

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**“SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS
CLÍNICAS”**

**CASO: FACULTAD DE MEDICINA, INSTITUTO DE
GENÉTICA, UNIDAD DE GENÉTICA MÉDICA**

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMÁTICA
MENCION INGENIERIA DE SISTEMAS INFORMÁTICOS

POR: NEIL ANGEL GRANEROS FLORES
TUTORA: PH.D. MARISOL TELLEZ RAMIREZ

La Paz - Bolivia
2024

HOJA DE CALIFICACIONES
UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

Proyecto de grado:
“SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS”
CASO: FACULTAD DE MEDICINA, INSTITUTO DE GENÉTICA, UNIDAD DE GENÉTICA MÉDICA

Presentado por: Neil Angel Graneros Flores.

Para optar el grado académico de Licenciado en Informática

Mención en Ingeniería de Sistemas Informáticos

Nota numeral:

Nota Literal:

Ha sido:

Director de la carrera de informática: PhD. José María Tapia Baltazar

Tutora: Ph.D. Marisol Tellez Ramirez

Asesor:

Tribunal:

Tribunal:

Tribunal:



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciéndola referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria

A Dios, por darme la vida llena de salud hasta ahora, por fortalecerme en momentos de dificultad y adversidad, iluminando mi camino a cada momento y abrirme las puertas a oportunidades de aprendizaje.

A mis señores padres Mary Flores Callisaya y Antonio Graneros Benavides, quienes me apoyaron día a día en cada etapa de mi vida, me dieron consejos de gran valor, me enseñaron valores y por darme todo su amor, gracias mamá y papá.

A mi hermana Vania Graneros Flores por ser un modelo a seguir y por ser una hermana ejemplar, a mis hermanos Gael y Abby por acompañarme en este camino.

A mis abuelos Dora Benavides, Angel Graneros y Teodora Callisaya por la enseñanza y cariño que brindaron en mi en cada momento.

A todas las personas que me han acompañado en este viaje que aún continúa, gracias. Prometo seguir demostrando que todavía tengo mucho más para dar.

Graneros F. Neil

Agradecimientos

Quisiera expresar mi profundo agradecimiento a la Universidad Mayor de San Andrés y a la Carrera de Informática, que no solo fueron mi segunda casa sino también el lugar donde adquirí valiosos conocimientos a lo largo de estos años. Su apoyo y enseñanzas han sido fundamentales en mi desarrollo personal y profesional.

A mi familia, en especial a mis Padres, Hermana, y hermanos, por toda su colaboración en todas las etapas de mi vida, por sus palabras de aliento y consejos que iluminaron mi camino en los momentos más difíciles.

A todos los docentes, quienes me concedieron sus conocimientos. Al M. Sc. Aldo Valdés y a la Ph.D. Marisol Téllez por la colaboración prestada en la elaboración del proyecto. Al Lic. Florencio Antonio por el recibimiento en la unidad RED-UMSALUD. Al Doctor Rafael Montaño del Instituto de Genética, quien me brindo su tiempo y cooperación en la elaboración del proyecto.

A Red Velvet, EXO y NCT por acompañarme en esas noches de desvelo, en cada instante alegrando el día con su música, muchas gracias por ser un apoyo en esos momentos complicados y difíciles.

A mis amigos y compañeros por darme consejos y aliento para salir adelante, mencionando a algunos: Priscila, Pilar, Kamil, Boris, Laura, David, Erick, Fernando entre otros, muchas gracias por su amistad.

*A los grupos de estudio GNU Linux y Ctrl+Dev por el conocimiento dado y compartido entre todos. **Muchas Gracias.***

neilgraneros11@gmail.com

RESUMEN

La administración eficiente de la información médica es crucial para la optimización de los procesos clínicos y la mejora de la atención al paciente en cualquier institución de salud. En respuesta a esta necesidad, el presente proyecto de grado se enfocó en el desarrollo e implementación de un sistema web destinado a la gestión de historias clínicas de la Unidad de Genética Médica del Instituto de Genética Médica, dependiente de la Facultad de Medicina. Este sistema busca facilitar el acceso y manejo de la información clínica de los pacientes, contribuyendo a la satisfacción y eficiencia del personal médico.

El cual, es capaz de administrar de manera integral las historias clínicas mediante módulos especializados que cubren distintas necesidades operativas y administrativas dentro del entorno médico. El proyecto fue desarrollado utilizando la metodología Agile Unified Process (AUP) para una gestión del desarrollo más flexible y adaptativa, y UWE para el modelado preciso de los componentes del sistema. Las tecnologías seleccionadas para la implementación incluyeron PostgreSQL, Node.js y React.js, asegurando una plataforma robusta, escalable y de fácil acceso para los usuarios finales.

El presente proyecto se estructura en seis capítulos que abordan distintos aspectos relacionados con el tema en cuestión. En el primer capítulo, se presenta el marco referencial, proporcionando un contexto y fundamentación para el estudio. Posteriormente, en el segundo capítulo, se explora en detalle el marco teórico, analizando conceptos y teorías relevantes para comprender mejor el problema de estudio. En el tercer capítulo, se aborda el marco aplicativo, donde se describen las metodologías y herramientas utilizadas para llevar a cabo el proyecto. Luego, en el cuarto capítulo, se examinan aspectos de calidad y seguridad relacionados con el desarrollo del sistema. En el quinto capítulo, se evalúan los costos y beneficios asociados al sistema. Finalmente, en el sexto capítulo, se exponen las conclusiones y recomendaciones obtenidas tras el desarrollo y la implementación del sistema, detallando las acciones llevadas a cabo y proporcionando directrices para futuras mejoras y acciones prácticas en el sistema.

Palabras Clave: Historia clínica, Gestión de Información Médica, Sistemas Web, Historia Clínica Electrónica.

Metodología: AUP y UWE

ABSTRACT

The efficient management of medical information is crucial for the optimization of clinical processes and the improvement of patient care in any health institution. In response to this need, this degree project focused on the development and implementation of a web-based system for the management of medical records of the Medical Genetics Unit of the Institute of Medical Genetics, under the Faculty of Medicine. This system seeks to facilitate the access and management of patients' clinical information, contributing to the satisfaction and efficiency of the medical staff.

It is able to comprehensively manage medical records through specialized modules that cover different operational and administrative needs within the medical environment. The project was developed using the Agile Unified Process (AUP) methodology for a more flexible and adaptive development management, and UWE for the accurate modeling of the system components. Technologies selected for implementation included PostgreSQL, Node.js and React.js, ensuring a robust, scalable and easily accessible platform for end users.

This project is structured in six chapters that address different aspects related to the topic in question. In the first chapter, the frame of reference is presented, providing a context and rationale for the study. Subsequently, in the second chapter, the theoretical framework is explored in detail, analyzing relevant concepts and theories to better understand the study problem. The third chapter deals with the application framework, describing the methodologies and tools used to carry out the project. Then, in the fourth chapter, quality and safety aspects related to the development of the system are examined. The fifth chapter evaluates the costs and benefits associated with the system. Finally, the sixth chapter presents the conclusions and recommendations obtained after the development and implementation of the system, detailing the actions carried out and providing guidelines for future improvements and practical actions in the system.

Keywords: Medical records, Medical Information Management, Web Systems, Electronic Medical Records.

Methodology: AUP and UWE

ÍNDICE

| | |
|--|----------|
| CAPÍTULO I MARCO REFERENCIAL | 1 |
| 1.1 Introducción..... | 1 |
| 1.2 Antecedentes..... | 2 |
| 1.2.1 Estado del Arte..... | 3 |
| 1.2.2 Trabajos Similares..... | 6 |
| 1.2.3 Antecedentes Institucionales..... | 10 |
| 1.3 Planteamiento del Problema | 11 |
| 1.3.1 Problemas secundarios | 12 |
| 1.3.2 Formulación del Problema..... | 13 |
| 1.4 Objetivos..... | 13 |
| 1.4.1 Objetivo General | 13 |
| 1.4.2 Objetivos Específicos..... | 13 |
| 1.5 Justificación | 14 |
| 1.5.1 Justificación Social..... | 14 |
| 1.5.2 Justificación Económica..... | 15 |
| 1.5.3 Justificación Tecnológica..... | 15 |
| 1.6 Alcances y Límites | 15 |

| | | |
|--|---|-----------|
| 1.6.1 | Alcance..... | 16 |
| 1.6.2 | Límites..... | 16 |
| CAPÍTULO II MARCO TEÓRICO | | 18 |
| 2.1 | Ingeniería de Software..... | 18 |
| 2.1.1 | El Proceso de Software | 19 |
| 2.1.2 | Modelos de Desarrollo de Software..... | 19 |
| 2.1.3 | Metodología Ágil en el Desarrollo de Software | 23 |
| 2.2 | Proceso Unificado Ágil (AUP)..... | 26 |
| 2.2.1 | Principios del Proceso Unificado Ágil (AUP). | 26 |
| 2.2.2 | Ciclo de Vida del Proceso Unificado Ágil (AUP). | 27 |
| 2.2.3 | Fases e Hitos del Ciclo de Vida. | 28 |
| 2.2.4 | Disciplinas..... | 32 |
| 2.2.5 | Hitos | 33 |
| 2.2.6 | Ventajas y desventajas de la implementación..... | 35 |
| 2.3 | Ingeniería Web | 35 |
| 2.3.1 | La Ingeniería Web Dirigida por Modelos | 36 |
| 2.3.2 | Metodologías en Ingeniería Web | 37 |
| 2.4 | UWE (UML-Based Web Engineering). | 39 |

| | |
|---|-----------|
| 2.4.1 Modelos - Fases de UWE..... | 41 |
| 2.5 Tecnologías de Software | 49 |
| 2.5.1 JavaScript | 49 |
| 2.5.2 PostgreSQL | 50 |
| 2.5.3 Node JS | 50 |
| 2.5.4 Sequelize ORM | 50 |
| 2.5.5 React JS..... | 51 |
| 2.5.6 Playwright | 51 |
| 2.6 Historia Clínica..... | 52 |
| 2.6.1 Características de una Historia Clínica | 53 |
| 2.6.2 Gestión de Historias Clínicas..... | 55 |
| 2.6.3 Historia Clínica Electrónica (HCE). | 57 |
| CAPÍTULO III MARCO APLICATIVO..... | 59 |
| 3.1 Introducción..... | 59 |
| 3.2 Fase de Inicio..... | 61 |
| 3.2.1 Situación actual | 62 |
| 3.2.2 Especificación de Requerimientos | 64 |
| 3.3 Fase de Elaboración..... | 66 |

| | | |
|---|--|------------|
| 3.3.1 | Descripción de los Actores..... | 66 |
| 3.3.2 | Modelo de Casos de Uso del Negocio | 67 |
| 3.3.3 | Modelo de objetos del negocio | 70 |
| 3.4 | Fase de Construcción..... | 74 |
| 3.4.1 | Casos de uso del Sistema | 74 |
| 3.4.2 | Especificaciones de Casos de Uso | 75 |
| 3.4.3 | Asignación de Casos de Uso a Módulos del Sistema | 85 |
| 3.4.4 | Modelo de Contenido..... | 86 |
| 3.4.5 | Modelo de Navegación | 96 |
| 3.4.6 | Modelo de Presentación | 97 |
| 3.4.7 | Modelo del flujo de proceso..... | 98 |
| 3.5 | Fase de Transición..... | 105 |
| 3.5.1 | Modelo de Despliegue..... | 106 |
| 3.5.2 | Interfaces de usuario | 107 |
| 3.5.3 | Pruebas End-To-End con Playwright..... | 115 |
| CAPÍTULO IV CALIDAD Y SEGURIDAD..... | | 118 |
| 4.1 | Introducción..... | 118 |
| 4.2 | Norma ISO/IEC 25010..... | 118 |

| | | |
|---|--|-----|
| 4.2.1 | Características de la norma ISO/IEC 25010 | 119 |
| 4.3 | Adecuación Funcional | 120 |
| 4.4 | Usabilidad..... | 125 |
| 4.5 | Fiabilidad..... | 127 |
| 4.6 | Mantenibilidad..... | 129 |
| 4.7 | Portabilidad..... | 130 |
| 4.8 | Seguridad..... | 132 |
| 4.8.1 | Descripción de criterios de evaluación (CE)..... | 135 |
| 4.8.2 | Métricas para cada subcaracterística..... | 138 |
| 4.8.3 | Ejecutar evaluación | 141 |
| 4.9 | Calidad Global..... | 142 |
| CAPÍTULO V COSTO BENEFICIO | 143 | |
| 5.1 | Introducción..... | 143 |
| 5.2 | Modelo COCOMO II | 143 |
| 5.2.1 | Líneas de código fuente (SLOC)..... | 144 |
| 5.2.2 | Definiciones de los tipos SLOC..... | 144 |
| 5.3 | Cálculo de costo de Software | 146 |
| 5.4 | Costo de Elaboración del Proyecto..... | 148 |

| | | |
|---|--|------------|
| 5.5 | Costo de Implementación del Proyecto | 149 |
| 5.6 | Costo Total | 149 |
| 5.7 | Cálculo de Valor Actual Neto - VAN | 150 |
| 5.8 | Relación Costo Beneficio..... | 151 |
| CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES | | 153 |
| 6.1 | Introducción..... | 153 |
| 6.2 | Conclusiones..... | 153 |
| 6.3 | Recomendaciones | 155 |
| REFERENCIAS BIBLIOGRÁFICAS | | 157 |
| ANEXOS | | 163 |

ÍNDICE DE FIGURAS

| | |
|--|-----|
| Figura 1.1 Estructura Orgánica del Instituto de Genética..... | 11 |
| Figura 2.1 Capas de la ingeniería de software | 18 |
| Figura 2.2 Modelo en Cascada..... | 20 |
| Figura 2.3 Modelo en Espiral..... | 21 |
| Figura 2.4 Modelo de Prototipo | 23 |
| Figura 2.5 Ciclo de vida de AUP | 28 |
| Figura 2.6 Fases e Hitos de AUP | 33 |
| Figura 2.7 Dependencia entre modelos | 41 |
| Figura 2.8 Modelo de Requerimientos | 43 |
| Figura 2.9 Modelo de Contenido..... | 44 |
| Figura 2.10 Modelo de Navegación | 45 |
| Figura 2.11 Modelo de Presentación | 46 |
| Figura 2.12 Estructura del Proceso..... | 47 |
| Figura 2.13 Flujo del Proceso | 48 |
| Figura 2.14 Relación entre modelos..... | 49 |
| Figura 3.1 Planificación de Integración Metodológica | 61 |
| Figura 3.2 Inicio de Sesión | 107 |
| Figura 3.3 Gestión de Usuarios | 108 |
| Figura 3.4 Gestión de Pacientes | 109 |
| Figura 3.5 Gestión de Citas..... | 110 |
| Figura 3.6 Lista de Pacientes | 111 |
| Figura 3.7 Gestión de Consultas Médicas | 111 |
| Figura 3.8 Gestión de Consultas Médicas - Consulta..... | 112 |
| Figura 3.9 Gestión de Exámenes de Laboratorio | 113 |
| Figura 3.10 Gestión de Historias Clínicas – Parte uno..... | 113 |
| Figura 3.11 Gestión de Historias Clínicas – Parte dos | 114 |
| Figura 3.12 Generar reporte de historia clínica..... | 114 |
| Figura 3.13 Prueba end-to-end..... | 117 |
| Figura 4.1 Norma ISO/IEC 25010. | 119 |
| Figura 5.1 Inserción de datos para COCOMO II | 147 |
| Figura 5.2 Resultado de estimación de costos | 148 |

ÍNDICE DE TABLAS

| | |
|--|-----|
| Tabla 1.1 Aplicación de TIC's en el subsector público | 3 |
| Tabla 1.2 Aplicación de TIC's en el subsector de la Seguridad Social | 4 |
| Tabla 1.3 Aplicación de TIC's en el subsector privado y actores | 5 |
| Tabla 3.1 Fases de implementación aplicando la metodología AUP | 59 |
| Tabla 3.2 Integración AUP y UWE..... | 60 |
| Tabla 3.3 Tabla de Requerimientos Funcionales | 64 |
| Tabla 3.4 Tabla de Requerimientos no funcionales | 65 |
| Tabla 3.5 Descripción de Actores | 67 |
| Tabla 3.6 Descripción de caso de uso del negocio - Registra Paciente..... | 68 |
| Tabla 3.7 Descripción de caso de uso del negocio - Programar Cita Médica..... | 69 |
| Tabla 3.8 Descripción de caso de uso del negocio - Atender Consulta Médica | 69 |
| Tabla 3.9 Descripción de caso de uso del negocio - Realizar Exámenes de Laboratorio | 70 |
| Tabla 3.10 Especificación – Acceso al Sistema..... | 75 |
| Tabla 3.11 Especificación – Administrar Cuentas de Usuario | 77 |
| Tabla 3.12 Especificación – Gestión de Pacientes | 78 |
| Tabla 3.13 Especificación – Gestión de Citas Médicas..... | 79 |
| Tabla 3.14 Especificación – Gestión de Consultas Médicas..... | 80 |
| Tabla 3.15 Especificación – Gestión de Exámenes de Laboratorio | 81 |
| Tabla 3.16 Especificación – Gestión de Historias Clínicas | 83 |
| Tabla 3.17 Especificación – Generar Reporte de Historia Clínica..... | 84 |
| Tabla 3.18 Tabla de asignación de casos de uso a módulos del sistema | 85 |
| Tabla 4.1 Parámetros de medida y cantidad..... | 121 |
| Tabla 4.2 Parámetros de medida y ponderación..... | 122 |
| Tabla 4.3 Valores de ajuste de complejidad..... | 123 |
| Tabla 4.4 Encuesta sobre usabilidad del sistema | 126 |
| Tabla 4.5 Valores de fiabilidad de cada módulo | 127 |
| Tabla 4.6 Información requerida por el IMS | 129 |
| Tabla 4.7 Cuestionario para la característica de Seguridad | 133 |
| Tabla 4.8 Descripción de criterios de evaluación (CE) | 135 |
| Tabla 4.9 Fórmulas para las subcaracterísticas de Seguridad | 140 |
| Tabla 4.10 Calidad Global del Sistema | 142 |
| Tabla 5.1 Costo de elaboración | 149 |
| Tabla 5.2 Costo Total..... | 149 |

Tabla 5.3 Calculo de VAN.....151

ÍNDICE DE DIAGRAMAS

| | |
|--|-----|
| Diagrama 3.1 Diagrama de Actividades del flujo actual del sistema..... | 63 |
| Diagrama 3.2 Casos de uso del negocio..... | 68 |
| Diagrama 3.3 Registrar Paciente..... | 71 |
| Diagrama 3.4 Programar Cita Médica..... | 72 |
| Diagrama 3.5 Atender Consulta Médica | 73 |
| Diagrama 3.6 Realizar Exámenes de Laboratorio | 73 |
| Diagrama 3.7 Casos de uso del sistema | 74 |
| Diagrama 3.8 Diagrama de Actividades - Acceso al Sistema | 75 |
| Diagrama 3.9 Diagrama de Actividades - Administrar Cuentas de Usuario..... | 77 |
| Diagrama 3.10 Diagrama de Actividades - Gestión de Pacientes | 78 |
| Diagrama 3.11 Diagrama de Actividades - Gestión de Citas Médicas | 79 |
| Diagrama 3.12 Diagrama de Actividades - Gestión de Consultas Médicas | 80 |
| Diagrama 3.13 Diagrama de Actividades - Gestión de Exámenes de Laboratorio | 81 |
| Diagrama 3.14 Diagrama de Actividades - Gestión de Historias Clínicas..... | 82 |
| Diagrama 3.15 Diagrama de Actividades - Generar Reporte de Historia Clínica | 84 |
| Diagrama 3.16 Diagrama de Clase de Análisis de Dominio | 87 |
| Diagrama 3.17 Diagrama Entidad Relación | 88 |
| Diagrama 3.18 Diagrama Entidad Relación - Parte 1 | 89 |
| Diagrama 3.19 Diagrama Entidad Relación - Parte 2 | 90 |
| Diagrama 3.20 Diagrama de Clase de Diseño del Sistema | 92 |
| Diagrama 3.21 Diagrama de Clase de Diseño del Sistema - Parte 1 | 93 |
| Diagrama 3.22 Diagrama de Clase de Diseño del Sistema - Parte 2 | 94 |
| Diagrama 3.23 Diagrama de Clase de Diseño del Sistema - Parte 3 | 95 |
| Diagrama 3.24 Diagrama de Navegación..... | 96 |
| Diagrama 3.25 Diagrama de Presentación | 97 |
| Diagrama 3.26 Flujo del Proceso de Acceso al Sistema | 99 |
| Diagrama 3.27 Flujo del Proceso de Gestión de Usuarios | 100 |
| Diagrama 3.28 Flujo del Proceso de Gestión de Pacientes | 101 |
| Diagrama 3.29 Flujo del Proceso de Gestión de Citas Médicas | 102 |
| Diagrama 3.30 Flujo del Proceso de Gestión de Consultas Médicas..... | 103 |
| Diagrama 3.31 Flujo del Proceso de Gestión de Exámenes de Laboratorio | 104 |
| Diagrama 3.32 Flujo del Proceso de Gestión de Historias Clínicas | 105 |
| Diagrama 3.33 Flujo del Proceso Generar Reporte de Historia Clínica | 105 |

Diagrama 3.34 Modelo de Despliegue 106

ÍNDICE DE ANEXOS

| | |
|---|-----|
| Anexo 1 Historia Clínica | 164 |
| Anexo 2 Historia Clínica 2 | 165 |
| Anexo 3 Archivo Access | 166 |

CAPÍTULO I MARCO REFERENCIAL

1.1 Introducción

En el ámbito médico, la historia clínica es un documento fundamental que recoge toda la información relevante sobre la salud de un paciente, incluyendo sus antecedentes médicos, diagnósticos, tratamientos y evolución clínica. Tradicionalmente, estos registros se han mantenido en formato papel y han sido parte esencial de la atención médica.

Sin embargo, en la actualidad, estamos presenciando una transformación digital que está teniendo un profundo impacto en el sector de la salud. Esta revolución tecnológica no solo ha cambiado la forma en que los profesionales médicos gestionan los datos de los pacientes, sino que también ha alterado la forma en que los pacientes interactúan con su atención médica.

A pesar de estos avances, muchos actores en el sector médico no están completamente conscientes del impacto que la tecnología puede generar, lo que puede llevar a percances y problemas en el uso de los sistemas digitales. Es fundamental que los gobiernos y las entidades de salud establezcan normativas que prioricen la salud digital para garantizar el uso efectivo de estas tecnologías.

Según el informe del simposio de la OMS sobre el futuro de sistemas de salud digitales en la región europea (World Health Organization. Regional Office for Europe, 2019), "La adopción de tecnologías digitales en el ámbito de la salud está ampliamente reconocida como crucial para el buen funcionamiento de los sistemas en salud y para capacitar a las personas en el marco de la transición hacia una atención integrada y centrada en la persona."

En este contexto, el proyecto de grado titulado "Sistema Web de Administración de Historias Clínicas. Caso: Facultad de Medicina, Instituto de Genética, Unidad de Genética Médica" se presenta como una solución para administrar las historias clínicas de los pacientes, mejorando la gestión de la información de las historias clínicas de los pacientes, permitiendo al personal médico agilizar su trabajo y brindar una atención más efectiva al paciente. Este esfuerzo se centra en la Unidad de Genética Médica del Instituto de Genética de la Facultad de Medicina de la Universidad Mayor de San Andrés, reconocida por sus contribuciones a la genética médica, citogenética y genética toxicológica.

Este trabajo se divide en varios capítulos, cada uno de los cuales se enfoca en aspectos específicos del sistema propuesto. En el Capítulo 1, se presenta el marco referencial y se introduce el contexto del problema que motiva este proyecto. El Capítulo 2 se presenta el marco teórico, donde se detalla la metodología de desarrollo del sistema y se exploran conceptos fundamentales que servirán de base para la creación del sistema. El Capítulo 3 se concentra en el diseño y desarrollo del sistema. El Capítulo 4 se enfoca en la calidad y seguridad del sistema. El Capítulo 5 se centra a determinar el costo beneficio del sistema. Finalmente, el Capítulo 6 ofrece las conclusiones y recomendaciones derivadas de este proyecto, proporcionando una visión global de los resultados obtenidos y orientación para futuros desarrollos.

1.2 Antecedentes

En este apartado se explora las circunstancias y eventos que preceden a la problemática que aborda el proyecto. Esto incluye un análisis del estado del arte, que presenta las contribuciones previas en el campo de estudio. También se examinarán trabajos similares que han enfrentado desafíos afines, así como el contexto nacional que influye en el problema específico. Los antecedentes institucionales se explorarán para

comprender el marco dentro del cual se desarrolla el proyecto. Finalmente, se presentará una lista de problemas identificados

1.2.1 Estado del Arte.

Según Gil López, E., y Medinaceli Díaz, K. (2019), en Bolivia se presenta una deficiencia general en el acceso a la información, ya que esta no se encuentra disponible en los sitios web oficiales. Además, algunas instituciones tanto públicas como privadas carecen de memorias institucionales anuales que permitirían acceder a la información.

En su estudio de campo basado en entrevistas, los autores generaron tablas que identifican la aplicación, uso y desarrollo de Tecnologías de la Información y Comunicación (TIC) en centros de salud públicos y privados en Bolivia.

La siguiente tabla presenta la aplicación de TIC en el subsector público de Bolivia. En ella, se detallan los sistemas y proyectos de cada entidad, junto con sus respectivas fechas de implementación.

Tabla 1.1 Aplicación de TIC's en el subsector público

| | <i>Entidad</i> | <i>Tema</i> | <i>Nombre y cargo</i> | <i>Fecha</i> |
|-------------------|---|---|---|-------------------------------|
| Subsector público | Ministerio de Salud | Dirección General de Planificación | Dr. Ronald Machaca | 21/07/2015 - 04/08/2015 |
| | Sistema Nacional de Información en Salud (SNIS) | SNIS-VE | Rocco Abruzzese Responsable nacional de Información y Producción de Servicios, nivel I | 17/06/2015 - 24/06/2015 |
| | | SNIS | Ing. Jorge Bailey e Ing. Gabriel Jiménez | 17/06/2015 |
| | | Software de Atención Primaria en Salud (SOAPS) | Ing. Mauricio Bustillos | 23/07/2015 |
| | | Sistema de Información Clínica Estadística (SICE) | Ing. Gabriel Jiménez | 03/08/2015 |
| | | Sistema de Información Administrativa Financiera (SIAF) | Ing. Gabriel Jiménez | 27/08/2015 |
| | Programa VIH-SIDA | Centro Departamental de Vigilancia y Referencia (CDVIR) | Dr. David Segurondo Responsable de CDVIR, La Paz | 20/07/2015 - 27/07/2015 |

Fuente: Gil López, E., & Medinaceli Díaz, K. (2019).

La tabla siguiente exhibe la implementación de TIC en el subsector de la seguridad social en Bolivia.

Tabla 1.2 Aplicación de TIC's en el subsector de la Seguridad Social

| | <i>Entidad</i> | <i>Tema</i> | <i>Nombre y cargo</i> | <i>Fecha</i> |
|----------------------------------|---|--|--|-------------------------------|
| Subsector de la seguridad social | Seguro Social Universitario (SSU) | Historia clínica | Jaime Riveros Encargado de bioestadística | 10/09/2015 |
| | | Manejo historia clínica | Lic. Sonia Apaza Jefe de enfermeras a.i. | 14/10/2015 |
| | | Archivo historia clínica | Lic. Elizabeth Saravia Encargada de la Unidad de Admisión, Archivo y Fichaje | 14/10/2015 |
| | Caja de Salud de la Banca Privada (CSBP) | Historia clínica electrónica | Dr. Gonzalo Maldonado Médico pediatra y director del Hospital Regional La Paz | 04/03/2015 - 30/07/2015 |
| | | Historia clínica electrónica | Dr. David Martínez Médico Traumatólogo | 22/07/2015 - 24/07/2015 |
| | | Software médico y Sistema Administrativo Médico (SAMI) | Dra. Tania Cherro Responsable del software médico | 25/08/2015 |
| | Corporación del Seguro Social Militar (Cossmil) | Archivo historia clínica | Lic. Katia A. de Auza Jefa de Archivo Clínico | 25/09/2015 - 30/09/2015 |
| | | Sistema Integrado de Seguimiento Hospitalario a Pacientes (SISHAP) | Coronel Grover Quiroga Director nacional de sistemas | 04/11/2015 |

Fuente: Gil López, E., & Medinaceli Díaz, K. (2019).

En último término, la tabla siguiente presenta la utilización de TIC en el subsector privado, además de identificar los actores involucrados y los detalles de sus respectivos proyectos en esta área.

Tabla 1.3 Aplicación de TIC's en el subsector privado y actores

| | <i>Entidad</i> | <i>Tema</i> | <i>Nombre y cargo</i> | <i>Fecha</i> |
|-------------------|--|---|---|--------------------------|
| Subsector privado | Hospital Arco Iris | Historia clínica electrónica sistema open HAI | Ing. Julio Alarcón Jefe de la Unidad de Sistemas | 04/08/2015 07/08/2015 |
| | | Historia clínica electrónica sistema open HAI | Dr. Igor Salvatierra Adjunto Enseñanza y director de Enseñanza e Investigación | 01/09/2015 |
| | | Archivo de la historia clínica | Sr. Rubén Heredia Auxiliar de Archivo | 08/09/2015 |
| Actores TIC | Agencia para el Desarrollo de la Sociedad de la Información en Bolivia (ADSIB) | Firma digital | Lic. Kantuta Muruchi Encargada de Planificación y Proyectos | 14/08/2015 |
| | | Firma digital | Ing. Sylvain Lesage Jefe de la Unidad de Innovación y Desarrollo | 02/09/2015 |
| | ENTEL | Telecentros | Ing. Rolando Álvarez Jefe de la Unidad de Telecentros | 28/07/2015 |
| | | | Ing. Wilson Cuellar Profesional de Desarrollo Rural | 28/07/2015 |
| | Ministerio de Planificación del Desarrollo | Dirección General de Gobierno Electrónico | Ing. Rodrigo Siles Director general de Gobierno Electrónico | 29/09/2015 |
| | Asociación de Bancos Privados de Bolivia (ASOBAN) | Administradora de Cámaras de Compensación y Liquidación (ACCL S.A.) | Ing. Ricardo Primintela Administrador de Sistemas | 21/09/2015 |

Fuente: Gil López, E., & Medinaceli Díaz, K. (2019).

De acuerdo con las conclusiones extraídas del artículo de Gil López, E., & Medinaceli Díaz, K. (2019) sobre la implementación de la historia clínica electrónica en los establecimientos de salud del Sistema Nacional de Salud de Bolivia, se observaron los siguientes hallazgos:

- La Caja de Salud de la Banca Privada ha desarrollado su propio software médico y sistema administrativo médico (SAMI) desde el año 2005. Este sistema cuenta con módulos específicos para cada usuario, lo que permite la generación de informes de exámenes, laboratorios, médicos y de seguimiento.
- La Corporación del Seguro Social Militar (Cossmil), creó el Sistema de Gestión Hospitalaria (SIGEH) en 2004. Desde 2014, se ha implementado el Sistema de

Información Integrado de Control y Seguimiento Hospitalario (SISHAP), que también se utiliza en las ciudades de Sucre y Puerto Suárez.

- El Hospital Arco Iris (HAI) adoptó el sistema openHAI. Este sistema incluye la Historia Clínica Electrónica (HCE) de hospitalización, terapia intensiva (UTI), emergencias y consulta externa desde septiembre de 2015. Además, ofrece un módulo especializado para almacenar imágenes de resonancias, tomografías, radiografías y otros estudios radiológicos.

1.2.2 Trabajos Similares.

1.2.2.1 Contexto Internacional

En el ámbito internacional, encontramos trabajos significativos que abordan problemáticas similares relacionadas con la administración de historias clínicas en entidades de salud. A continuación, se presentan ejemplos relevantes.

Proyecto de Jácome, F. (2015): “Desarrollo del Sistema de Administración de Historias Clínicas del Departamento Médico de la Empresa Imprenta Mariscal Cia. Ltda”

- En este proyecto, se abordó la problemática de la falta de herramientas tecnológicas en una organización, lo que llevaba a la realización manual de tareas y generaba desorganización, pérdida de información, duplicidad de historias clínicas y demoras en la búsqueda de registros. Se utilizó la metodología Rational Unified Process (RUP) y el Lenguaje Unificado de Modelado (UML) para el modelado del sistema, con herramientas como Filemaker (Front End) y FileMaker Server (Back End y Base de Datos). El proyecto se desarrolló en la Carrera de Ingeniería en Sistemas Informáticos de la Universidad Israel, Quito, Ecuador.

Proyecto de Ortiz, O. L. E. (2022): “Desarrollo de un sistema web de gestión de historias clínicas en un consultorio privado de medicina general”

- En este proyecto, la principal problemática abordada se centró en la atención y acceso a la información de los pacientes. La recolección de datos de pacientes se volvía complicada debido a la utilización de registros físicos y formularios manuscritos poco legibles, lo que propiciaba la generación de información incorrecta. El objetivo del proyecto fue estandarizar, organizar y automatizar los procesos de diagnóstico y administración de las historias clínicas de los pacientes en el Área de Consulta Externa. Facilitando el acceso a la información, reduciendo los errores en la digitalización de datos y estandarizando los formularios de registro.
- Para el desarrollo de este sistema, se aplicó la metodología Rational Unified Process (RUP) y se utilizó el Lenguaje Unificado de Modelado (UML) para el modelado del sistema. Las herramientas empleadas incluyeron MySQL 5.0.45 como sistema de gestión de bases de datos, JDK 1.6.0 como framework de desarrollo (Front End) y GlassFish V 2.0 update 2 como servidor de aplicaciones. El proyecto se llevó a cabo en la Facultad De Ingeniería De Sistemas de la Escuela Politécnica Nacional en Quito, Ecuador.

Proyecto de Padilla, P. J. (2018): “Diseño e implementación de un sistema de administración y consulta de historias clínicas electrónicas (HCE) mediante el uso de tecnología Webservices en diversos entes de salud del municipio de Santa Cruz de Lorica - Córdoba”.

- Este proyecto abordó desafíos significativos que afectan la implementación de sistemas de historia clínica electrónica (HCE) en centros de salud. Entre los obstáculos identificados se encontraban el crecimiento exponencial de la información, problemas de compatibilidad de archivos y consideraciones de costos, entre otros. El objetivo principal del proyecto fue diseñar e

implementar un sistema de administración y consulta de historias clínicas electrónicas (HCE) mediante la aplicación de tecnología Webservices.

- Este enfoque involucró el desarrollo de tecnologías Webservices para acceder y administrar la información, además de la creación de pruebas de funcionamiento para garantizar su eficacia. Para llevar a cabo este proyecto, se siguió la metodología Rational Unified Process (RUP). Las herramientas utilizadas incluyeron lenguajes de programación como C y C++ en el entorno Visual Studio, así como tecnologías web como HTML 5, JavaScript y CSS. El proyecto se llevó a cabo en la Carrera de Ingeniería de Sistemas, Facultad de Ingenierías de la Universidad de Córdoba, Colombia.

1.2.2.2 Contexto Nacional

A nivel nacional, se han desarrollado numerosos proyectos relacionados con la administración de historias clínicas. A continuación, se detallan algunos ejemplos destacados:

Proyecto de Paz, G. C. A. (2020): "SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS."

- La problemática central abordada en este proyecto estaba relacionada con la gestión de historias clínicas en formato físico, lo que ocasionaba molestias tanto al personal médico como a los pacientes. Esto generaba expedientes clínicos desactualizados, pérdida de información, duplicación de expedientes y retrasos en la atención médica. El objetivo principal de este proyecto fue desarrollar un sistema web de administración de historias clínicas. La metodología utilizada incluyó SCRUM y UWE-UML para el desarrollo y modelado del sistema. En cuanto a las herramientas, se empleó el lenguaje de programación PHP (Versión 7.4.6), la base de datos MariaDB (Versión

10.1.21) y Bootstrap (Versión 4.5) para la parte de Front End. Este proyecto se llevó a cabo en la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Proyecto de Flores, G. P. (2018):"Sistema web de administración de historias clínicas Caso: centro médico quirúrgico Erzengel."

- En este caso, los problemas identificados en el centro de salud incluyeron la redacción manual de historias clínicas, la duplicidad de expedientes, el deterioro de documentos físicos, retrasos en la atención de pacientes debido a la búsqueda y registro manuales de información, y expedientes clínicos desactualizados. El objetivo de este proyecto fue desarrollar un sistema web de administración de historias clínicas que mejora las tareas de los especialistas en salud y proporciona una atención más óptima a los pacientes. Para su desarrollo, se aplicó la metodología SCRUM, y se utilizaron herramientas como PHP como lenguaje de programación, Bootstrap para la parte del Frontend y PostgresSQL como gestor de base de datos. Este proyecto se llevó a cabo en la carrera de Informática de la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés.

Proyecto de Ponce, Y. Y. A. (2016) "Software como servicio para la administración de historias."

- El objetivo principal de este proyecto fue desarrollar un software como servicio que permitiera la administración de historias clínicas, facilitando el control y seguimiento de la información de los pacientes. Se aplicó la metodología UWE, con énfasis en los modelos de diseño web. Para el desarrollo del sistema, se empleó el framework Django y el lenguaje de programación Python para crear el webservice. Además, se utilizó Heroku como servidor en la nube. Este proyecto se desarrolló en la carrera de

Informática de la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés.

Cada uno de estos proyectos abordó desafíos específicos en la administración de historias clínicas y aplicó diferentes metodologías y herramientas para lograr sus objetivos.

1.2.3 Antecedentes Institucionales

El Instituto de Genética, perteneciente a la Facultad de Medicina, Enfermería, Nutrición y Tecnología Médica de la Universidad Mayor de San Andrés, está ubicado en la ciudad de La Paz, específicamente en la Av. Saavedra N° 2246, Miraflores, Edificio de la Facultad de Medicina, Piso 9. Este instituto, dirigido por la M. Sc. Dra. Ximena Aguilar Mercado, comenzó sus operaciones el 15 de junio de 1972 y se ha establecido como el principal centro de referencia a nivel nacional en el campo de la genética. En la actualidad, el Instituto de Genética se divide en tres áreas fundamentales: interacción social, docencia e investigación. Estas áreas engloban unidades especializadas en Genética Médica, Citogenética y Genética Toxicológica, y brindan servicios tanto a la comunidad estudiantil como a la ciudadanía en general.

