

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

SISTEMA WEB DE ADMINISTRACION DE HISTORIAS CLÍNICAS

CASO: (HOSPITAL DE LA MUJER)

Para optar al título de Licenciado en Ingeniería de Sistemas

Mención: INFORMATICA Y COMUNICACIONES

Postulante: Cesar Ariel Gonzales Paz

Tutor Metodológico: MSc. Ing. Enrique Flores Baltazar

Tutor Especialista: Ing. Cesar Ríos Camacho

Tutor Revisor: Ing. Elías Carlos Hidalgo Mamani

EL ALTO – BOLIVIA

2020

AGRADECIMIENTO

Agradezco a la Universidad Pública de El Alto, Carrera Ingeniería de Sistemas.

A mi familia, en especial a mis Padres, Abuelo y mi pequeño hermano, 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 Ing. Enrique Flores Baltazar, por dar su tiempo y paciencia en la revisión del proyecto.

Al Ing. Elías Carlos Hidalgo Mamani por la colaboración prestada en las etapas del proyecto como tutor revisor.

Al personal administrativo por su orientación en la entrega de la documentación.

Al Ing. Cesar Ríos Camacho, quien me brindo su tiempo y cooperación.

Al plantel médico y administrativo del Hospital de la Mujer, por permitirme realizar este proyecto y hacerlo realidad.

A mis amigos y compañeros por darme consejos y aliento para salir adelante.

Por sobre todo agradecer a Dios, quien ilumino y me dio su guía en este gran proceso de formación

Muchas Gracias.

Dedicatoria

A mis padres Mario y Nancy, a mi abuelo Alberto y a mi pequeño hermano Rodrigo, que me brindaron todo su apoyo incondicional día tras día y en todo momento.

A toda mi familia y amigos que siempre están ahí brindándome su ayuda y consejos.

RESUMEN

El presente proyecto de grado denominado “Sistema Web de administración de Historias Clínicas”, permite almacenar los datos relevantes del paciente, ha sido desarrollado para los Consultorios dependientes del Hospital de la Mujer, cuya actividad principal es brindar atención médica eficaz y eficiente.

El propósito del proyecto, es la realización de la transición de la gestión de la información de la Historia Clínica tradicional del paciente, con la sustitución de un sistema informático, que permita almacenar y procesar cantidades de datos, además de servicios propios del Hospital.

El proceso de investigación que se utilizó para el desarrollo del proyecto es la investigación científica, explicativa, experimental y metodológica.

El sistema contempla procesos de Admisión de Pacientes, Asignación de Atención, Consulta Médica, Historiales clínicos, Registro de Enfermería, Registro Medico y Reportes.

En la implementación se utilizó como gestor de base de datos María DB, el lenguaje de programación PHP con tecnologías, Ajax, java script, servidor XAMPP).

Finalmente, mediante el análisis de resultados se determina que el sistema desarrollado cumple con los objetivos planteados.

Palabras Clave: Historia, datos, investigación, gestión, metodológica, procesos, implementación.

ABSTRACT

The present degree project called "Web System for the administration of Clinical Histories", allows the storage of relevant patient data, has been developed for the offices of the Women's Hospital, whose main activity is to provide effective and efficient medical care.

The purpose of the project is to carry out the transition of the information management of the traditional Clinical History of the patient, with the replacement of a computer system that allows storing and processing amounts of data, in addition to the Hospital's own services.

The research process that was used for the development of the project is scientific, explanatory, experimental and methodological research.

The system includes processes for Patient Admission, Attention Assignment, Medical Consultation, Medical Records, Nursing Registry, Medical Registry and Reports.

In the implementation, Maria DB, the PHP programming language with technologies, Ajax, java script, XAMPP server, was used as the database manager).

Finally, through the analysis of results, it is determined that the developed system meets the stated objectives.

Key Words: History, data, research, management, methodological, processes, implementation.

