todos los usuarios



5.1 PÁGINA PRINCIPAL.

Como página principal se tiene algunos gráficos que representan los datos que se tienen en el sistema, con respecto a la cantidad de casos que se reciben. Cabe

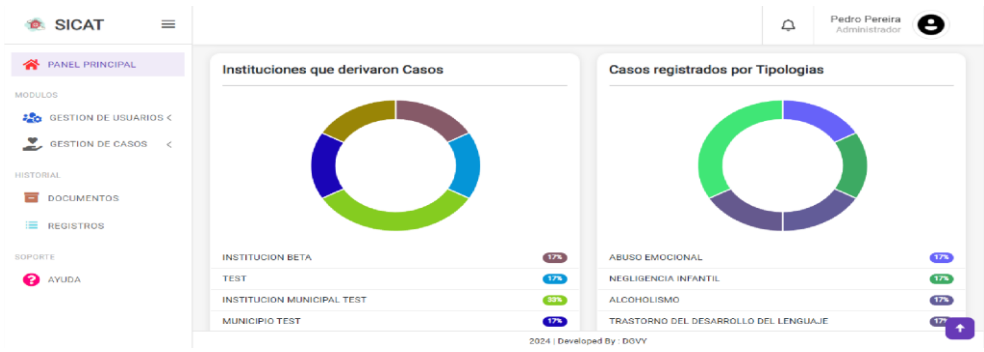
recalcar que estos datos se muestran acorde a los roles de usuario, es decir, los Administradores y Jefes pueden ver los gráficos de todos los datos del sistema, mientras que los Terapeutas solo podrán ver los gráficos de todos los casos a los que hayan sido asignados.

Figura 6.

Página principal del sistema, gráficos acerca de los casos registrados

Figura 7.

Página principal del sistema, gráficos acerca de los casos registrados



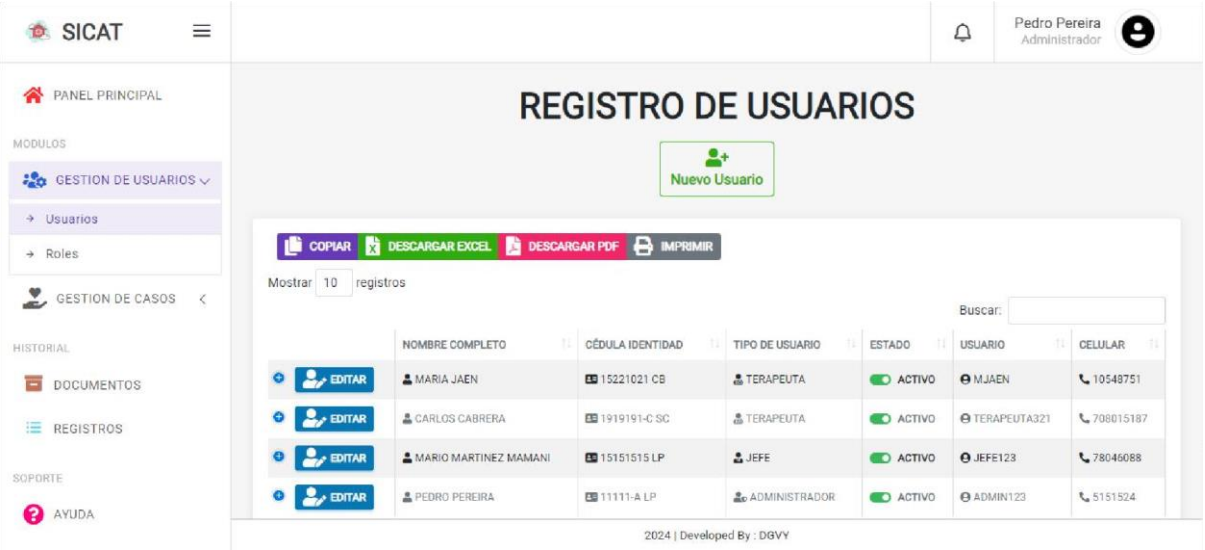
5.2 MÓDULO DE USUARIOS / MÓDULO DE TERAPEUTAS.