Misión:

Ser un centro líder en investigación científica en salud genética a nivel nacional, con capacidad para planificar, ejecutar y evaluar proyectos de investigación y desarrollo tecnológico, manteniendo un fuerte compromiso social, que presta servicios docentes, asistenciales y de laboratorio de alta especialidad equipados con tecnología avanzada.

Visión:

Ser un Centro de Referencia Nacional, dotado de alta calificación profesional y tecnológica, capaz de estudiar, evaluar e investigar los problemas genéticos de la población boliviana

A continuación, se presenta un organigrama que ilustra la estructura del instituto en las áreas previamente mencionadas.

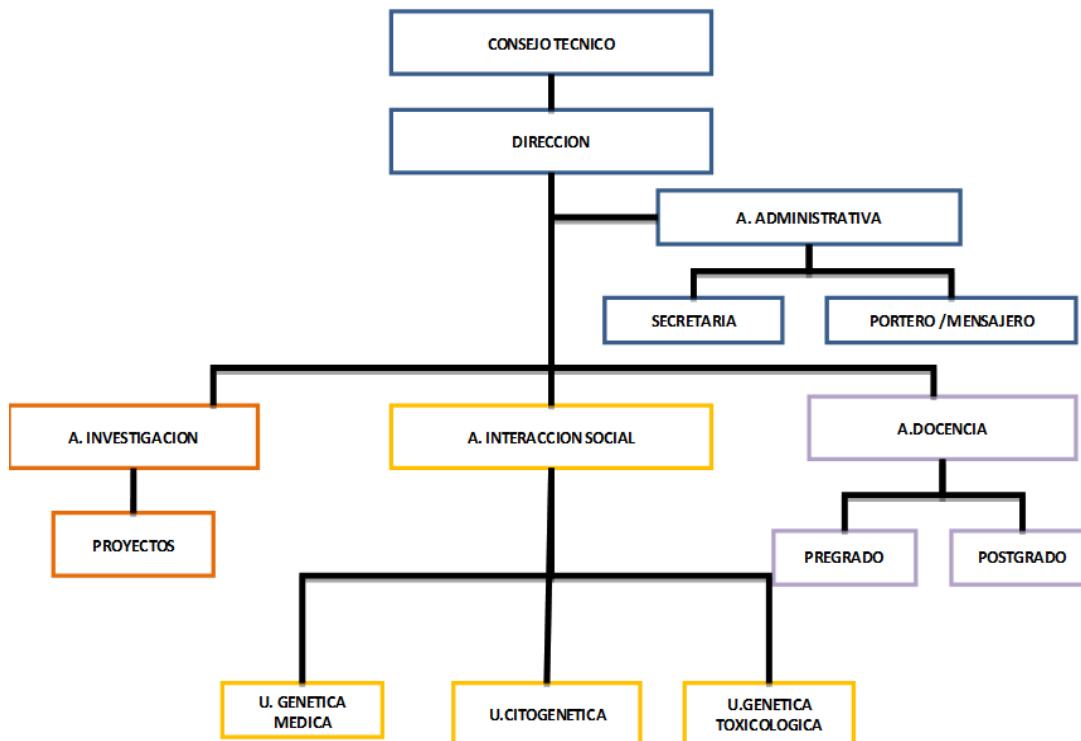


Figura 1.1 Estructura Orgánica del Instituto de Genética.

Fuente: Inst. Genética, 2022.

La Unidad de Genética Médica tiene la responsabilidad de brindar atención y seguimiento a los pacientes, la cual está integrada en el marco del proyecto de Telemedicina, esta se encuentra altamente activa en la actualidad desarrollando proyectos destinados a beneficiar a la población.

1.3 Planteamiento del Problema

La gestión de historias clínicas en el Instituto de Genética de la Universidad Mayor de San Andrés se ha visto enfrentada a una serie de desafíos que impactan negativamente

en la eficiencia y la calidad del servicio ofrecido. La carencia de un sistema informático adecuado ha resultado en un proceso manual para el registro de datos de pacientes (anexo 1 y anexo 2), los cuales se realizan y almacenan en formatos físicos y/o en archivos de Excel. Además, la ausencia de un formato estandarizado para las historias clínicas y la ausencia de un sistema centralizado han generado dificultades en el acceso y gestión de la información de los pacientes. Incluso, en un intento previo de resolver esta problemática, se implementó una base de datos en Access en 2012 (ver anexo 3), pero esta solución resultó ineficaz y se retornó al sistema anterior basado en formularios impresos y registros en archivos Word y Excel.

Estos desafíos evidencian un problema significativo en la gestión de historias clínicas en el Instituto de Genética. Actualmente, el proceso es altamente deficiente, ya que se basa en el uso de formatos físicos y archivos Excel para el registro de datos, lo que conlleva a la pérdida de información, conflictos internos y limitaciones en el acceso a los datos.

1.3.1 Problemas secundarios

Mediante el análisis realizado a la Unidad de Genética Médica, se pudo identificar los siguientes problemas secundarios:

- El registro de los datos clínicos de los pacientes se realiza de forma manual en formato físico y en hojas de cálculo Excel, lo que resulta en la pérdida de información.
- La carencia de un procedimiento efectivo para el registro de citas médicas ocasiona retrasos en la atención de los pacientes, ya que el personal encargado de esta tarea no comunica adecuadamente esta información a los médicos responsables.

- La descentralización de la información genera duplicidad de datos y registros de los pacientes, ocasionando a la vez perdida de la información que dificulta su seguimiento clínico.
- La búsqueda manual de historias clínicas puede llegar a consumir mucho tiempo, lo que afecta negativamente la eficiencia de los procesos clínicos y administrativos.
- Los doctores encargados tienen dificultades para acceder a la información clínica necesaria de los pacientes lo que ocasiona el retraso en el tratamiento o diagnóstico adecuado.

1.3.2 Formulación del Problema

Teniendo en cuenta lo descrito anteriormente surge la siguiente pregunta:

¿Cómo facilitar la gestión de información de historias clínicas para la unidad de genética médica del instituto de genética?

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar un sistema web de administración de historias clínicas para la Unidad de Genética Médica del Instituto de Genética.

1.4.2 Objetivos Específicos

- Establecer un proceso de registro sistematizado y validado de los datos clínicos de los pacientes, con el fin de garantizar la integridad y precisión de la información antes de su almacenamiento.

- Establecer un proceso que facilite el registro de citas médicas, con el fin de reducir los retrasos en la atención de los pacientes.
- Centralizar la gestión de registros de los pacientes mediante la implementación de una base de datos centralizada, con el propósito de reducir la duplicidad de registros y asegurar la integridad de la información para facilitar el seguimiento clínico efectivo.
- Implementar un proceso de acceso a la información de datos clínicos de los pacientes dentro del sistema de gestión de historias clínicas que permita a los usuarios encontrar rápidamente la información necesaria, reduciendo el tiempo dedicado a la búsqueda manual y mejorando la eficiencia de los procesos clínicos y administrativos.
- Establecer un proceso para consultas médicas que facilite a los médicos agilizar el flujo de atención médica, permitiendo recuperar rápidamente los datos necesarios para un tratamiento o diagnóstico adecuado

1.5 Justificación

1.5.1 Justificación Social

La automatización de los procesos de registro, actualización y búsqueda de datos mediante un sistema de administración de historias clínicas para el Instituto de Genética conlleva beneficios significativos. En primer lugar, esta automatización optimiza el tiempo, lo que se traduce en una atención más rápida y precisa para los pacientes. Y, en segundo lugar, la gestión eficiente de las historias clínicas mejora la atención al permitir un acceso más completo a la información de los pacientes, lo que resulta en diagnósticos más certeros. Esto también reduce los conflictos generados por la discrepancia de datos entre diferentes unidades, garantizando una atención más coherente y uniforme para todos los pacientes. Además, simplifica las tareas diarias del personal médico y administrativo,

al automatizar los procesos de registro y búsqueda de datos, se reduce la carga de trabajo manual, permitiendo que el personal se concentre en brindar una atención de calidad.

1.5.2 Justificación Económica

La implementación de este sistema conlleva ahorros significativos en gastos de material, como hojas de papel, folders, y formularios para las historias clínicas, además de liberar espacio físico que antes se ocupaba para el almacenamiento de archivos en papel. Es importante destacar que no se ha generado ningún costo adicional para el Instituto de Genética por la adquisición del sistema, evitando así los gastos sustanciales comunes en sistemas similares.

Además, el desarrollo del sistema se basó en tecnologías y frameworks de software libre (open source), lo que permitió reducir los gastos de licencias y adquisición de software propietario. En adición, no fue necesario contratar personal adicional para la administración del sistema, ya que el Instituto y la facultad cuentan con personal que colaborará en esta función de manera efectiva. Esta combinación de ahorros y eficiencia en el uso de recursos garantiza la viabilidad económica del proyecto.

1.5.3 Justificación Tecnológica

El proyecto es justificable tecnológicamente, ya que el instituto cuenta con recursos informáticos adecuados para el acceso, administración y mantenimiento del sistema. Además, se están realizando mejoras en la infraestructura de red, lo que incluye la actualización del cableado, con el objetivo de mejorar la velocidad de conexión. Estas mejoras permitirán un despliegue más eficiente del sistema de administración de historias clínicas en el Instituto de Genética.

1.6 Alcances y Límites

1.6.1 Alcance

Basado en las principales necesidades del instituto de genética, es necesario delimitar los alcances, para lo cual se debe especificar el área de acción que se cubrirá. Por tanto, el presente proyecto cubre los siguientes alcances:

- El proyecto implica en desarrollar un sistema web de administración de historias clínicas, donde se almacena la información médica de los pacientes.
- Los módulos que se implementaron son:
 - Módulo de registro de usuarios.
 - Módulo de registro de pacientes.
 - Módulo de registro de citas.
 - Módulo de consulta.
 - Módulo de historia clínica.
 - Módulo de reporte.
 - Módulo de autenticación y de autorización.
- Se creo una base de datos relacional que centralizará los datos.
- La información fue obtenida con la autorización del personal a cargo.

1.6.2 Límites

Los límites del presente proyecto se indican a continuación:

- El proyecto solo lleva a cabo el análisis, diseño y desarrollo de los módulos de acuerdo con los requerimientos de la institución.
- Las actividades económicas y estadísticas no son consideradas, ya que el enfoque se centra únicamente en la información del paciente y los procesos relacionados con la historia clínica.

- El acceso al sistema está restringido solo al personal autorizado, excluyendo a los pacientes.
- El sistema fue desarrollado y es utilizado por la unidad de Genética Medica, sin considerar su aplicación en otras unidades del Instituto.

CAPÍTULO II MARCO TEÓRICO

2.1 Ingeniería de Software

Según Fritz Bauer (1969, citado en Pressman, 2010), la ingeniería de software se define como 'el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales'. Esta definición sirve como base para comprender los principios fundamentales que pueden aplicarse al desarrollo de software.

Por otro lado, la IEEE (1993, citado en Pressman, 2010) realiza una definición más completa: "La ingeniería de software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. 2) El estudio de enfoques según el punto 1."

Pero Pressman nos da una definición más objetiva, disciplina, pero también adaptabilidad y agilidad (Pressman, 2010)." La ingeniería de software es una tecnología con varias capas, cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad".



Figura 2.1 Capas de la ingeniería de software

Fuente: Pressman, 2010

2.1.1 El Proceso de Software

Pressman (2010) explica que el proceso de desarrollo de software es un enfoque adaptable que permite a las personas elegir un conjunto de tareas para realizar el trabajo. Este enfoque tiene como objetivo entregar el software de manera oportuna y con alta calidad para satisfacer las necesidades de los usuarios finales.

Pressman (2010) identifica una estructura de proceso general para la ingeniería de software, la cual consta de cinco actividades: **comunicación, planificación, modelado, construcción y despliegue.**

Las cinco actividades descritas por Pressman resultan útiles para la comprensión y la implementación en proyectos de desarrollo de software, independientemente de su complejidad. Estas actividades son empleadas de forma recurrente a lo largo del proyecto, con cada iteración representando un progreso en la creación del software. A medida que se realizan más iteraciones, el software se perfecciona gradualmente, completando con los requerimientos que este debe cumplir.

2.1.2 Modelos de Desarrollo de Software

Los modelos de desarrollo de software ofrecen una estructura que se utiliza para administrar el proceso de construcción de software. Estos modelos establecen flujos de trabajo claros, lo que es fundamental para alcanzar el éxito en cualquier proyecto. En la fase previa a la entrega del software, es necesario completar una serie de fases, determinar su secuencia y los plazos correspondientes; precisamente para abordar estos aspectos, se han diseñado los modelos de desarrollo de software. Estos modelos se centran en los procesos técnicos y administrativos, contribuyendo significativamente a la creación del producto final, ya que supervisan todas las actividades involucradas.

2.1.2.1 Modelo en Cascada

Pressman (2010) nos indica que, el modelo en cascada, a veces llamado ciclo de vida clásico, sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado.

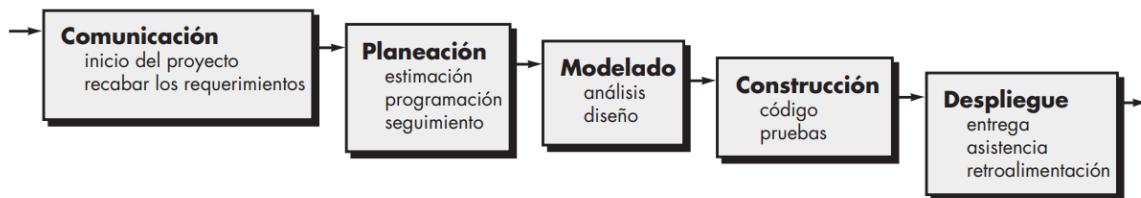


Figura 2.2 Modelo en Cascada

Fuente: Pressman, 2010

En este modelo, el proceso se divide en fases secuenciales (como se ve en la figura 2.2), y cada fase debe completarse antes de pasar a la siguiente. Estas fases incluyen la planificación, el diseño, la implementación, las pruebas y el mantenimiento. El modelo en cascada es especialmente adecuado para proyectos en los que los requisitos están bien definidos desde el principio y cambian poco a lo largo del proyecto. Aunque proporciona una estructura ordenada, puede tener dificultades para adaptarse a los cambios en los requisitos, lo que lo hace menos flexible en comparación con los modelos evolutivos y ágiles.

2.1.2.2 Modelo en Espiral

(Pressman, 2010) nos indica que, el modelo espiral es un modelo evolutivo del proceso de desarrollo de software que combina la naturaleza iterativa de los prototipos

con los aspectos controlados y sistemáticos del modelo en cascada. Este modelo permite un desarrollo rápido de versiones cada vez más completas.

De acuerdo con Boehm (como se cita en Pressman, 2010), el modelo de desarrollo en espiral es un generador de procesos que se enfoca en la gestión de riesgos y guía la ingeniería concurrente en sistemas intensivos en software. Tiene dos características distintivas principales. La primera es el enfoque cíclico para el crecimiento incremental del grado de definición de un sistema y su implementación, mientras que disminuye su grado de riesgo. La otra es un conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.

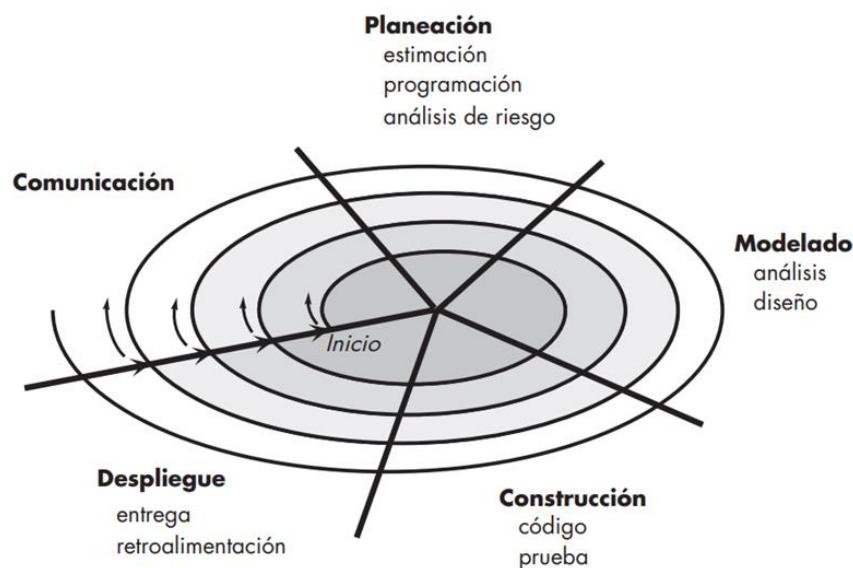


Figura 2.3 Modelo en Espiral

Fuente: Pressman, 2010

El proceso en el modelo en espiral es dividido por el equipo de desarrollo en un conjunto de actividades estructurales. Las actividades se desarrollan en espiral, y la elección de estas actividades se basa en el análisis de riesgos específicos. Cada iteración

en este modelo requiere una cuidadosa consideración de los objetivos, las alternativas y diversos factores, como la experiencia personal, los requisitos, las estrategias de gestión del sistema, entre otros. El modelo en espiral ofrece dos enfoques de planificación, uno angular, centrado en el avance del software, y otro radial, que implica un aumento en los costos a medida que se realizan más iteraciones (ver figura 2.3), lo que permite abordar de manera controlada las incertidumbres y riesgos asociados al desarrollo de software. Esta metodología se adapta a lo largo de todo el ciclo de vida del software, lo que la convierte en una herramienta valiosa para proyectos que enfrentan incertidumbre y cambios constantes.

2.1.2.3 Modelo de Prototipos

(Admin,2023) nos dice, este modelo implica la creación de un prototipo o modelo de trabajo del software antes de desarrollar la versión final, y se utiliza para obtener comentarios y retroalimentación del cliente. Un ejemplo práctico del modelo de prototipos podría ser el desarrollo de un software para la gestión de inventarios. En este caso, se crearía un prototipo inicial para presentar al cliente, para recibir retroalimentación y ajustar la solución antes de pasar al desarrollo completo del software.

El propósito principal de este enfoque es proporcionar a los usuarios una vista preliminar de una sección del software.

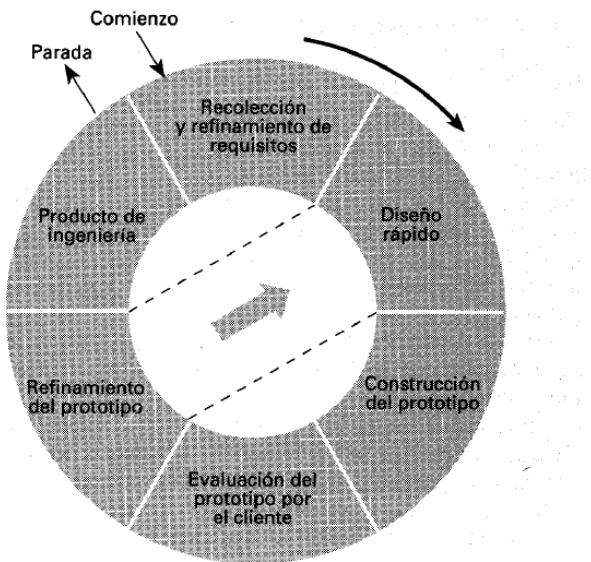


Figura 2.4 Modelo de Prototipo

Fuente: EcuRed, s. f.

Se basa principalmente en el método de prueba y error, ya que, si el usuario no está satisfecho con alguna parte del prototipo, esto se considera un error y se debe corregir hasta que el usuario esté contento con el resultado. Además, la construcción del prototipo se realiza en un corto período de tiempo, utilizando las herramientas adecuadas y minimizando los costos, ya que una vez que el prototipo recibe la aprobación, se inicia el desarrollo real del software (ver figura 2.4).

2.1.3 Metodología Ágil en el Desarrollo de Software

La Metodología Ágil en el Desarrollo de Software es un enfoque altamente flexible y colaborativo para la creación de software que ha ganado amplia aceptación en la industria. Se diferencia de los modelos de desarrollo tradicionales, como el Modelo en Cascada, al enfocarse en la adaptabilidad, la colaboración constante y la respuesta ágil al cambio. La metodología ágil promueve la entrega continua de software funcional y valora la comunicación constante con los clientes y los miembros del equipo. Uno de los manifiestos más conocidos de la Metodología Ágil es el "Manifiesto Ágil" (Beck et al.

2001), que destaca valores clave, como la satisfacción del cliente a través de la entrega temprana y continua de software útil, y la bienvenida al cambio incluso en etapas tardías del desarrollo. Este enfoque se traduce en ciclos de desarrollo cortos, prototipado constante y la capacidad de responder rápidamente a las necesidades cambiantes del cliente.

Los métodos ágiles se caracterizan por su enfoque en iteraciones de desarrollo, las cuales suelen tener una duración de una a cuatro semanas. Cada ciclo iterativo incluye actividades como planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. A diferencia de buscar agregar una gran cantidad de funcionalidades en cada iteración, el objetivo principal es producir un prototipo libre de errores al final de cada ciclo. Al concluir una iteración, el equipo se reúne nuevamente para evaluar y reajustar las prioridades del proyecto. Además, los métodos ágiles promueven la comunicación cara a cara sobre la documentación, y consideran que el progreso real se mide mejor mediante el software funcional, priorizando así la entrega de valor a lo largo del proceso.

Algunos ejemplos de estos enfoques incluyen Extreme Programming (XP), Scrum, la familia de Metodologías Crystal, Feature Driven Development, Proceso Unificado Rational (RUP), Agile Unified Process (AUP) y OpenUP. Cada uno de estos métodos ágiles ofrece un conjunto específico de prácticas y enfoques para adaptarse a diversas situaciones de desarrollo de software, lo que brinda a los equipos la capacidad de ajustarse rápidamente a cambios y requisitos en evolución.

2.1.3.1 Conceptos clave de metodologías ágiles de desarrollo.

La metodología ágil se centra en el trabajo en equipo, la comunicación constante y la entrega temprana de software. El Manifiesto Ágil es una declaración de valores y principios fundamentales para el desarrollo ágil de software. El manifiesto establece que los desarrolladores deben priorizar "individuos e interacciones sobre procesos y

herramientas, software funcionando sobre documentación exhaustiva, colaboración con el cliente sobre negociación contractual y responder al cambio sobre seguir un plan" (Beck et al., 2001).

(Schwaber & Sutherland, 2017) nos explica que una de las principales características de las metodologías ágiles es la entrega temprana y continua de software. En lugar de esperar a tener un producto completo antes de presentarlo al cliente, los equipos de desarrollo entregan versiones funcionales del software en iteraciones cortas llamadas sprints. Cada sprint generalmente tiene una duración de una a cuatro semanas.

La comunicación constante entre los miembros del equipo es crucial para el éxito de las metodologías ágiles. Las reuniones diarias de pie, conocidas como reuniones de scrum, son una forma efectiva de mantener a todos informados sobre el progreso del proyecto y cualquier problema que surja. Además, las metodologías ágiles fomentan la colaboración entre el equipo de desarrollo y el cliente a lo largo del proyecto.

2.1.3.2 Ventajas y desventajas de la implementación de metodologías ágiles.

Las metodologías ágiles tienen muchas ventajas potenciales, como la entrega temprana y continua de software, la capacidad de adaptarse a los cambios en los requisitos del proyecto y la mejora de la comunicación y la colaboración entre el equipo de desarrollo y el cliente.

Sin embargo, la implementación de las metodologías ágiles también puede tener desventajas. Una de las principales preocupaciones es la falta de documentación exhaustiva. A diferencia del enfoque tradicional de cascada, las metodologías ágiles no requieren una documentación detallada de cada fase del proyecto. Esto puede ser un

problema si los miembros del equipo cambian o si se requiere una auditoría posterior del proyecto.

Otra desventaja potencial es la falta de planificación detallada. Aunque las metodologías ágiles promueven la capacidad de adaptarse a los cambios, algunas organizaciones pueden sentirse incómodas con la falta de un plan detallado y estructurado desde el inicio del proyecto.

2.2 Proceso Unificado Ágil (AUP).

El Proceso Unificado Ágil (AUP, por sus siglas en inglés) es una metodología de desarrollo de software que combina elementos del Proceso Unificado de Rational (RUP) y de las metodologías ágiles que fue desarrollado por Scott Ambler desde 2002 hasta 2006.

(Ambler, 2006) nos dice que AUP es una metodología iterativa e incremental que se enfoca en la entrega temprana de software funcional a través de iteraciones cortas. En este marco teórico se discutirán los conceptos clave del AUP, así como las ventajas y desventajas de su implementación.

2.2.1 Principios del Proceso Unificado Ágil (AUP).

AUP se fundamenta en los siguientes principios, de acuerdo con EcuRed (s. f.):

- **Simplicidad:** Todo se describe concisamente utilizando poca documentación, no miles de ellas.
- **Agilidad:** El ajuste a los valores y principios de La Alianza Ágil.
- **Centrarse en actividades de alto valor:** La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.
- **Herramienta de la independencia:** Usted puede usar cualquier conjunto de herramientas que desea con el AUP. Se sugiere utilizar las herramientas

más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.

- **Usted querrá adaptar este producto para satisfacer sus propias necesidades:** La metodología AUP es un producto de fácil uso utilizando cualquier herramienta. No es necesario comprar una herramienta especial, o tomar un curso, para adaptar esta metodología.

Estos conceptos se resumen en:

- **Comunicación continua:** Se fomenta la colaboración y la comunicación continua entre los miembros del equipo de proyecto y los interesados.
- **Entrega temprana:** Se prioriza la entrega temprana de funcionalidades, para obtener retroalimentación temprana y ajustar el desarrollo del proyecto de manera adecuada.
- **Desarrollo iterativo e incremental:** Se llevan a cabo iteraciones cortas y frecuentes, para ajustar el desarrollo del proyecto de manera oportuna.
- **Flexibilidad y adaptabilidad:** Se busca ser flexible y adaptable, para poder hacer frente a los cambios que puedan surgir durante el proyecto.

2.2.2 Ciclo de Vida del Proceso Unificado Ágil (AUP).

La Figura 2.5 ilustra el ciclo de vida de AUP (Proceso Unificado Ágil). Este proceso se repite a lo largo de una serie de ciclos que representan la evolución de un sistema, donde cada ciclo corresponde a una versión del sistema, y a medida que avanzamos en ciclos posteriores, se añaden funcionalidades, se realizan mejoras y se perfecciona el software.

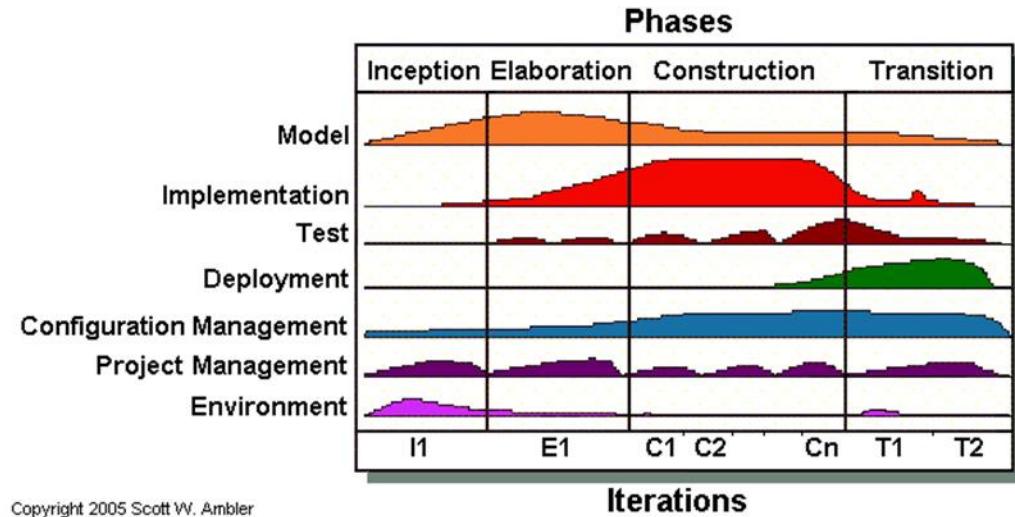


Figura 2.5 Ciclo de vida de AUP

Fuente: Ambler, 2006

2.2.3 Fases e Hitos del Ciclo de Vida.

El Proceso Unificado Ágil (AUP) se compone de cuatro fases principales (ver figura 2.6), y al término de cada fase, se debe alcanzar un hito o meta para concluir y avanzar a la fase siguiente. Cada fase se divide en iteraciones, y en cada iteración se desarrolla secuencialmente un conjunto de disciplinas o flujos de trabajo. Según la descripción de Ambler (2006), estas fases son las siguientes:

2.2.3.1 Fase de Inicio.

Esta fase se enfoca en el establecimiento de los objetivos del proyecto y la identificación de las partes interesadas clave. Además, se lleva a cabo un análisis preliminar de los riesgos y se delinean con claridad la visión y el alcance del proyecto. Esta etapa generalmente es la más breve en el ciclo de desarrollo y se recomienda no prolongarla innecesariamente.

De acuerdo con (Ambler, 2006), los principales objetivos de la fase inicial son el consenso de las partes interesadas en torno a los objetivos del proyecto y obtener financiación. Las principales actividades de esta fase son:

- Definir el alcance del proyecto.
- Estimar el coste y el calendario.
- Definir los riesgos.
- Determine la viabilidad del proyecto.
- Preparar el entorno del proyecto.

Para salir de la fase inicial, (Ambler, 2006) indica que el equipo debe superar el hito llamado “**Objetivos del Ciclo de Vida**” (LCO por sus siglas en inglés). Las principales cuestiones son si el equipo comprende suficientemente el alcance del esfuerzo y si las partes interesadas desean financiar el proyecto. Si el equipo supera este hito, el proyecto pasa a la fase de “Elaboración”. En caso contrario, el proyecto puede reorientarse o cancelarse directamente.

2.2.3.2 Fase de Elaboración.

En esta fase se lleva a cabo una planificación detallada y se crea un modelo arquitectónico preliminar del sistema. También se identifican los requisitos del sistema y se realiza una evaluación detallada de riesgos.

Según (Ambler, 2006) el propósito principal de la fase de Elaboración es probar la arquitectura del sistema a desarrollar. El objetivo es garantizar que el equipo pueda realmente desarrollar un sistema que satisfaga los requisitos, desarrollando software funcional y de alta calidad que cumpla con varios casos de uso de alto riesgo (desde un punto de vista técnico) para demostrar que el sistema es técnicamente viable.

En esta etapa, el equipo también se prepara para la fase de Construcción. A medida que dominan la arquitectura del sistema, comienzan a configurar el entorno para la construcción, que implica la adquisición de hardware, software y herramientas. Desde la perspectiva de la gestión de proyectos, se abordan aspectos como la dotación de personal, recursos y la finalización de los planes de comunicación y colaboración, especialmente importante si el equipo va a estar distribuido físicamente.

La fase de Elaboración culmina al superar el hito de “**Arquitectura del ciclo de vida**” (LCA por sus siglas en inglés). El problema principal que se aborda con este hito es si el equipo ha demostrado que tiene un prototipo funcional de principio a fin que demuestre que el equipo tiene una estrategia viable para construir el sistema y que las partes interesadas están preparadas para continuar financiando el proyecto. Si el equipo supera este hito, el proyecto pasa a la fase de “Construcción”, de lo contrario, el proyecto puede ser redireccionado o cancelado.

2.2.3.3 Fase de Construcción

En esta fase se desarrolla y prueba el software. Se realizan iteraciones cortas y se lleva a cabo una evaluación continua del progreso y de los riesgos del proyecto.

Según (Ambler, 2006) El objetivo de la fase de construcción es desarrollar el sistema hasta el punto en que esté listo para las pruebas de preproducción. En fases anteriores, se han identificado la mayoría de los requisitos y se ha establecido la arquitectura del sistema. Ahora el énfasis se centra en priorizar y comprender los requisitos, diseñar una solución y luego codificar y probar el software. Si es necesario, se implementan versiones tempranas del sistema, ya sea interna o externamente, para obtener comentarios de los usuarios.

Para salir de la fase de construcción, se debe superar el hito de “**Capacidad operativa inicial**” (IOC por sus siglas en inglés). El problema principal aquí es si la

versión actual del sistema está lista para pasar a su entorno de prueba de preproducción para las pruebas de aceptación y del sistema. Si el equipo supera este hito, el proyecto pasa a la fase de “Transición”, de lo contrario, puede ser redireccionado o cancelado.

2.2.3.4 Fase de Transición

En esta fase se realiza la implementación del software y se entregan los productos finales al cliente. También se lleva a cabo una evaluación del proyecto y se identifican las lecciones aprendidas para mejorar futuros proyectos.

(Ambler, 2006) nos indica que, la fase de transición se centra en poner el sistema en producción. Es posible que se realicen pruebas exhaustivas durante esta fase, incluidas las pruebas beta. Aquí se lleva a cabo el ajuste fino del producto, así como el retrabajo para abordar defectos importantes (las partes interesadas pueden optar por aceptar la existencia de algunos defectos conocidos en la versión actual).

El tiempo y el esfuerzo invertidos en la Transición varían de un proyecto a otro. El lanzamiento de un sistema completamente nuevo puede implicar la compra y configuración de hardware, mientras que la actualización de un sistema existente puede implicar conversiones de datos y una amplia coordinación con la comunidad de usuarios. Cada proyecto es diferente.

Para salir de la fase de transición, se debe superar el hito de “**Lanzamiento del producto**” (PR por sus siglas en inglés). La cuestión principal aquí es si el sistema se puede implementar de forma segura y eficaz en producción. Si el equipo supera este hito, el proyecto pasa a producción. Si el proyecto falla en cualquiera de las áreas anteriores, el proyecto puede ser redirigido o cancelado (algunos proyectos son tan desastrosos que ni siquiera desea instalarlos).

2.2.4 Disciplinas

(Ambler, 2006) Las disciplinas se realizan de manera iterativa, definiendo las actividades que realizan los miembros del equipo de desarrollo para construir, validar y entregar software funcional que satisfaga las necesidades de sus partes interesadas (ver figura 2.5). Las disciplinas son:

- **Modelo.** El objetivo de esta disciplina es comprender el negocio de la organización, el dominio del problema que aborda el proyecto e identificar una solución viable para abordar el dominio del problema.
- **Implementación.** El objetivo de esta disciplina es transformar su(s) modelo(s) en código ejecutable y realizar un nivel básico de pruebas, en particular pruebas unitarias.
- **Prueba.** El objetivo de esta disciplina es realizar una evaluación objetiva para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema funcione según lo diseñado y verificar que se cumplan los requisitos.
- **Despliegue.** El objetivo de esta disciplina es planificar la entrega del sistema y ejecutar el plan para que el sistema esté disponible para los usuarios finales.
- **Gestión de configuración.** El objetivo de esta disciplina es gestionar el acceso a los productos de trabajo de su proyecto. Esto incluye no sólo el seguimiento de las versiones de los productos de trabajo a lo largo del tiempo, sino también el control y la gestión de sus cambios.
- **Gestión de Proyectos.** El objetivo de esta disciplina es dirigir las actividades que se desarrollan en el proyecto. Esto incluye gestionar riesgos, dirigir personas (asignar tareas, rastrear el progreso, etc.) y coordinar con personas y sistemas fuera del alcance del proyecto para asegurarse de que se entregue a tiempo y dentro del presupuesto.

- **Entorno.** El objetivo de esta disciplina es respaldar el resto del esfuerzo garantizando que el proceso, la orientación (estándares y directrices) y las herramientas (hardware, software, etc.) adecuados estén disponibles para el equipo según sea necesario.

2.2.5 Hitos

(Ambler, 2006) nos indica que en AUP, se presentan cuatro hitos. Al llegar a cada uno de estos puntos, que marcan el cierre de una fase, es recomendable llevar a cabo una "revisión de hitos" para confirmar que el equipo ha cumplido satisfactoriamente con los criterios establecidos. Estos cuatro hitos son:

- Objetivos del ciclo de vida (Lifecycle Objectives - LCO)
- Arquitectura del ciclo de vida (Lifecycle Architecture - LCA)
- Capacidad operativa inicial (Initial Operating Capacity - IOC)
- Lanzamiento del producto (Product Release - PR)

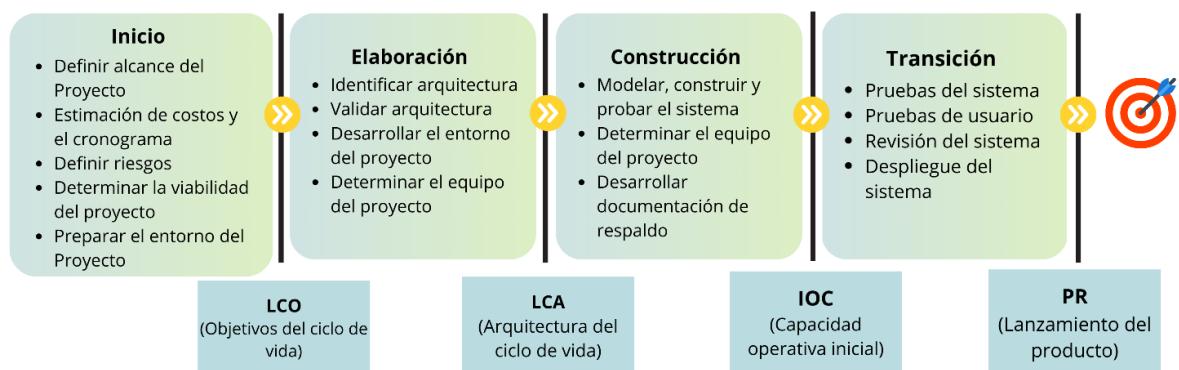


Figura 2.6 Fases e Hitos de AUP

Fuente: Ambler, 2006

(Ambler, 2006) Indica que, en cada hito, se deben cumplir ciertas tareas y acordar actividades específicas:

- **Hito de la fase inicial: objetivos del ciclo de vida (LCO).** En este hito, las partes interesadas evalúan el estado del proyecto. Deberán ponerse de acuerdo en lo siguiente:
 - Concurrencia de alcance.
 - Definición de requisitos iniciales.
 - Concurrencia de planes.
 - Aceptación del riesgo.
 - Aceptación del proceso.
 - Viabilidad.
 - Plano del proyecto.
 - Cumplimiento de cartera.
- **Hito de la fase de elaboración: Arquitectura del ciclo de vida (LCA).** En este hito, las partes interesadas evalúan el estado del proyecto. Deberán ponerse de acuerdo en lo siguiente:
 - Estabilidad de la visión.
 - Estabilidad de la arquitectura.
 - Aceptación del riesgo.
 - Viabilidad.
 - Plano del proyecto.
 - Cumplimiento empresarial.
- **Hito de la fase de construcción: Capacidad operativa inicial (COI).** En este hito, las partes interesadas deben acordar lo siguiente:
 - Estabilidad del sistema.
 - Partes interesadas preparadas.
 - Aceptación del riesgo.
 - Aceptación de costos y estimaciones.
 - Plano del proyecto.
 - Cumplimiento empresarial.