INDICE DE CONTENIDO GENERAL

CAPITULO 1	1
1. MARCO PRELIMINAR	1
1.1. INTRODUCCION.....	1
1.2. ANTECEDENTES.....	4
1.2.1. Antecedentes de la institución.....	4
1.2.2. Trabajos afines al proyecto Internacional.....	5
1.3. PLANTEAMIENTO DEL PROBLEMA.....	7
1.3.1. Problema principal.....	8
1.3.2. Problemas secundarios.....	8
1.3.3. Formulación del problema.....	8
1.4. OBJETIVOS.....	9
1.4.1. Objetivo general	9
1.4.2. Objetivo específico.....	9
1.5. JUSTIFICACIONES	10
1.5.1. Justificación técnica	10
1.5.2. Justificación económica	10
1.5.3. Justificación social.....	10
1.6. METODOLOGÍA DE DESARROLLO.....	11
1.6.1. Metodología SCRUM	11
1.6.2. Metodología UWE	11
1.6.3. Modelo de costos	13
1.6.4. Calidad de Software.....	14
1.6.5. Pruebas de software	15
1.6.6. Técnicas de recopilación de datos	16
1.7. LÍMITES Y ALCANCES	17
1.7.1. Límites.....	17
1.7.2. Alcances.....	17
1.8. APORTES.....	18
1.8.1. Institucional	18

1.8.2. Académico	18
.....	19
MARCO TEORICO	19
CAPITULO 2.....	20
2. MARCO TEÓRICO.....	20
2.1. INTRODUCCIÓN.....	20
2.1.1. Dato.....	20
2.1.2. Información	23
2.1.3. Sistema	26
2.1.4. Sistema de información.....	32
2.1.5. Ingeniería web.....	38
2.1.6. Aplicación web	39
2.1.7. Información clínica	40
2.1.8. Historia clínica.....	40
2.1.9. Atención de salud.....	42
2.1.10. Paciente o usuario en salud	42
2.1.11. Control.....	42
2.1.12. Autenticar	43
2.2. INGENIERIA DE SOFTWARE.....	43
2.2.1. Metodologías de desarrollo del software.....	45
2.2.2. Arquitectura de software	50
2.2.3. Seguridad de software	62
2.2.4. Calidad de software.....	64
2.3. HERRAMIENTAS	71
2.4. MÉTRICAS DE CALIDAD.....	74
2.4.1. Familia de estándares ISO/IEC 9126.....	75
2.4.2. Estándar ISO/IEC 9126 – 1	75
2.4.3. Modelos de referencia.....	76
2.4.4. Modelos de calidad	77
2.5. MODELO DE COSTOS DE SOFTWARE.....	87

2.5.1. Modelo COCOMO II	87
.....	100
CAPITULO 3.....	101
3. MARCO APLICATIVO	101
3.1. INTRODUCCION.....	101
3.2. PRE-GAME.....	101
3.2.1. Concepción	101
3.2.2. Indagación y elaboración - PRODUCTO BACKLOG.....	102
3.2.3. Negociación	104
3.2.4. Especificación	105
3.3. GAME	109
3.3.1. Sprint 1: Acceso al Sistema	109
3.3.2. Sprint 2: Registro.....	113
3.3.3. Sprint 3: Orden de Atención	119
3.3.4. Sprint 4: Historias Clínicas	124
3.3.5. Sprint 5: Resultados	129
3.3.6. Sprint 6: Gestión de Servicios	134
3.3.7. Sprint 7: Reportes	139
3.3.8. Sprint 8: Gestión de Usuarios	145
3.4. POST-GAME	149
3.4.1. Pruebas.....	149
CAPITULO 4.....	152
4. METRICAS DE CALIDAD	152
4.1. INTRODUCCION.....	152
4.2. CARACTERISTICAS PROPUESTAS POR LA ISO 9126.....	152
4.2.1. Funcionalidad.....	152
4.2.2. Confiabilidad	156
4.2.3. Usabilidad	158
4.2.4. Mantenibilidad	158
4.2.5. Portabilidad	159

4.3. METODOLOGIA WEB SITE QEM.....	160
4.3.1. Criterio de preferencia de calidad elemental	163
4.3.2. Especificación de atributos.....	163
4.3.3. Definición e implementación de la evaluación elemental	164
4.3.4. Operadores de LSP para el modelado de relaciones lógicas.....	167
CAPITULO 5.....	170
5. EVALUACION DE COSTOS Y BENEFICIOS	170
5.1. INTRODUCCION.....	170
5.2. ANALISIS DE COSTOS	170
5.2.1. Estimación con el método COCOMO.....	170
5.2.2. Estimación método VAN Y TIR	174
CAPITULO 6.....	178
6.1. CONCLUSIONES Y RECOMENDACIONES.....	178
6.2. RECOMENDACIONES.....	178
ANEXOS.....	183