En este módulo se puede visualizar la tabla de registros y el estado de los usuarios registrados en el sistema.

Esta vista solo es accesible para usuarios con el rol: Administrador o Jefe, siendo que los primeros pueden registrar a cualquier usuario en el sistema y los segundos solo pueden registrar Terapeutas.

Figura 8.

Vista principal del Módulo de Usuarios



5.2.1 Registrar un Nuevo Usuario:

Paso 1: Para registrar a un nuevo Usuario primero debe hacer clic al botón

Nuevo Usuario. Figura 9.

Botón para el registro de un Nuevo Usuario

Paso 2: Debe rellenar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 10.

Ventana donde se encuentra el formulario de registro de usuarios

Paso 3: Presionar el Botón de Registrar para guardar el registro, caso contrario presione Cancelar.

Figura 11.

Botones para Registrar o Cancelar el registro de un nuevo usuario

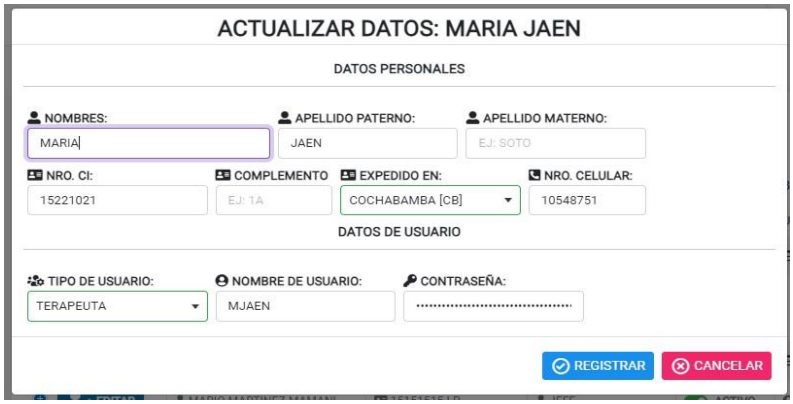
5.2.2 Actualizar Usuario:

Paso 1: Para actualizar los datos de un Usuario seleccionado primero debe hacer clic al botón correspondiente Figura 12.

Botones para la actualización de datos de un usuario

Paso 2: Debe editar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro. Figura 13.

Ventana para la actualización de datos de un usuario



Paso 3: Presionar el Botón de Registrar para actualizar el registro, caso contrario presione Cancelar. Figura 14.

Botones para Actualizar o Cancelar la actualización de un usuario

5.2.3 Cambiar el Estado de un Usuario:

Paso 1: Para actualizar el estado de un Usuario (actualizar el estado, permite denegar los permisos de acceso al sistema a cualquier usuario dependiendo de la situación), debe hacer clic en el switch asignado.

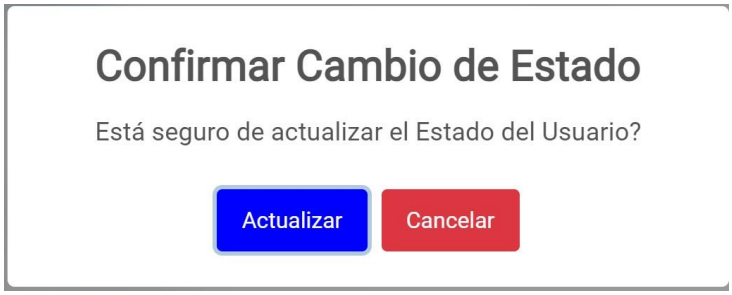
Figura 15.

Botón para actualizar el estado de un usuario

Paso 2: Presione el botón Actualizar para confirmar el Cambio de Estado, caso contrario presione Cancelar.

Figura 16.

Botones para la confirmación del cambio estado de un usuario



5.3 MÓDULO DE ROLES DE USUARIO.

En este módulo solo es visible para el usuario con rol Administrador, donde se puede visualizar la tabla de registros con los roles de usuario establecidos. Figura 17.