- **Hito de la fase de transición: lanzamiento del producto (PR).** En este hito, las partes interesadas deben acordar lo siguiente:
 - Aceptación de las partes interesadas del negocio.
 - Aceptación de operaciones.
 - Aceptación de soporte.
 - Aceptación de costos y estimaciones.

2.2.6 Ventajas y desventajas de la implementación.

El Proceso Unificado Ágil (AUP) tiene muchas ventajas potenciales, como la entrega temprana y continua de software funcional, la capacidad de adaptarse a los cambios en los requisitos del proyecto y la mejora de la calidad del software a través de la retroalimentación del cliente.

Sin embargo, la implementación de AUP también puede tener desventajas. Una de las principales preocupaciones es la complejidad del proceso. AUP tiene una estructura definida que puede ser difícil de entender y aplicar en algunos proyectos. Además, la implementación de AUP puede ser costosa debido a la necesidad de entrenamiento y herramientas específicas para la metodología.

Otra desventaja potencial es la falta de documentación detallada en algunas fases del proceso. A diferencia del enfoque tradicional de cascada o de RUP, AUP se enfoca en la entrega temprana de software funcional y puede tener menos documentación en algunas fases del proyecto. Esto puede ser un problema si se requiere una auditoría posterior del proyecto.

2.3 Ingeniería Web

(Escribano Santamarina, 2010) La ingeniería web es una nueva área de la ingeniería del software que abarca procesos, técnicas y modelos orientados a los entornos

Web. Consiste en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad.

(Pinzon, 2017) La ingeniería web se describe cómo: "El uso de principios científicos, de ingeniería y de gestión con un enfoque sistemático con el objetivo de desarrollar, desplegar con éxito el mantenimiento de alta calidad de los sistemas basados en la Web y aplicaciones".

(Pressman, 2002) Los sistemas y aplicaciones (WebApps) basados en Web hacen posible que una población extensa de usuarios finales disponga de una gran variedad de contenido y funcionalidad. La ingeniería Web no es un clónico perfecto de la ingeniería del software, pero toma prestado muchos de los conceptos y principios básicos de la ingeniería del software, dando importancia a las mismas actividades técnicas y de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado que dicta un enfoque disciplinado para el desarrollo de un sistema basado en computadora.

2.3.1 La Ingeniería Web Dirigida por Modelos

(Vargas, 2017) El enfoque MDWE (Ingeniería Web Guiada por Modelos) provee metodologías y herramientas para el diseño y desarrollo de aplicaciones Web. Estas herramientas dirigen diferentes problemáticas usando modelos separados (navegación, presentación, datos, etc.), y son soportados por modelos compilados que producen la mayoría de las páginas y lógica de las aplicaciones Web basados en estos modelos.

(Escribano Santamarina, 2010) La Ingeniería Web Dirigida por Modelos (MDWE) es la aplicación de la Arquitectura Dirigida por Modelos al campo del desarrollo de aplicaciones web donde puede resultar especialmente útil debido a la evolución continua de las tecnologías y plataformas web.

En esta dirección existen propuestas de modelado de las cuales podemos destacar dos vertientes importantes:

- Metodologías orientadas al diseño navegacional cuyo objetivo es construir aplicaciones hipermedia en sistemas estáticos. La mayoría de estas aproximaciones están basadas en el Modelo Relacional clásico, o bien en extensiones de éste. Algunos ejemplos destacables de estas iniciativas son OOHDM, WebML, ADM, AutoWeb y RMM.
- El otro grupo de aproximaciones se basan en la idea de extender los métodos de desarrollo orientados a aplicaciones dinámicas tratando de introducir la semántica de la hipermedia como característica inherente a este nuevo tipo de sistemas software. Este tipo de aproximaciones de introducir características navegacionales al modelo OO. En este grupo podemos encontrar los métodos UWE, WSDM, EORM, OOW y OO-Method.

2.3.2 Metodologías en Ingeniería Web

En el ámbito de la ingeniería web, las metodologías desempeñan un papel fundamental para garantizar el éxito de los proyectos de desarrollo de aplicaciones web. Estas metodologías aportan estructura, contribuyen a la definición efectiva de los requisitos y simplifican la implementación, lo que se traduce en la creación de aplicaciones web más eficientes y satisfactorias para los usuarios.

(Escribano Santamarina, 2010) destaca las siguientes metodologías:

- **Web Services Distributed Management (WSDM):** Es una especificación basada en servicios web para gestionar y monitorizar el estado de otros servicios. Es un estándar OASIS (Organization for the Advancement of Structured Information Standards), y WSDM consiste en dos especificaciones:

- Management Using Web Services (MUWS): define como representar y como acceder a las interfaces de gestión de recursos expuestos como servicios web. Define un conjunto básico de operaciones de gestión sobre los servicios, tales como identificación, métricas, configuración y relaciones, además de un formato de eventos estándar.
- Management Of Web Services (MOWS): define como manejar servicios web como recursos y como describir y acceder a las capacidades de gestión utilizando MUWS. Esta especificación permite a las aplicaciones de gestión de servicios web interoperar entre sí.
- **WebML (Web Modeling Language):** Es una metodología de modelado visual de aplicaciones web, centrada especialmente en las aplicaciones de uso intensivo de datos, separando el contenido de la información en páginas, navegación y presentación, que se pueden definir y desarrollar de forma independiente. Permite la especificación de operaciones de manipulación de datos para actualizar la aplicación. Cuenta con cuatro perspectivas: el Modelo Estructural (de los datos de la aplicación), el Modelo de Hipertexto (para cada hipertexto describe qué páginas lo componen, y cómo navegan), el Modelo de Presentación (disposición y apariencia gráfica), y el Modelo de Personalización (para definir operaciones específicas para usuarios ó grupos de usuarios, ya que se almacenan como entidades en el Modelo Estructural). Dispone de una herramienta CASE (WebRatio).
- **OOWS:** Es una extensión del método OO-Method (ya basado en modelos), al cual se les añaden a las técnicas de modelado conceptual, capacidades de expresar la hipermedia inherente a las aplicaciones web. Consta de cinco modelos, que se especifican a continuación:
 - El Modelo de Objetos define la estructura y las relaciones estáticas entre clases identificadas en el dominio del problema.

- En el Modelo Dinámico se describen las posibles secuencias de servicios y los aspectos relacionados con la interacción entre objetos.
- El Modelo Funcional captura la semántica asociada a los cambios de estado entre los objetos motivados por la ocurrencia de eventos o servicios.
- El Modelo de Navegación define la semántica navegacional asociada a las clases de los objetos del modelo.
- El Modelo de Presentación captura los requisitos básicos de presentación de información. Está fuertemente basado en el modelo de navegación y permite definir, de una manera abstracta, la estructura lógica de presentación de los objetos navegacionales en la interfaz de usuario.
- **NDT (Navigational Development Techniques):** Es una metodología orientada a la Ingeniería Dirigida por Modelos (MDE). Aunque en un principio se centraba en las fases de ingeniería de requisitos y análisis, se ha ido ampliando a otras fases del ciclo de vida. Define un conjunto de metamodelos para las fases de requisitos y análisis, y una serie de transformaciones y reglas que permiten obtener los modelos de análisis a partir de ellos. Los metamodelos se representan a partir de MOF (MetaObject Facility), mientras que las transformaciones se definen mediante QVT (Query/View/Transformation). Consta de una herramienta denominada NDT Suite

2.4 UWE (UML-Based Web Engineering).

(Nieves-Guerrero et al., 2014) UWE es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Language) para sus modelos y sus métodos, lo que facilita la transición. La metodología define claramente la construcción de cada uno de los elementos del modelo.

(Jacobson et al., 2006) UWE, o Ingeniería Web basada en UML (UML-based Web Engineering), es un método de ingeniería de software ampliamente utilizado para el desarrollo de aplicaciones web. Se basa en el Lenguaje Unificado de Modelado (UML) y proporciona una base sólida para el diseño y desarrollo de sistemas web.

(Nora Koch, 2016). UWE es un método de ingeniería del software para el desarrollo de aplicaciones web basado en UML. Cualquier tipo de diagrama UML puede ser usado, ya que UWE es una extensión de UML.

(Escribano Santamarina, 2010) UWE (UML-based Web Engineering) sirve para modelar aplicaciones web, y presta una especial atención a la sistematización y personalización (sistemas adaptativos). Provee de perfiles UML, metamodelos, un proceso de desarrollo dirigido por modelos, y herramientas de soporte para el diseño sistemático de aplicaciones web (ArgoUWE y MagicUWE). Utiliza notación basada en UML 2.0 (OMG): para aplicaciones Web en general y para aplicaciones adaptativas en particular.

(Rotta et al., 2016) UWE utiliza notación y diagramas UML para el análisis y diseño de aplicaciones Web. UWE cubre aspectos de navegación, presentación, procesos

de negocios, así como también, aspectos de adaptación. En la figura 2.7 se muestran las dependencias que existe entre los distintos tipos de modelos en UWE.

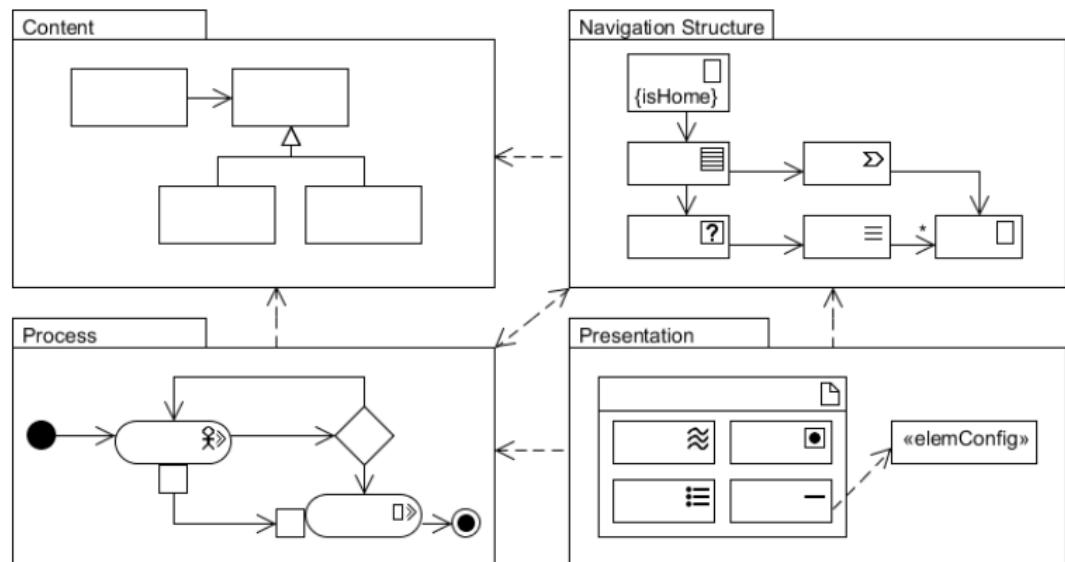


Figura 2.7 Dependencia entre modelos

Fuente: Nora Koch, 2016.

2.4.1 Modelos - Fases de UWE.

(Galiano, 2012) UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas

La metodología UWE se caracteriza por su enfoque basado en modelos, lo que significa que se generan varios tipos de modelos a lo largo del proceso de desarrollo web para representar diferentes aspectos del sistema web. Estos modelos se utilizan como parte integral de cada fase del proceso, es por eso que fase y modelo dentro de esta metodología se ven altamente relacionadas.

Las fases, modelos o etapas a utilizar son:

2.4.1.1 Fase de Requerimientos.

(Busch, 2016) El modelo de requisitos (a veces también llamado modelo de casos de uso) define la funcionalidad de una aplicación y los actores que la utilizan a un alto nivel de abstracción. Para ello se pueden utilizar diagramas de casos de uso.

(Nieves-Guerrero et al., 2014) Plasma los requisitos funcionales de la aplicación Web mediante un modelo de casos de uso.

Una de las primeras actividades en la construcción de aplicaciones Web es la identificación de los requisitos, y en UWE se especifican mediante el modelo de requerimientos, que involucra el modelado de casos de uso con UML. El diagrama de casos de uso está conformado por los elementos actor y caso de uso. Los actores se utilizan para modelar los usuarios de la aplicación Web que para este caso de estudio son los diferentes tipos de usuarios (anónimo, consultor, tutor, alumno) que pueden interactuar con el mismo. Los casos de uso se utilizan para visualizar las diferentes funcionalidades que la aplicación tiene que proporcionar, como son: crear a un nuevo usuario, identificar al usuario, realizar una búsqueda, realizar la composición de un nuevo objeto y guardar el objeto compuesto.

(Galiano, 2012) 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.

En la Figura 2.8 se presenta un ejemplo del diagrama de casos de uso para una aplicación web.

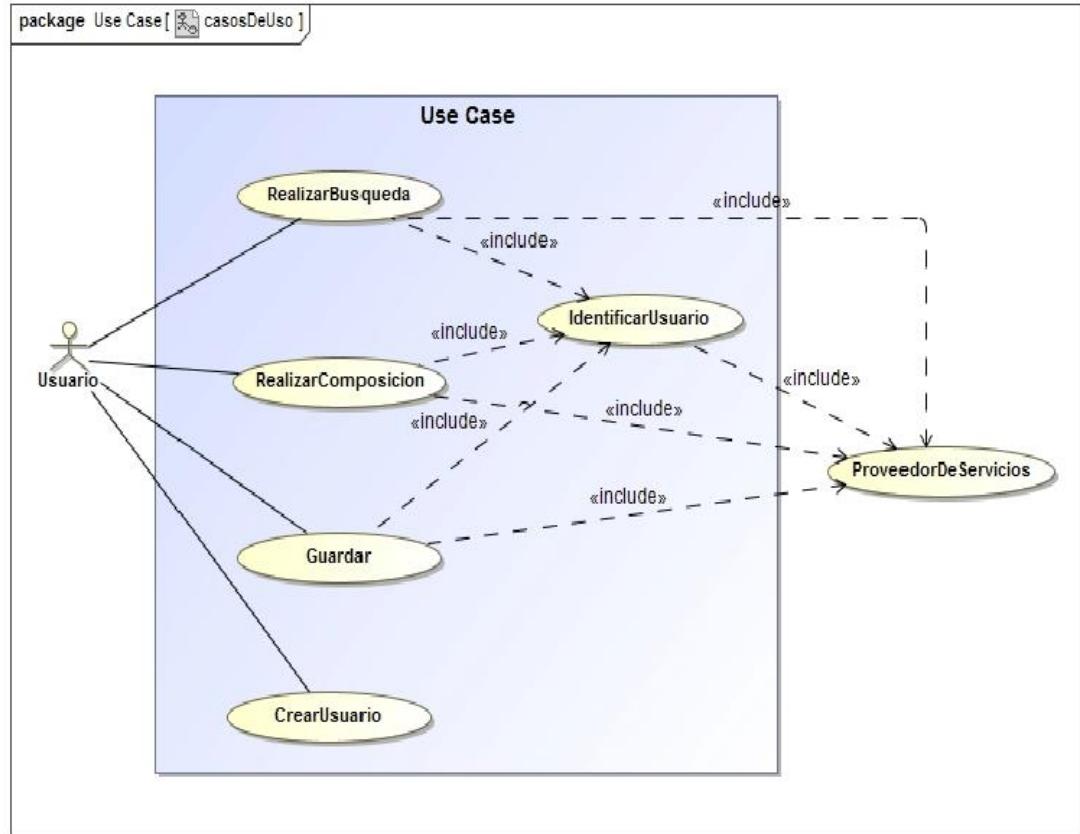


Figura 2.8 Modelo de Requerimientos

Fuente: Nieves-Guerrero et al., 2014

2.4.1.2 Modelo de Contenido.

(Busch, 2016) El modelo de contenido contiene la estructura de datos utilizada por la aplicación. "Estructura de datos" se refiere a la estructura de clases importantes en una aplicación orientada a objetos o a la estructura lógica de los datos en una base de datos y, en muchas ocasiones, ambas están estrechamente relacionadas.

(Nieves-Guerrero et al., 2014) El objetivo del modelo de contenido es proporcionar una especificación visual de la información en el dominio relevante para la aplicación Web. Este es un diagrama UML normal de clases, por ello se debe pensar en las clases que son necesarias para el caso de estudio presentado.

En la figura 2.9 se presenta un ejemplo del diagrama de clases para el modelo de contenido.

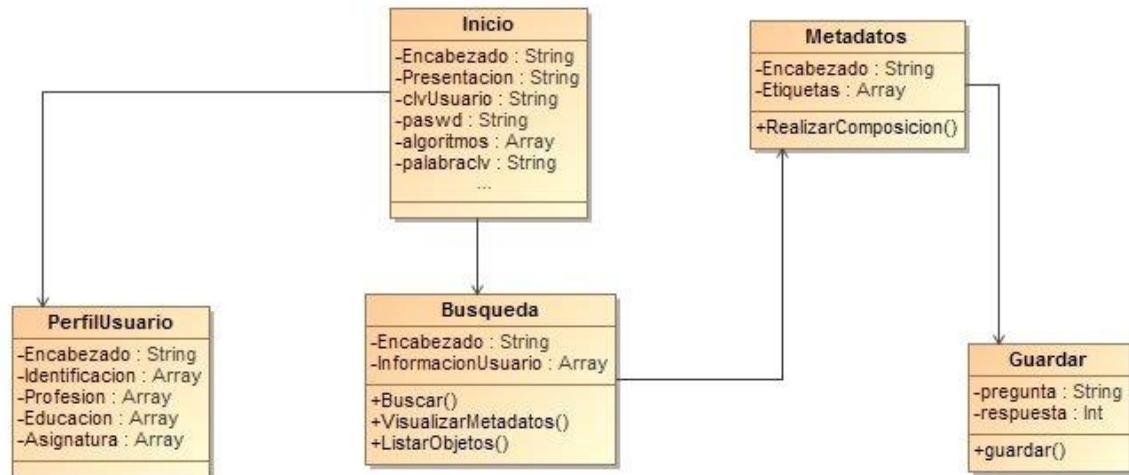


Figura 2.9 Modelo de Contenido

Fuente: Nieves-Guerrero et al., 2014

2.4.1.3 Modelo de Navegación.

(Busch, 2016) El modelo de navegación define el flujo de navegación de la aplicación, así como las políticas de control de acceso a la navegación.

(Galiano, 2012) 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.

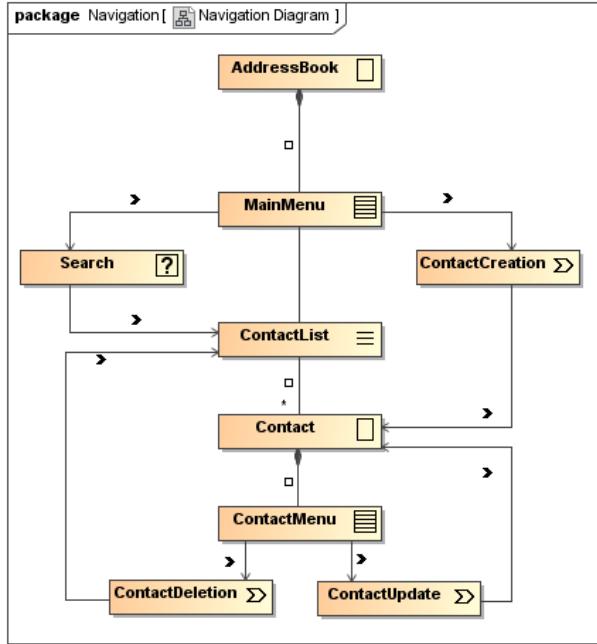


Figura 2.10 Modelo de Navegación

Fuente: Nora Koch, 2016.

(Nieves-Guerrero et al., 2014) Este diagrama se modela con base en el análisis de los requisitos y el modelo de contenido. UWE provee diferentes estereotipos para el modelado de navegación, en la figura 2.10 se presenta un ejemplo del modelo de navegación.

2.4.1.4 Modelo de Presentación.

(Busch, 2016) El modelo de presentación de UWE esboza bloques de construcción lógicos de la interfaz de usuario de una aplicación web. Los diagramas de estructura compuesta pueden utilizarse para expresar la composición como clases y propiedades UML anidadas.

(Galiano, 2012) 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.

(Nieves-Guerrero et al., 2014) El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de una aplicación Web. Se basa en el modelo de navegación y en los aspectos concretos de la interfaz de usuario (IU). En la figura 2.11 se presenta un ejemplo del modelo de presentación.

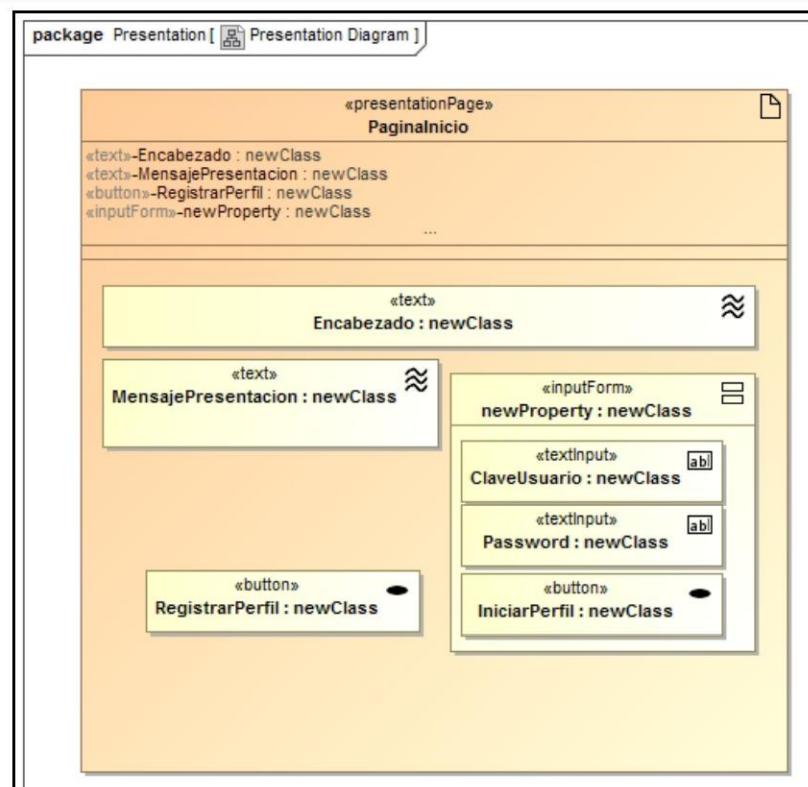


Figura 2.11 Modelo de Presentación

Fuente: Nieves-Guerrero et al., 2014

2.4.1.5 Modelo de Proceso

(Rotta et al., 2016) 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.

(Nieves-Guerrero et al., 2014) El modelo del proceso representa el aspecto que tienen las acciones de las clases de proceso. En este modelo se tienen dos tipos de modelos:

- **Modelo de estructura del proceso.** Es representado por un diagrama de clases donde se describen las relaciones entre las diferentes clases de proceso. La figura 2.12 presenta la aplicación del modelo para el caso de estudio analizado.

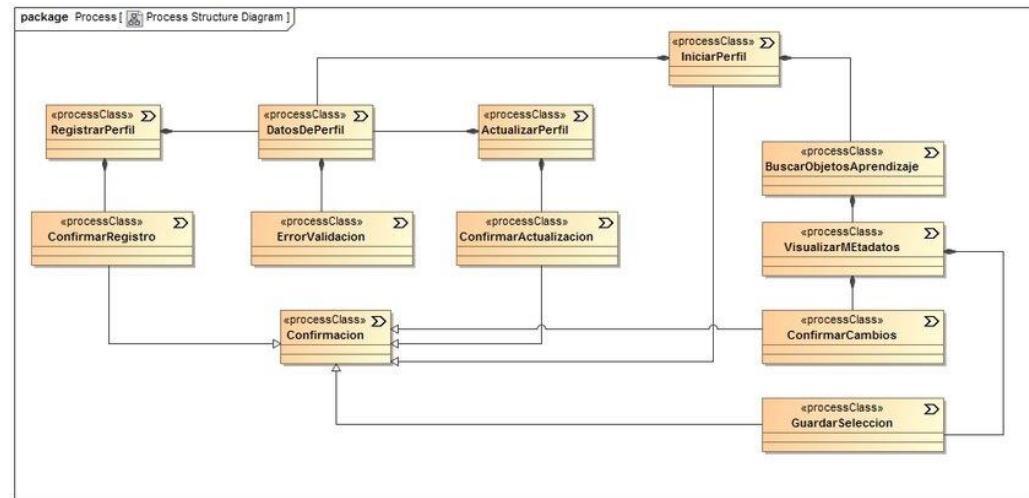


Figura 2.12 Estructura del Proceso

Fuente: Nieves-Guerrero et al., 2014

- **Modelo del flujo del proceso.** Siguiendo el principio de la utilización de UML se han refinado los requisitos con los diagramas de actividad UML. Los diagramas de actividades incluyen actividades, actores responsables de estas actividades (opcional) y elementos de flujo de control. Ellos pueden ser enriquecidos con flujos de objetos que muestran objetos relevantes para la entrada o salida de esas actividades (ver figura 2.13).

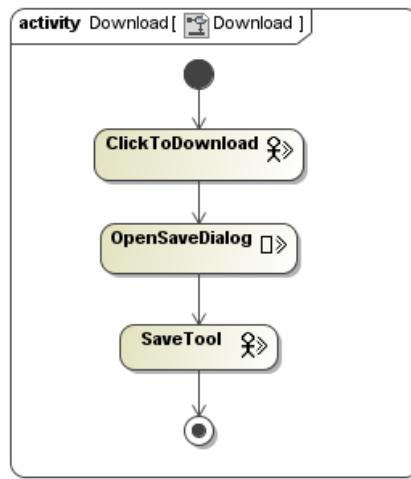


Figura 2.13 Flujo del Proceso

Fuente: Nora Koch, 2016

2.4.1.6 Relación entre Modelos.

(Nieves-Guerrero et al., 2014) nos explica sobre la relación entre cada modelo. UWE provee diferentes modelos que permite describir una aplicación Web desde varios puntos de vista abstractos, dichos modelos están relacionados tal como se ilustra en la figura 2.14. Cada uno de estos modelos se representa como paquetes UML, dichos paquetes son procesos relacionados que pueden ser refinados en iteraciones sucesivas durante el desarrollo del UWE

El análisis de requisitos en UWE se modela con casos de uso. Está conformado por los elementos actor y caso de uso. En este sentido, los actores se utilizan para modelar los usuarios de la aplicación Web. El modelo de contenido es el modelo conceptual del dominio de aplicación tomando en cuenta los requerimientos especificados en los casos de uso y se representa con un diagrama de clases. Basado en el análisis de requisitos y el modelo de contenido se obtiene el modelo de navegación. Éste se representa con clases de navegación. Basado en el modelo de navegación y en los aspectos de la interfaz usuario (requisitos), se obtiene el modelo de presentación. Dicho modelo describe la estructura de la interacción del usuario con la aplicación Web. El modelo de navegación puede ser

extendido mediante clases de procesos. El modelo del proceso representa el aspecto que tienen las acciones de las clases de proceso.

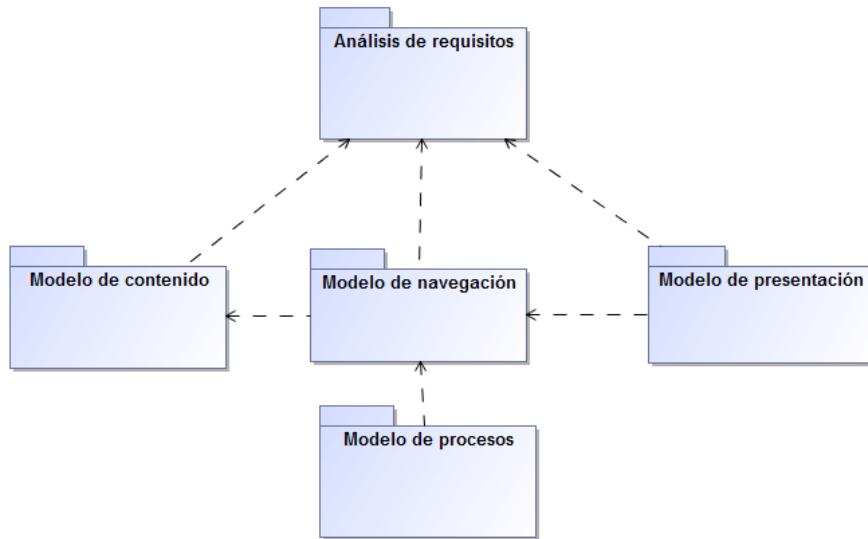


Figura 2.14 Relación entre modelos

Fuente: Nieves-Guerrero et al., 2014

2.5 Tecnologías de Software

En esta sección se presentan las tecnologías de software empleadas durante el desarrollo del sistema.

2.5.1 JavaScript

(MDN, 2023) JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.

2.5.2 PostgreSQL

(IBM, s. f.) PostgreSQL, es una base de datos de código abierto que tiene una sólida reputación por su fiabilidad, flexibilidad y soporte de estándares técnicos abiertos. A diferencia de otros RDMBS (sistemas de gestión de bases de datos relacionales), PostgreSQL soporta tipos de datos relacionales y no relacionales. Esto la convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles actualmente.

Desarrollada originalmente en 1986 como continuación de INGRES (un proyecto de base de datos relacional SQL de código abierto que comenzó a principios de la década de 1970), POSTGRES, ahora conocida como PostgreSQL, fue una creación de Michael Stonebraker, profesor de informática en Berkeley. En 1994, el proyecto agregó soporte para SQL y, poco después, surgió PostgreSQL.

2.5.3 Node JS

(Wikipedia, 2024) Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.⁴ Fue creado por Ryan Dahl.

2.5.4 Sequelize ORM

(Singh, 2023) Un ORM (Mapeador Objeto-Relacional) funciona como un asistente que facilita la manipulación de datos de una base de datos mediante objetos. Esta herramienta simplifica la interacción con los datos y permite realizar consultas de manera

más intuitiva. Además, el uso de un ORM optimiza las consultas específicas a la base de datos (SQL), lo que posibilita su reutilización y mantenimiento ordenado a lo largo del tiempo.

(Sequelize, 2024) Sequelize es una herramienta ORM de Node.js basada en promesas para Postgres, MySQL, MariaDB, SQLite, Microsoft SQL Server, Oracle Database, Amazon Redshift y Snowflake's Data Cloud. Cuenta con un sólido soporte de transacciones, relaciones, carga ansiosa y perezosa, replicación de lectura y mucho más.

(Singh, 2023) Sequelize es una herramienta gratuita para Node.js que ayuda a los desarrolladores de JavaScript a trabajar más fácilmente con ciertos tipos de bases de datos. Es como un ayudante que sabe cómo hablar con bases de datos como MySQL y Postgres, para que no tengas que preocuparte tanto de cómo hacerlo tú mismo.

Sequelize proporciona una capa de abstracción sobre bases de datos relacionales, haciendo más fácil interactuar con bases de datos usando objetos JavaScript en lugar de consultas SQL en bruto.

2.5.5 React JS

(Webman, 2024) React JS es una librería de JavaScript desarrollada y mantenida por Facebook en el año 2013, que permite construir interfaces de usuario para aplicaciones web. Esta trabaja directamente en la capa de la “Vista”, pudiendo crear elementos interactivos que se renderizan eficientemente en el navegador gracias al uso de algunas características importantes como el uso de componentes, JSX y el virtual DOM.

2.5.6 Playwright

(Fateh, 2024) Playwright (a veces denominado Playwright.js) es una herramienta de pruebas web integrales que elimina la complejidad de probar interfaces web en distintos

navegadores y tamaños de pantalla. Cuenta con una documentación exhaustiva, un fuerte apoyo de la comunidad y se centra en la experiencia del desarrollador, con características como una sintaxis de prueba expresiva y paquetes nativos para TypeScript, JavaScript, Python, .NET y Java.

Las pruebas web, también conocidas como pruebas de extremo a extremo y pruebas de navegador, son el proceso automatizado de asegurarse de que una página o aplicación web aparece y funciona según lo previsto mediante la representación de la aplicación y la simulación de las interacciones del usuario, como clics y pulsaciones de teclas. A medida que se ejecutan las interacciones mediante scripts, los resultados se cotejan con un conjunto de reglas y se notifica al desarrollador cualquier fallo, lo que elimina la necesidad de realizar pruebas manuales en distintos entornos.

Playwright ejecuta sus pruebas en un navegador headless, por lo que funciona con cualquier marco frontend, incluidos Angular, React, Vue.js o simplemente HTML y JavaScript.

2.6 Historia Clínica

(Luna Orosco & Carrasco, 2008) La historia clínica es el conjunto documental básico y fundamental para el registro de las características del proceso salud-enfermedad que se presenta en un determinado paciente, sirviendo además como instrumento testimonial que evalúa las condiciones de funcionamiento de cualquier servicio de salud. Su interpretación y análisis tienen un uso y aplicación polivalente, ya sea en lo puramente clínico-nosológico, como en lo académico, investigativo, epidemiológico, estadístico, administrativo, ético y jurídico.

Según Laín Entralgo (1978, citado en Sarsanedas Castellano, s. f.) la historia clínica es el documento fundamental donde se recoge la información confiada por el

enfermo al médico, para obtener el diagnóstico el tratamiento y la posible curación de la enfermedad.

Según Huffman (1981, citado en Sarsanadas Castellano, s. f.) la es historia clínica es información o conjunto de documentos resultantes de la práctica asistencial realizada en torno a un individuo concreto, cualquiera que sea la naturaleza física del documento que soporte la información.

(Sarsanadas Castellano, s. f.) Conjunto de documentos, en cualquier tipo de soporte, donde se registran los hechos relativos a la salud-enfermedad de una persona y la actividad de los profesionales sanitarios relacionada con estos hechos. Esto permite, en cada momento, disponer de un relato global de la atención prestada.

La historia clínica, es un documento donde se plasma la comunicación que tiene el paciente con el médico, para expresarle sus dolencias o enfermedades, es una forma de interrogatorio que da lugar a la generación de datos que pueden ayudar a crear un diagnóstico y un análisis, del cual se puede hacer procesos estadísticos.

Es una herramienta vital para los profesionales de la salud, ya que les permite llevar un registro completo y preciso del estado de salud del paciente y su evolución en el tiempo. Además, la historia clínica también puede ser utilizada para fines estadísticos, investigación médica, y para proporcionar información valiosa a otros profesionales de la salud que puedan estar involucrados en el tratamiento del paciente.

2.6.1 Características de una Historia Clínica.

De acuerdo con (Guzmán & Arias, 2012) las características principales de una historia clínica son:

- **Su práctica es obligatoria.** Ningún acto médico hospitalario o de consultorio debe efectuarse sin su correspondiente registro en la historia clínica. En las

instituciones de salud se exige la historia clínica como elemento indispensable para ejercer una medicina de calidad. Por otro lado, en caso de complicaciones (salvo en algunos casos de extrema urgencia y corto tiempo disponible), su ausencia no tiene excusa.

- **Es irreemplazable.** La escritura de la historia no puede ser reemplazada por la memoria del médico. Es lógico que no se puedan conocer detalles de cada paciente, ni por el número ni por la complejidad individual de cada ser humano.
- **Es privada y pertenece al paciente.** Aquí se integran los conceptos de confidencialidad, secreto profesional e información.

(Velito & Tejada, 2010) las principales características de una historia clínica son:

- **Confidencialidad.** El secreto médico es uno de los deberes principales del ejercicio médico cuyo origen se remonta a los tiempos más remotos de la historia y que hoy mantiene toda su vigencia. La obligación de secretos es uno de los temas del derecho sanitario que más preocupa dada la creciente dificultad de su mantenimiento, el secreto no es absoluto, en la práctica médica pueden surgir situaciones de conflicto entre el deber de secreto y el principio de beneficencia del médico, fuerte protección legal del derecho a la intimidad. El secreto médico, la confidencialidad e intimidad y la historia clínica, son tres cuestiones que se implican recíprocamente y se relacionan. La historia clínica es el soporte documental biográfico de la asistencia sanitaria administrada a un paciente, por lo que es el documento más privado que existe en una persona.
- **Seguridad.** Debe constar la identificación del paciente, así como de los facultativos y personal sanitario que intervienen a lo largo del proceso asistencial.

- **Disponibilidad.** Aunque debe preservarse la confidencialidad y la intimidad de los datos en ella reflejada, debe ser así mismo un documento disponible, facilitándose en los casos legalmente contemplados, su acceso y disponibilidad.
- **Es Única.** La historia clínica debe ser única para cada paciente por la importancia de cara a los beneficios que ocasiona al paciente la labor asistencial y la gestión y economía sanitaria.
- **Es Legible.** Una historia clínica mal ordenada y difícilmente inteligible perjudica a todos, a los médicos, porque dificulta su labor asistencial y a los pacientes por los errores que pueden derivarse de una inadecuada interpretación de los datos contenidos en la historia clínica.

2.6.2 Gestión de Historias Clínicas.

(Chero-Farro et al., 2017) La correcta administración y gestión de las historias clínicas contribuyen a mejorar la calidad de atención de los pacientes. Además, ayuda a la recolección de datos para estudios de investigación, elaboración de estadísticas y sirve de enseñanza para estudiantes y profesionales.

Según Carrión (2015, citado en Davis, 2018) la gestión de historias clínicas se inicia cuando el paciente verifica si cuenta con historia, en tal caso si no se crea una historia la cual se llenará toda su información necesaria e importante con un código único que sea correlativo y será una forma de identificación para ser atendido, y si cuenta solo registra su cita (Carrión, 2015).

Falagán y Nogueira (2003, citado en Davis, 2018) la gestión de historias clínicas es el proceso en el cual permite agilizar y facilitar la atención de los usuarios para su recogimiento de los datos, el registro se realiza tradicional en carpetas donde el medico registrar atenciones y recomendaciones.