INDICE DE TABLAS

CAPITULO 2

Tabla 2. 1 Familia del estándar ISO/IEC.....	75
Tabla 2. 2 Características y Subcaracterísticas	77
Tabla 2. 3 Criterios de Evaluación	86
Tabla 2. 4 Productividad Modelo.....	89
Tabla 2. 5 Clasificación de puntos Objeto.....	94
Tabla 2. 6 Peso de un Punto Objeto	94

CAPITULO 3

Tabla 3. 1 Tareas para la obtención de requisitos	101
Tabla 3. 2 Requisitos Funcionales	102
Tabla 3. 3 Requisitos No Funcionales	103
Tabla 3. 4 Tabla de Iteraciones.....	104
Tabla 3. 5 Actores del Sistema	106
Tabla 3. 6 Planificación del Sprint 1	109
Tabla 3. 7 Sprint Backlog Acceso al Sistema	111
Tabla 3. 8 Validar Información del Usuario	112
Tabla 3. 9 Generar menú según Rol.....	112
Tabla 3. 10 Planificación del Sprint 2.....	113
Tabla 3. 11 Sprint Backlog Registro.....	116
Tabla 3. 12 Registro de Pacientes	116
Tabla 3. 13 Generar menú según Rol.....	117
Tabla 3. 14 Planificación del Sprint 3.....	119
Tabla 3. 15 Sprint Backlog Atención	121
Tabla 3. 16 Nueva Orden.....	122
Tabla 3. 17 Planificación del Sprint 4.....	124
Tabla 3. 18 Sprint Backlog Registro Historial.....	126
Tabla 3. 19 Registro Historial Clínico.....	127

Tabla 3. 20 Planificación del Sprint 5.....	129
Tabla 3. 21 Sprint Backlog Resultados.....	131
Tabla 3. 22 Buscar Resultados.....	132
Tabla 3. 23 Generar PDF de Resultado.....	132
Tabla 3. 24 Planificación del Sprint 6.....	134
Tabla 3. 25 Sprint Backlog Servicios.....	137
Tabla 3. 26 Adicionar Servicio.....	137
Tabla 3. 27 Modificar Servicio.....	138
Tabla 3. 28 Eliminar Servicio.....	138
Tabla 3. 29 Planificación del Sprint 7.....	140
Tabla 3. 30 Sprint Backlog Reportes.....	142
Tabla 3. 31 Lista de Usuarios.....	142
Tabla 3. 32 Lista de Pacientes.....	143
Tabla 3. 33 Lista de Servicios.....	143
Tabla 3. 34 Planificación del Sprint 8.....	145
Tabla 3. 35 Sprint Backlog Usuarios.....	147
Tabla 3. 36 Visualizar información de Usuarios.....	147
Tabla 3. 37 Habilitar o deshabilitar Usuarios.....	147
Tabla 3. 38 Prueba de Registro de Pacientes.....	149
Tabla 3. 39 Prueba de aceptación de Información.....	150
Tabla 3. 40 Prueba de aceptación de Reportes.....	150

CAPITULO 4

Tabla 4. 1 Síntesis para Hallar el punto de función.....	153
Tabla 4. 2 Datos - Ponderación Media.....	153
Tabla 4. 3 Niveles de Ponderación.....	154
Tabla 4. 4 Factores Escala y Valores.....	155
Tabla 4. 5 Componentes del Sistema.....	157
Tabla 4. 6 Resultados de medir la Usabilidad.....	158
Tabla 4. 7 Funcionamiento en Navegadores.....	160

Tabla 4. 8	Árbol de Requerimientos para Calidad	160
Tabla 4. 9	Resultados de Usabilidad	164
Tabla 4. 10	Resultados de Funcionalidad	165
Tabla 4. 11	Resultados Confiabilidad	165
Tabla 4. 12	Resultados de Eficiencia	166
Tabla 4. 13	Resultados de Portabilidad	166
Tabla 4. 14	Resultados de Mantenibilidad	166
Tabla 4. 15	Resultados Globales	167
CAPITULO 5		
Tabla 5. 1	Coeficiente COCOMO.....	171
Tabla 5. 2	Multiplicadores de Esfuerzo	172
Tabla 5. 3	Resumen Global	174
Tabla 5. 4	Inversión y Flujos Netos.....	175

INDICE DE FIGURAS

CAPITULO 2

Figura 2. 1 Ciclo de Procesamiento de Datos.....	22
Figura 2. 2 WebQem Usabilidad	81