Vista principal del módulo de Roles de Usuario

5.4 MÓDULO DE GESTIÓN DE CASOS.

En este módulo se puede visualizar la tabla de registros (como se puede observar en la Figura 18, la tabla de registros de los casos está organizada según su estado, así puede revisar el estado de cada caso según presente cambios con el tiempo) y el estado de los casos que se registren.

Este módulo es accesible para todos los roles de usuario, sin embargo, los Administradores y Jefes pueden realizar mas acciones que los Terapeutas. (Los Administradores y Jefes pueden registrar nuevos casos, actualizar los datos del caso, registrar los afectados que cada caso puede tener, asignar a un terapeuta para dicho caso y cancelar o finalizar el caso dependiendo de la situación, los Terapeutas a los que se les asignaron casos pueden ver los detalles del mismo, y de la lista de afectados que tengan cada caso seleccionar cuales de ellos están dispuestos a tomar las sesiones de terapia y sean pacientes).

Figura 18.

Vista principal del módulo de gestión de casos

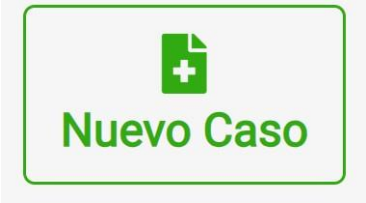
5.4.1 Registrar un Nuevo Caso:

Paso 1: Para registrar un nuevo Caso primero debe hacer clic al botón

Nuevo Caso.

Figura 19.

Botón para el registro de un nuevo Caso



Paso 2: Debe rellenar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 20.

Ventana donde se encuentra el formulario de registro de casos

Paso 3: Presionar el Botón de Registrar para guardar el registro, caso contrario presione Cancelar.

Figura 21.

Botones para Registrar o Cancelar el registro de un nuevo Caso

5.4.2 Actualizar Caso:

Paso 1: Para actualizar los datos de un Caso seleccionado primero debe hacer clic al menú de acciones disponibles y luego al botón correspondiente.

Figura 22.

Botones para la actualización de datos de un caso

Paso 2: Debe editar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro. Figura 23.

Ventana para la actualización de datos de un caso

Paso 3: Presionar el Botón de Registrar para actualizar el registro, caso contrario presione Cancelar. Figura 24.

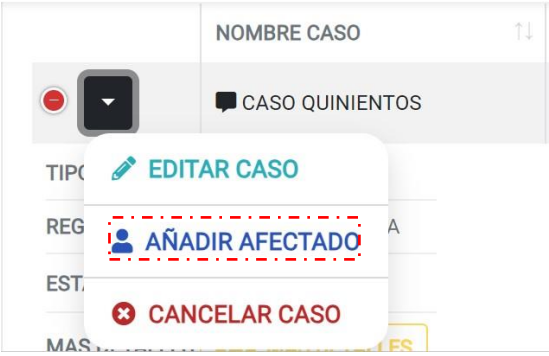
Botones para Actualizar o Cancelar la actualización de un caso

5.4.3 Añadir afectado/s:

Paso 1: Para añadir a los afectados que pertenecen a un caso registrado primero debe hacer clic al menú de acciones disponibles y luego al botón correspondiente.

Figura 25.

Botón para añadir afectados



Paso 2: Debe rellenar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 26.

Ventana donde se encuentra el formulario para el registro de afectados

Paso 3: Presionar el Botón de Registrar para guardar el registro, caso contrario presione Cancelar (Al registrar como mínimo a un afectado, se tendrán también nuevas opciones como ser el asignar a un terapeuta para que se encargue del caso). Figura 27.