Espinoza (2015, citado en Davis, 2018) la gestión de historias clínicas ha ido evolucionando al traspasar los años actualmente existen sistemas informático que permiten reducir los procesos manuales que son las carpetas donde la información de un paciente hacia tener un repositorio de almacenamiento de todas los historiales clínicos dándoles un código de identificación único ,que permite la agilización de los registro de las citas y las búsquedas con dichos códigos ,también el almacenamiento de los resultados de los exámenes ,esto minimiza los tiempos de ejecución del proceso.

Según (Torres Lloveras & Sarsanadas Castellanos, s. f.) la gestión de historias clínicas debe cubrir los siguientes puntos:

- **Planificación.** Para planificar, es necesario disponer de datos morbilidad atendida por los hospitales. Estos datos, se obtienen en gran medida de la historia clínica,
- **Administración.** El análisis de la utilización de recursos en el tratamiento de los distintos procesos es una actividad fundamental para una correcta administración. Para llevarlo a cabo, es necesario disponer de información sobre el proceso patológico del paciente (diagnósticos) y la actividad de los profesionales (procedimientos). Esta información se obtiene únicamente de la historia clínica.
- **Evaluación.** La valoración de la calidad de la asistencia prestada se lleva a cabo a partir de datos que únicamente se registran en la historia clínica.

(Juliá, 2020) señalan los beneficios derivados de implementar una gestión efectiva en las historias clínicas.

- Todos los datos son accesibles desde un solo lugar
- Se cumple con la ley de protección de datos
- Personalizar las fichas de historia clínica
- Seguimiento personalizado

- Conexión con otros módulos
- Alto nivel de escalabilidad

Algunas de las tecnologías y herramientas utilizadas en la gestión de historias clínicas incluyen sistemas de información de historias clínicas electrónicas (EHR), sistemas de información de gestión de prácticas médicas (PM) y sistemas de información de gestión de pacientes (PMS).

2.6.3 Historia Clínica Electrónica (HCE).

Con la llegada de la tecnología y la digitalización de los procesos médicos, surgió la historia clínica electrónica (HCE) como una alternativa más eficiente y segura para la gestión de la información médica de los pacientes. En este marco teórico se describirán las características y ventajas de la HCE, así como su impacto en la gestión de las historias clínicas.

Según (Chaudhry et al., 2006) una HCE es un registro digital longitudinal de la información de salud del paciente, que incluye información demográfica, historia médica, medicamentos prescritos, alergias, notas del médico, resultados de pruebas, radiografías y otros informes de diagnóstico.

(Zhou et al., 2015) las HCE tienen una serie de beneficios en comparación con las historias clínicas en papel. Las HCE son más precisas y completas, y permiten un acceso más rápido y fácil a la información del paciente. Además, se pueden compartir con varios proveedores de atención médica y se pueden acceder desde cualquier lugar con una conexión a Internet.

Según indica (Ghiglia, 2019), la HCE es el centro de cualquier sistema de información en salud, su función es adquirir datos clínicos relacionados con el paciente para su uso y análisis por parte del equipo de salud. Un sistema de HCE permite:

- Acceso electrónico inmediato a la información de salud personal o poblacional solamente de usuarios autorizados.
- Provisión de bases de conocimiento y sistemas de soporte para la toma de decisiones que mejoren la calidad, seguridad y eficiencia de la atención de los pacientes.
- Dar soporte efectivo en la eficiencia de los procesos para brindar cuidados de salud.

(Torres Lloveras & Sarsanedas Castellanos, s. f.) nos indica que la HCE es la interacción del profesional clínico y el conocimiento existente en el sistema de información. Esta interacción se concreta según lo siguiente:

- Registrar y consultar datos clínicos de un paciente individual.
- Gestionar el proceso asistencial de un paciente (programar actividad sanitaria relativa a ese paciente).
- Soporte en la toma de decisiones:
- Trabajar con datos de grupos de pacientes para obtener indicadores.

Según (Organización Mundial de la Salud, 2013) para implementar una HCE, es necesario utilizar sistemas de información especializados. Los sistemas de información en salud son sistemas electrónicos diseñados para recolectar, procesar, almacenar y distribuir información relacionada con la salud. Estos sistemas incluyen sistemas de gestión de información hospitalaria (HIMS), sistemas de información de gestión de pacientes (PIMS) y sistemas de información clínica (CIS).

CAPÍTULO III MARCO APlicativo

3.1 Introducción

En el presente capítulo se realiza el análisis y diseño para el “Sistema de Administración de Historias Clínicas”, haciendo uso de los instrumentos, métodos, técnicas, y herramientas mencionadas en el capítulo anterior.

Para su desarrollo, utilizaremos la metodología AUP, abordando sus fases de iniciación, elaboración, construcción y transición de manera iterativa.

Tabla 3.1 Fases de implementación aplicando la metodología AUP

| FASE | APLICACIÓN |
|--------------|--|
| Iniciación | Análisis de los requerimientos, definir la arquitectura en base a los requerimientos del cliente |
| Elaboración | <ul style="list-style-type: none">• Definir requerimientos funcionales.• Definir casos de uso del negocio.• Definir los modelos. |
| Construcción | <ul style="list-style-type: none">• Casos de uso del sistema.• Codificación del software• Implementar la solución. |
| Transición | Liberar el sistema. |

Esta metodología nos permite desarrollar el sistema de manera progresiva en cada etapa. Además, emplearemos la metodología UWE para modelar el sistema con un enfoque web.

La integración de ambas metodologías se detalla en la tabla 3.2, mostrando el proceso de integración entre ellas.

Tabla 3.2 Integración AUP y UWE

| Fase de Inicio | Fase de Elaboración | Fase de Construcción | Fase de Transición |
|--|--|--|--------------------|
| <ul style="list-style-type: none"> • Modelo AUP • Análisis de requerimientos | <ul style="list-style-type: none"> • Modelo de requerimientos UWE • Modelo de contenido UWE para el negocio. | <ul style="list-style-type: none"> • Modelo de contenido UWE • Modelo de presentación UWE • Modelo de navegación UWE • Modelo de proceso UWE | Producto |

La siguiente figura detalla la planificación y la estructura de la integración de ambas metodologías para el desarrollo del sistema.

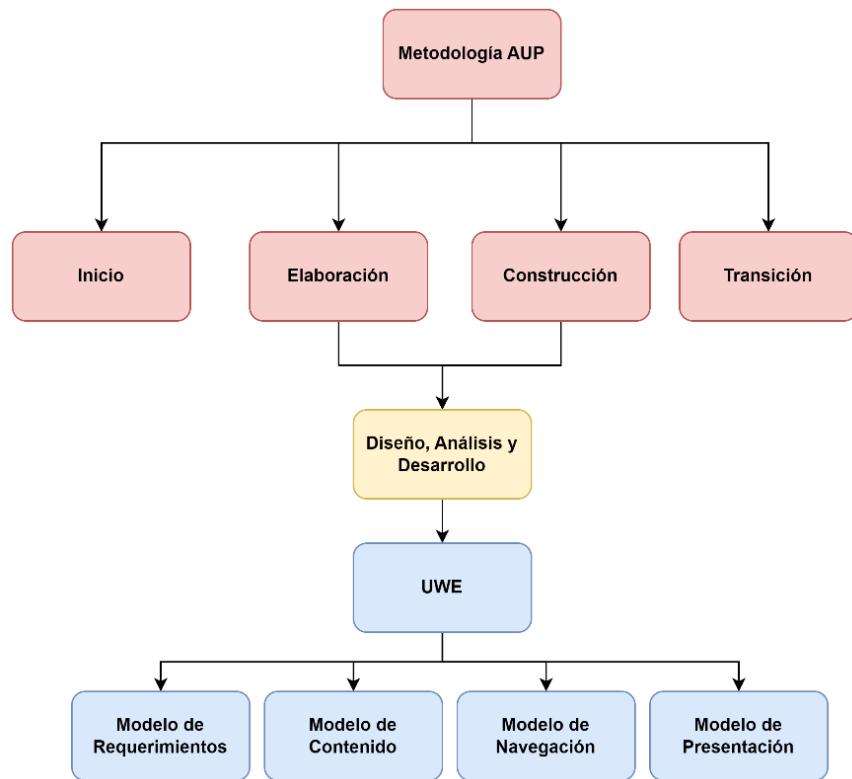


Figura 3.1 Planificación de Integración Metodológica

3.2 Fase de Inicio

La fase inicial tiene como propósito identificar los requerimientos, creando historias de usuario que se convertirán luego en requisitos para el desarrollo del software. Esta etapa es breve y se enfoca en las actividades de modelado empresarial. Los requisitos obtenidos para esta fase son los siguientes:

- **Entrevista:** Entrevistas directas con la directora del Instituto de Genética y el médico a cargo de la unidad de genética médica.
- **Observación:** Observación de los procesos que el personal realiza para la obtención, registro y acceso a la información de los datos clínicos de un paciente.

- **Documentación:** El instituto proporcionó documentación de anteriores gestiones, así como un archivo de Access que servía como una alternativa para almacenar información, aunque no era utilizado. Se obtuvieron modelos de historias clínicas de distintas administraciones (ver anexo 1 y 2), además de archivos de Excel con datos de pacientes.

3.2.1 Situación actual

Actualmente, la Unidad de Genética Médica del Instituto de Genética de la Universidad Mayor de San Andrés no dispone de un sistema integrado para administrar la información de las historias clínicas de los pacientes. Por esta razón, toda la información se gestiona manualmente, utilizando documentos Excel y medios físicos como hojas de papel (ver anexo 1 y 2).

El diagrama 3.1 detalla el proceso actual para el registro de historias clínicas. En él se delinean las decisiones involucradas, desde la identificación inicial del paciente hasta la consolidación de su información médica, ofreciendo una visión clara del flujo de trabajo que actualmente lleva a cabo el instituto.

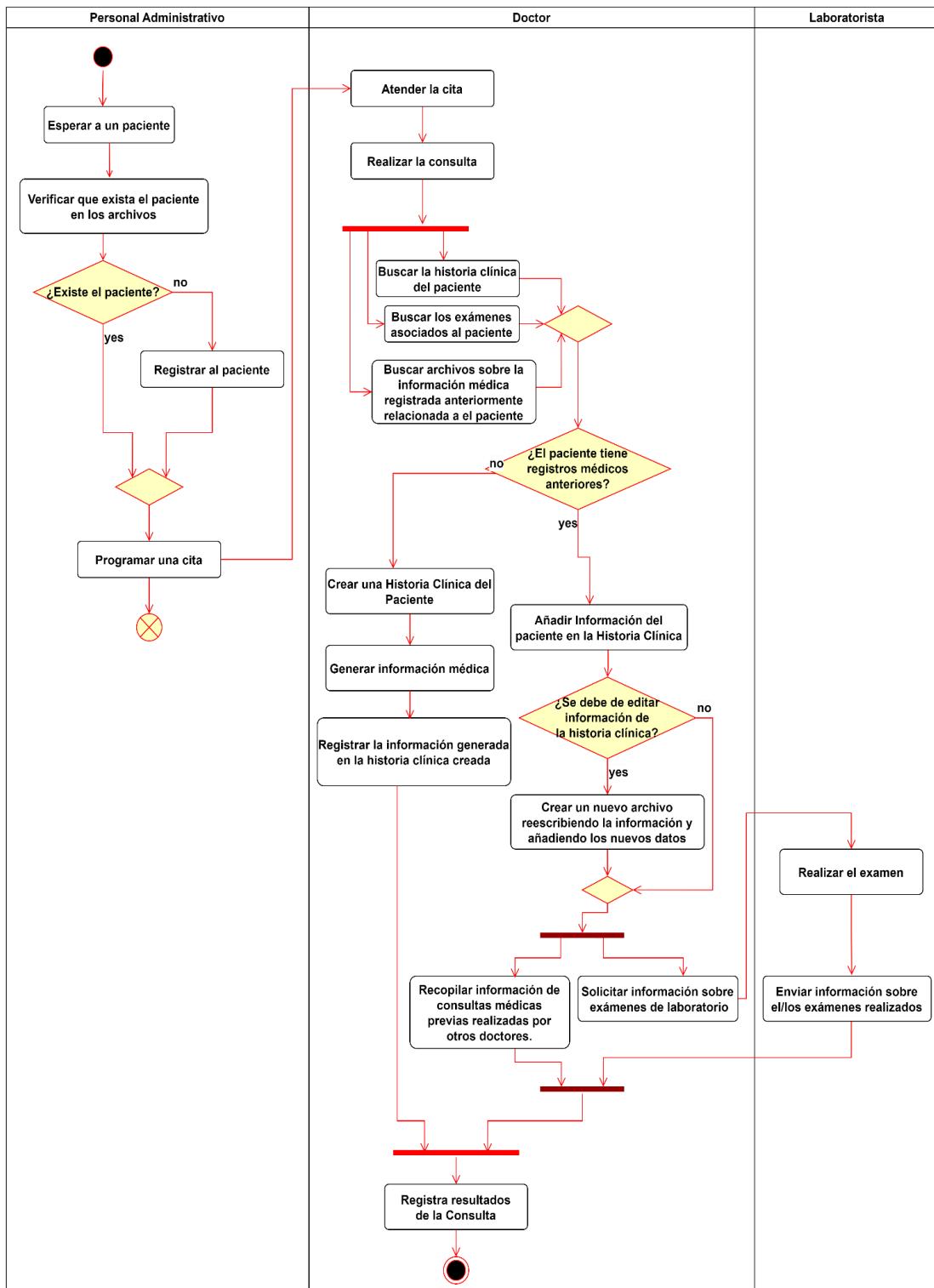


Diagrama 3.1 Diagrama de Actividades del flujo actual del sistema

3.2.2 Especificación de Requerimientos

En este apartado se presentan los requerimientos que deberían ser cumplidos por el sistema. Los requerimientos se obtuvieron a través de entrevistas realizadas al personal administrativo, doctores y laboratoristas del Instituto de Genética, involucrados en el proyecto.

Con la información obtenida anteriormente, se analizó y se obtuvieron los siguientes requerimientos.

3.2.2.1 Requerimientos Funcionales

Tabla 3.3 Tabla de Requerimientos Funcionales

| ID | Requerimiento | Descripción | Prioridad |
|-----|--|---|-----------|
| RF1 | El administrador podrá crear y gestionar cuentas de usuario dentro del sistema. | El sistema permitirá el registro de nuevos usuarios según el rol. | 5 |
| RF2 | Cada usuario podrá iniciar sesión con un correo válido. | El sistema permitirá el inicio de sesión de usuarios únicamente a través de una dirección de correo electrónico y contraseña que haya sido validada y registrada previamente. | 5 |
| RF3 | El personal administrativo podrá registrar a los pacientes. | El sistema permitirá el registro de pacientes. | 5 |
| RF4 | El personal administrativo podrá modificar los datos esenciales de un paciente, además de poder buscarlos utilizando su número de carnet de identidad. | El sistema permitirá editar solo la información de los datos personales del paciente, además de facilitar la búsqueda mediante el carnet de identidad. | 4 |
| RF5 | El personal administrativo podrá agendar una cita. | El sistema posibilitará el registro de una cita, lo cual incluirá la selección de un | 5 |

| | | | |
|------|--|---|---|
| | | doctor y la elección de una fecha para dicha cita. | |
| RF6 | El doctor podrá ver a los pacientes registrados, además de buscarlos por su carnet de identidad. | El sistema podrá listar a los pacientes, además de facilitar la búsqueda mediante el carnet de identidad. | 4 |
| RF7 | El doctor podrá visualizar la historia clínica de un paciente. | El sistema permitirá la visualización de la información de la historia clínica de un paciente. | 4 |
| RF8 | El doctor podrá visualizar las citas que debe atender. | El sistema permitirá la visualización de las citas. | 4 |
| RF9 | El doctor podrá atender una cita, es decir, llevar a cabo una consulta médica. Además de editar o crear la historia clínica en el proceso de dicha consulta. | El sistema permitirá el registro de una consulta en la historia clínica, incluyendo la capacidad de actualizar la información de la misma o crearla si así fuera el caso. | 5 |
| RF10 | El laboratorista podrá registrar la descripción de los exámenes realizados a un paciente en su historia clínica. | El sistema permitirá registrar descripciones de exámenes en la historia clínica, brindando la posibilidad de editar dichas descripciones. | 5 |
| RF11 | El laboratorista y el personal administrativo podrán visualizar la historia clínica de un paciente. | El sistema permitirá visualizar la información de una historia clínica. | 4 |
| RF12 | Generación de reportes. | Los datos registrados en la historia clínica podrán ser impresos. | 4 |

3.2.2.2 Requerimientos no Funcionales

Tabla 3.4 Tabla de Requerimientos no funcionales

| ID | Requerimientos | Descripción | Prioridad |
|------|----------------|--|-----------|
| RFN1 | USABILIDAD | Debe ser fácil de usar, con la ayuda de interfaces intuitivas. | 5 |

| | | | |
|------|--------------|--|---|
| RFN2 | SEGURIDAD | <p>El ingreso y visualización del sistema estará restringido bajo autenticación y autorización por roles. Las contraseñas de los usuarios deberán almacenarse de forma segura usando algoritmos de hash modernos.</p> <p>El acceso a historias clínicas estará restringido solo a personal autorizado.</p> | 5 |
| RFN3 | EFICIENCIA | <p>El almacenamiento de datos debe ser eficiente para asegurar que las historias clínicas se guarden y recuperen de manera rápida y segura.</p> | 5 |
| RFN4 | PORTABILIDAD | <p>El sistema debe brindar comodidad al usuario y a las demás áreas que necesitan acceder al sistema. Además de funcionar en toda la infraestructura que cuenta el Instituto de Genética.</p> | 5 |

3.3 Fase de Elaboración

En esta etapa, se examinan los requerimientos y se construye un prototipo de arquitectura que abarca las secciones más críticas o significativas del sistema. Se realizará el modelado de requerimientos, describiendo los casos de uso del negocio.

3.3.1 Descripción de los Actores

La identificación de actores nos permite, en líneas generales, reconocer a aquellos individuos que participan en la interacción con el sistema. Estos usuarios contribuyen o reciben información para mejorar el rendimiento del sistema, la siguiente tabla describe los actores que interactuarán con el sistema.

Tabla 3.5 Descripción de Actores

| Actores | Descripción |
|--------------------------------|--|
| Administrador | El usuario con mayores privilegios, se encarga del registro de usuarios y tiene acceso a todo el sistema |
| Personal Administrativo | El usuario que se encarga del registro de pacientes, registro de historias clínicas y de registrar citas. |
| Doctor | El usuario con más privilegios después del administrador, atiende una consulta, hace seguimiento a la historia clínica de un paciente, modificando y añadiendo información a la misma. |
| Laboratorista | El usuario que se encarga de añadir información sobre exámenes realizados a la historia clínica de un paciente |

3.3.2 Modelo de Casos de Uso del Negocio

Los casos de uso del negocio describen las acciones y procesos que se llevan a cabo en una organización para lograr sus objetivos y cumplir con sus requisitos. Estos casos de uso se enfocan en el punto de vista del negocio y cómo el sistema puede apoyar y mejorar estos procesos. El diagrama 3.2 describe los casos de uso del negocio del Instituto de Genética relacionado a la historia clínica de un paciente.

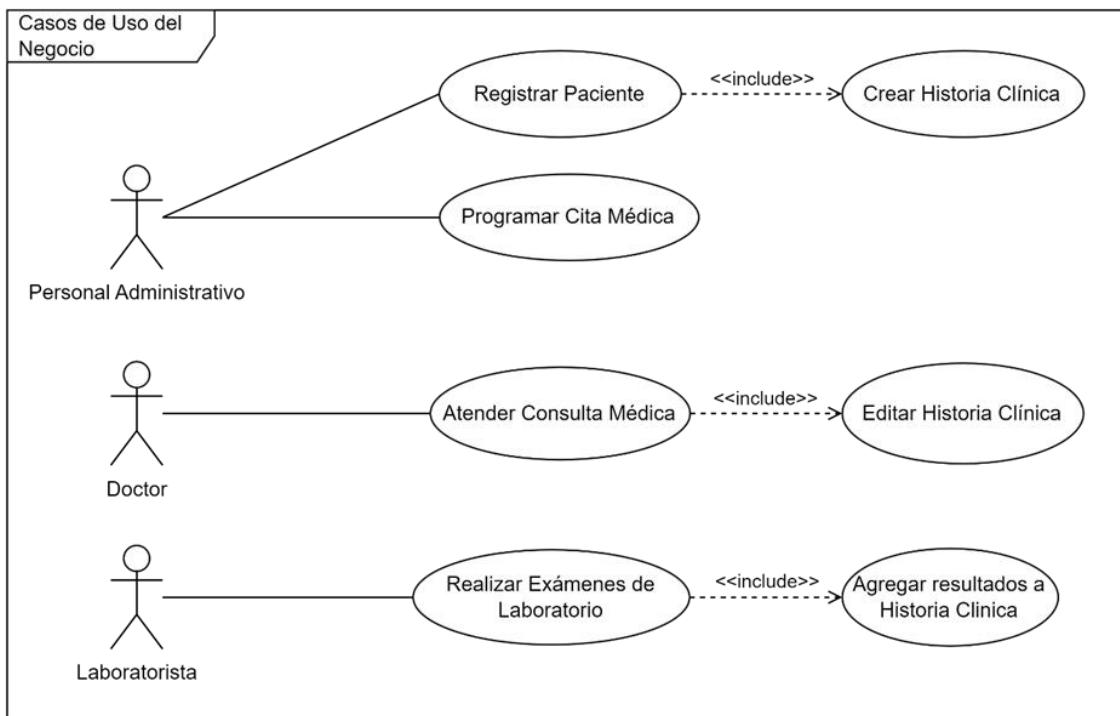


Diagrama 3.2 Casos de uso del negocio

3.3.2.1 Descripción de los casos de uso del negocio

En las siguientes tablas se detallan los casos de uso, incluyendo la interacción entre los diversos actores y los procesos correspondientes.

Tabla 3.6 Descripción de caso de uso del negocio - Registra Paciente

| CASO DE USO: | REGISTRAR PACIENTE |
|--------------|---|
| ACTORES: | Personal Administrativo |
| TIPO: | Primario |
| DESCRIPCIÓN: | El personal administrativo verifica si el paciente está registrado. |

| | |
|--|--|
| | <p>Si no lo está, se procede a registrar al paciente creando su historia clínica con sus datos personales.</p> <p>Si el paciente ya está registrado, se omite el registro.</p> |
|--|--|

Tabla 3.7 Descripción de caso de uso del negocio - Programar Cita Médica

| | |
|---------------------|---|
| CASO DE USO: | PROGRAMAR CITA MÉDICA |
| ACTORES: | Personal Administrativo |
| TIPO: | Primario |
| DESCRIPCIÓN: | Se programa una cita médica para el paciente registrado. Se asigna fecha y hora, y se elige al doctor para atender la cita. |

Tabla 3.8 Descripción de caso de uso del negocio - Atender Consulta Médica

| | |
|---------------------|--|
| CASO DE USO: | ATENDER CONSULTA MÉDICA |
| ACTORES: | Doctor |
| TIPO: | Primario |
| DESCRIPCIÓN: | El doctor atiende la consulta médica, puede agregar o editar información en la historia clínica del paciente. Al finalizar, genera un resumen de hallazgos/descripciones de la consulta y lo agrega a la historia clínica. |

Tabla 3.9 Descripción de caso de uso del negocio - Realizar Exámenes de Laboratorio

| | |
|---------------------|--|
| CASO DE USO: | REALIZAR EXÁMENES DE LABORATORIO |
| ACTORES: | Laboratorista |
| TIPO: | Primario |
| DESCRIPCIÓN: | El laboratorista realiza los exámenes requeridos para el paciente. Agrega los resultados de los exámenes a la historia clínica del paciente. |

3.3.3 Modelo de objetos del negocio

El modelo de objetos del negocio representa la ejecución de cada caso de uso del negocio, detallando los actores internos, la información que manipulan y los flujos de trabajo relacionados con cada caso de uso. Los diagramas correspondientes ofrecerán una visión específica de los flujos asociados a cada actor y caso de uso, proporcionando una representación visual de los procesos en cuestión.

3.3.3.1 Registrar Paciente

En el diagrama 3.3 nos muestra el flujo de actividades relacionadas con el proceso de registro de un paciente. Incluye pasos como verificar si el paciente está registrado, registrar a un nuevo paciente y crear su historia clínica en caso de que no esté registrado.

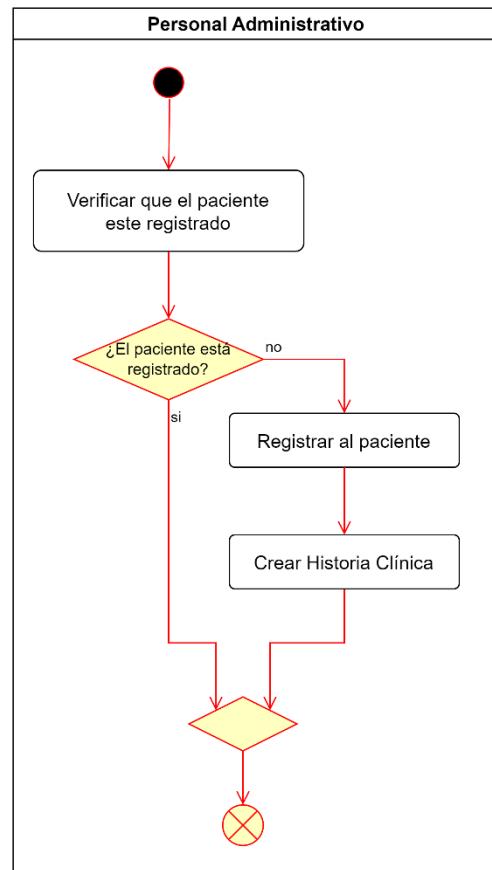


Diagrama 3.3 Registrar Paciente

3.3.3.2 Programar Cita Médica

El diagrama 3.4 representa el flujo de actividades relacionadas para la programación de una cita médica.

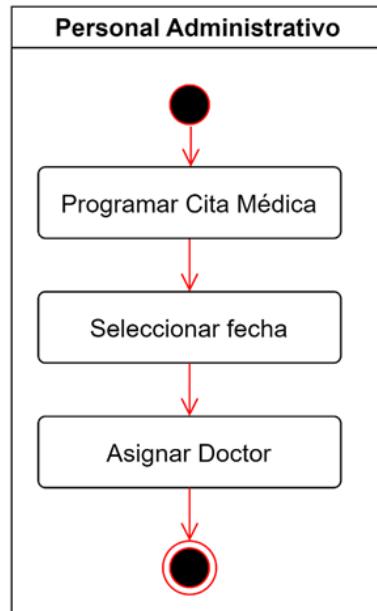


Diagrama 3.4 Programar Cita Médica

3.3.3.3 Atender Consulta Médica

El diagrama 3.5 representa el flujo de actividades relacionadas con el proceso de atender una consulta médica. La representación visual proporcionada por el diagrama de actividad ofrece una comprensión clara y detallada de los pasos involucrados en la atención médica, permitiendo una visión general de los procesos y el flujo de trabajo que el doctor realiza.

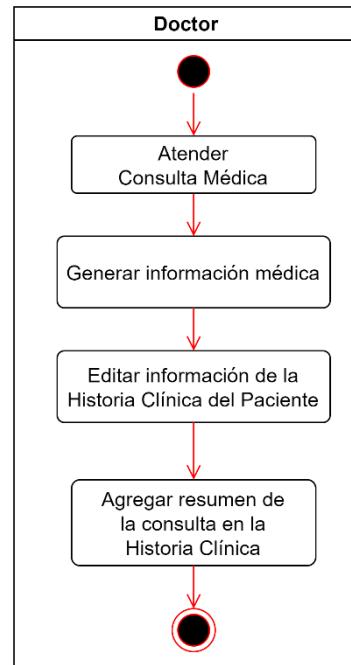


Diagrama 3.5 Atender Consulta Médica

3.3.3.4 Realizar Exámenes de Laboratorio

El diagrama 3.6 representa el flujo de actividades que realiza el laboratorista para realizar exámenes de laboratorio y agregar los resultados a la historia clínica del paciente.



Diagrama 3.6 Realizar Exámenes de Laboratorio

3.4 Fase de Construcción

Durante la fase de construcción, se completan el análisis y diseño de todos los casos de uso, mejorando y refinando el Modelo de Análisis y Diseño.

3.4.1 Casos de uso del Sistema

El modelo de casos de uso del sistema describe las funcionalidades y los actores que interactúan con el sistema. Se suele representar a través de los Diagramas de Casos de Uso. El diagrama 3.7 describe los casos de uso del sistema, ilustrando las distintas funciones que cada actor realiza dentro del sistema.

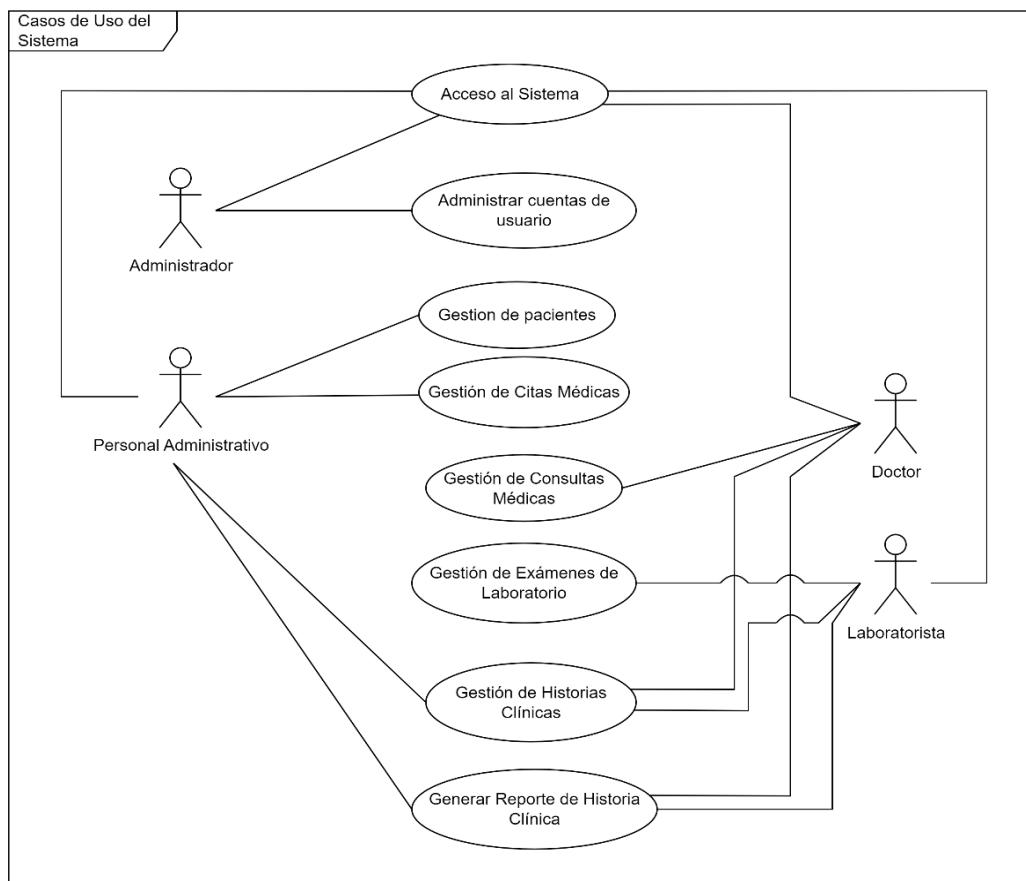


Diagrama 3.7 Casos de uso del sistema

3.4.2 Especificaciones de Casos de Uso

En esta sección se presentan las especificaciones de los casos de uso asociados a los procesos visualizados en el diagrama de casos de uso del sistema. Estas descripciones ofrecen una visión detallada de los distintos procesos que los actores realizan dentro del sistema.

3.4.2.1 Acceso al Sistema

El diagrama 3.8 nos muestra el flujo de los procesos relacionados al caso de uso de acceso al sistema, ofreciendo una visión estructurada de cómo interactúan los diferentes elementos del sistema para permitir el acceso de los usuarios.

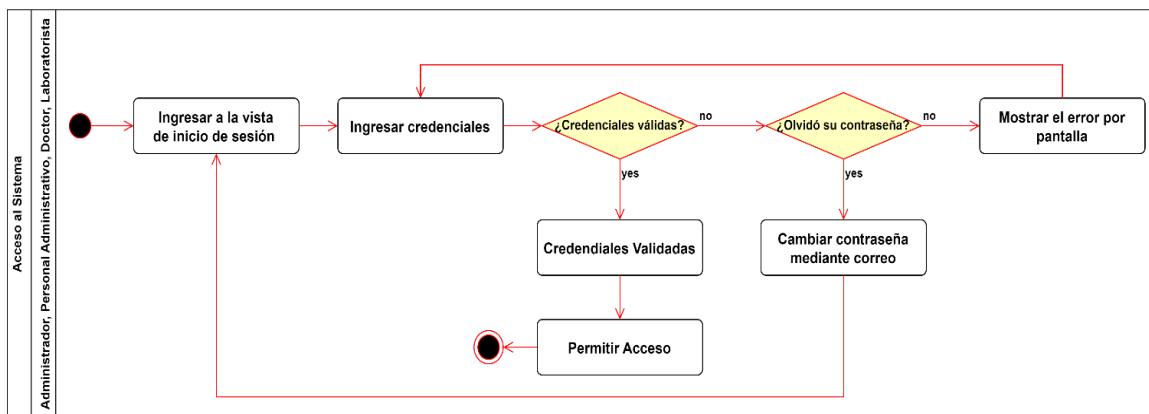


Diagrama 3.8 Diagrama de Actividades - Acceso al Sistema

En la tabla 3.10 se describe el caso de uso de acceso al sistema.

Tabla 3.10 Especificación – Acceso al Sistema

| Nombre | Acceso al Sistema |
|-------------|--|
| Descripción | Describe el proceso de acceso al sistema a través de un navegador web. |

| | |
|---------------------------------------|---|
| Actores | Administrador, Personal administrativo, Doctor y Laboratorista. |
| Precondición | El sistema está en funcionamiento y disponible para el acceso. El usuario tiene credenciales válidas. |
| Post Condiciones | El usuario accede al sistema y puede interactuar con las funciones asignadas a su rol. |
| Flujo normal | <ul style="list-style-type: none"> • El usuario ingresa al sistema web • El usuario ingresa su correo y contraseña. • El campo correo debe ser un correo válido, • El campo contraseña debe ser mayor a cinco caracteres. • El sistema verifica las credenciales del usuario. • Si las credenciales son correctas, se permite el acceso al sistema. |
| Flujo alternativo | <ul style="list-style-type: none"> • En caso de que el usuario digite mal sus credenciales o que no sean credenciales válidas, el sistema informará sobre el error. • Si el usuario olvido su contraseña hay un apartado donde puede generar una nueva contraseña a través de su correo. |
| Requisitos no- funcionales | Seguridad y autenticación para el inicio de sesión. |

3.4.2.2 Administrar cuentas de usuario

El diagrama 3.9 nos muestra el flujo de los procesos relacionados al caso de uso de administrar cuentas de usuario, ofreciendo una visión estructurada de los procesos que interactúan para la gestión de cuentas de usuario del sistema.

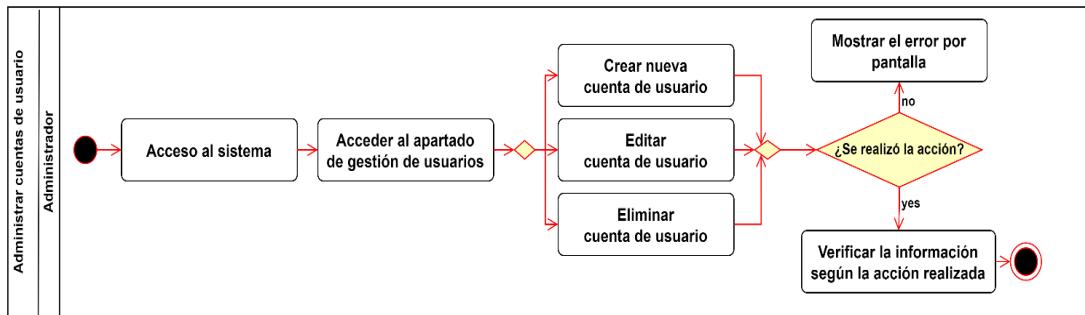


Diagrama 3.9 Diagrama de Actividades - Administrar Cuentas de Usuario

En la tabla 3.11 se describe el caso de uso de administrar cuentas de usuario.

Tabla 3.11 Especificación – Administrar Cuentas de Usuario

| Nombre | Administrador Cuentas de Usuario |
|---------------------------|--|
| Descripción | Describe el proceso de gestionar (ingresar, actualizar o eliminar) a los usuarios que tendrán acceso al sistema. |
| Actores | Administrador. |
| Precondición | El usuario tiene permisos de administrador y debe estar autenticado dentro del sistema. |
| Post Condiciones | Las cuentas de usuario se crean, modifican o eliminan según la acción realizada por el administrador. |
| Flujo normal | <ul style="list-style-type: none"> El administrador crea, edita o elimina cuentas de usuario. Se realizan validaciones para garantizar la integridad y seguridad de las cuentas. |
| Flujo alternativo | En caso de encontrar algún error, el sistema informara que error se ha encontrado |
| Requisitos no-funcionales | Se necesita un nivel de acceso y autorización más alto para esta acción. |

3.4.2.3 Gestión de pacientes

El diagrama 3.10 nos muestra el flujo de los procesos relacionados al caso de uso de gestión de pacientes.

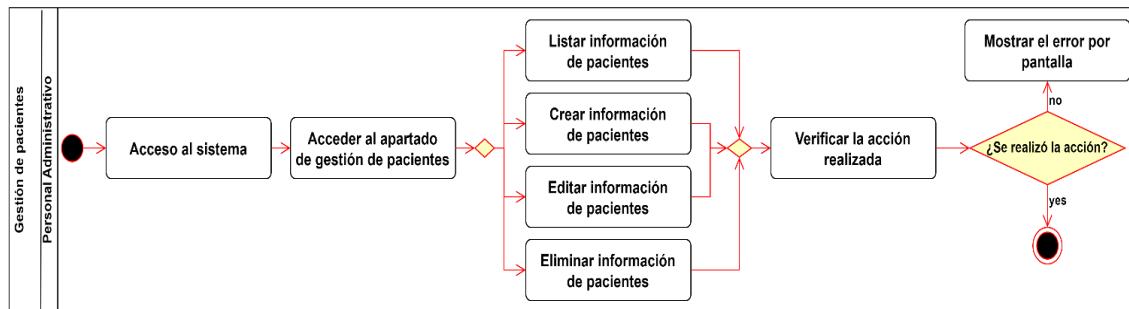


Diagrama 3.10 Diagrama de Actividades - Gestión de Pacientes

En la tabla 3.12 se describe el caso de uso de gestión de pacientes.