CAPITULO 3

Figura 3. 1 Arquitectura del Sistema.....	105
Figura 3. 2 Diagrama Casos de uso Principal.....	107
Figura 3. 3 Modelo de Base de Datos del Sistema	108
Figura 3. 4 Diagrama UC Acceso al sistema	110
Figura 3. 5 Diagrama de Secuencia Acceso al Sistema	111
Figura 3. 6 Interfaz Gráfica Sprint 1	113
Figura 3. 7 Diagrama de Casos de uso Registro	114
Figura 3. 8 Diagrama de Secuencia Registro	115
Figura 3. 9 Interfaz gráfica Registro Paciente	118
Figura 3. 10 Interfaz de Usuario Registro	118
Figura 3. 11 Diagrama UC Orden de atención.....	120
Figura 3. 12 Diagrama de secuencia Orden de Laboratorio	121
Figura 3. 13 Interfaz Gráfica Nueva Orden	123
Figura 3. 14 Interfaz de Usuario Orden Pendiente.....	123
Figura 3. 15 Diagrama UC Historial Clínico	124
Figura 3. 16 Diagrama de secuencia de uso Historial Clínico.....	125
Figura 3. 17 Interfaz Gráfica Historia Clínica Enfermería.....	127
Figura 3. 18 Interfaz Gráfica Historia Clínica Médico Especialista.....	128
Figura 3. 19 Diagrama de Casos de Uso Resultados	130
Figura 3. 20 Diagrama de Secuencia Resultados.....	131
Figura 3. 21 Interfaz Gráfica 1 Sprint 4	133
Figura 3. 22 Reportes de Resultados	134
Figura 3. 23 Diagramas UC Gestión de Servicios.....	135

Figura 3. 24 Diagrama de secuencia Gestión de Servicios	136
Figura 3. 25 Interfaz Gráfica Sprint 6	139
Figura 3. 26 Interfaz Gráfica Sprint 6	139
Figura 3. 27 Diagrama UC Reportes.....	140
Figura 3. 28 Diagrama de Secuencia Reportes	141
Figura 3. 29 Reportes de Usuarios	144
Figura 3. 30 Reportes de Pacientes.....	144
Figura 3. 31 Reporte de Servicios.....	144
Figura 3. 32 Diagrama UC Gestión de Usuarios.....	145
Figura 3. 33 Diagrama de secuencia Gestión de Usuarios.....	146
Figura 3. 34 Interfaz Gráfica Sprint 8	148
CAPITULO 4	
Figura 4. 1 Características de Alto Nivel.....	167

MARCO PRELIMINAR

1. MARCO PRELIMINAR

1.1. INTRODUCCION

El escenario ecuménico actual está experimentando cambios y un crecimiento significativo en el uso de la tecnología, incorporando la informática a todas las actividades del que hacer del hombre en cuanto al manejo de información para el seguimiento de procesos, generando modelos y metodologías que permitan mejorar la productividad y alcanzar los objetivos de la organización.

En esta evolución constante en cuánto al desarrollo de sistemas informáticos, hace que las organizaciones y empresas de toda índole, opten por éstas para agilizar sus procesos de información y sean de gran consideración en la toma de decisiones, debiendo estar disponibles en cualquier momento de tal forma que sea la más oportuna, confiable y efectiva posible, para brindar un mejor servicio.

La informática aplicada a la medicina utiliza teorías, métodos y técnicas dirigidas a solucionar, describir y analizar problemas del área médica.

De forma continua los hospitales y centros de salud trabajan tras la búsqueda de soluciones que optimicen la atención de personas como una estrategia para ahorrar tiempo en actividades rutinarias asociadas con el manejo de datos en historias clínicas, que permitan mejorar los diagnósticos, el seguimiento e interacción con nuevos y antiguos pacientes para brindar información y atención de mejor calidad.

En dicho escenario se presenta como una herramienta perteneciente a las tecnologías de información, el desarrollo de sistemas web, que permitan un registro informatizado de los datos sociales, preventivos y médicos de un paciente, de modo que dichos informes estén disponibles para los profesionales médicos que trabajan en los centros de salud.

Los sistemas de información médica se organizan en conjuntos de elementos que interactúan para procesar los datos y la información de salud de las personas, con el objetivo de realizar un intercambio eficiente de información entre los actores del sistema de salud, buscando optimizar la toma de decisiones en todos los niveles (atención, planificación, gestión, etc.). Uno de estos elementos es la Historia Clínica Electrónica (HCE).