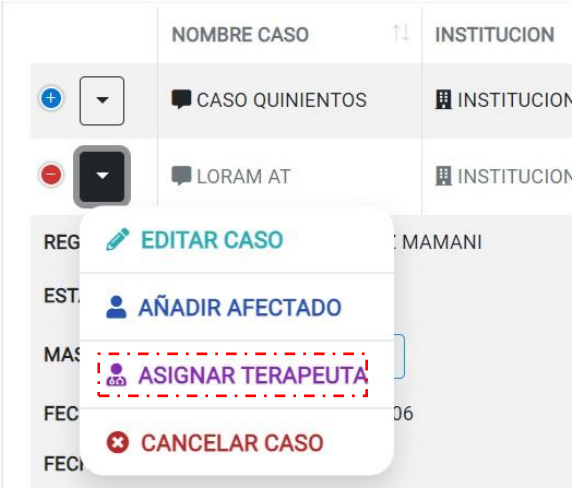
Botones para Registrar o Cancelar el registro de un nuevo Caso

5.4.4 Asignar Terapeuta:

Paso 1: Para asignar a un terapeuta, en principio se debe registrar como mínimo a un afectado, caso contrario esta opción permanecerá bloqueada. Si ya se tienen afectados registrados, primero debe hacer clic al menú de acciones disponibles y luego al botón correspondiente.

Figura 28.

Botón para asignar un terapeuta



Paso 2: Debe seleccionar del listado de los terapeutas a uno que tenga horarios disponibles para llevar a cabo las sesiones de terapia, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 29.

Ventana donde se encuentra el formulario para la selección del terapeuta

Paso 3: Presionar el Botón de Registrar para guardar el registro, caso contrario presione Cancelar (Una vez se asigna un terapeuta al caso, este cambia su estado de Pendiente a Aprobado y a partir de ese momento el

Terapeuta es quien se encarga de los procesos siguientes). Figura 30.

Botones para Registrar o Cancelar la asignación del terapeuta

5.4.5 Concluir y Cancelar Caso:

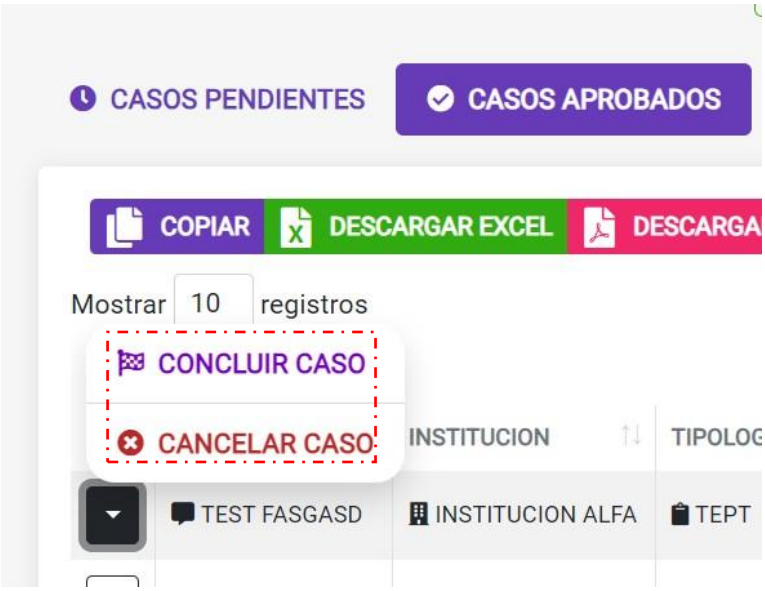
Paso 1: La opción de cancelar se encuentra disponible desde un inicio, debido a diversas situaciones que se puedan presentar es posible cancelar el proceso de un caso, por

otro lado, la opción de concluir se encuentra disponible solo después de que se asigna un terapeuta al caso.

Primero debe hacer clic en el/los botones correspondientes.

Figura 31.

Botones para concluir o cancelar un caso



Paso 2: Presione el botón Actualizar para confirmar el Cambio de Estado, caso contrario presione Cancelar.

Figura 32.

Botones para la confirmación del cambio estado del caso

5.4.6 Seleccionar Pacientes:

Paso 1: Para realizar la selección de los pacientes, en principio esto solo lo puede realizar un Terapeuta.

Primero debe hacer clic en el botón correspondiente.

Figura 33.

Botón para seleccionar los pacientes

Paso 2: Debe seleccionar del listado de los afectados registrados a aquellos que confirmen su disponibilidad y asistencia a las sesiones de terapia, esto se realiza tickeando en el recuadro de la columna Seleccionar, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 34.