Tabla 3.12 Especificación – Gestión de Pacientes

| Nombre | Gestión de Pacientes |
|-------------------------|---|
| Descripción | Describe el proceso de gestionar (ingresar, actualizar o eliminar) a los pacientes dentro del sistema. |
| Actores | Personal Administrativo. |
| Precondición | El personal administrativo tiene acceso a los datos de los pacientes y debe estar autenticado dentro del sistema. |
| Post Condiciones | Se realizan cambios (creación, edición o eliminación) en la información de los pacientes en el sistema. |
| Flujo normal | <ul style="list-style-type: none">• El personal administrativo puede listar, crear, editar o eliminar información de pacientes.• Se verifican los datos ingresados para garantizar su exactitud. |

| | |
|----------------------------------|---|
| Flujo alternativo | En caso de encontrar algún error, el sistema informara que error se ha encontrado |
| Requisitos no-funcionales | Acceso controlado y restricciones en la edición o eliminación de datos sensibles. |

3.4.2.4 Gestión de citas médicas

El diagrama 3.11 nos muestra el flujo de los procesos relacionados al caso de uso de gestión de citas médicas.

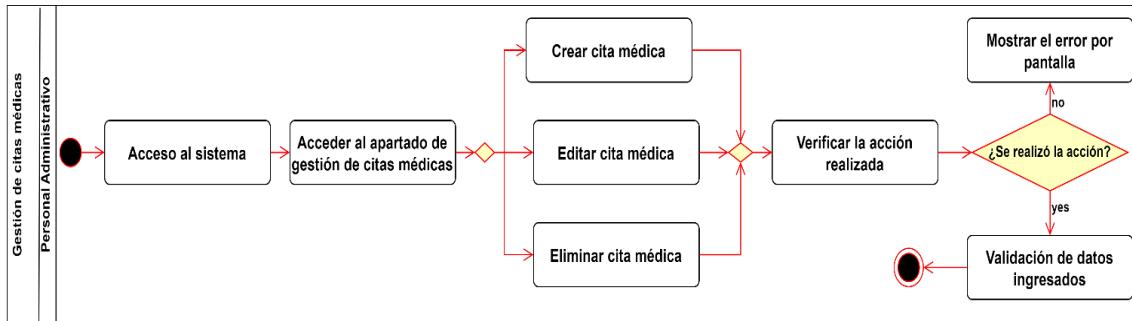


Diagrama 3.11 Diagrama de Actividades - Gestión de Citas Médicas

En la tabla 3.13 se describe el caso de uso de gestión de citas médicas.

Tabla 3.13 Especificación – Gestión de Citas Médicas

| Nombre | Gestión de Citas Médicas |
|-------------------------|---|
| Descripción | Describe el proceso de gestionar (crear, actualizar o eliminar) citas médicas dentro del sistema. |
| Actores | Personal Administrativo. |
| Precondición | El personal administrativo requiere autenticarse en el sistema para acceder a los datos de los pacientes y a la lista de médicos disponibles. |
| Post Condiciones | Se programan citas médicas para los pacientes en el sistema, asignando un doctor, fecha y hora. |

| | |
|----------------------------------|---|
| Flujo normal | <ul style="list-style-type: none"> El personal administrativo crea, edita o elimina citas médicas, asignando médicos, fecha y hora. Se asegura la consistencia y precisión de los datos ingresados. |
| Flujo alternativo | En caso de encontrar algún error, el sistema informará que error se ha encontrado |
| Requisitos no-funcionales | Acceso restringido para evitar manipulaciones indebidas de las citas. |

3.4.2.5 Gestión de consultas médicas

El diagrama 3.12 nos muestra el flujo de los procesos relacionados al caso de uso de gestión de consultas médicas.

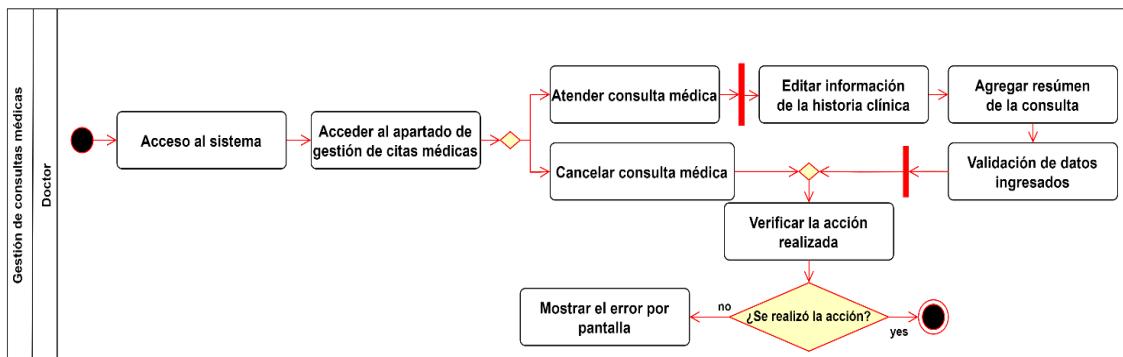


Diagrama 3.12 Diagrama de Actividades - Gestión de Consultas Médicas

En la tabla 3.14 se describe el caso de uso de gestión de consultas médicas.

Tabla 3.14 Especificación – Gestión de Consultas Médicas

| Nombre | Gestión de Consultas Médicas |
|--------------------|---|
| Descripción | Describe el proceso de atender una consulta médica a un paciente. |

| | |
|----------------------------------|--|
| Actores | Doctor. |
| Precondición | El doctor tiene acceso al sistema y a citas médicas asignadas. |
| Post Condiciones | Se llevan a cabo consultas médicas programadas y se actualiza la información de la historia clínica. |
| Flujo normal | <ul style="list-style-type: none"> • El doctor puede agregar resúmenes a las consultas dentro de la historia clínica de un paciente. • Durante la consulta, el doctor edita la información de la historia clínica. |
| Flujo alternativo | En caso de encontrar algún error, el sistema informara que error se ha encontrado |
| Requisitos no-funcionales | Seguridad en el acceso y restricciones claras para la modificación de datos sensibles. |

3.4.2.6 Gestión de exámenes de laboratorio

El diagrama 3.13 nos muestra el flujo de los procesos relacionados al caso de uso de gestión de exámenes de laboratorio.

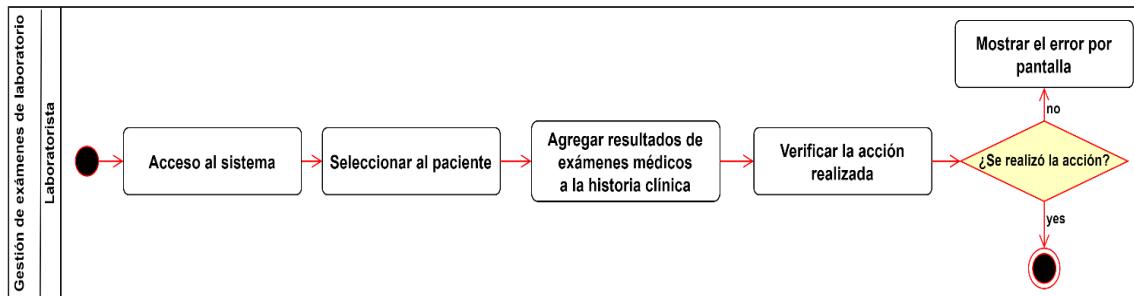


Diagrama 3.13 Diagrama de Actividades - Gestión de Exámenes de Laboratorio

En la tabla 3.15 se describe el caso de uso de gestión de exámenes de laboratorio.

Tabla 3.15 Especificación – Gestión de Exámenes de Laboratorio

| Nombre | Gestión de Exámenes de Laboratorio |
|----------------------------------|--|
| Descripción | Describe el proceso de agregar un examen de laboratorio a la historia clínica de un paciente. |
| Actores | Laboratorista. |
| Precondición | El laboratorista tiene acceso al sistema y la historia clínica del paciente. |
| Post Condiciones | Se añade información de exámenes médicos a la historia clínica del paciente. |
| Flujo normal | <ul style="list-style-type: none"> • El laboratorista selecciona al paciente y agrega los resultados de los exámenes médicos a su historial. • Se garantiza la precisión y confidencialidad de los datos ingresados. |
| Flujo alternativo | En caso de encontrar algún error, el sistema informará que error se ha encontrado |
| Requisitos no-funcionales | Acceso limitado a la información del paciente y resguardo de datos sensibles. |

3.4.2.7 Gestión de historias clínicas

El diagrama 3.14 nos muestra el flujo de los procesos relacionados al caso de uso de gestión de historias clínicas.

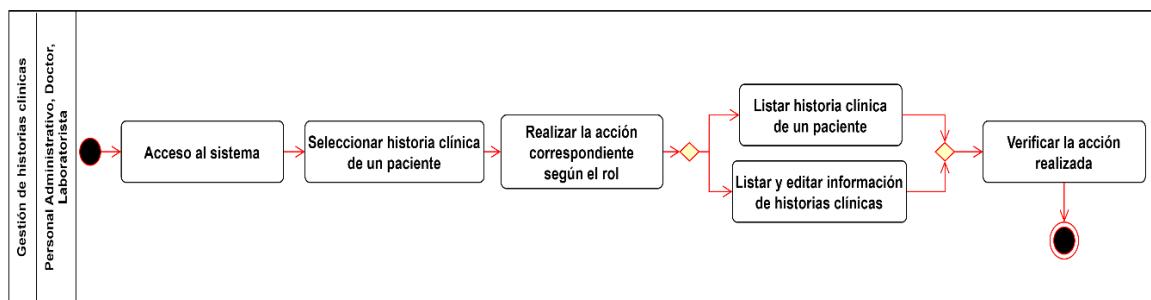


Diagrama 3.14 Diagrama de Actividades - Gestión de Historias Clínicas

En la tabla 3.16 se describe el caso de uso de gestión de historias clínicas.

Tabla 3.16 Especificación – Gestión de Historias Clínicas

| Nombre | Gestión de Historias Clínicas |
|---------------------------------------|--|
| Descripción | Describe el proceso de gestionar la información de una historia clínica, limitando las acciones según el rol de cada actor. |
| Actores | Personal Administrativo, Doctor y Laboratorista |
| Precondición | El personal administrativo, el laboratorista y el doctor tienen acceso al sistema. |
| Post Condiciones | Se accede y edita la información de las historias clínicas dentro de los límites de cada rol. |
| Flujo normal | <ul style="list-style-type: none">• El personal administrativo y el laboratorista listan la información de las historias clínicas.• El doctor puede listar y editar la información de las historias clínicas.• Se asegura la integridad y privacidad de los datos al acceder y modificar la información. |
| Flujo alternativo | En caso de encontrar algún error, el sistema informará que error se ha encontrado |
| Requisitos no- funcionales | Acceso limitado a la información del paciente y resguardo de datos sensibles. |

3.4.2.8 Generar reporte de historia clínica

El diagrama 3.15 nos muestra el flujo de los procesos relacionados al caso de uso de generar reporte de historia clínica.

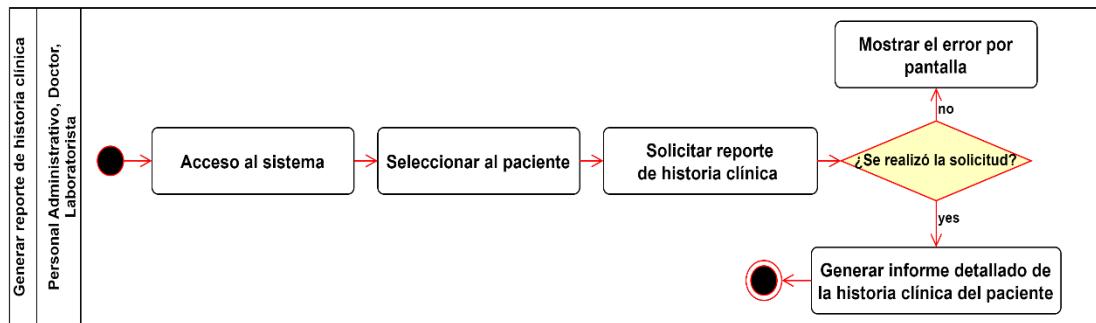


Diagrama 3.15 Diagrama de Actividades - Generar Reporte de Historia Clínica

En la tabla 3.17 se describe el caso de uso de generar reporte de historia clínica.

Tabla 3.17 Especificación – Generar Reporte de Historia Clínica

| Nombre | Generar Reporte de Historia Clínica |
|----------------------------------|--|
| Descripción | Describe el proceso de generar un reporte de la historia clínica de un paciente. |
| Actores | Personal Administrativo, Doctor y Laboratorista |
| Precondición | El personal administrativo, el laboratorista y el doctor tienen acceso al sistema y a la información de un paciente. |
| Post Condiciones | Se genera un informe detallado de la historia clínica del paciente. |
| Flujo normal | <ul style="list-style-type: none"> Los usuarios seleccionan al paciente y solicitan el reporte de su historia clínica. Se genera un informe detallado que contiene la información relevante de la historia del paciente. |
| Flujo alternativo | En caso de encontrar algún error, el sistema informará qué error se ha encontrado |
| Requisitos no-funcionales | Acceso limitado a la información y generación de reportes con datos completos y actualizados. |

3.4.3 Asignación de Casos de Uso a Módulos del Sistema

En este apartado, nos enfocamos en la distribución estructurada de los casos de uso dentro de los módulos específicos del sistema. Cada módulo en el sistema ha sido diseñado para encapsular un conjunto de funcionalidades relacionadas a los casos de uso ya descritos, facilitando así la modularidad y mantenibilidad del software. La asignación precisa de casos de uso a módulos específicos es crucial para garantizar que todas las funcionalidades del sistema estén bien organizadas y que cada componente se responsabilice de sus operaciones designadas.

La siguiente tabla detalla la asignación de cada caso de uso al módulo correspondiente, proporcionando una visión clara de cómo se estructura y se distribuyen las responsabilidades dentro del sistema.

Tabla 3.18 Tabla de asignación de casos de uso a módulos del sistema

| Casos de uso | Módulos del Sistema | Actor |
|----------------------------------|---|--|
| • Acceso al sistema | Módulo de autenticación y de autorización | <ul style="list-style-type: none">• Administrador• Personal Administrativo• Doctor• Laboratorista |
| • Administrar cuentas de usuario | Módulo de registro de usuarios | <ul style="list-style-type: none">• Administrador |
| • Gestión de pacientes | Módulo de registro de pacientes. | <ul style="list-style-type: none">• Personal Administrativo |

| | | |
|---|-----------------------------|--|
| <ul style="list-style-type: none"> • Gestión de citas médicas | Módulo de registro de citas | <ul style="list-style-type: none"> • Personal Administrativo |
| <ul style="list-style-type: none"> • Gestión de consultas médicas | Módulo de consulta | <ul style="list-style-type: none"> • Doctor |
| <ul style="list-style-type: none"> • Gestión de historias clínicas • Gestión de exámenes de laboratorio | Módulo de historia clínica | <ul style="list-style-type: none"> • Personal Administrativo • Doctor • Laboratorista |
| <ul style="list-style-type: none"> • Generar reporte de historia clínica | Módulo de reporte | <ul style="list-style-type: none"> • Personal Administrativo • Doctor • Laboratorista |

3.4.4 Modelo de Contenido

El modelo de contenido se puede derivar en clases conceptuales que representan el dominio del problema, pero esta fase es más específica y se relaciona más directamente con los diagramas de clases de análisis y diseño.

3.4.4.1 Diagrama de Clase de Análisis de Dominio

El diagrama de clases de análisis de dominio representa las clases esenciales del dominio del problema, las cuales se basan en los casos de uso del negocio.

El diagrama 3.16 nos muestra el diagrama de clases de análisis de dominio, el cual se basa en los casos de uso del negocio.

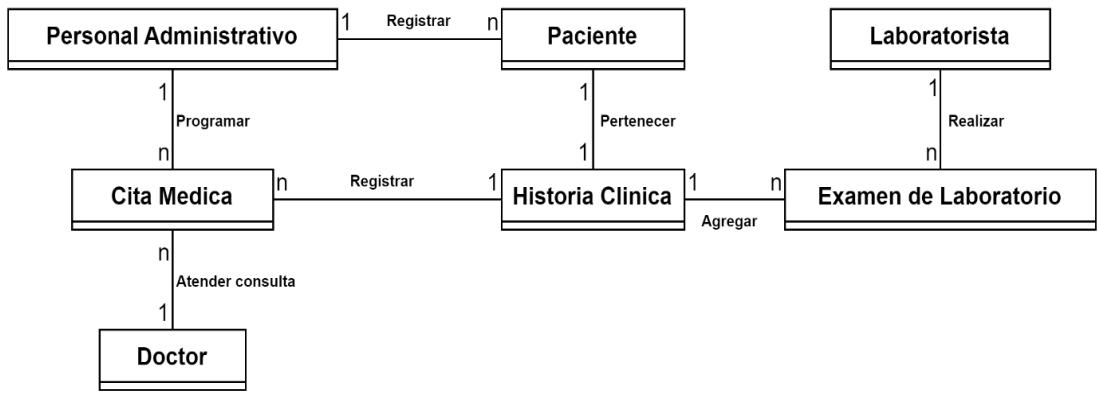


Diagrama 3.16 Diagrama de Clase de Análisis de Dominio

3.4.4.2 Modelo de Datos

El diseño del modelo de datos es fundamental para la estructuración de la información en un sistema. Este modelo describe la organización y relaciones entre los datos de manera lógica y física, anticipando la persistencia de la información en una base de datos relacional.

- **Diagrama de entidad-relación**

Es una representación gráfica utilizada en el diseño de bases de datos. En él, se identifican entidades (objetos o conceptos del mundo real, como personas, lugares o cosas), relaciones entre estas entidades y atributos que describen las características de las entidades. El diagrama 3.17 nos muestra el diseño de la base de datos.

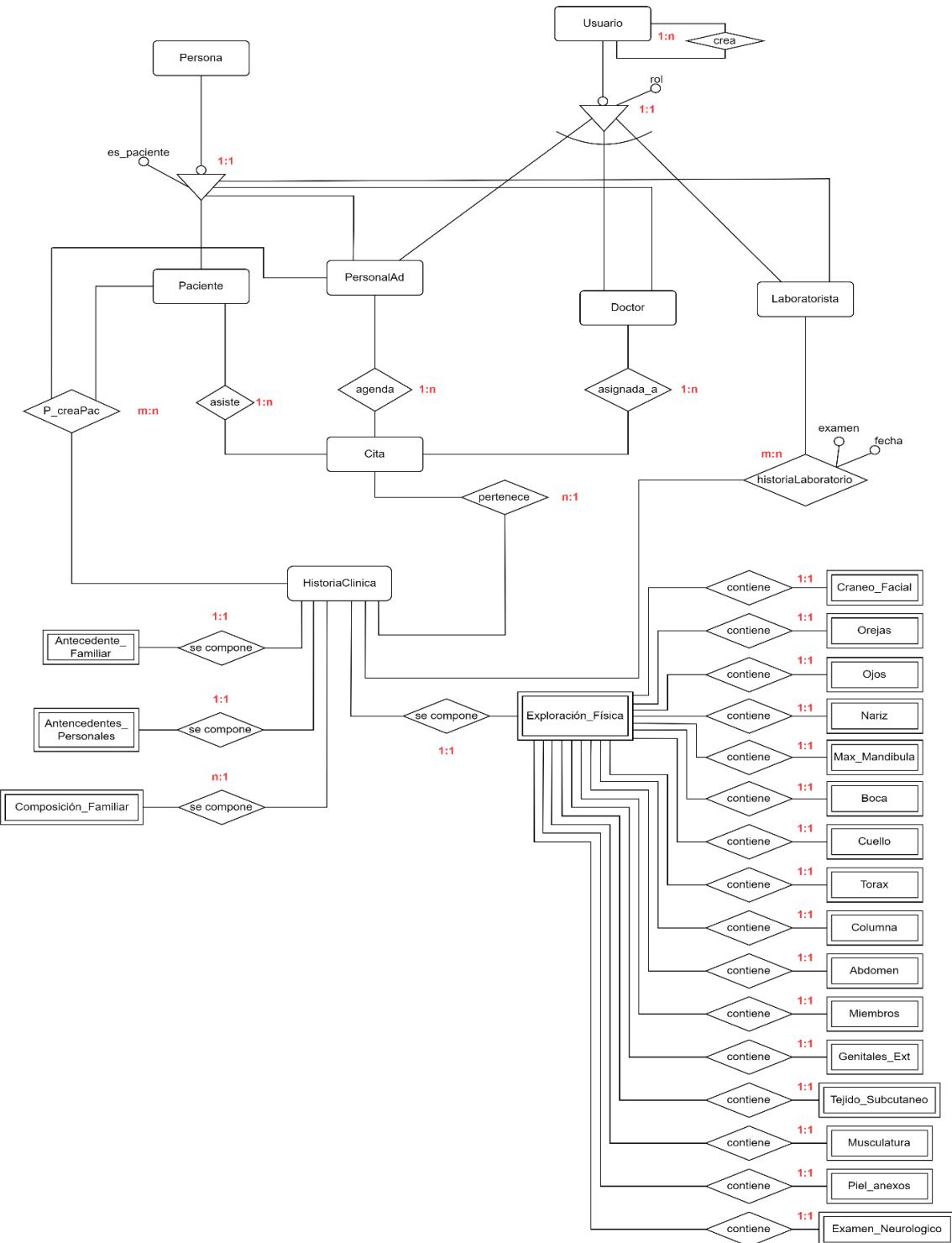


Diagrama 3.17 Diagrama Entidad Relación

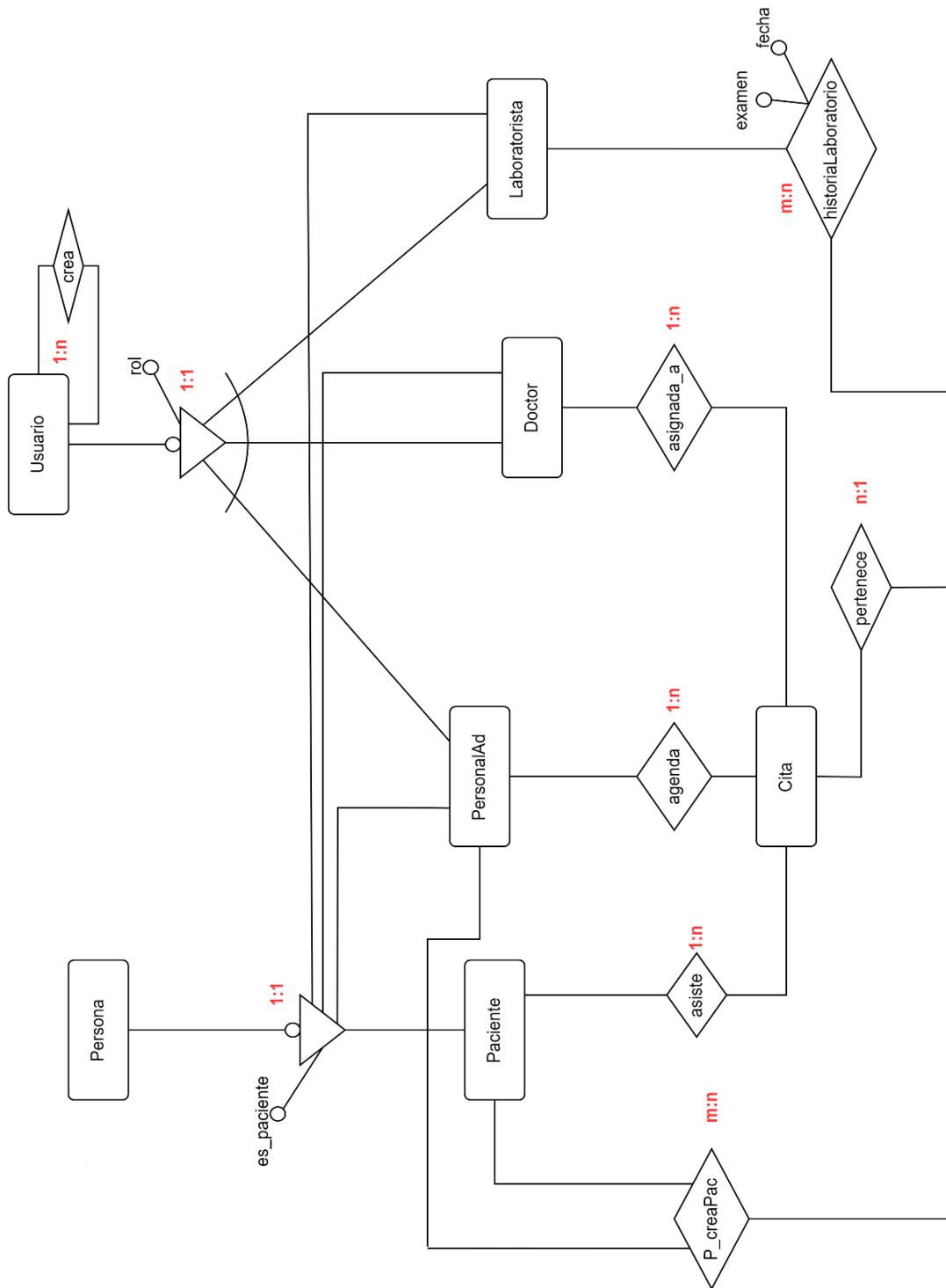


Diagrama 3.18 Diagrama Entidad Relación - Parte 1

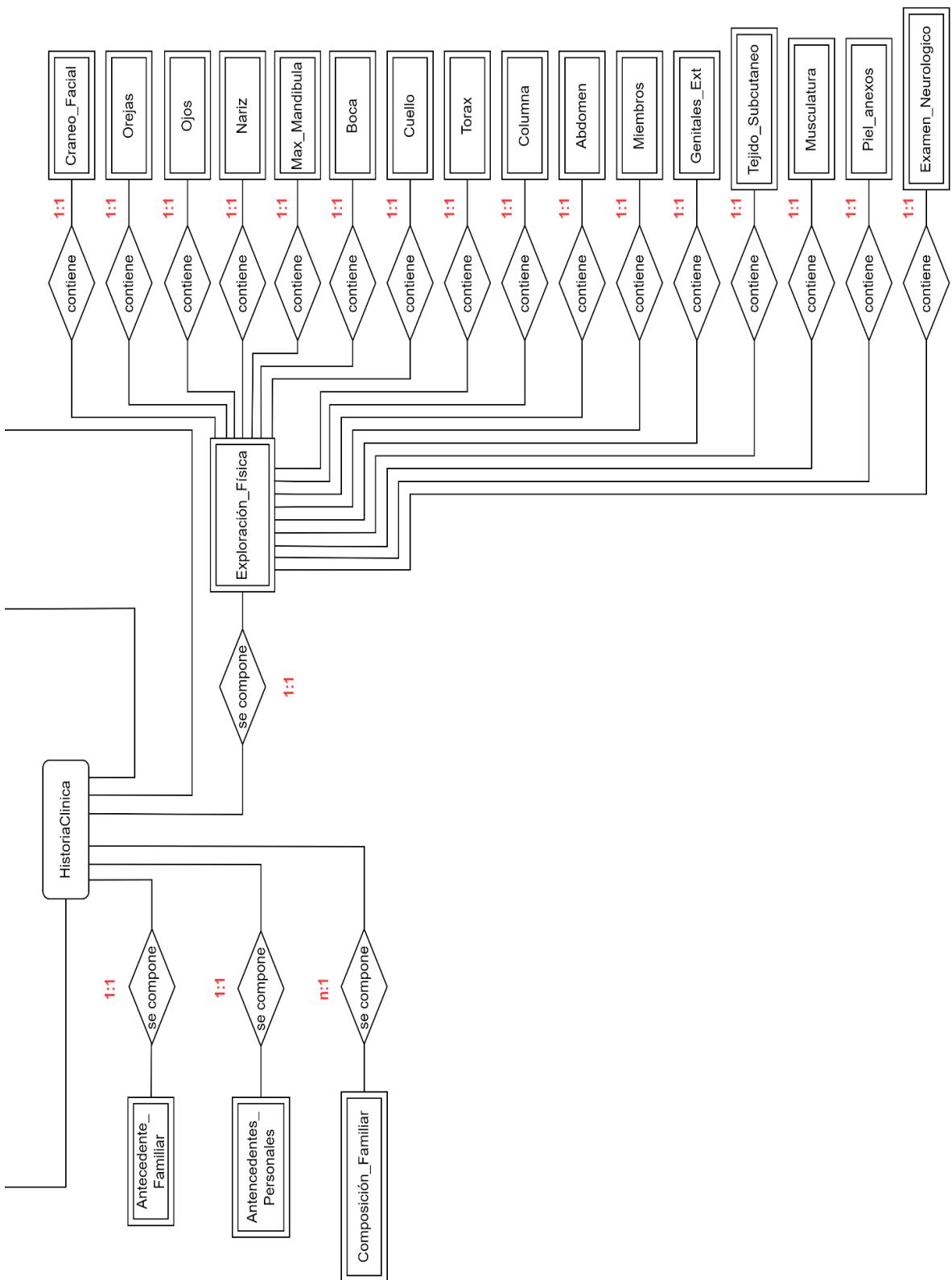


Diagrama 3.19 Diagrama Entidad Relación - Parte 2

- **Diagrama de Clase de Diseño del Sistema**

El Diagrama de Clase de Diseño del Sistema es una representación gráfica que muestra la estructura de clases del sistema, derivada de los casos de uso y el diagrama entidad-relación. Este diagrama ilustra cómo se implementan las clases para cumplir con los requisitos del sistema, detallando las relaciones, atributos y métodos de cada clase.

El diagrama 3.20 presenta esta estructura de clases, siendo una base fundamental que guía la implementación del sistema, originada en la estructura conceptual y los requerimientos identificados en el diagrama entidad-relación.