(Plazzotta Fernando 2016:6).

Una HCE debe conceptualizarse como el aplicativo que utilizan los miembros del equipo de Salud para registrar su quehacer asistencial; debe ser el lugar primario para la carga y consulta de toda la información clínica.

Existen muchos términos relacionados con el concepto de HCE, tales como Registro Médico Electrónico, Registro Médico Computarizado, Ficha Clínica Electrónica, Historia Clínica Digital, Informática o Informatizada, pudiendo todos ellos ser utilizados indistintamente.

En general, no se utiliza el término digital para diferenciarlo de la Historia Clínica Digitalizada (generada en papel y capturada digitalmente) y se prefiere el término Historia Clínica Electrónica, que es la denominación más utilizada internacionalmente y en la mayoría de los idiomas.

(Domínguez, 2002).

Una HCE es un registro de información de salud que reside en un sistema electrónico específicamente diseñado para recolectar, almacenar, manipular y dar soporte a los usuarios para el acceso a datos seguros y completos, alertas, recordatorios y sistemas clínicos de soporte para la toma de decisiones, brindando información clínica importante y pertinente para el cuidado de los pacientes.

El crecimiento exponencial de la población en la ciudad de La Paz, hizo recurrir a la necesidad de tener sistemas de control automatizado en el área de salud, en el cual ya no sea necesario el archivar las historias clínicas de los pacientes en folders donde se guarda la información importante sobre el estado de salud del paciente.

Los hospitales públicos y centros de salud a cargo del Servicio Departamental de Salud, están tomando conciencia de la importancia de estos sistemas de información electrónica, sin embargo, a medida del crecimiento poblacional, también crecen sus requerimientos y necesidades, se debe mantener actualizada la información para una óptima planificación, y cubrir con dichas necesidades.

El “Hospital de la Mujer” de la ciudad de La Paz, es una de tantas instituciones pertenecientes al Servicio Departamental de Salud que busca mejorar la atención y administración, mediante la automatización e integración de sus procesos, en cuanto al manejo de información asociada a los pacientes, principalmente en el registro, seguimiento médico y guías hospitalarias.

Para ello se desarrollará un sistema web para la administración de historias clínicas, en la cual se podrá almacenar toda la información de distintos pacientes, sin el temor que la información brindada por el paciente se pierda, más al contrario se guarde de manera segura.

La metodología ágil empleada para el desarrollo de este sistema de gestión documental ha sido Scrum, el cual es un método de gestión de proyectos basado en un proceso iterativo e incremental que puede adaptarse a cualquier tipo de proyecto, ya que es utilizado en proyectos de entorno complejos, donde se desea obtener resultados rápidos y la productividad es lo más importante.

1.2. ANTECEDENTES

1.2.1. Antecedentes de la institución

El “Hospital de la mujer” es la unidad de nivel desconcentrado perteneciente al Servicio Departamental de Salud (SEDES), ubicado en la ciudad de La Paz, dedicado a la atención de la población femenina en servicios de Ginecología, Obstetricia, Terapia Intensiva, Neonatología, Anestesiología y complementos como Ecografía, Farmacia, Laboratorio, Anatomía Patológica, Transfusión de Sangre y Derivados.

El Servicio Departamental de Salud fue creado el 1 de septiembre de 1965 con el nombre de “UNIDAD SANITARIA DE LA PAZ” a través del D.S. N.º 07299, bajo la gestión del Director de Fomento de la Salud y Salud Rural del Ministerio de Previsión Pública, Dr. Walter Julio Fortún.

El 8 de febrero de 1996 a través del D.S. 24237 La Unidad Sanitaria de La Paz cambió de denominación y pasó a ser “Dirección Departamental de Salud” (DIDES) y como una entidad desconcentrada del Ministerio de Salud.

A partir del 8 de febrero de 1996 en cumplimiento de la Ley 1654 de Descentralización Administrativa, la DIDES pasa a depender administrativamente y orgánicamente a la Prefectura del Departamento de La Paz.

El 2 de septiembre de 1997 a través del D.S. N.º 29833 la DIDES cambia de denominativo y pasa a ser, UNIDAD DEPARTAMENTAL DE SALUD (UDES).

Finalmente, por D.S. N.º 25060 de 2 de julio de 1998 pasa a ser SERVICIO DEPARTAMENTAL DE SALUD (SEDES).