Ventana donde se encuentra el formulario para la selección de los pacientes

Paso 3: Presionar el Botón de Registrar para guardar el registro, caso contrario presione Cancelar (Una vez se selecciona a los pacientes la opción desaparecerá y otra tomará su lugar). Figura 35.

Botones para Registrar o Cancelar la asignación del terapeuta

5.4.7 Subir Documentos:

Paso 1: Para subir documentos relacionados con el caso, en principio esto solo lo puede realizar un Terapeuta.

Primero debe hacer clic en el botón correspondiente.

Figura 36.

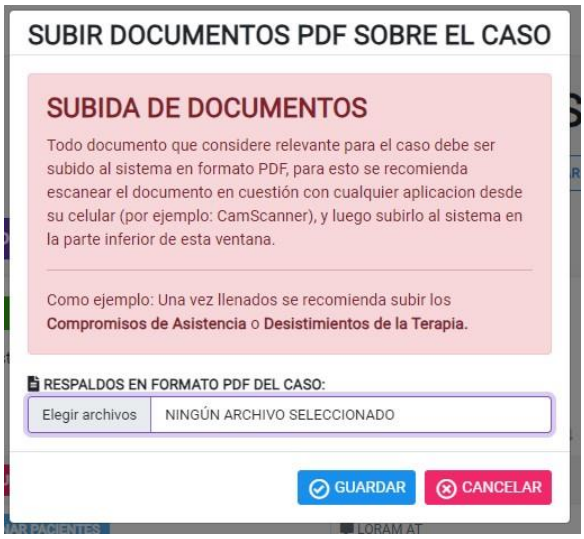
Botón para seleccionar los pacientes

Paso 2: Debe subir documentos en pdf sobre todo que respalden la decisión de los afectados sea si estos deciden no tomar las sesiones de terapia su debido Desistimiento y si deciden tomar las sesiones su debido Compromiso

de Asistencia, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 37.

Ventana donde se encuentra el formulario para subir documentos



Paso 3: Presionar el Botón de Registrar para guardar el registro, caso contrario presione Cancelar. Figura 38.

Botones para Registrar o Cancelar la asignación del terapeuta

5.4.8 Más Detalles:

Este botón le permitirá ver mas información con respecto a los casos. Figura 39.

Botón para visualizar más detalles

Figura 40.

Ventana con más información sobre los casos

5.5 MÓDULO DE AFECTADOS / PACIENTES.

En este módulo se puede visualizar la tabla de registros y el estado de las dependencias de los afectados y los pacientes registrados en el sistema.

Es accesible por todos los roles de usuario, sin embargo, los Administradores y Jefes pueden ver a los afectados (un afectado siempre relacionado a un caso) mientras que los Terapeutas solo podrán visualizar a los afectados que hayan decidido tomar terapias y se vuelvan Pacientes.

Figura 41.

Vista principal del módulo de afectados / pacientes desde un Administrador

5.5.1 Editar Afectado:

Paso 1: Esta opción solo está disponible para Administradores y Jefes. Para actualizar los datos de una Afectado seleccionado primero debe hacer clic al botón correspondiente Figura 42.

Botones para la actualización de datos de un afectado

Paso 2: Debe editar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 43.

Ventana para la actualización de datos de un afectado

Paso 3: Presionar el Botón de Registrar para actualizar el registro, caso contrario presione Cancelar. Figura 44.

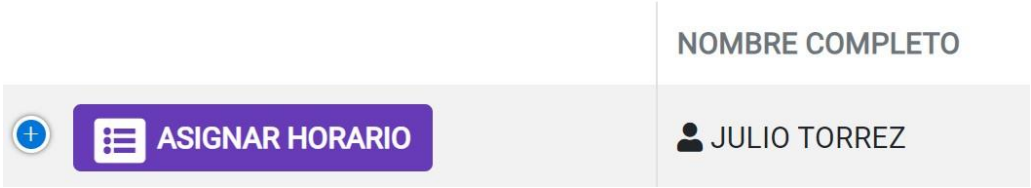
Botones para Actualizar o Cancelar la actualización de una dependencia

5.5.2 Asignar Horario:

Paso 1: Esta opción solo está disponible para Terapeutas.