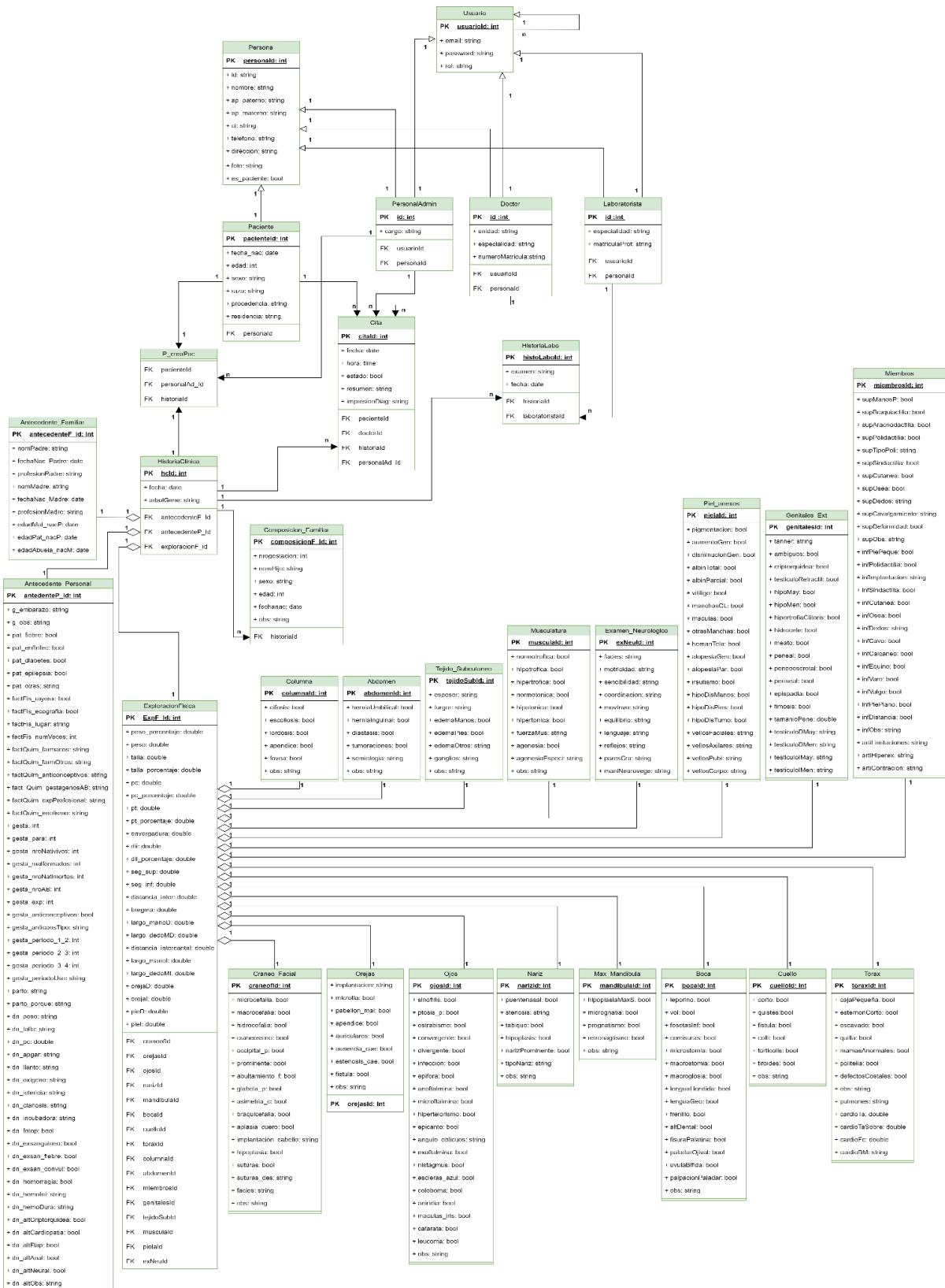


Diagrama 3.20 Diagrama de Clase de Diseño del Sistema

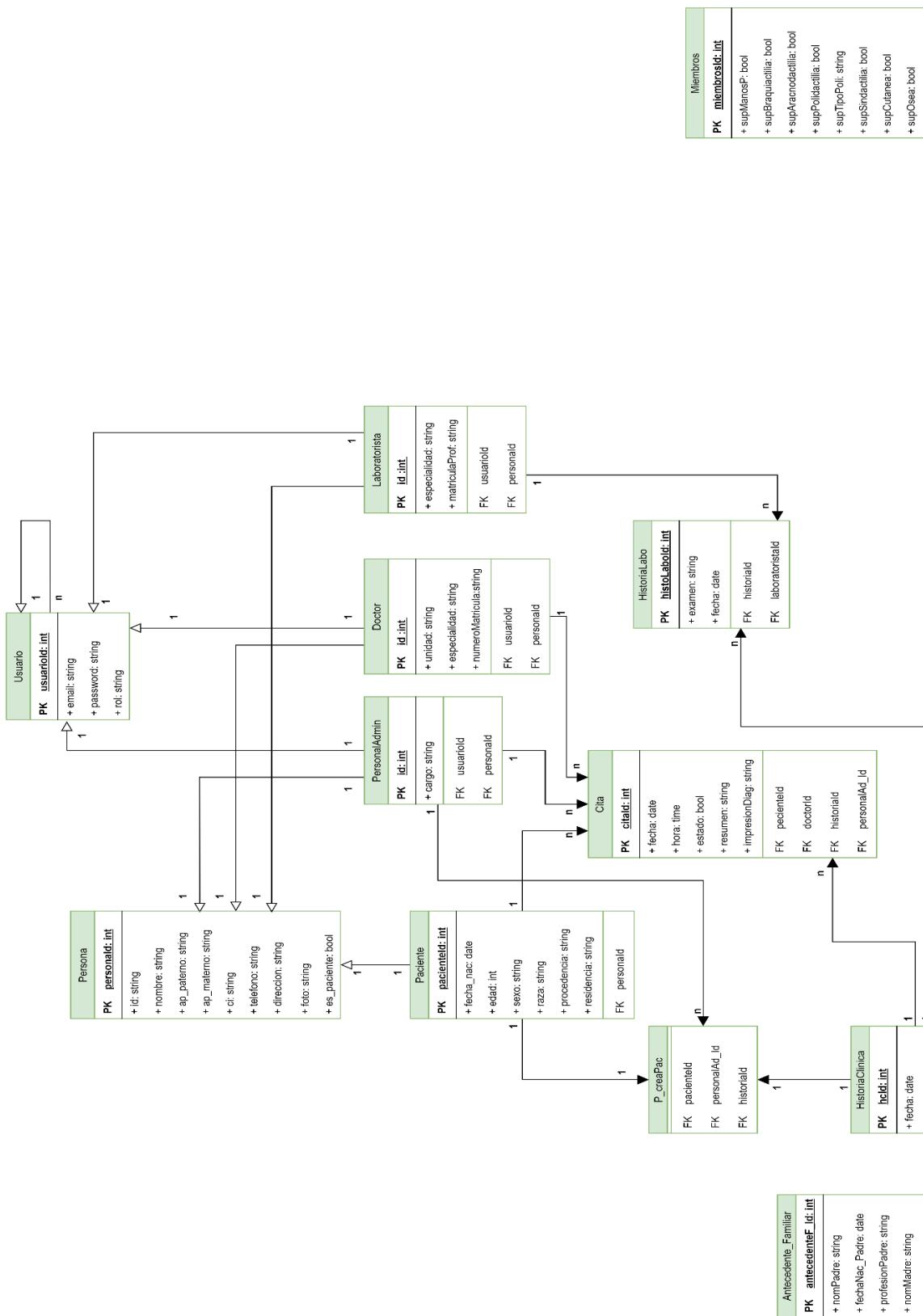


Diagrama 3.21 Diagrama de Clase de Diseño del Sistema - Parte 1

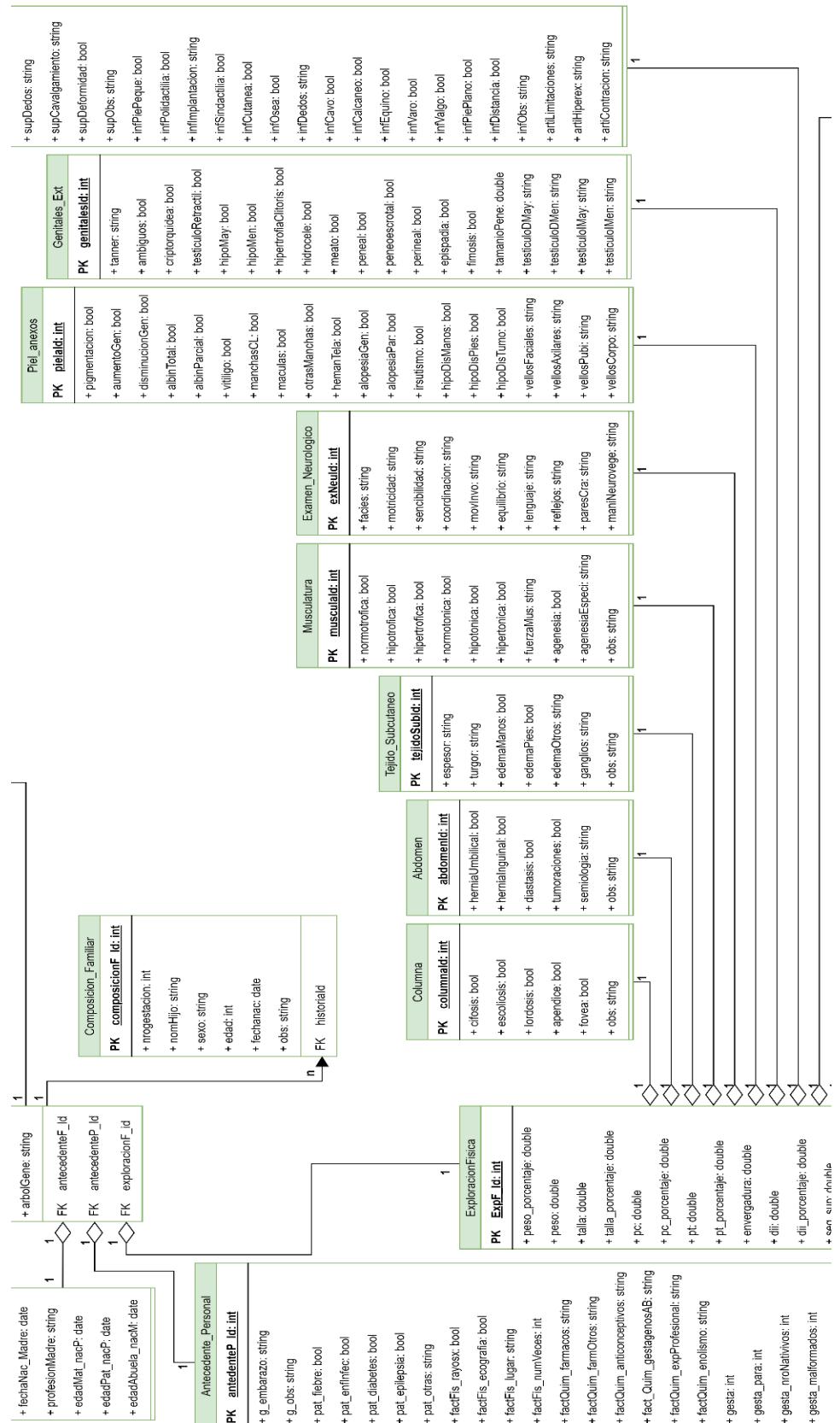
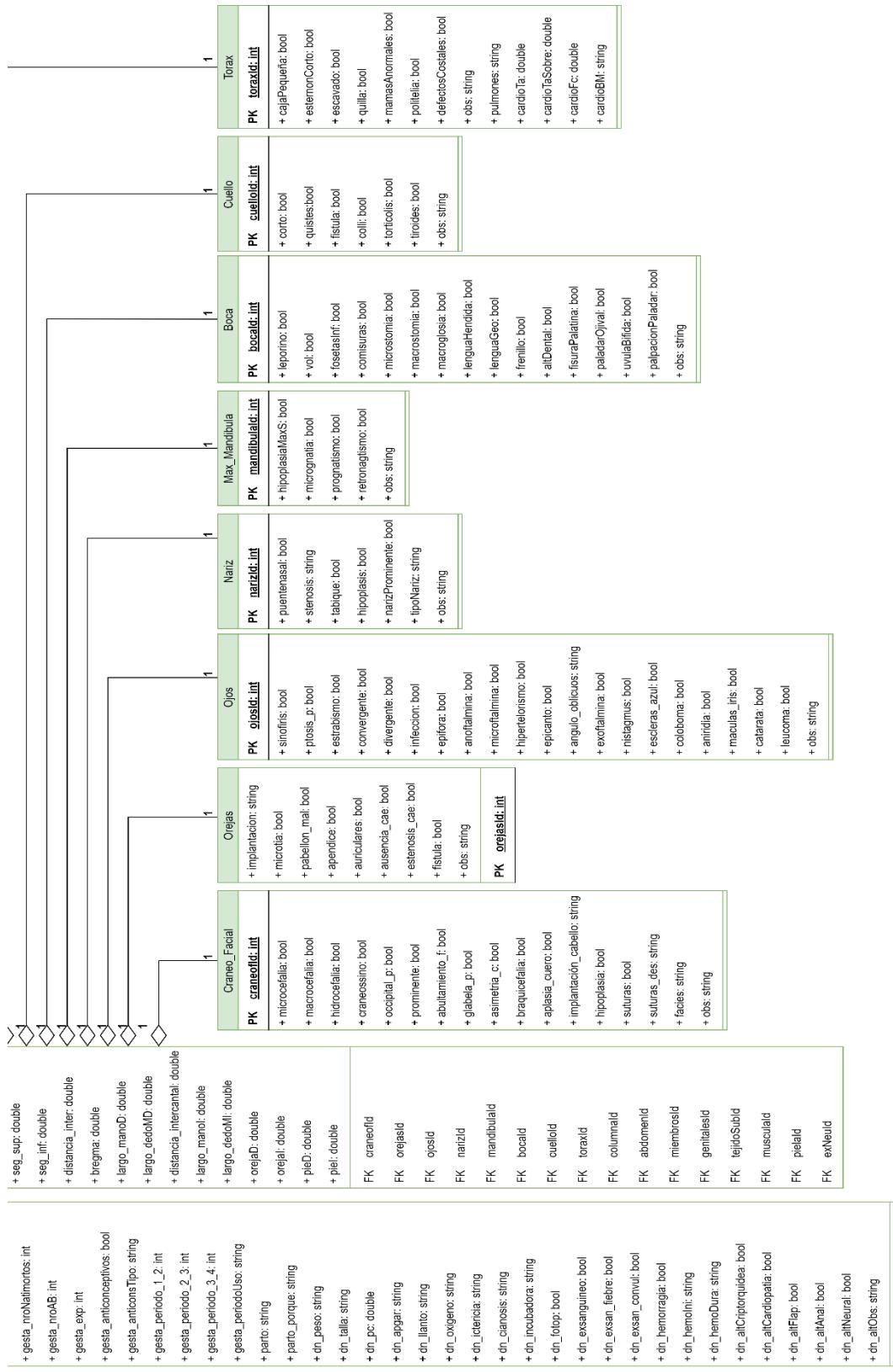


Diagrama 3.22 Diagrama de Clase de Diseño del Sistema - Parte 2



3.4.5 Modelo de Navegación

Este modelo describe la manera en que los usuarios interactuarán con el sistema según el rol asignado.

Existen cuatro formas de navegar: como administrador, como personal administrativo, como doctor y como laboratorista.

Dependiendo del rol asignado, podrán acceder a distintos espacios y opciones de navegación. En todos los casos, es necesario que el usuario esté autenticado para realizar las tareas específicas asignadas a cada rol dentro del sistema.

En el diagrama 3.24 representa el esquema de navegación del sistema web, mostrando sus diversas opciones de desplazamiento.

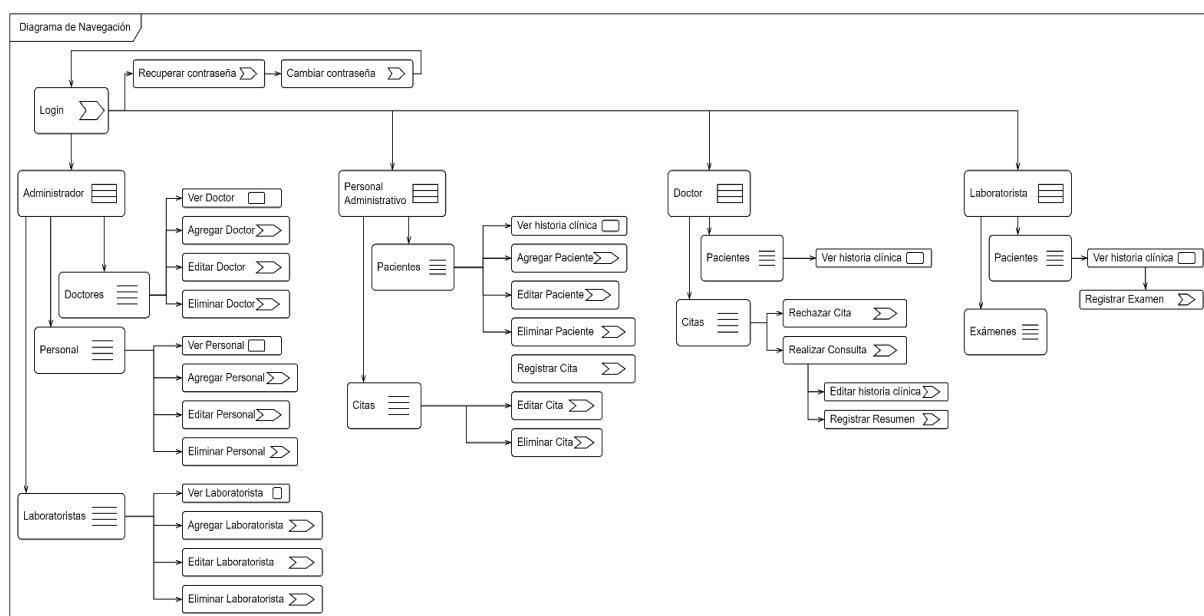


Diagrama 3.24 Diagrama de Navegación

3.4.6 Modelo de Presentación

El modelo de presentación describe la manera en que se presentarán y mostrarán las interfaces de usuario. Este modelo se emplea para diseñar y estructurar la apariencia visual de las pantallas finales del sistema.

Este modelo proporciona una visión general de cómo serán las interfaces que interactuarán con los usuarios en el sistema.

El diagrama 3.25 nos muestra los componentes de presentación presente en el sistema.

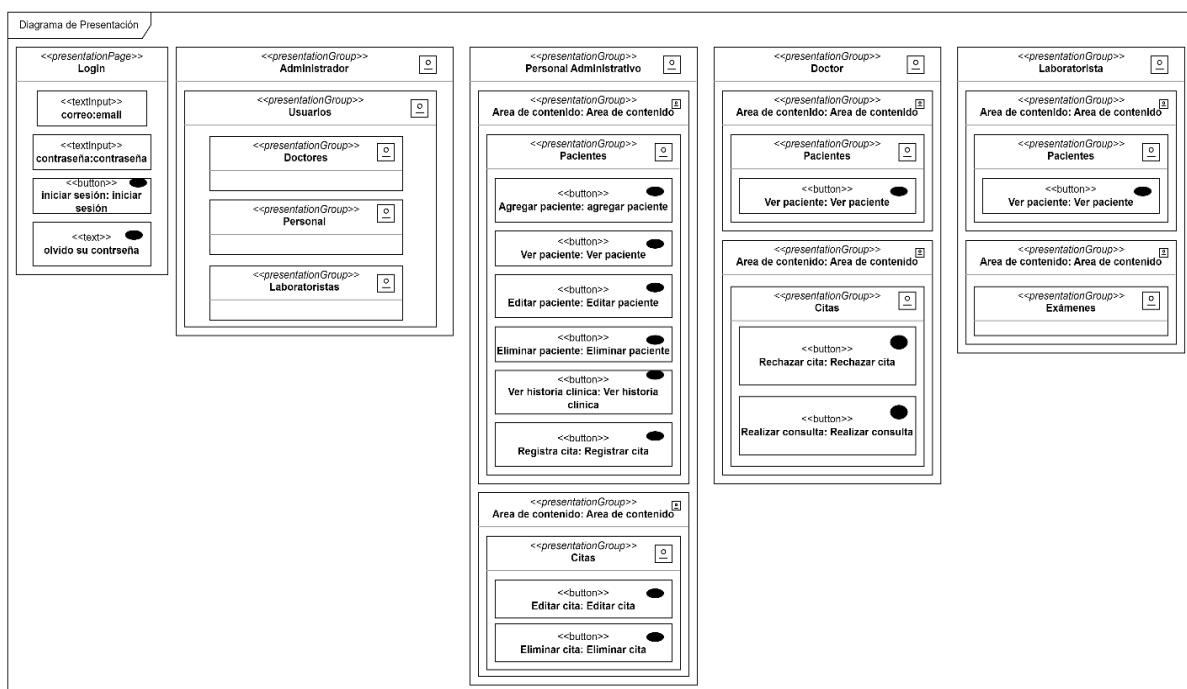


Diagrama 3.25 Diagrama de Presentación

3.4.7 Modelo del flujo de proceso

El modelo de flujo de procesos detalla las distintas operaciones que el sistema llevará a cabo en interacción con el usuario. Este diagrama, desglosa y representa visualmente cada uno de los procesos, brindando una visión clara y detallada de las interacciones entre el sistema y el usuario.

Los siguientes diagramas nos muestran el flujo de los procesos dentro del sistema.

a) Acceso al Sistema

En este proceso se debe ingresar el correo y la contraseña, si es que existe el usuario se dará un acceso al sistema, y si no existe se mandará un mensaje de error y se volverá a intentar la autenticación del usuario. En el diagrama 3.26 tenemos al flujo del proceso de autenticación.

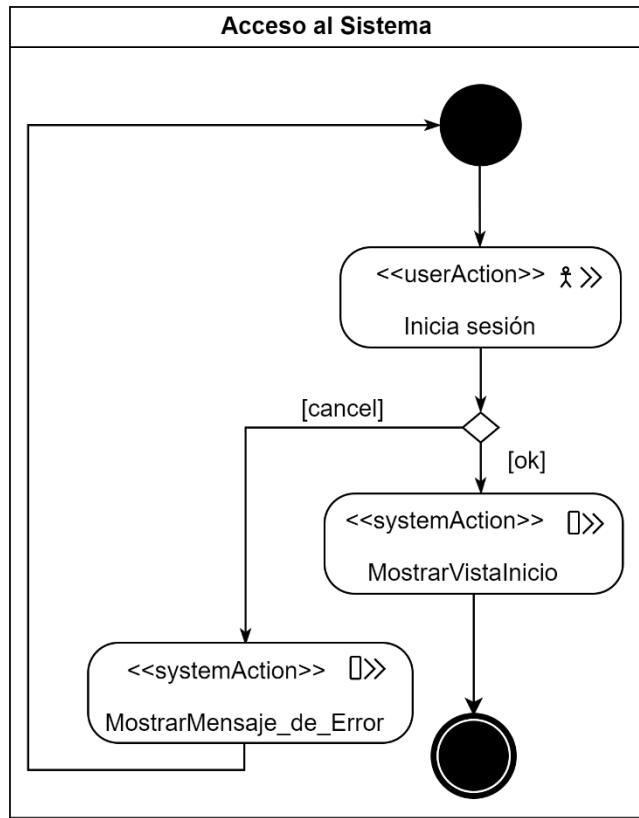


Diagrama 3.26 Flujo del Proceso de Acceso al Sistema

b) Gestión de Usuarios

En este proceso el administrador gestiona a los usuarios dentro del sistema según el rol asignado, este puede ser de tipo doctor, personal administrativo y laboratorista. En el diagrama 3.27 tenemos al flujo del proceso de gestión de usuarios.

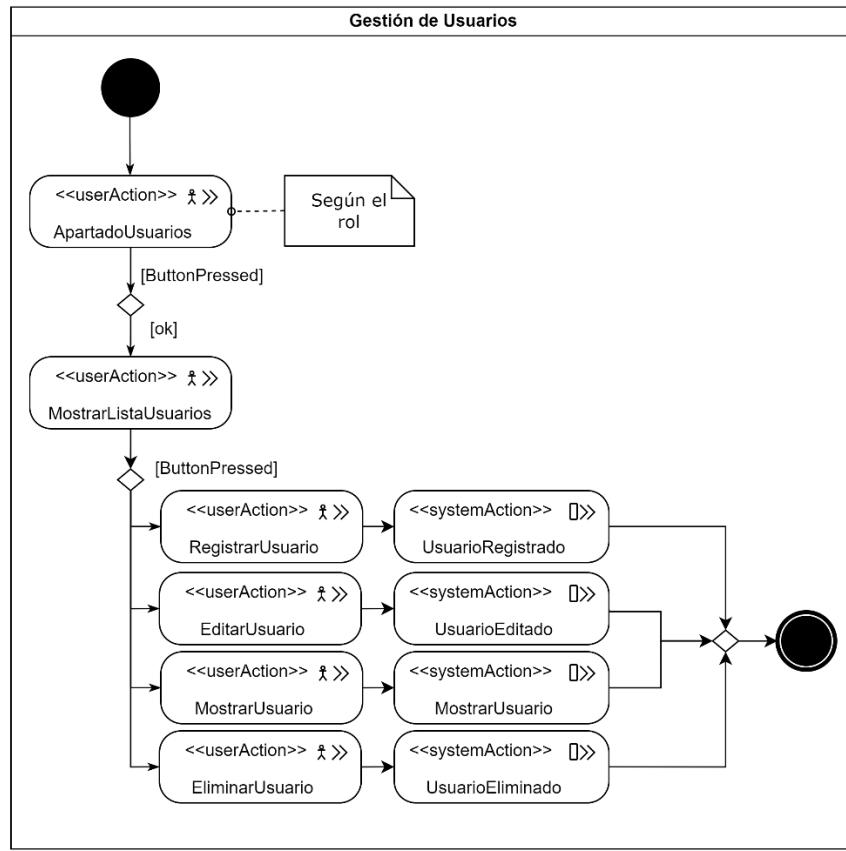


Diagrama 3.27 Flujo del Proceso de Gestión de Usuarios

c) Gestión de Pacientes

En este proceso el personal administrativo gestiona a los pacientes dentro del sistema. En el diagrama 3.28 tenemos al flujo del proceso de gestión de pacientes.

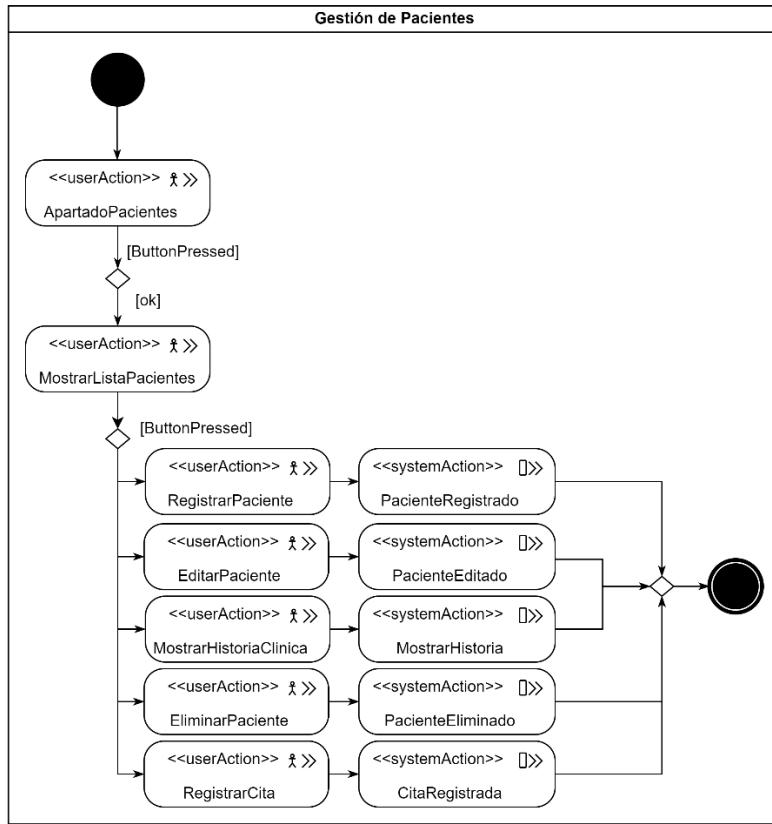


Diagrama 3.28 Flujo del Proceso de Gestión de Pacientes

d) Gestión de Citas Médicas

En este proceso el personal administrativo gestiona a las citas médicas dentro del sistema. En el diagrama 3.29 tenemos al flujo del proceso de gestión de citas médicas.

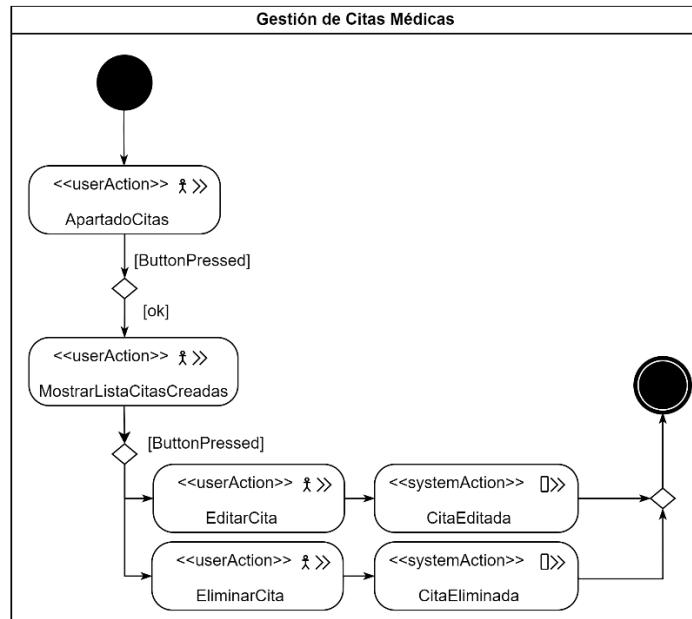


Diagrama 3.29 Flujo del Proceso de Gestión de Citas Médicas

e) Gestión de Consultas Médicas

En este proceso el doctor atiende a las citas que el personal administrativo agendó, el proceso de atender a las citas se le denomina consulta, es por eso que el proceso tiene el nombre gestión de consultas médicas. En el diagrama 3.30 tenemos al flujo del proceso de gestión de consultas médicas.

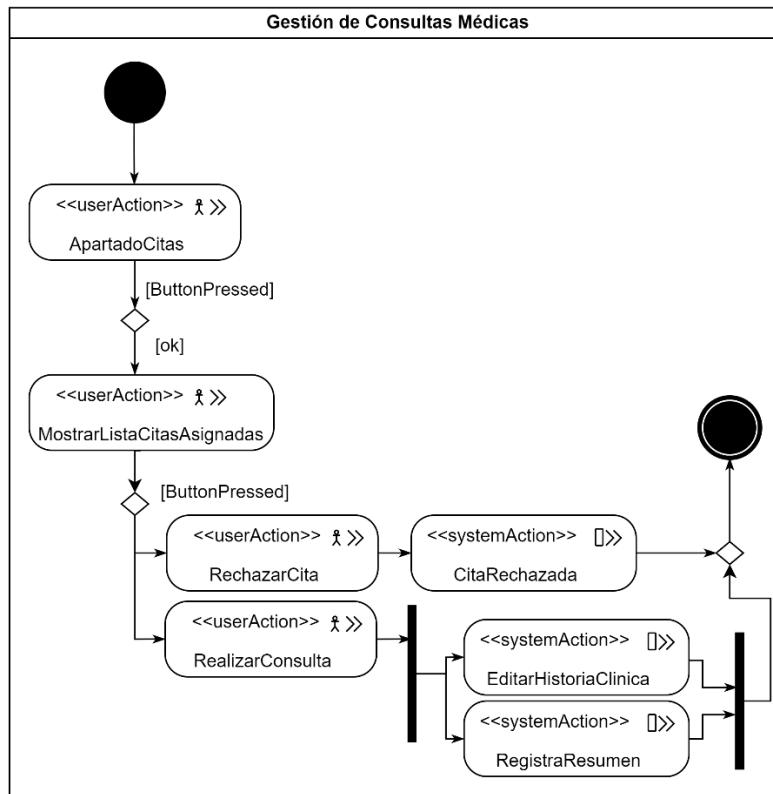


Diagrama 3.30 Flujo del Proceso de Gestión de Consultas Médicas

f) Gestión de Exámenes de Laboratorio

En este proceso el laboratorista gestiona a los exámenes de laboratorio que se registran o se editan dentro del sistema. En el diagrama 3.31 tenemos al flujo del proceso de gestión de exámenes de laboratorio.

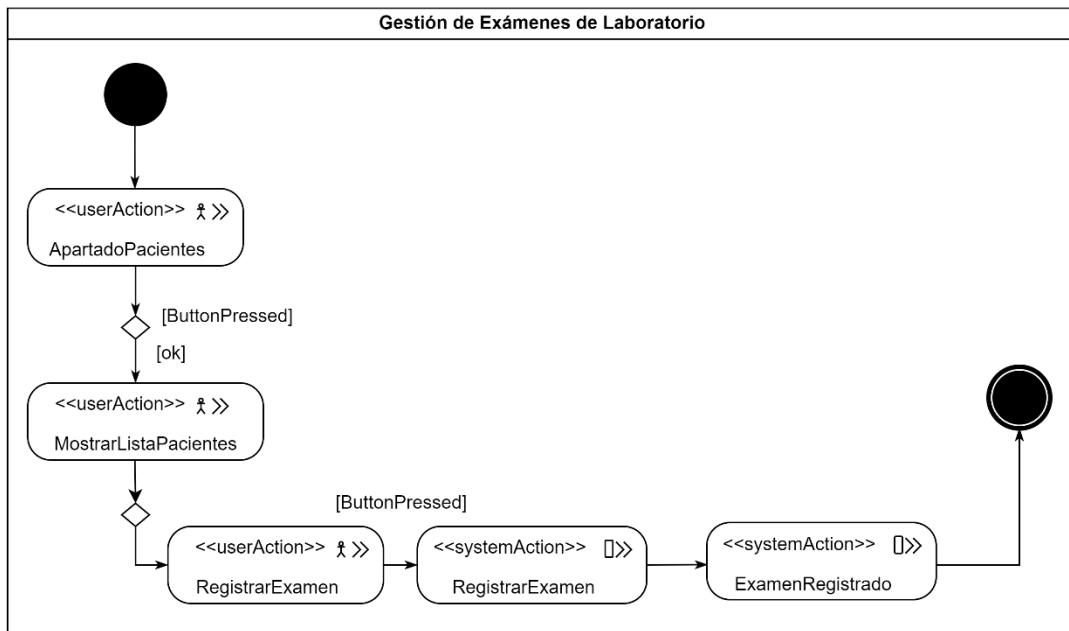


Diagrama 3.31 Flujo del Proceso de Gestión de Exámenes de Laboratorio

g) Gestión de Historias Clínicas

En este proceso, el nivel de acceso a las historias clínicas está determinado por el rol del usuario. El doctor posee el mayor nivel de acceso, permitiéndole ver y editar las historias clínicas. Por otro lado, los usuarios con roles de personal administrativo y laboratorista tienen permisos limitados, solo pueden visualizar las historias clínicas. El diagrama 3.32 muestra el flujo del proceso de gestión de historias clínicas.

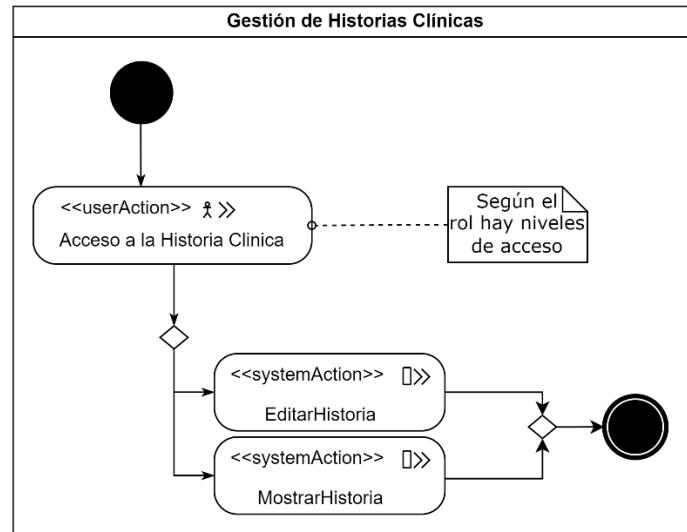


Diagrama 3.32 Flujo del Proceso de Gestión de Historias Clínicas

f) Generar reporte de historia clínica

En este proceso los actores del sistema pueden generar un reporte de la historia clínica de un paciente.



Diagrama 3.33 Flujo del Proceso Generar Reporte de Historia Clínica

3.5 Fase de Transición

Durante esta fase, se desarrolla el modelo de despliegue, que detalla la configuración y operación del sistema. También se examinan el proceder del sistema, sus

vistas y funcionamiento. Finalmente, se efectúan pruebas con la herramienta Playwright para asegurar la funcionalidad del sistema.

3.5.1 Modelo de Despliegue

El modelo de despliegue describe la conexión entre las máquinas y el servidor en el entorno de producción, destacando la contribución de cada elemento al funcionamiento y uso del sistema. El modelo se centra en las relaciones de red y la topología usada para la ejecución del sistema.

En la Diagrama 3.34 se presenta el modelo de despliegue, ofreciendo una representación visual de la interconexión entre las máquinas y el servidor, lo que mejora la comprensión de la estructura de red implementada.

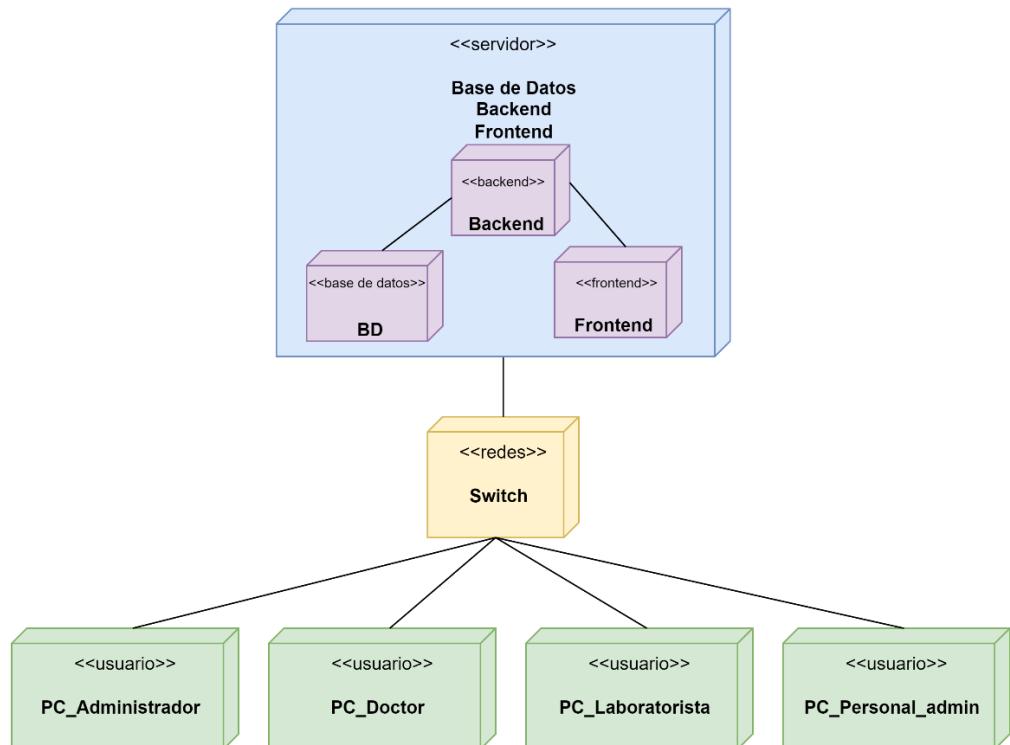


Diagrama 3.34 Modelo de Despliegue

3.5.2 Interfaces de usuario

En esta sección se proporcionará una visión del sistema mediante las interfaces de usuario. Estas interfaces proporcionarán una representación visual de las diversas funcionalidades y características disponibles para los usuarios.

a) Acceso al sistema

El acceso al sistema requiere que el usuario inicie sesión utilizando el correo electrónico y la contraseña asignados previamente. Es necesario completar este paso para acceder a las funcionalidades del sistema.



Figura 3.2 Inicio de Sesión

b) Gestión de usuarios

El administrador dispone de los procesos mencionados previamente, como la gestión de usuarios. Al seleccionar una opción según el rol del usuario, se despliega una lista de usuarios registrados en el sistema. En este apartado, el administrador puede ejecutar las acciones descritas previamente y que el sistema llevará a cabo.

| COD | APELLIDO PATERNO. | APELLIDO MATERNO. | NOMBRE | CI | ESPECIALIDAD | CORREO ELECTRONICO | CREADO EN | OPCIONES |
|-----|-------------------|-------------------|---------------|----------|--------------------|--------------------------|--------------------------|----------|
| 1 | Holder | Gilliam | Jeri Bradford | 12345678 | laboratorio medico | laboratorista1@email.com | 2023-09-26T22:31:51.388Z | |
| 2 | Holder | Groneros | Malu Pelucas | 12345687 | laboratorio medico | laboratorista2@email.com | 2023-11-17T16:22:56.565Z | |

Figura 3.3 Gestión de Usuarios

c) Gestión de pacientes

El personal administrativo dispone de los procesos de gestión de pacientes y citas médicas. En la sección de pacientes, puede llevar a cabo las acciones mencionadas anteriormente, las cuales son realizadas por el sistema.

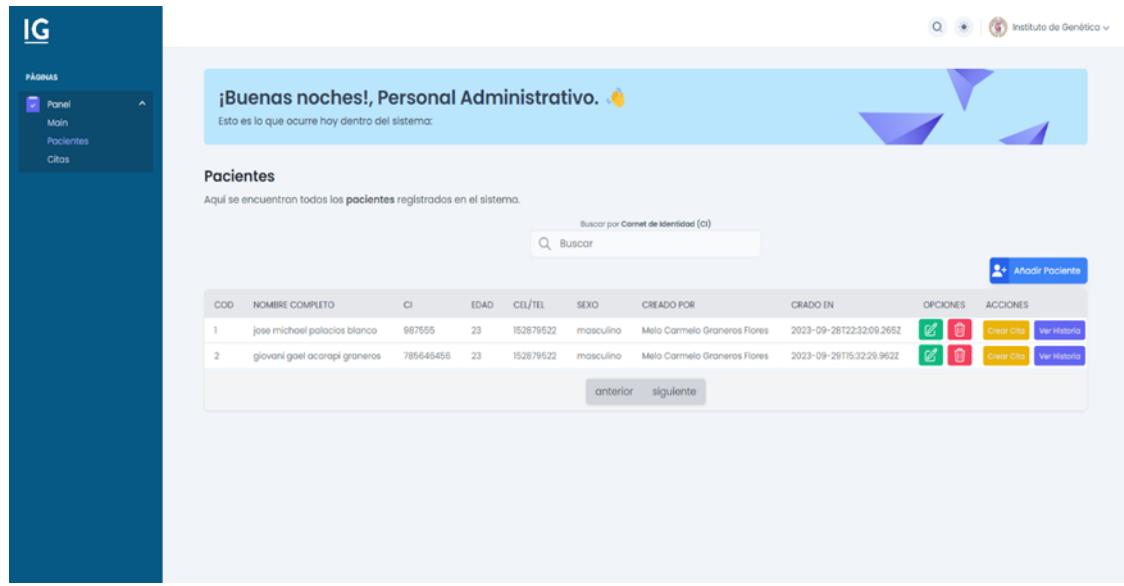


Figura 3.4 Gestión de Pacientes

d) Gestión de citas médicas

En la sección de citas, el personal administrativo podrá acceder a las citas que agendó, además de realizar las acciones descritas anteriormente.

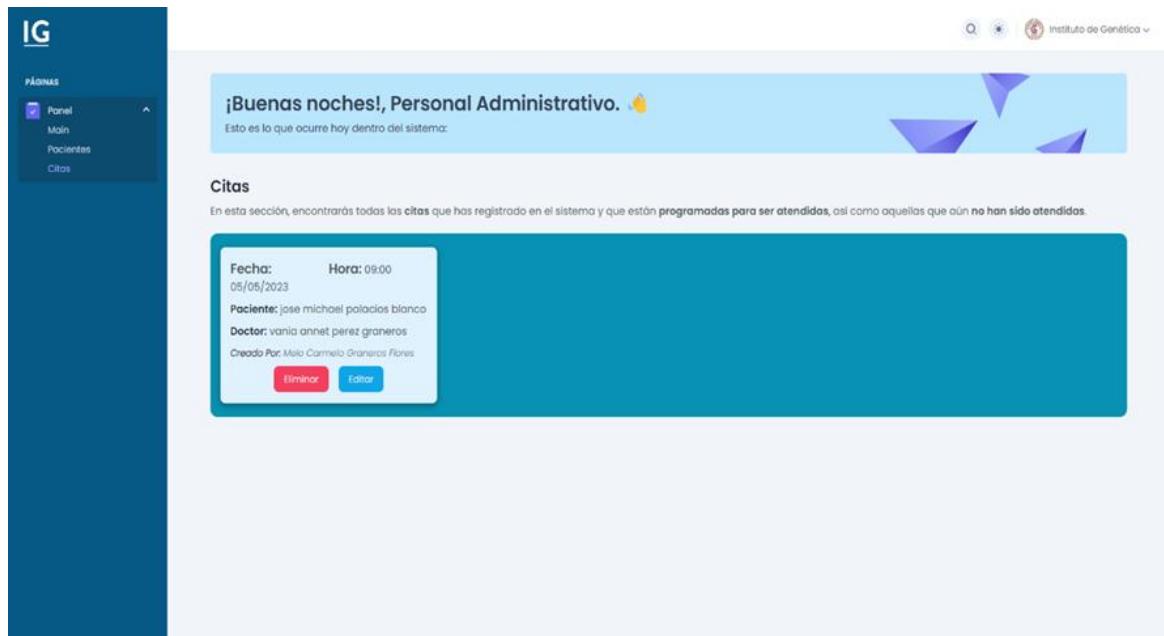


Figura 3.5 Gestión de Citas

e) Gestión de consultas médicas

El doctor tiene acceso a los procesos de gestión de consultas médicas y de historias clínicas, con un nivel de acceso más elevado en comparación con otros actores dentro del sistema, en la sección pacientes este podrá visualizar la historia clínica de un paciente.

Figura 3.6 Lista de Pacientes

En la sección de citas, el doctor tendrá la opción de realizar una consulta o rechazar una cita previamente creada.