MISION:

Atención y resolución de referencia Departamental y Nacional, que brinda servicios de salud especializada con calidad y calidez en las especialidades de Obstetricia, Ginecología, Neonatología y Medicina Critica, contribuyendo a

disminuir la morbi-mortalidad materna-neonatal mediante acciones de promoción de la salud, asistencia sanitaria oportuna y atención médica en función a RR.HH., calificados en el marco de la integración docente asistencial forma profesionales en salud tanto en pregrado como en postgrado.

VISIÓN:

Al 2025 el Hospital de la Mujer Acreditado de tercer nivel de referencia departamental y nacional, con alta capacidad resolutive y de respuesta en la atención a la demanda de usuarias que acuden para una resolución satisfactoria a sus problemas de salud, contando con suficientes RR.HH., infraestructura acorde al nivel de atención y equipamiento médico moderno necesario y asimismo promoviendo la investigación científica en beneficio de la comunidad y de toda la población usuaria.

1.2.2. Trabajos afines al proyecto Internacional

[Oscar David Doria Urango, 2015].

“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”.

El objetivo que persigue este proyecto es el diseñar e implementar 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 Salud del municipio de Santa Cruz de Lorica-Córdoba. Utilizando la metodología RUP, las herramientas que se usa en este proyecto son; los lenguajes como C, C++ en el entorno Visual Studio, además de HTML 5, JavaScript y CSS. Realizado en la Facultad de Ingenierías de la Universidad de Córdoba, Montería-Córdoba.

[Carlos Rey Gutarra Mejía, Roberto Carlos Quiroga Rosas, 2014].

“Implementación de un Sistema de Historias Clínicas Electrónicas para el Centro de Salud Perú 3ra Zona”.

El objetivo principal consiste implementar un Sistema de Historias Clínicas electrónicas en el Centro de Salud Perú 3ra Zona. Utilizando la metodología Scrum, las herramientas que se usó en el proyecto son Rational Rose 2003, Microsoft Windows 7, Ms Project 2010, para la programación Eclipse, Java, Servidor Web TOMCAT, Mysql Se desarrolló en la Facultad de Ingeniería y Arquitectura de la Universidad SAN Martín de Porres, Lima – Perú.

[Ricard Sabartés Fortuny, 2013].

“Historia Clínica Electrónica en un departamento de Obstetricia, Ginecología y Reproducción: Desarrollo e Implementación. Factores Clave”.

El objetivo principal consiste en analizar, diseñar, desarrollar e implementar una historia clínica Electrónica en un centro de Obstetricia, Ginecología y Reproducción a través de una correcta metodología, además de conocer los factores clave de un proceso de implementación de una Historia Clínica Electrónica. Utilizando la RUP, las herramientas que se uso en el proyecto son Netbeans, JavaScript, Servidor Web Apache, Oracle. Se desarrolló Facultad de Medicina de la Universidad Autónoma de Barcelona, ciudad de la Bellaterra-España.

Nacional

[Paola Gutierrez Flores, 2017].

“Sistema web de administración de Historias Clínicas Caso: Centro Médico Quirúrgico Erzangel”.

El objetivo es el desarrollar un sistema Web de administración de Historias Clínicas, que permita mejorar las tareas de registro, búsqueda y elaboración de reportes que se puedan realizar de manera rápida y eficiente, facilitando de

manera oportuna y precisa del registro de datos y manejo de la información en los consultorios al médico especialista. Utilizando la metodología SCRUM, las herramientas que se usó en el proyecto son el lenguaje de marcado HTML, Javascript, Bootstrap, Postgresql para el diseño, el servidor Web, PHP, Apache. Realizado en la carrera de Informática perteneciente a la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés, La Paz – Bolivia.

[Sara Patricia Huanca Cantuta, 2015].

“Sistema Web de control de Pagos, Citas e Historiales Clínicos Caso: Clínica Dental Lavadent”.

Cuyo objetivo general es el desarrollar un Sistema Web de Control de pagos, citas e historiales clínicos para la clínica dental Lavadent, que permita tener un buen manejo de información odontológica en constante actualización, y así garantizar la preferencia de antiguos y nuevos pacientes. Se utilizó la metodología Scrum-UWE. las herramientas necesarias son HTML, VRML, XML. Lenguajes de programación como java, php, jsp. Desarrollado en la carrera de Informática perteneciente a la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés, La Paz – Bolivia.