Una vez un afectado de un caso, decide tomar las sesiones de terapia se debe asignar un horario para que el asista, primero debe hacer clic al botón correspondiente Figura 45.

Botones para la asignación de horarios para un paciente



Paso 2: Debe rellenar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 46.

Ventana para la asignación de horario

Paso 3: Presionar el Botón de Registrar para actualizar el registro, caso contrario presione Cancelar.

Figura 47.

Botones para Actualizar o Cancelar la asignación de horario

5.5.3 Ver Horario Asignado:

Este botón le permitirá ver más información con respecto a los horarios de los pacientes. Figura 48.

Botón para visualizar el horario asignado

Figura 49.

Ventana con más información sobre el horario

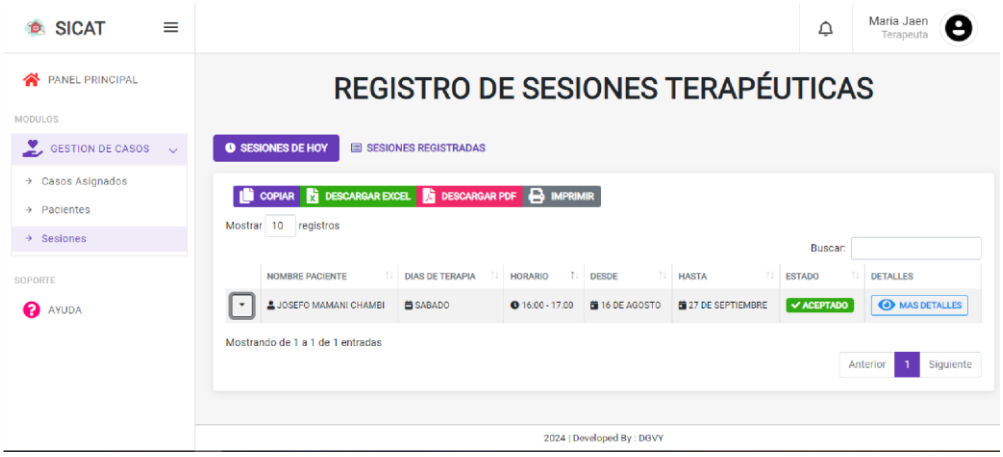
5.6 MÓDULO DE SESIONES DE TERAPIA.

En este módulo se puede visualizar la tabla de registros y el estado de las sesiones que lleva cada terapeuta.

Solo los usuarios con rol Terapeuta, pueden visualizar este modulo y registrar los datos que consideren pertinentes para cada sesión.

Figura 50.

Vista principal del módulo de sesiones de terapia



5.6.1 Registrar Observaciones de una Sesión:

Paso 1: Para registrar las observaciones de una sesión actual se debe presionar el botón correspondiente. Figura 51.

Botón para el registro de observaciones en una sesión de terapia

Paso 2: Debe rellenar los campos disponibles en la ventana emergente, el sistema verificará el llenado de los campos obligatorios y dependiendo de los mismos, le permitirá guardar el registro.

Figura 52.

Ventana donde se encuentra el formulario de registro de observaciones

Paso 3: Presionar el Botón de Registrar para guardar el registro, caso contrario presione Cancelar. Figura 53.

Botones para Registrar o Cancelar el registro de observaciones

5.6.2 Más Detalles:

Este botón le permitirá ver mas información con respecto a los casos. Figura 54.

Botón para visualizar más detalles

Figura 55.

Ventana con más información sobre las sesiones

5.7 MÓDULO DE HISTORIAL DE DOCUMENTOS.

En este módulo se puede visualizar la tabla de registros y el estado de todos los documentos almacenados en el sistema, solo está disponible para los usuarios con rol Administrador.

Figura 56.

Vista principal del módulo de documentos

5.8 MÓDULO DE HISTORIAL DE REGISTROS.

En este módulo se puede visualizar la tabla de registros y el estado de todos los cambios realizados en el sistema, solo está disponible para los usuarios con rol Administrador. Figura 57.

Vista principal del módulo de registros