Figura 3.7 Gestión de Consultas Médicas

Durante una consulta médica, el sistema guiará al usuario a través del flujo necesario para llevar a cabo la consulta. Este flujo asegura que se sigan todos los pasos requeridos para completar la consulta con éxito. Al finalizar la consulta, los datos se registrarán automáticamente en la historia clínica del paciente correspondiente.

Figura 3.8 Gestión de Consultas Médicas - Consulta

f) Gestión de exámenes de laboratorio

El laboratorista cuenta con acceso a los procesos de gestión de exámenes de laboratorio y de historias clínicas. En la sección de pacientes, puede visualizar las historias clínicas de los pacientes y agregar exámenes de laboratorio según sea necesario.

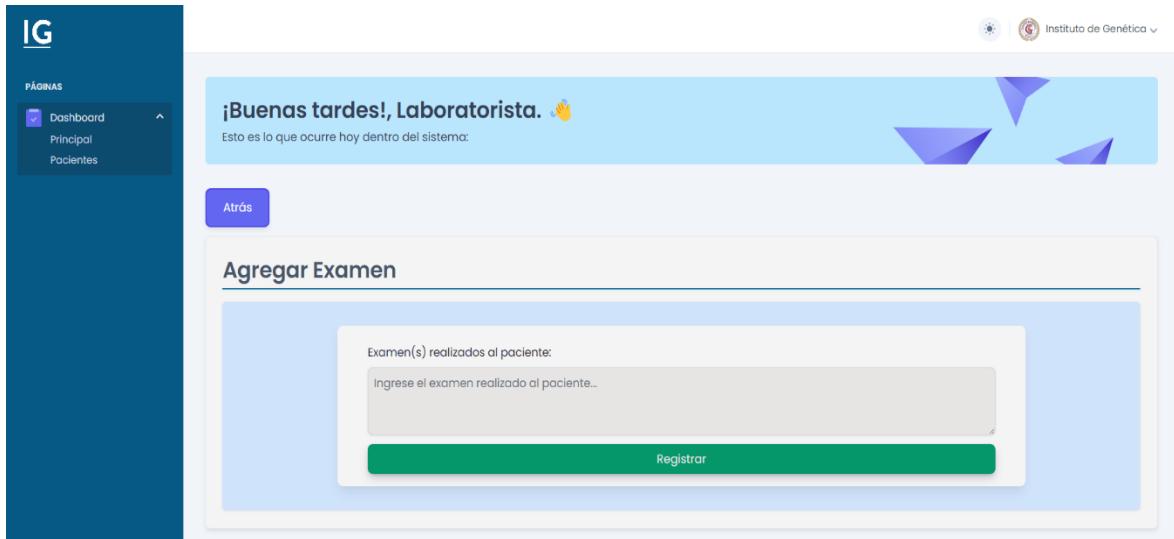


Figura 3.9 Gestión de Exámenes de Laboratorio

g) Gestión de Historias Clínicas

La historia clínica proporciona una visión completa de la información relacionada con un paciente, detallando los resultados de cada aspecto médico. Además, refleja los datos generados en otros procesos realizados por los actores del sistema, (doctor, laboratorista y personal administrativo), los cuales tienen acceso a este apartado.

Figura 3.10 Gestión de Historias Clínicas – Parte uno

Figura 3.11 Gestión de Historias Clínicas – Parte dos

h) Generar reporte de historia clínica

Cada actor del sistema tiene la opción de generar el reporte de la historia clínica de un paciente.

Figura 3.12 Generar reporte de historia clínica

3.5.3 Pruebas End-To-End con Playwright

En este apartado exploraremos el proceso de evaluación integral del sistema, creando escenarios de uso reales para garantizar su funcionamiento óptimo en un entorno de producción. Estas pruebas abarcan la interacción de todos los componentes del sistema, desde la entrada de datos hasta la salida final, asegurando así su integridad y eficacia. Para lo cual, nos ayudaremos de la herramienta Playwright.

(QALified, 2022) Las pruebas end-to-end se pueden definir simplemente como un procedimiento que se ejecuta para productos complejos. Este tipo de pruebas confirman que la aplicación funciona tal como se espera analizando todos sus componentes. Por ende, se simula la experiencia del usuario de principio a fin.

Estas pruebas se realizan al final del desarrollo del software para garantizar la integridad, funcionalidad y rendimiento del sistema en condiciones similares a las del entorno de producción, utilizando datos reales o simulados para validar su comportamiento.

Para validar la integración y funcionalidad de todos los componentes del sistema, se llevó a cabo una prueba end-to-end. Para cada caso de uso relacionado en cada módulo, se generó una prueba con la siguiente descripción: se realizaron 10 pruebas por cada caso, introduciendo en lote datos aleatorios ya validados. El objetivo era verificar la cantidad de registros almacenados y detectar cualquier error. Los resultados de cada caso se detallan en la siguiente tabla.

Tabla 3.19 Resultados Prueba end-to-end

| Casos de uso | Resultados |
|---------------------|------------------------------------|
| • Acceso al sistema | De 10 pruebas solo se falló en una |

| | |
|---------------------------------------|------------------------------------|
| • Administrar cuentas de usuario | De 10 pruebas solo se falló en dos |
| • Gestión de pacientes | De 10 pruebas solo se falló en dos |
| • Gestión de citas médicas | De 10 pruebas solo se falló en una |
| • Gestión de consultas médicas | De 10 pruebas solo se falló en una |
| • Gestión de historias clínicas | De 10 pruebas solo se falló en dos |
| • Gestión de exámenes de laboratorio | De 10 pruebas solo se falló en una |
| • Generar reporte de historia clínica | De 10 pruebas solo se falló en una |
| Total | 11 fallos |

Entonces, de un total de 80 pruebas (8 casos * 10 pruebas por caso), 11 pruebas fallaron.

Para calcular el porcentaje de pruebas que fallaron, podemos usar la fórmula:

$$(Porcentaje de pruebas fallidas) = (11 / 80) * 100\% \approx 13.75\%$$

Por lo tanto, aproximadamente el 13.75% de las pruebas fallaron.

Para determinar el porcentaje de pruebas que funcionaron correctamente, simplemente restamos el porcentaje de pruebas fallidas del 100%

$$\text{Porcentaje de pruebas funcionales} = 100\% - 13.75\% = 86.25\%$$

Entonces, aproximadamente el **86.25%** de las pruebas funcionaron correctamente.

Por lo tanto, el sistema mostró un alto grado de funcionamiento, con aproximadamente el 86.25% de las pruebas ejecutadas sin errores significativos.

Estos resultados destacan la eficacia y la integridad del sistema en escenarios operativos simulados, respaldando su capacidad para satisfacer los casos de uso y brindar una experiencia confiable a los usuarios finales.

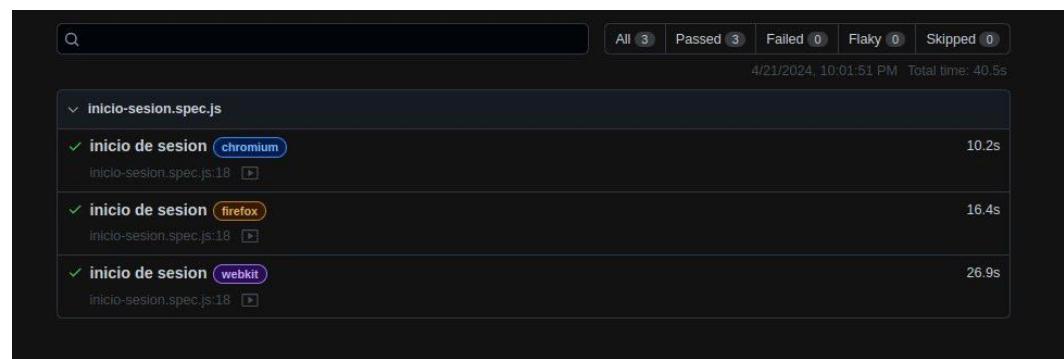


Figura 3.13 Prueba end-to-end

CAPÍTULO IV CALIDAD Y SEGURIDAD

4.1 Introducción

Este capítulo se enfoca en abordar los aspectos fundamentales de calidad y seguridad en el desarrollo de sistemas, específicamente utilizando como marco de referencia la norma ISO/IEC 25010. Se explorarán los criterios y requisitos esenciales para garantizar la calidad y la seguridad en los sistemas, proporcionando un análisis detallado de cómo esta norma se aplica en el proyecto.

4.2 Norma ISO/IEC 25010

(ISO 25010, s. f.) 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 subcaracterísticas.

(Britton, 2021) La norma ISO 25010 describe un modelo de calidad del producto compuesto por ocho características (que a su vez se subdividen en subcaracterísticas) relacionadas con las propiedades estáticas del software y las propiedades dinámicas del sistema informático.

Las características y subcaracterísticas proporcionan una terminología coherente para especificar, medir y evaluar la calidad de los sistemas y productos informáticos.

También proporcionan un conjunto de características de calidad con las que se pueden comparar los requisitos de calidad establecidos para comprobar su integridad.

La estructura del modelo de calidad del producto establecido por la norma ISO/IEC 25010 se conforma mediante los ocho atributos de calidad que se presentan en la siguiente figura:



Figura 4.1 Norma ISO/IEC 25010

Fuente: ISO 25010, s. f.

4.2.1 Características de la norma ISO/IEC 25010

(ISO 25010, s. f.) El modelo de calidad que describe la norma ISO/IEC 25010 son los siguientes:

- **Adecuación Funcional:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.
- **Eficiencia de desempeño:** Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.
- **Compatibilidad:** Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.

- **Usabilidad:** Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.
- **Fiabilidad:** Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.
- **Seguridad:** Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.
- **Mantenibilidad:** Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
- **Portabilidad:** Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

4.3 Adecuación Funcional

La capacidad funcional se refiere a la habilidad del software para ofrecer las funciones requeridas para cumplir tanto con las necesidades específicas como con las implícitas al ser utilizado en situaciones particulares. Este aspecto del sistema no es directamente medible, por lo que, para evaluar la adecuación funcional, se emplea la métrica de punto función. Esta métrica se basa en la determinación de cinco características de dominios de información. Los valores para estas características se definen de la siguiente manera.

- **Número de entradas de usuario.** Se registra cada ingreso de usuario que suministra diversos datos dirigidos hacia la aplicación. Estos ingresos deben distinguirse de las solicitudes, las cuales se contabilizan por separado.

- **Número de salidas de usuario.** Cada resultado que ofrece información específica a la aplicación y es recibido por el usuario se toma en cuenta. En este contexto, las salidas pueden referirse a informes, pantallas, mensajes de error, entre otros.
- **Número de peticiones de usuario.** Cada petición se contabiliza por separado en el sistema.
- **Número de archivos.** Se registra cada archivo de un grupo lógico de datos que puede formar parte de una base de datos o ser un archivo independiente.
- **Número de interfaces externas.** Se contabilizan todas las interfaces legibles por la computadora que se utilizan para transmitir información a otro sistema específico.

Para calcular el punto función se usó la siguiente formula:

$$PF = Cuenta\ Total * (0.65 + 0.01 * \sum F_i)$$

Donde:

- **PF:** Medida de adecuación funcional.
- **Cuenta total:** Es la suma de los siguientes datos: N° de entradas, N° de salidas, N° de peticiones, N° de archivos y N° de interfaces externas.
- **0.65:** Confiabilidad del proyecto, varia el 1% al 100% es decir de (0 a 1)
- **0.01:** Error mínimo aceptable de complejidad
- **$\sum F_i:$** Son los valores de ajuste de complejidad, donde ($1 \leq i \leq 14$)

Analizando todas las interfaces que tiene el sistema, se obtuvieron los siguientes datos:

Tabla 4.1 Parámetros de medida y cantidad

| Parámetros de medida | Cantidad |
|-----------------------------|----------|
| Nº de entradas de usuario | 24 |
| Nº de salidas de usuario | 22 |
| Nº de peticiones de usuario | 33 |
| Nº de archivos | 7 |
| Nº de interfaces externas | 1 |

Después de recopilar la información de la tabla 4.1, se llevó a cabo el cálculo de la suma total utilizando el promedio de los factores de ponderación, tal como se refleja en la siguiente tabla:

Tabla 4.2 Parámetros de medida y ponderación

| Parámetros de medida | Cantidad | Factores de Ponderación | Total |
|-----------------------------|----------|-------------------------|-------|
| Nº de entradas de usuario | 24 | 4 | 96 |
| Nº de salidas de usuario | 22 | 5 | 110 |
| Nº de peticiones de usuario | 33 | 4 | 132 |
| Nº de archivos | 7 | 10 | 70 |
| Nº de interfaces externas | 1 | 7 | 7 |
| CUENTA TOTAL | | | 415 |

La cuenta total de los puntos fusión obtenida se debe ajustar en función a las características ambientales del sistema. Los valores de ajuste de complejidad Fi basados en las respuestas a las preguntas formuladas de la siguiente tabla:

Tabla 4.3 Valores de ajuste de complejidad

| Nº | Factores | 0: sin influencia | 1: incidental | 2: moderado | 3: medio | 4: significativo | 5: esencial | Fi |
|----|---|-------------------|---------------|-------------|----------|------------------|-------------|----|
| 1 | ¿Se requiere comunicación de datos? | | | | | | x | 5 |
| 2 | ¿Requiere el sistema copias de seguridad y de recuperaciones fiables? | | | | x | | | 3 |
| 3 | ¿Es crítico el rendimiento? | | | | x | | | 3 |
| 4 | Existen funciones de procesos distribuidos? | | | | | x | | 4 |
| 5 | ¿Se utilizaron los archivos maestros de forma interactiva? | | | | x | | | 3 |
| 6 | ¿Son complejas las entradas, las salidas y/o peticiones? | | | | x | | | 3 |
| 7 | ¿Es complejo el procesamiento interno? | | | | | x | | 4 |

| | | | | | | | |
|---------------------------------|--|--|--|---|---|---|-----------|
| 8 | ¿Sera ejecutado el sistema en un SO existente y fuertemente utilizado? | | | | | x | 5 |
| 9 | ¿Requiere el sistema entrada de datos interactiva? | | | | x | | 4 |
| 10 | ¿Requiere la entrada de datos interactiva que se utilicen varias pantallas o varias operaciones? | | | x | | | 3 |
| 11 | ¿Se ha diseñado el código para ser reutilizable? | | | | x | | 4 |
| 12 | ¿Están incluidas en el diseño la conversión y la instalación? | | | x | | | 3 |
| 13 | ¿Se ha diseñado el sistema para soportar diferentes instalaciones en diferentes organizaciones? | | | | x | | 4 |
| 14 | ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario? | | | | | x | 5 |
| Factor de ajuste de complejidad | | | | | | | 53 |

Una vez que se obtuvieron los valores para las variables de las fórmulas de los puntos función, se procedió a realizar los cálculos correspondientes para calcular la medida de adecuación funcional.

$$PF = Cuenta\ Total * \left(0.65 + 0.01 * \sum F_i \right)$$

$$PF = 415 * (0.65 + 0.01 * 53)$$

$$PF = 415 * (1.18)$$

$$PF = 489.7$$

Se compararon los puntos función con su valor máximo calculando los puntos función con valores de ajuste de complejidad hasta su máximo, que es un total de 70.

$$PF = Cuenta\ Total * \left(0.65 + 0.01 * \sum F_i \right)$$

$$PF = 415 * (0.65 + 0.01 * 70)$$

$$PF = 415 * (1.35)$$

$$PF = 560.25$$

Después de haber calculado ambos valores se tiene que la funcionalidad real es:

$$Adecuación\ funcional = \left(\frac{489.7}{560.25} \right) * 100\ %$$

$$Adecuación\ funcional = 87\%$$

Este resultado quiere decir que el sistema, satisface las necesidades de adecuación funcional en un 87% y en un 13% no satisface dichas necesidades.

4.4 Usabilidad

La usabilidad se refiere a la facilidad y atractivo con que un software puede ser comprendido, aprendido y utilizado. Para evaluar el nivel de usabilidad del sistema, se

llevó a cabo una encuesta con la participación de 10 personas. A continuación, se presentan los resultados obtenidos en dicha encuesta:

Tabla 4.4 Encuesta sobre usabilidad del sistema

| Nº | Pregunta | Respuestas | | Nro. % de Si |
|-----------------|---|------------|----|--------------|
| | | Si | No | |
| 1 | ¿Aprendió rápido a usar el sistema? | 8 | 2 | 80 |
| 2 | ¿Las pantallas que vio fueron de su agrado? | 10 | 0 | 100 |
| 3 | ¿Las pantallas que vio fueron fáciles de comprender? | 10 | 0 | 100 |
| 4 | ¿El sistema responde rápido a sus solicitudes? | 9 | 1 | 90 |
| 5 | ¿El sistema le facilitara el trabajo? | 9 | 1 | 90 |
| 6 | ¿El sistema reduce su tiempo de trabajo? | 10 | 0 | 100 |
| 7 | ¿Es fácil navegar por las distintas opciones? | 10 | 0 | 100 |
| 8 | ¿Las operaciones que se realizan no son complicadas? | 9 | 1 | 90 |
| 9 | ¿El sistema le proporciono las respuestas requeridas? | 10 | 0 | 100 |
| 10 | ¿El sistema no presento errores? | 9 | 1 | 90 |
| Promedio | | | | 94 |

El nivel de usabilidad del sistema evaluado a través de una encuesta con la participación de 10 personas, ha obtenido un resultado del 94%. Esto indica que el software es altamente comprensible, fácil de aprender y usar, según la percepción de los usuarios que participaron en la encuesta.

4.5 Fiabilidad

La fiabilidad se refiere a la habilidad del software para funcionar de manera adecuada en condiciones específicas y durante un período de tiempo determinado. En este análisis se evaluó el nivel de fiabilidad del sistema, considerando inicialmente la fiabilidad de cada módulo o subsistema por separado.

Para calcular la fiabilidad de cada módulo se usó la formula:

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

Donde:

- **R (t):** Fiabilidad de un componente o subsistema t.
- **λ :** Tasa de constantes de fallo: $\lambda = \left(\frac{\text{Nº de fallas de acceso}}{\text{Nº total de accesos al sistema}} \right)$.
- **t:** Periodo de operación de tiempo.
- **$e^{-\lambda t}$:** Probabilidad de falla de un componente o subsistema en el tiempo t.

Luego de realizar pruebas de cada módulo en un tiempo de 4 horas continuas se logró llenar la siguiente tabla:

Tabla 4.5 Valores de fiabilidad de cada módulo

| Nº | Módulo | λ | t | R(t) |
|----|--|-----------|--------|------|
| 1 | Módulo de autenticación y de autorización. | 0.012 | 4 Hrs. | 0.95 |
| 2 | Módulo de registro de usuarios. | 0.015 | 4 Hrs. | 0.94 |
| 3 | Módulo de registro de pacientes. | 0.013 | 4 Hrs. | 0.95 |
| 4 | Módulo de registro de citas. | 0.015 | 4 Hrs. | 0.94 |

| | | | | |
|----------|-----------------------------|-------|--------|------|
| 5 | Módulo de consulta. | 0.017 | 4 Hrs. | 0.93 |
| 6 | Módulo de historia clínica. | 0.015 | 4 Hrs. | 0.94 |
| 7 | Módulo de reporte. | 0.014 | 4 Hrs. | 0.95 |

Para calcular la fiabilidad del sistema completo, se observó que el módulo de autenticación y de autorización operan en serie con los demás módulos. Por otro lado, el resto de los módulos funcionan en paralelo, lo que significa que su funcionamiento es independiente de los otros. Esta consideración nos lleva a que la fiabilidad del sistema estaría dada por la fórmula:

$$Fiabilidad = R_s * R_p$$

Donde:

$$R_s = R_1 = 0.95 \quad \text{y} \quad R_p = \frac{\sum_{i=2}^7 (R_i * P_i)}{\sum_{i=2}^7 P_i}$$

En la fórmula de R_p , la variable P_i representa la contribución del equipo de desarrollo en el desarrollo del módulo. Como la participación fue al 100% entonces $P_i = 1$, así se tiene el siguiente resultado:

$$R_p = \frac{\sum_{i=2}^7 R_i}{6} = \frac{0.94 + 0.95 + 0.94 + 0.93 + 0.94 + 0.95}{6} = \frac{5.65}{6} = 0.94$$

Por lo tanto, la fiabilidad del sistema está dada por:

$$Fiabilidad = 0.95 * 0.94 = 0.89 = 89\%$$

Se puede concluir que hay una probabilidad del 11% de que el sistema experimente algún tipo de error en caso de que se supere un periodo prolongado de uso. Esto podría deberse a posibles fallos en la conexión entre el sistema y la base de datos, conexiones del cliente al sistema, utilización incorrecta por parte del usuario o errores en la introducción de datos.

4.6 Mantenibilidad

La mantenibilidad se refiere a la capacidad del software para ser modificado, adaptándose a correcciones, mejoras, y cambios en el entorno y requerimientos funcionales. Con el propósito de evaluar la calidad de mantenimiento del sistema, se empleará el índice de madurez del software (IMS), el cual proporciona una medida de la estabilidad de un producto de software. La fórmula para calcular el índice de madurez es la siguiente:

$$IMS = \frac{M_t - (F_a + F_b + F_c)}{M_t}$$

Donde:

- ***Mt***: Número de módulos en la versión actual.
- ***Fa***: Número de módulos en la versión actual que se han cambiado.
- ***Fb***: Número de módulos en la versión actual que se han añadido.
- : Número de módulos en la versión anterior que se han borrado en la versión actual.

Se ha recopilado la información necesaria para la fórmula, obteniendo la información que se muestra en la siguiente tabla:

Tabla 4.6 Información requerida por el IMS

| Información | Valor |
|-------------|-------|
| M_t | 7 |
| F_a | 0 |
| F_b | 0 |
| F_c | 0 |

Ahora calculemos el IMS, usando los valores obtenidos:

$$\text{IMS} = \frac{7 - (0 + 0 + 0)}{7}$$

$$\text{IMS} = \frac{7}{7}$$

$$\text{IMS} = 1 * 100\% = 100\%$$

Con ese resultado se concluyó que el sistema, tiene un índice de madurez de software del 100%. Por lo tanto, el sistema cumple la característica de mantenibilidad al 100%.

4.7 Portabilidad

La portabilidad se refiere a la habilidad del software para ser transferido de un entorno a otro. Con el fin de evaluar la portabilidad del sistema, se empleará una fórmula específica que cuantifica el nivel de portabilidad que posee un software.

$$GP = 1 - \frac{ET}{ER}$$

Donde:

- **GP:** Grado de portabilidad.

- **ET:** Representa el esfuerzo de adaptación para trasladar el software a otro entorno.
- **ER:** Corresponde al esfuerzo requerido para la adaptación total del software al nuevo entorno.
- Si **GP > 0:** La portabilidad es más rentable que el redesarrollo.
- Si **GP = 1:** La portabilidad es perfecta.
- Si **GP < 0:** El redesarrollo es más rentable que la portabilidad.

Para llevar el sistema de historias clínicas a otro entorno, se necesita las siguientes especificaciones mínimas:

- Memoria extraíble de 4GB.
- Servidor con capacidad de almacenamiento mayor a 8GB, memoria RAM de 4GB y acceso a red local.
- Instalar las dependencias necesarias en el servidor para ejecutar el sistema (Docker).

Para calcular ET y ER se debe realizar las siguientes acciones:

- **Esfuerzo de Adaptación (ET):** Se refiere al esfuerzo específico requerido para instalar las dependencias (Docker) en el servidor. Considerando que esta es la única acción específica que se menciona para adaptar el sistema al nuevo entorno, se podría asignar un valor de una escala de 1 al 10, siendo 10 el valor máximo de esfuerzo. En este contexto elegimos **1 como el valor para ET**, el cual será el esfuerzo para instalar las dependencias de Docker.
- **Esfuerzo Requerido (ER):** En este contexto, debería considerarse el esfuerzo total para llevar a cabo la portabilidad del sistema, incluyendo la obtención y configuración del servidor con las especificaciones mínimas, además de la instalación de las dependencias (Docker). En este caso, debido a que las especificaciones del servidor también son requerimientos

adicionales para la portabilidad, se podría asignar un valor mayor para ER. En este contexto elegimos **7 unidades para el valor de ER.**

Con esta información requerida para la fórmula, se procede a calcular el grado de portabilidad:

$$GP = 1 - \frac{1}{7}$$

$$GP = 1 - 0.14$$

$$GP = 0.86 * 100\%$$

$$GP = 86\%$$

El análisis llevado a cabo revela que el sistema exhibe un grado de portabilidad del 86%, lo que sugiere una adaptabilidad significativa para ser trasladado a otro entorno con una alta eficiencia.

4.8 Seguridad

La seguridad, dentro del marco de la norma ISO 25010, es un componente esencial que se refiere a la capacidad del software para preservar la confidencialidad, integridad y disponibilidad de la información, así como para prevenir accesos no autorizados y garantizar la protección de los datos.

(Calabrese et al., 2017) Como caso de estudio se tomó la característica Seguridad que contiene las subcaracterísticas: Confidencialidad, Integridad, No-Repudio, Responsabilidad y Autenticidad.

- **CONFIDENCIALIDAD:** Evalúa la capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.

- **INTEGRIDAD:** Evalúa la capacidad del sistema o componente para prevenir accesos modificaciones no autorizados a datos o programas de ordenador.
- **NO-REPUDIO:** Evalúa la capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- **RESPONSABILIDAD:** Evalúa la capacidad de rastrear de forma inequívoca las acciones de una entidad.
- **AUTENTICIDAD:** Evalúa la capacidad de demostrar la identidad de un sujeto o un recurso.

Se establecieron 33 preguntas de respuesta verdadero/falso basadas en las subcaracterísticas de la seguridad mencionadas previamente. Estas preguntas se detallan en la siguiente tabla:

Tabla 4.7 Cuestionario para la característica de Seguridad

| ID | Pregunta |
|-----|---|
| P1 | ¿Se requiere que la contraseña posea al menos 8 caracteres? |
| P2 | ¿Se requiere que la contraseña posea letras mayúsculas y minúsculas? |
| P3 | ¿Se requiere que la contraseña posea números y letras? |
| P4 | ¿Se requiere que la contraseña posea caracteres especiales? |
| P5 | ¿El sistema utiliza conexión segura mediante HTTPS? |
| P6 | ¿La base de datos posee los datos encriptados? |
| P7 | ¿El sistema permite acceder a funcionalidades en las cuales no se tiene permiso? |
| P8 | ¿El sistema permite que cualquier persona tenga acceso a la base de datos? |
| P9 | ¿El sistema permite que cualquier persona tenga acceso al código del servidor de la aplicación? |
| P10 | ¿Cualquier persona tiene acceso al servidor físico? |

| | |
|------------|--|
| P11 | ¿Cualquier persona tiene acceso al servidor remoto? |
| P12 | ¿El sistema posee redireccionamientos hacia sitios no seguros? |
| P13 | ¿El sistema solicita una confirmación de registro mediante un mail a la hora de registrarse? |
| P14 | ¿El sistema permite que cualquier persona pueda modificar la base de datos? |
| P15 | ¿El sistema permite que cualquier persona pueda modificar el código del servidor de la aplicación? |
| P16 | ¿El sistema permite inyecciones SQL? |
| P17 | ¿El sistema posee un historial de acciones realizadas? |
| P18 | ¿El sistema posee algoritmos de cifrado de datos? |
| P19 | ¿El sistema posee un mecanismo criptográfico, como firma digital? |
| P20 | ¿El sistema solicita confirmación a la hora de realizar una acción? |
| P21 | ¿El sistema posee una protección con certificados SSL? |
| P22 | ¿El sistema da aviso cuando se accede desde una ubicación desconocida? |
| P23 | ¿El sistema informa vía mail las operaciones realizadas? |
| P24 | ¿El sistema guarda un registro de fecha y hora de ingreso al mismo? |
| P25 | ¿El sistema registra el tipo de navegador y sistema de operación utilizado para ingresar al sitio? |
| P26 | ¿El sistema registra la dirección IP desde la cual se ingresa al sitio? |
| P27 | ¿El sistema realiza una comprobación de identidad mediante un certificado digital? |
| P28 | ¿El sistema posee un sistema de verificación en dos pasos? |
| P29 | ¿Es requerida una clave de segundo nivel para el ingreso al sistema? |
| P30 | ¿El sistema realiza una comprobación de identidad mediante datos biométricos? |
| P31 | ¿El sistema realiza una comprobación de identidad mediante tarjeta de coordenadas? |
| P32 | ¿El sistema realiza una comprobación de identidad mediante credenciales? |

| | |
|------------|---|
| P33 | ¿El sistema realiza una comprobación de identidad mediante una firma electrónica? |
|------------|---|

Fuente: Calabrese et al., 2017.

4.8.1 Descripción de criterios de evaluación (CE)

(Calabrese et al., 2017) Para alcanzar el objetivo de calcular el valor de seguridad dentro del sistema, se combinaron las respuestas de las preguntas de manera lógica para asignar un puntaje a cada uno de los CE (criterios de evaluación).

La siguiente tabla nos muestra las formas de calcular cada una de las subcaracterísticas a partir de las preguntas realizadas en la tabla 4.7, en el apartado de puntos se calcula el valor según las respuestas, para calcular se deben responder a las preguntas, estas pueden ser verdad o falso, después se calculan con la formula y nos da los valores de 0 a 1.

Tabla 4.8 Descripción de criterios de evaluación (CE)

| ID | Nombre | Descripción | Fórmula | Ptos. Según las respuestas |
|-----|--------------------|--|--|----------------------------|
| C-1 | Conexiones seguras | Una conexión se considera segura si se utiliza HTTPS y si no se tienen redireccionamientos hacia sitios no seguros | $P5 \& \sim P12 = V$ | 1 |
| C-2 | Control de acceso | Se debe controlar que no se permita acceder a funcionalidades sin autorización, tampoco a la base de datos, al código de la aplicación ni a los servidores, físico ni remoto | $si\ P7\ P8\ P9\ P10\ P11 = F$ | 1 |

| | | | | |
|-------------|------------------------------|---|---------------------------|------------|
| C-3 | Encriptación de datos | Los datos de la base de datos deben estar encriptados | P6 = V | 1 |
| C-4 | Contraseña de bajo nivel | La contraseña se considera de bajo nivel si posee menos de 8 caracteres, no posee letras mayúsculas y minúsculas, no posee letras y números y no posee caracteres especiales | P1 P2 P3 P4 = F | 0 |
| | Contraseña de medio nivel | La contraseña se considera de medio nivel si posee al menos 8 caracteres o letras mayúsculas y minúsculas o letras y números. | P1 P2 P3 P4 = V | 0.5 |
| | Contraseña de alto nivel | La contraseña se considera de alto nivel si posee al menos 8 caracteres, letras mayúsculas y minúsculas, letras y números y caracteres especiales | P1 & P2 & P3 & P4 = V | 1 |
| I-5 | Prevención de accesos | Se debe prevenir que no se permita acceder a funcionalidades sin autorización, tampoco a la base de datos ni al código de la aplicación, y que no se permitan inyecciones SQL | P7 P8 P9 P16 = F | 1 |
| I-6 | Prevención de modificaciones | Se debe prevenir que no se permita modificar datos de la base de datos ni modificar el código de la aplicación sin autorización | P14 P15 = F | 1 |
| I-7 | Confirmación de datos | Se debe realizar una confirmación de registro por mail | P13 = V | 0 |
| NR-8 | Operaciones realizadas | Se debe poseer un historial de acciones realizadas o las mismas deben ser enviadas por mail | P17 P23 = V | 1 |

| | | | | |
|--------------|------------------------------|--|---------------------------------------|----------|
| NR-9 | Mecanismos de cifrado | Se debe poseer un algoritmo de cifrado de datos o un mecanismo criptográfico, como firma digital, o una protección con certificados SSL | P18 P19 P21 = V | 1 |
| NR-10 | Confirmación de acciones | Se debe solicitar una confirmación al realizar una determinada acción | P20 = V | 1 |
| NR-11 | Registro de ubicación | Se debe informar si se accedió al sistema desde una ubicación desconocida | P22 = V | 0 |
| R-12 | Registro de acciones y datos | Se debe poseer un historial de acciones realizadas, o un registro de fecha y hora de ingreso al sistema o de la dirección IP desde la cual se ingresa o del tipo de navegador y sistema de operación utilizado | P17 P24 P25 P26 = V | 1 |
| R-13 | Control de ubicación | Se debe dar aviso cuando se accede al sistema desde una ubicación desconocida | P22 = V | 0 |
| A-14 | Comprobación de identidad | El sistema debe realizar una comprobación de identidad mediante alguno de los siguientes métodos: datos biométricos, tarjeta de coordenadas, credenciales, firma electrónica o certificado digital | P27 P30 P31 P32 P33 = V | 1 |
| A-15 | Comprobaciones adicionales | Se debe poseer un sistema de verificación en dos pasos, o se debe requerir una clave de segundo nivel para el ingreso al sistema o una confirmación de registro mediante un mail | P28 P29 P13 = V | 1 |

Fuente: Calabrese et al., 2017.

4.8.2 Métricas para cada subcaracterística

(Calabrese et al., 2017) Los CE se combinaron para establecer las métricas que cumplen con los objetivos de las subcaracterísticas. Para cada una de ellas se especificó un nombre, un propósito, un método de aplicación, valores de entrada y la fórmula correspondiente.

Se combinaron los CE para definir las métricas que satisfacen los objetivos de las subcaracterísticas. Para cada una se definió un nombre, un propósito, un método de aplicación, valores de entradas y fórmula aplicada.

- **Confidencialidad.**
 - **Métrica:** Confidencialidad
 - **Propósito:** ¿Cuán eficiente es el sistema a la hora de proteger el acceso de datos e información no autorizados, ya sea accidental o deliberadamente?
 - **Método de aplicación:** Contestar las preguntas de los CE correspondientes a la subcaracterística "Confidencialidad" y calcular la puntuación obtenida, sumando los puntajes de los CE referenciados que cumplan con la meta esperada. "Puntaje total" hace referencia al máximo puntaje que se puede obtener.
 - **Entradas:** $A = \text{Puntaje obtenido}$. $B = \text{Puntaje total}$.
 - **Fórmula:** $X = A/B$
 - **Observaciones:** Los CE a utilizar son: C-1, C-2, C-3 y C-4
- **Integridad.**
 - **Métrica:** Integridad
 - **Propósito:** ¿Cuán capaz es el sistema a la hora de prevenir accesos o modificaciones no autorizados a datos o programas de ordenador?
 - **Método de aplicación:** Contestar las preguntas de los CE correspondientes a la subcaracterística "Integridad" y calcular la puntuación obtenida,

sumando los puntajes de los CE referenciados que cumplan con la meta esperada. “Puntaje total” hace referencia al máximo puntaje que se puede obtener.

- **Entradas:** $A = \text{Puntaje obtenido}$. $B = \text{Puntaje total}$.
- **Fórmula:** $X = A/B$
- **Observaciones:** Los CE a utilizar son: I-5, I-6 e I-7.
- **No-Repudio.**
 - **Métrica:** No-Repudio
 - **Propósito:** ¿Cuán capaz es el sistema de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente?
 - **Método de aplicación:** Contestar las preguntas de los CE correspondientes a la subcaracterística "No-Repudio" y calcular la puntuación obtenida, sumando los puntajes de los CE referenciados que cumplan con la meta esperada. “Puntaje total” hace referencia al máximo puntaje que se puede obtener.
 - **Entradas:** $A = \text{Puntaje obtenido}$. $B = \text{Puntaje total}$.
 - **Fórmula:** $X = A/B$
 - **Observaciones:** Los CE a utilizar son: NR-8, NR-9, NR-10 y NR-11
- **Responsabilidad.**
 - **Métrica:** Responsabilidad
 - **Propósito:** ¿Cuán capaz es el sistema de rastrear de forma inequívoca las acciones de una entidad?
 - **Método de aplicación:** Contestar las preguntas de los CE correspondientes a la subcaracterística "Responsabilidad" y calcular la puntuación obtenida, sumando los puntajes de los CE referenciados que cumplan con la meta esperada. “Puntaje total” hace referencia al máximo puntaje que se puede obtener.

- **Entradas:** $A = \text{Puntaje obtenido}$. $B = \text{Puntaje total}$.
- **Fórmula:** $X = A/B$
- **Observaciones:** Los CE a utilizar son: R-12 y R-13.
- **Autenticidad.**
 - **Métrica:** Autenticidad
 - **Propósito:** ¿Cuán capaz es el sistema de demostrar la identidad de un sujeto o un recurso?
 - **Método de aplicación:** Contestar las preguntas de los CE correspondientes a la subcaracterística "Autenticidad" y calcular la puntuación obtenida, sumando los puntajes de los CE referenciados que cumplan con la meta esperada. "Puntaje total" hace referencia al máximo puntaje que se puede obtener.
 - **Entradas:** $A = \text{Puntaje obtenido}$. $B = \text{Puntaje total}$.
 - **Fórmula:** $X = A/B$
 - **Observaciones:** Los CE a utilizar son: A14 y A15.

Las fórmulas correspondientes a cada subcaracterística, basadas en los puntajes obtenidos en la tabla 4.8, se muestran en la siguiente tabla, proporcionando un resultado específico para cada subcaracterística.

Tabla 4.9 Fórmulas para las subcaracterísticas de Seguridad

| MÉTRICA | FÓRMULA | RESULTADO |
|-------------------------|---------------------------|-----------------------|
| CONFIDENCIALIDAD | $(C1+C2+C3+C4) / 4$ | $0.75 * 100\% = 75\%$ |
| INTEGRIDAD | $(I5+I6+I7) / 3$ | $0.67 * 100\% = 67\%$ |
| NO REPUDIO | $(NR8+NR9+NR10+NR11) / 4$ | $0.75 * 100\% = 75\%$ |
| RESPONSABILIDAD | $(R12+R13) / 2$ | $0.50 * 100\% = 50\%$ |

| | | |
|-------------------------|---------------|--------------------------|
| AUTENTICIDAD | (A14+A15) / 2 | 1*100% = 100% |
| Porcentaje Total | | 73.4% |

Fuente: Calabrese et al., 2017.

4.8.3 Ejecutar evaluación

Para ejecutar la evaluación de la característica de seguridad y sus subcaracterísticas, (Calabrese et al., 2017) nos da estos criterios de aceptación para cada subcaracterística:

- Inaceptable: $0 \leq X < 40$
- Mínimamente aceptable: $40 \leq X < 60$
- Rango objetivo: $60 \leq X < 90$
- Excede los requerimientos: $90 \leq X \leq 100$

El propósito se considerará aceptado si todas las subcaracterísticas se encuentran entre los rangos mínimamente aceptables y excede los requerimientos.