1.3. PLANTEAMIENTO DEL PROBLEMA

La ineficacia del manejo de historiales clínicos tradicionales de los pacientes en el Hospital de la Mujer, donde se almacenan de forma física anexando a estos los resultados, análisis y demás. Se debe a factores como la necesidad de repetir análisis por pérdida de resultados, la ausencia de documentación para un tratamiento prescrito, la no documentación de diagnósticos o por no estar disponible el historial del paciente a la hora de la consulta, o también la falta de seguridad y salvaguarda de la información.

1.3.1. Problema principal

Por la falta de un sistema de información para el manejo de Historial y seguimiento clínico, la labor del personal se hace cada vez más ardua debido al tiempo que lleva elaborarlas, ocasionando a la vez molestias y perjuicios en los pacientes asistentes.

1.3.2. Problemas secundarios

- La información de los pacientes al ser redactada manualmente, en algunos casos se encuentra en un lenguaje incomprensible en la historia clínica, lo cual trae dificultad al leer la misma que puede ser mal interpretada.
- Expedientes clínicos desactualizados y no disponibles en el momento preciso, debido a que la información del paciente no está bien clasificada, complicando así la obtención de datos al momento.
- El traslado continuo y recuperación de las carpetas que contienen información del paciente de un expediente clínico se realiza con lentitud, generando así un riesgo de dispersión, deterioro y pérdida tanto de información como de tiempo en la atención del paciente.
- Se puede tropezar con la existencia de múltiples expedientes clínicos referentes a un mismo paciente, teniendo un volumen masivo de papeles que ocupan espacio físico.
- La demora para ser atendidos y el mal manejo de la información va a generar un malestar en los pacientes.

1.3.3. Formulación del problema

Teniendo en cuenta lo anteriormente mencionado surge la pregunta problemática:

¿Cómo mejorar el proceso de administración de historias clínicas de los pacientes, de manera que se pueda agilizar el tiempo de registro, consulta, búsqueda, y proceso de emisión de reportes de manera eficiente?

1.4. OBJETIVOS

1.4.1. Objetivo general

Desarrollar un sistema web de administración de Historias Clínicas que permitan mejorar las tareas de registro, búsqueda y elaboración de reportes de manera rápida y eficiente, facilitando la entrega oportuna y precisa del registro de datos y manejo de la información en los consultorios del médico especialista, brindando una mejor atención a los pacientes.

1.4.2. Objetivo específico

- Establecer requerimientos de los usuarios.
- Centralizar la información a través de una base de datos, evitando así demoras en cuanto a su registro y actualización de información necesaria para cada médico.
- Desarrollar los módulos de registro del paciente, módulo consulta médica, módulo de diagnóstico, módulo de historial clínico, módulo de hoja de control de enfermería, módulo registro médico, módulo de reporte, módulo de internación.
- Definir autenticación de usuarios donde se asignará roles y niveles de acceso para los usuarios, con el fin de resguardar la información.
- Efectuar el registro de historial clínico, haciendo visible los datos personales de cada paciente, con el seguimiento del tipo de tratamiento según el caso.

1.5. JUSTIFICACIONES

1.5.1. Justificación técnica

El presente proyecto para el desarrollo de un Sistema Web de administración de Historia Clínica, para los consultorios del Hospital de la Mujer, se justifica porque el Departamento de Sistemas cuenta con los recursos informáticos y tecnológicos, como equipos completos (computadoras Pentium D y Core 2Duo), además de una infraestructura de redes que facilitarán el mantenimiento del sistema, el cual se desarrollará en un lenguaje que no requiere la compra de licencia, como PHP y un gestor de base de datos como MariaDB, disponible en un servidor que se ubicará dentro del centro corporativo, este servidor tendrá una gran capacidad de procesamiento y almacenamiento.

1.5.2. Justificación económica

El proyecto se justifica desde el punto de vista económico, porque el desarrollo y posible implementación del Sistema, reducirá gastos en material (formularios para historiales, sobres manila, libros de consultas y libros de enfermería), con lo que se ahorrará en costos. Además de contar con los recursos necesarios para la implementación del sistema, evitando un costo adicional.

1.5.3. Justificación social

El desarrollo del presente proyecto de grado, beneficiará al Hospital de la Mujer al contar con un sistema web, que Administre y Controle los Historiales Clínicos de la población femenina, reduciendo notablemente el tiempo de atención, para así mejorar las condiciones de trabajo y evitar las tareas manuales para el personal brindando una información veraz, rápida y oportuna a los médicos, enfermeras y pacientes.