Finalmente, interpretamos los criterios de aceptación basándonos en los resultados obtenidos:

La característica de seguridad se considera que cumple con el propósito de la evaluación ya que todas sus subcaracterísticas se encuentran en un rango de aceptación.

Ya que las subcaracterísticas Confidencialidad, Integridad y No Repudio están en el rango objetivo. Responsabilidad en el rango mínimamente aceptable y Autenticidad en el rango que excede las expectativas.

Por lo tanto, el sistema se considera adecuado para su implementación, ya que la evaluación revela que cumple satisfactoriamente con los criterios establecidos para la

característica de seguridad, garantizando la integridad, confidencialidad y no repudio requeridos, además de un nivel superior de autenticidad frente a las expectativas previstas.

4.9 Calidad Global

Después de obtener los porcentajes de los distintos atributos del sistema según lo estipulado en la norma de calidad ISO/IEC 25010, se procedió a evaluar la calidad global del sistema, presentada en la siguiente tabla:

Tabla 4.10 Calidad Global del Sistema

| Característica | Valor de (%) |
|-----------------------|--------------|
| Adecuación funcional | 87 |
| Usabilidad | 94 |
| Fiabilidad | 89 |
| Mantenibilidad | 100 |
| Portabilidad | 86 |
| Seguridad | 73 |
| Calidad Global | 88.17 |

Después de evaluar las distintas características según la norma ISO/IEC 25010, se determinó que la Calidad Global del “Sistema Web de Administración de Historias Clínicas Caso: Facultad de Medicina, Instituto de Genética, Unidad de Genética Médica” alcanza un porcentaje del **88.17 %**.

CAPÍTULO V COSTO BENEFICIO

5.1 Introducción

Este capítulo aborda el tema del Costo-Beneficio, proporcionando un análisis detallado sobre esta importante métrica en el ámbito específico del sistema web de administración de historias clínicas, utilizando la metodología COCOMO II. Se exploran diversos aspectos relacionados con la evaluación de costos y beneficios, así como su relevancia en la toma de decisiones estratégicas.

5.2 Modelo COCOMO II

(Boehm, 2000) El modelo COCOMO II es una herramienta de estimación de costos de software desarrollada por Barry Boehm en la década de 1990. Este modelo se basa en los principios del modelo original COCOMO, pero con mejoras significativas. El Modelo COCOMO II se utiliza para estimar el costo, el esfuerzo y el tiempo necesarios para desarrollar un proyecto de software y es ampliamente utilizado en la industria del software.

(Migani, 2014) Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.

- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

Cocomo II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura. Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores descriptos, 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.

5.2.1 Líneas de código fuente (SLOC)

(Clark & Madachy, 2015) La medida común del tamaño del software utilizada en este manual y en los modelos de costes son las líneas de código fuente (SLOC). SLOC son declaraciones lógicas de código fuente que consisten en declaraciones de datos y ejecutables.

5.2.2 Definiciones de los tipos SLOC

Esta herramienta utiliza una serie específica de parámetros para llevar a cabo el cálculo de las estimaciones necesarias. (Clark & Madachy, 2015) Estos parámetros incluyen:

- **SLOC (Source Lines of Code):** Aquí se espera que ingreses la cantidad de líneas de código fuente que estás desarrollando para tu proyecto. Esta métrica puede variar según el lenguaje de programación y la complejidad del proyecto. Es una medida crítica para la estimación de costos.

- **% Design Modified:** Indica el porcentaje de modificaciones en el diseño. Si estás creando un nuevo diseño desde cero, este valor será del 100%. Si no estás haciendo modificaciones al diseño, el valor será del 0%.
- **% Code Modified:** Representa el porcentaje de modificaciones en el código. Si estás creando nuevo código sin basarte en código previo, será del 100%. Si no hay modificaciones, este valor será del 0%.
- **% Integration Required:** Indica el porcentaje de integración necesaria con otros módulos o sistemas. Si estás trabajando de forma independiente, será del 0%. Si se necesita una integración significativa, será un valor mayor.
- **Assessment and Assimilation (0% - 8%):** Esto se refiere al esfuerzo requerido para evaluar y asimilar el software existente. El rango de valores va desde el 0% hasta el 8% según la complejidad de esta tarea.
- **Software Understanding (0% - 50%):** Indica el porcentaje de comprensión del software. Va desde 0% (ninguna comprensión) hasta 50% (total comprensión).
- **Unfamiliarity (0-1):** Esta es una medida subjetiva que representa el nivel de familiaridad del equipo con la tecnología, el dominio del proyecto o las herramientas utilizadas. Va desde 0 (totalmente familiarizado) hasta 1 (totalmente no familiarizado).

Además, esta herramienta hace referencia a las tres categorías de software que deben considerarse en el proceso de desarrollo:

- **New (Nuevo):** Se refiere a la porción del software que se está creando completamente desde cero para el proyecto actual, sin utilizar ningún código existente o heredado.
- **Reused (Reutilizado):** Representa la porción del software que se está adaptando o reutilizando de proyectos anteriores o de componentes de

software ya existentes. Se trata de código que ya ha sido probado y validado previamente.

- **Modified (Modificado):** Esta categoría incluye el software existente que está siendo modificado o adaptado para cumplir con los requerimientos del proyecto actual. Puede tratarse de código heredado que se está modificando o mejorando.

5.3 Cálculo de costo de Software

Para calcular el costo total mediante COCOMO II, se empleó una herramienta web desarrollada por softwarecost.org. Esta herramienta ofrece una implementación integral del Modelo de Costos Constructivos COCOMO II y proporciona funcionalidades para estimar los costos del desarrollo de software.

Para determinar el costo total, se empleará el recuento de líneas de código (SLOC):

- **SLOC = 3235 líneas de código.**

Adicionalmente, se completarán los datos esenciales para el servicio utilizando la información recopilada por el equipo de desarrollo basada en el número de líneas de código.

Además de los datos relacionados con las líneas de código, el modelo requiere una serie de características adicionales, como la flexibilidad en el desarrollo, la estructura arquitectónica, el trabajo en equipo, el tamaño de la base de datos, la complejidad del producto, la capacidad del personal, las restricciones temporales, de almacenamiento y el uso de herramientas de software, entre otros aspectos que el equipo de desarrollo debe detallar.

Por último, se debe proporcionar el coste por persona-mes en dólares, calculado en función del entorno laboral y las condiciones del país. En este caso, se consideró el salario correspondiente a un programador junior, dando como resultado:

- **Coste por persona-mes = 5000 bolivianos = 722.93 dólares.**

Ya obtenidos estos valores calculados y por parte del equipo de desarrollo, usamos la herramienta para calcular las estimaciones de costo:

COCOMO II - Constructive Cost Model

| | | | | | | | | | |
|--|--------------------------------------|---|--------------------------------|-------------------------------|------------------------|--|-----------------------------------|---------------------|--|
| Software Size | Sizing Method | Source Lines of Code <input type="button" value="▼"/> | | | | | | | |
| | | SLOC | % Design Modified | % Code Modified | % Integration Required | Assessment and Assimilation (0% - 8%) | Software Understanding (0% - 50%) | Unfamiliarity (0-1) | |
| New | | 3235 | | | | | | | |
| Reused | | 620 | | | | | | | |
| Modified | | 720 | 3 | 4 | | | 48 | 1 | |
| Software Scale Drivers | | | | | | | | | |
| Precedentedness | | Nominal | Architecture / Risk Resolution | Nominal | Process Maturity | <input type="button" value="Nominal ▼"/> | | | |
| Development Flexibility | | Nominal | Team Cohesion | High | | | | | |
| Software Cost Drivers | | | | | | | | | |
| Product | | Personnel | | Platform | | | | | |
| Required Software Reliability | Nominal | Analyst Capability | Nominal | Time Constraint | High | <input type="button" value="High ▼"/> | | | |
| Data Base Size | Nominal | Programmer Capability | Nominal | Storage Constraint | Nominal | <input type="button" value="Nominal ▼"/> | | | |
| Product Complexity | Low | Personnel Continuity | Nominal | Platform Volatility | Nominal | <input type="button" value="Nominal ▼"/> | | | |
| Developed for Reusability | High | Application Experience | Nominal | Project | | | | | |
| Documentation Match to Lifecycle Needs | Nominal | Platform Experience | Nominal | Use of Software Tools | High | <input type="button" value="High ▼"/> | | | |
| | | Language and Toolset Experience | Nominal | Multisite Development | Nominal | <input type="button" value="Nominal ▼"/> | | | |
| | | | | Required Development Schedule | Nominal | | | | |
| Maintenance | <input type="button" value="Off ▼"/> | | | | | | | | |
| Software Labor Rates | | | | | | | | | |
| Cost per Person-Month (Dollars) | 722.93 | | | | | | | | |
| <input type="button" value="Calculate"/> | | | | | | | | | |

Figura 5.1 Inserción de datos para COCOMO II

Fuente: Madachy, R. (s. f.).

Una vez completada la información solicitada por la herramienta, se llevó a cabo el cálculo y la revisión de los resultados que ofrece. Estos resultados se detallan en la figura 5.2:

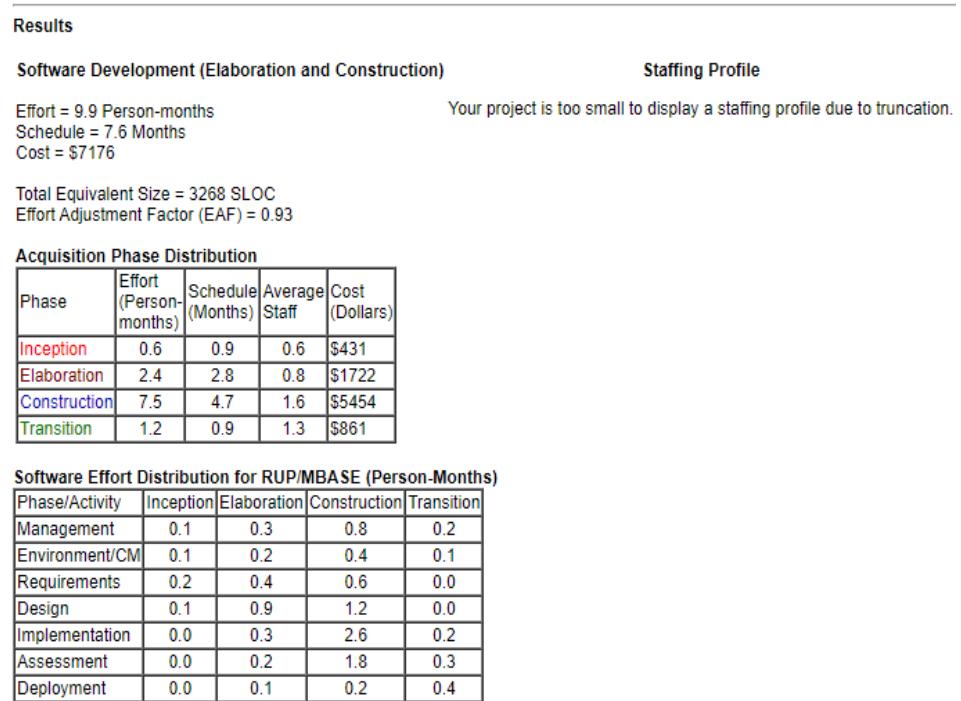


Figura 5.2 Resultado de estimación de costos

Fuente: Madachy, R. (s. f.).

La información proporcionada por la herramienta constituye datos significativos, incluyendo el esfuerzo estimado de 9.9 personas-mes, un tiempo de desarrollo estimado de 7.6 meses, la cantidad promedio de personas requeridas para llevar a cabo el proyecto es 1, y el coste final del proyecto se estima en **7,176 dólares**.

5.4 Costo de Elaboración del Proyecto

Los costos de elaboración del proyecto se detallan en la tabla 5.1:

Tabla 5.1 Costo de elaboración

| Descripción | Costo \$us |
|------------------------|------------|
| Análisis del proyecto | 200 |
| Pasajes | 80 |
| Material de escritorio | 50 |
| Otros | 10 |
| Total | 340 |

5.5 Costo de Implementación del Proyecto

La Unidad de Genética Médica del Instituto de Genética dispone de los equipos requeridos, junto con un servidor suministrado por la Facultad de Medicina de la Universidad Mayor de San Andrés. Por lo tanto, el costo es de 0.

5.6 Costo Total

El costo total es la suma de todos los costos, de software, de elaboración e implementación, tal como se detalla en la tabla 5.2.

Tabla 5.2 Costo Total

| Descripción | Costo \$us |
|----------------------|------------|
| Costo de software | 7,176 |
| Costo de elaboración | 340 |

| | |
|-------------------------|--------------|
| Costo de implementación | 0 |
| Costo Total | 7,516 |

De esa forma se define el costo del proyecto en 7,516 dólares.

5.7 Cálculo de Valor Actual Neto - VAN

El Valor Actual Neto (VAN) es un procedimiento que permite calcular el valor presente de un determinado número de flujos de futuros ingresos y egresos que tendrá el proyecto. Este enfoque implica descontar el momento actual, es decir, ajustar por el valor del tiempo usando una taza. A este resultado se le resta la inversión inicial, proporcionando así el valor actual neto.

$$VAN = \sum \frac{Ganancias}{(1 + k)^n} - l_0$$

Donde:

- **VAN:** Valor Actual Neto.
- **Ganancias:** Ingreso del flujo anual.
- **l_0 :** Es el valor del desembolso inicial de la inversión.
- **k :** Tasa de descuento o tasa de interés al préstamo.
- **n :** Es el número de periodos considerados.

Los valores de ganancia esperados para el presente proyecto a lo largo de un período de cinco años se determinan utilizando una tasa de interés del 10% que corresponde a una tasa de préstamo. Estos cálculos se emplean para obtener el Valor Actual Neto (VAN) y se presentan de la siguiente manera:

$$inversion = 7,516$$

$$k = 10\%$$

Para calcular los valores de ganancia esperada, se recopiló información del instituto y se aplicaron las fórmulas detalladas en la siguiente tabla:

Tabla 5.3 Calculo de VAN

| Año | Ganancia | $(1 + k)^n$ | $\frac{Ganancias}{(1 + k)^n}$ |
|--------------|--------------|-------------|-------------------------------|
| 1 | 0 | 1.1 | 0 |
| 2 | 3000 | 1.21 | 2,479.34 |
| 3 | 6000 | 1.331 | 4,507.89 |
| 4 | 7000 | 1.4641 | 4,781.09 |
| 5 | 8000 | 1.61051 | 4,967.37 |
| Total | 24000 | 6.72 | 16,735.69 |

Aplicando la fórmula.

$$VAN = \sum \frac{Ganancias}{(1 + k)^n} - l_0$$

$$VAN = 16735.69 - 7516$$

$$VAN = 9219.69$$

Se tiene que el valor obtenido es mayor a cero por lo que la inversión es factible en un principio y el proyecto es rentable.

5.8 Relación Costo Beneficio

Para hallar la relación del costo beneficio se utiliza la siguiente formula:

$$RCB = \frac{costo}{beneficio}$$

Reemplazando los valores encontrados anteriormente, resulta lo siguiente:

$$RCB = \frac{7516}{4000} = 1.88$$

Del resultado obtenido se concluye que la ganancia por cada 1 dólar invertido la ganancia es 0.88 centavos de dólar. Por tanto, se concluye que el proyecto es viable.

CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES

6.1 Introducción

En este capítulo, se expondrán las conclusiones y recomendaciones derivadas del presente proyecto titulado “SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS CASO: FACULTAD DE MEDICINA, INSTITUTO DE GENÉTICA, UNIDAD DE GENÉTICA MÉDICA”. Por tanto, se analizarán los resultados obtenidos durante el desarrollo del proyecto. Y finalmente, se ofrecerán recomendaciones específicas para futuras mejoras del sistema y posibles áreas de investigación que puedan surgir a partir de este trabajo.

6.2 Conclusiones

La gestión efectiva de la información de historias clínicas en un establecimiento de salud es fundamental para garantizar la calidad de la atención médica, mejorar la eficiencia de los procesos clínicos y administrativos, y ayuda a que los médicos tomen decisiones clínicas informadas de los pacientes. A lo largo de este proyecto, se ha examinado en profundidad la relevancia de esta gestión mediante el desarrollo del sistema web de administración de historias clínicas.

Inicialmente, la gestión de historias clínicas en la Unidad de Genética Médica del Instituto de Genética de la Universidad Mayor de San Andrés presentaba un manejo ineficiente de las historias clínicas de los pacientes, ocasionando la pérdida de información, conflictos internos y limitaciones en el acceso a los datos, ya que se realizaba de manera manual y almacenaban en formatos físicos y/o en archivos de Excel. En base a lo mencionado, el presente proyecto surge como una respuesta a estas problemáticas.

El objetivo general de desarrollar un sistema web para la gestión de historias clínicas para la Unidad de Genética Medica del Instituto de Genética, ha sido alcanzado

satisfactoriamente y dicho sistema se encuentra en actual funcionamiento en la unidad mencionada.

En cuanto al cumplimiento de los objetivos específicos, se abordaron de la siguiente manera:

- Se estableció un proceso de registro sistematizado y validado para el registro de datos clínicos de los pacientes, con el fin de mantener la integridad de sus datos e información.
- Se implementó un módulo en el sistema para facilitar el registro de citas médicas, agilizando la atención de los pacientes.
- Se implementó una base de datos centralizada para la gestión de registros de los pacientes, facilitando su seguimiento clínico.
- Se facilitó el acceso a la información de datos clínicos a través del sistema de administración de historias clínicas, permitiendo a los usuarios encontrar rápidamente la información necesaria.
- Se optimizó el flujo de atención médica mediante la aplicación de un proceso que permite recuperar rápidamente los datos necesarios para las consultas médicas.

Por otro lado, las problemáticas planteadas en el presente proyecto fueron satisfactoriamente solucionadas con la entrega del sistema, solucionando la pregunta:

¿Cómo facilitar la gestión de información de historias clínicas para la unidad de genética médica del instituto de genética?

Y los problemas secundarios identificados, fueron solucionados con los módulos que componen el sistema, los cuales se mencionan a continuación:

- Módulo de autenticación y de autorización: Este módulo garantiza la integridad del sistema al controlar quién tiene acceso y qué acciones

pueden realizar. Este módulo es fundamental para asegurar la seguridad de los datos y la protección de la información sensible, asegurando así la confidencialidad y la adecuada gestión de los permisos de los usuarios en el sistema.

- Módulo de registro de usuarios: Se realiza el registro de usuarios que interactuarán con el sistema.
- Módulo de registro de pacientes: Se realiza el registro y seguimiento a los pacientes dentro del sistema.
- Módulo de registro de citas: Este módulo registra una cita médica con la fecha, hora y un doctor asignado.
- Módulo de consulta: Este módulo realiza la atención de una cita que en términos médicos se denomina consulta, el medico realiza una consulta con los datos de un paciente.
- Módulo de historia clínica: Este módulo se encarga de dar acceso a la información de un paciente, según el nivel de acceso de acuerdo al rol del usuario.
- Módulo de reporte: Este módulo genera el reporte de la historia clínica de un paciente.
- Módulo de autenticación y de autorización: Este módulo garantiza la seguridad del sistema al verificar la identidad de los usuarios y controlar sus niveles de acceso a las funciones y datos del sistema.

Por tanto, se concluye que el presente proyecto fue cubierto y desarrollado satisfactoriamente, cumpliendo los requerimientos solicitados por el personal del instituto de genética. Y actualmente, el sistema está en pleno funcionamiento en la Unidad de Genética Médica.

6.3 Recomendaciones

Se puede incorporar un módulo de estadísticas con la capacidad de generar datasets. Estos conjuntos de datos resultarán útiles para facilitar procesos estadísticos que contribuirán significativamente al sistema.

Aunque se haya utilizado un gestor de base de datos relacional, se sugiere considerar en el futuro migrar hacia un modelo NoSQL debido a la gestión de grandes volúmenes de datos a largo plazo. Este cambio permitiría una mejor escalabilidad y flexibilidad para manejar la creciente cantidad de datos.

Se ha implementado el sistema en un servidor local ubicado dentro del edificio de la Facultad de Medicina. Se sugiere migrar hacia un entorno de servidor en la nube para evitar trámites burocráticos con la unidad responsable del servidor. Además, al optar por un servidor en la nube, se garantiza un acceso más ágil y a menor costo. Esto también conllevaría a tener el sistema independiente y aislado de los servidores de la Facultad de Medicina, ofreciendo una mayor autonomía y seguridad a la infraestructura del sistema.

Se ha observado que la combinación de las metodologías AUP (Proceso Unificado Ágil) y UWE (Entorno Web Unificado) ha sido beneficiosa para el desarrollo y diseño del sistema actual. Sin embargo, para futuras implementaciones, se recomienda explorar y adoptar metodologías contemporáneas adicionales como Scrum, DevOps, o Design Thinking, que están más alineadas con las tendencias actuales en el desarrollo y diseño de sistemas web. La inclusión de estas metodologías permitirá mantener el proceso de desarrollo ágil y adaptativo, abordando de manera más eficiente las demandas cambiantes del entorno tecnológico y las necesidades de los usuarios.

REFERENCIAS BIBLIOGRÁFICAS

- Ambler, S. (2002). The Agile Unified Process (AUP) Home Page. (s. f.). Ambyssoft Inc. <http://www.ambyssoft.com/unifiedprocess/agileUP.html> (Acceso: 30/06/2023) URL actualizada: <https://www.ic.unicamp.br/~ariadne/mc436/1s2017/Agile.html>
- Admin. (2023, 28 marzo). Metodologías de desarrollo software con ejemplos. DiagramasUML.com. <https://diagramasuml.com/metodologias-de-desarrollo-software-con-ejemplos/> (Acceso: 29/09/2023)
- Ambler, S. W. (2006). Agile UP Product. Ambyssoft Inc. May 13, 2006. <http://www.ambyssoft.com/unifiedprocess/agileUP.html> (Acceso: 13/08/2022)
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. <http://agilemanifesto.org/principles.html>
- Busch, M. (2016). Evaluating & Engineering: an Approach for the Development of Secure Web Applications. Fakultät für Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universität München. https://www.pst_ifi.lmu.de/~busch/thesisMarianneBusch.pdf
- Boehm, B. W. (2000). Software cost estimation with COCOMO II. Prentice-Hall PTR.
- Britton, J. (2021, 6 mayo). What is ISO 25010? Perforce Software. <https://www.perforce.com/blog/qac/what-is-iso-25010#:~:text=ISO25010%20describes%20two%20quality%20models,a%20particular%20context%20of%20use.>
- Chero-Farro, D., Cabanillas-Olivares, A., & Fernández-Mogollón, J. (2017). Historia clínica como herramienta para mejora del proceso de atención. Revista De Calidad Asistencial. <https://doi.org/10.1016/j.cali.2016.05.005>
- Chaudhry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., . . . Shekelle, P. G. (2006). Systematic review: Impact of health information technology on quality, efficiency, and costs of medical care. Annals of Internal Medicine, 144(10), 742-752. doi: 10.7326/0003-4819-144-10-200605160-00125 <https://pubmed.ncbi.nlm.nih.gov/16702590/>

- Calabrese, J., Muñoz, R., Pasini, A., Esponda, S., Boracchia, M., & Pesado, P. (2017). Asistente para la evaluación de características de calidad de producto de software propuestas por ISO/IEC 25010 basado en métricas definidas usando el enfoque GQM. Instituto de Investigación en Informática LIDI (III-LIDI) Facultad de Informática – Universidad Nacional de La Plata. http://sedici.unlp.edu.ar/bitstream/handle/10915/63778/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Clark, B., & Madachy, R. (Eds.). (2015). Software Cost Estimation Metrics Manual for Defense Systems. Haymarket, VA: Software Metrics Inc.
- Davis, R. G. (2018, 12 septiembre). Gestión de historias clínicas en el área de infectología de un hospital estatal, Lima, 2018. <https://repositorio.uwiener.edu.pe/handle/20.500.13053/2358> (Acceso: 05/04/2023)
- EcuRed. (s. f.). Modelo de prototipos - ECURED. Modelo de prototipos. https://www.ecured.cu/Modelo_de_prototipos (Acceso: 29/09/2023)
- EcuRed. (n.d.). Agile Unified Process - ECURED. https://www.ecured.cu/Agile_Unified_Process (Acceso: 02/10/2023)
- Escribano Santamarina, Á. (2010). Ingeniería web dirigida por modelos. Escuela Superior de Informática, Universidad de Castilla-La Mancha. http://www.inf-cr.uclm.es/www/mpolo/asig/trabajos0910/Ingenieria_web_basada_en_modelos.pdf (Acceso 30/10/2022).
- Fateh, D. (2024, 12 abril). How to start automated browser testing with Playwright. Contentful. <https://www.contentful.com/blog/playwright-automated-browser-testing/>
- Flores, G. P. (2018, 26 junio). Sistema web de administración de historias clínicas Caso: “centro médico quirúrgico Erzengel”. <https://repositorio.umsa.bo/xmlui/handle/123456789/16755>
- Galiano, L. (2012, octubre). Metodología UWE aplicada a mi solución informática de mi proyecto. Recuperado 8 de julio de 2023, de <http://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>
- Gil López, E., & Medinaceli Díaz, K. (2019). La protección del derecho a la intimidad en la historia clínica electrónica en Bolivia: una perspectiva para Latinoamérica. *Boletín mexicano de derecho comparado*, 52(154), 489-511

<https://pdfs.semanticscholar.org/04b9/f771e0883eaa2941daff20ad08ea1f3f2392.pdf>

Ghiglia, M. M. C. (2019). Historia clínica electrónica herramienta para la continuidad de asistencia. Revista médica del Uruguay.
<https://doi.org/10.29193/rmu.35.3.6>

Guzmán, F. & Arias, C. (2012). La historia clínica: elemento fundamental del acto médico. Revista Colombiana de Cirugía, vol.27(no.1),
http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=s2011-75822012000100002.

IBM. (s. f.). ¿Qué es PostgreSQL?. <https://www.ibm.com/mx-es/topics/postgresql>

ISO 25010. (s. f.). <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
(Acceso 22/10/2023)

Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G. (2006). Object-Oriented Software Engineering. A Use Case Driven Approach. Pearson.

Jácome, F. (2015, 2 septiembre). Repositorio Digital Universidad Israel: Desarrollo del Sistema de Administración de Historias Clínicas del Departamento Médico de la Empresa Imprenta Mariscal Cia.Ltda.
<http://repositorio.uisrael.edu.ec/handle/47000/584>

Juliá, S. (2020, 23 septiembre). Cómo gestionar historias clínicas de pacientes. Nubimed. <https://www.nubimed.com/blog/como-gestionar-historias-clinicas-pacientes/> (Acceso: 05/04/2023)

QAlified. (2022, 28 noviembre). Pruebas End-To-End: definición, ejemplos y herramientas. <https://qalified.com/es/blog/end-to-end-testing/#:~:text=Las%20pruebas%20end-to-end%20se%20pueden%20definir%20simplemente%20como,usuario%20de%20principio%20a%20fin.>

Luna Orosco, J., & Carrasco, M. (2008). Norma Técnica para el Manejo del Expediente Clínico. Dr. Javier Luna Orosco.
https://www.minsalud.gob.bo/images/Documentacion/dgss/Area_de_Calidad/64%20Norma%20expediente%20clinico.pdf

Madachy, R. (s. f.). COCOMO II - Constructive Cost model.
<http://softwarecost.org/tools/COCOMO/>

MDN. (2023, 24 julio). MDN Web Docs.
<https://developer.mozilla.org/es/docs/Web/JavaScript>

Migani, S. (2014). UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE. *unsj.*
https://www.academia.edu/4853589/UN_MODELO_DE_ESTIMACION_DE_PROYECTOS_DE_SOFTWARE

Nieves-Guerrero, C., Ucán-Pech, J., & Domínguez, V. H. M. (2014). UWE en sistema de recomendación de objetos de aprendizaje. Aplicando Ingeniería Web: un método en caso de estudio. *Revista latinoamericana de ingeniería de software*, 2(3), 137. <https://doi.org/10.18294/relais.2014.137-143>

Nora Koch. (2016). ¿Qué es UWE? UWE - Tutorial.
https://uwe.pst_ifi.lmu.de/teachingTutorial.html (Acceso: 30/06/2023)

Organización Mundial de la Salud. (2013). Sistemas de información de salud. Recuperado el 18 de febrero de 2023, de https://www.who.int/health-topics/health-systems#tab=tab_2

Ortiz, O. L. E. (2022, 6 febrero). Repositorio Digital - EPN: Desarrollo de un sistema web de gestión de historias clínicas en un consultorio privado de medicina general. <https://bibdigital.epn.edu.ec:443/handle/15000/22122>

Padilla, P. J. (2018, 23 enero). Repositorio Institucional Unicordoba: Diseño e implementación de un sistema de administración y consulta de historias clínicas electrónicas (HCE) mediante el uso de tecnología Webservices en diversos entes de salud del municipio de Santa Cruz de Lorica - Córdoba. <https://repositorio.unicordoba.edu.co/handle/ucordoba/482>

Paz, G. C. A. (2020, 4 noviembre). *Repositorio Institucional UPEA: SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS.*
<http://repositorio.upea.bo/handle/123456789/67>

Pinzon, O. (2017). Ingeniería Web: Una Metodología para el Desarrollo de Aplicaciones Web Escalables y Sostenibles. LACCEI. https://www.laccei.org/LACCEI2017-BocaRaton/student_Papers/SP277.pdf (Acceso: 08/03/2023).

Ponce, Y. Y. A. (2016, 5 octubre). Software como servicio para la administración de historias clínicas. <https://repositorio.umsa.bo/xmlui/handle/123456789/8179>

Pressman, R. S. (2002). INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO (5o). McGraw-Hill/Interamericana de España, S. A. U.

<https://pdfcoffee.com/qdownload/ingenieria-del-software-5ta-edicion-roger-s-pressman-freelibrosorgpdf-6-pdf-free.html> (Acceso: 08/03/2023)

Pressman, R. S. (2010). INGENIERIA DE SOFTWARE (7o edición revisada). McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V. A Subsidiary of The McGraw-Hill Companies, Inc.

Rotta, D. P., Pallotta, G. S., Klikailo, H. E., & Belloni, E. A. (2016). Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web. 19o Concurso de Trabajos Estudiantiles. Recuperado 9 de junio de 2023, de http://sedici.unlp.edu.ar/bitstream/handle/10915/58143/Documento_completo.pdf-PDFA.pdf?sequence=1

Sarsanadas Castellano, E. (s. f.). Sistema de información clínico: historia clínica. https://openaccess.uoc.edu/bitstream/10609/139235/2/Documentación%20clínica_Módulo%202_Sistema%20de%20información%20clínico_%20historia%20clínica.pdf (Acceso: 02/04/2023)

Schwaber, K., & Sutherland, J. (2017). The Scrum Guide.

Sequelize. (2024 ,12 abril). Sequelize v6. <https://sequelize.org/docs/v6/#:~:text=Sequelize%20is%20a%20promise-based,loading%2C%20read%20replication%20and%20more>.

Singh, A. (2023, 29 agosto). “Node.js Database Magic: Exploring the Powers of Sequelize ORM”. https://medium.com/@rishi_2701/node-js-database-magic-exploring-the-powers-of-sequelize-orm-a22a521d9d9d

Torres Lloveras, P., & Sarsanadas Castellanos, E. (s. f.). La historia clínica (HC). https://openaccess.uoc.edu/bitstream/10609/70507/5/Documentación%20clínica_Módulo%202_%20La%20historia%20clínica%20%28HC%29.pdf (Acceso: 05/04/2023)

Vargas, M. S. (2017, 30 abril). INGENIERIA WEB GUIADA POR MODELOS. <http://marcelosalasvargas.blogspot.com/2017/04/ingenieria-web-guiada-por-modelos.html> (Acceso: 09/03/2023)

Velito, A. S., & Tejada Soriano, S. R. (2010). La historia clínica como instrumento de calidad. <http://www.auditoriamedicaohoy.com/biblioteca/La%20historia%20clínica%20como%20instrumento%20de%20calidad%20Tejada%20Velito.pdf> (Acceso: 05/04/2023)

Webman. (2024, 10 marzo). ¿Qué es React? Trebol-IT. <https://www.trebol-it.com/que-es-react/>

Wikipedia. (2024, February 18). Node.js. Wikipedia, La Enciclopedia Libre. <https://es.wikipedia.org/wiki/Node.js>

World Health Organization. Regional Office for Europe. (2019, 10 de octubre). Future of digital health systems: report on the WHO symposium on the future of digital health systems in the European region: Copenhagen, Denmark, 6–8 de Ferbrero de 2019. <https://apps.who.int/iris/handle/10665/329032>

Zhou, L., Soran, C. S., Jenter, C. A., Volk, L. A., Orav, E. J., & Bates, D. W. (2015). The relationship. <https://europepmc.org/article/pmc/pmc2705247>

ANEXOS

Dow 1
2

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE MEDICINA, ENFERMERIA, NUTRICIÓN Y TECNOLOGÍA MÉDICA
INSTITUTO DE GENÉTICA

PARA PORTADORES O SOSPECHOSOS DE SÍNDROME DE DOWN (HC 1/IG)

I. IDENTIFICACIÓN:

Nombre: _____ Fecha de Nacimiento: ____/____/____ Edad: _____
 Sexo: M ♂ F ♀ Dirección: _____ Fecha: ____/____/____
 Teléfono: _____ REGISTRO IG: _____

II. MOTIVO DE CONSULTA:

III. ANTECEDENTES FAMILIARES:

Padre: Nombre _____ Edad: _____ Ocupación: _____
 Edad cuando nació el propósito _____ Obs: _____
 Madre: Nombre _____ Edad: _____ Ocupación: _____
 Edad cuando nació el propósito _____ Obs: _____
 Gestación _____ Para _____ AB _____ Cesárea _____ Malformados _____
 Nº nativos _____ Nº natimortos _____ Obs: _____
 Uso de anticonceptivos: SI NO Tipo: _____
 Período de uso _____ Tiempo de uso _____
 Abuela materna: Edad cuando nació la madre _____

IV. ANTECEDENTES PRENATALES:

| Agente | Nombre, vía, dosis | Periconcep. | EXPOSICIÓN A TERATÓGENOS | | | Observaciones |
|---------------|--------------------|-------------|--------------------------|---------|---------|---------------|
| | | | 1º trim | 2º trim | 3º trim | |
| Biológico | | | | | | |
| Químico | | | | | | |
| Físico | | | | | | |
| Fármacos | | | | | | |
| Enf. agudas | | | | | | |
| Enf. Crónicas | | | | | | |
| Otros | | | | | | |

Observaciones: _____

Ecografías prenatales: Si No Semana (mes, trimestre) de realización: _____
 Translucencia nucal: _____ Hueso nasal: _____ Otros hallazgos: _____

HISTORIA CLÍNICA : PACIENTES AMBULATORIOS DE GENÉTICA MÉDICAI. FILIACIÓN:

Nombre Suz María Muñoz F.Nac. / / Sexo: Raza
 F. de Consulta / / Procedencia _____ Residencia _____
 Dirección _____ Tel. _____

II. ANTECEDENTES FAMILIARES:

Padre _____ F.Nac. / / R Prof. _____
 Madre _____ F.Nac. / / R Prof. _____
 Edad materna cuando nació el propósito _____ del padre _____
 Edad de la abuela cuando nació la madre _____

III. MOTIVO DE CONSULTA:IV. ANTECEDENTES PERSONALES:

1. Gestación - a término pre-término post-término Obs. _____
2. Patologías en el embarazo - Fiebre Enf.infec. Diabetes Epilepsia
 otras enfermedades crónicas _____
3. Factores físicos durante el embarazo - Rayos X: Si No Ecografía: Si No
 Lugar donde se realizó _____ N° de veces _____
4. Factores químicos durante la gestación
 Fármacos anticonvulsivantes: _____ otros _____
 Anticonceptivos orales _____ Gestágenos para evitar AB. _____
 Exposición Profesional _____
- Enolismo _____
5. Gestas _____ Para _____ N° Nativivos _____
 Malformados _____ N° Nativmortos _____ N° AB. _____ Exp. _____
 Ind. de uso de anticonceptivos: Si No Tipo: _____
 Período entre 1^{ra} y 2^{da} gestación: _____ 2^{da} y 3^{ta}: _____ 3^{ta} y 4^{ta}: _____
 Período de uso _____
6. Parto eutóxico Cesárea Forceps ¿Por qué? _____
7. Datos del nacimiento: Peso: _____ Talla: _____ PC: _____ APGAR: _____
 Llanto - Inmediato Tardio No sabe Cianosis - Si No No sabe
 Oxígeno terapia - Si No No sabe Incubadora - Si No No sabe
 Ictericia Neonatal - Si No No sabe Fototerapia - Si No
 Exsanguíneo - Si No Fiebre Convulsiones
 Hemorragias - Si No No sabe Inicio: _____ Duración: _____
 Otras alteraciones: Criptorquidea Cardiopatía Congénita FLAP Atresia anal
 Defectos del tubo Neural Obs. _____

Personalizado

LOGO I.G.H. LOCALIDAD RAZA PACIENTES DOWN OTRAS ENFERMEDADES TOTALES DE PACIENTES TOTALES DE DOWN

| DATOS PERSONALES | | Foto | | | | |
|--|---|--|--|---------------------------------------|---------------------------------------|--|
| Registro | 4127 B1 C1 | | | | | |
| Nombre | Gladis | | | | | |
| Apellido Paterno | MONTANO | | | | | |
| Apellido Materno | Lopez | | | | | |
| Sexo | Masculino | | | | | |
| Raza | Blanca | | | | | |
| Fecha Nacimiento | 24-dic-62 | | | | | |
| Edad | 43 Año(s) | | | | | |
| | Días Meses Años | | | | | |
| Procedencia | Oruro | | | | | |
| Residencia | La Paz | | | | | |
| Dirección | Av. 6 de agosto #2549 Edif El Carmán 4B | | | | | |
| Teléfono | 2434022 | | | | | |
| <input checked="" type="checkbox"/> Enfermedad de Down | | | | | | |
| <input type="button" value="Agregar nuevo registro"/> | | <input type="button" value="Eliminar registro"/> | | | | |
| <input type="button" value="Primer"/> | | <input type="button" value="Anterior"/> | <input type="button" value="Siguiente"/> | <input type="button" value="Último"/> | <input type="button" value="Buscar"/> | |
| Registro: 14 1 de 648 > >> Sin filtrar Buscar | | | | | | |

Anexo 3 Archivo Access

DOCUMENTACIÓN