1.6. METODOLOGÍA DE DESARROLLO

1.6.1. Metodología SCRUM

La metodología a emplear en este proyecto es la de SCRUM, éste es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de inicio para definir el proceso de desarrollo que se ejecutará durante el proyecto.

En dicho proceso se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto, SCRUM también se utiliza cuando es necesario identificar y solucionar ineficiencias.

Esta metodología de trabajo promueve la innovación, motivación, que forma parte de este proyecto, los beneficios que se tiene con dicha metodología son; el cumplimiento de las expectativas, flexibilidad en cambios, mayor calidad de software y reducción de riesgos.

Las características del proceso de SCRUM en el proyecto actual son:

- La primera y última fase que consiste en procesos definidos, se observará todos los procesos definidos para regularizar las expectativas del Hospital de la Mujer.
- Se desarrollará iteraciones llamadas "Sprint", donde el equipo de desarrollo decide que funcionalidad incluir o no, además de definir el tiempo necesario para terminar el proyecto.

1.6.2. Metodología UWE

Por otra parte, se trabajará con la metodología UWE, que está especializada en la especificación de aplicaciones adaptativas y por tanto hace hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas del usuario.

Siguiendo las fases que tiene la UWE tenemos:

- **Captura, análisis y especificación de requisitos**, que debe tener el sistema web para la administración de historias clínicas.
- **Diseño del Sistema**, donde es importante definir el cumplimiento de requisitos y la estructura que debe tener la aplicación web.
 - Diagrama de Casos de Uso.
 - Diagrama Conceptual.
 - Diagrama Físico.
 - Diagrama de Clases.
 - Modelo Navegacional.
 - Modelo de presentación.
- **Codificación del Software**, Durante esta etapa se realizan las tareas que se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- **Pruebas**, Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- **Instalación o fase de Implementación**, es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.
- **Mantenimiento**, Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

1.6.3. Modelo de costos

1.6.3.1. COCOMO II

Este es un modelo que permite estimar el costo, el esfuerzo, y el horario en la planificación de una nueva actividad de desarrollo de software.

COCOMO II es la última extensión importante a la COCOMO 81, está compuesto por tres modelos denominados:

- **Composición de Aplicación**, se emplea en desarrollos de software durante la etapa de prototipación.
- **Diseño Temprano**, se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto.
- **Post-Arquitectura**, se aplica en la etapa de desarrollo propiamente dicho, después que se define la arquitectura del sistema, y en la etapa de mantenimiento.

Este modelo utiliza:

- a) Puntos Función y/o Líneas de Código Fuente, para estimar tamaño, con modificadores que contemplan el reúso, con y sin traducción automática, y el "desperdicio" (breakage).
- b) Un conjunto de 17 atributos, denominados factores de costo, que permiten considerar características del proyecto referentes al personal, plataforma de desarrollo, etc., que tienen injerencia en los costos.
- c) Cinco factores que determinan un exponente, Estos factores reemplazan los modos Orgánico, Semi acoplado y Empotrado del modelo COCOMO '81.

CARACTERÍSTICAS:

- Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.
- Representa el más extenso modelo empírico para la estimación de software.
- Existen herramientas automáticas que estiman costos basados en COCOMO como ser: Costar, COCOMO 81.

OBJETIVOS:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de hoy como en el futuro.
- 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 evaluarán el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

1.6.4. Calidad de Software

1.6.4.1. Metodología WebQem

Se utilizará esta metodología para medir la calidad del sistema.

Luis Olsina publicó el Estándar internacional para la evaluación de la Calidad de los productos de Software el año 1999, denominado "Metodología cuantitativa para la Evaluación y Comparación de la calidad de Sitios Web" de manera

general se la denominó “Web-Site QEM”, esta metodología está basada en el ISO 9126 que contempla un modelo jerárquico de requerimientos de calidad. Esta metodología tiene a su descripción un conjunto de fases, actividades, productos y modelos.

1.6.5. Pruebas de software

1.6.5.1. Pruebas de caja negra

En los estándares para Software Testing definidos por ISTQB (International Software Testing Qualifications Board), las técnicas de pruebas de caja negra son utilizadas para realizar pruebas funcionales, basadas en las funciones o características del sistema y su interacción con otros sistemas o componentes. Se pueden utilizar técnicas basadas en especificación para identificar las condiciones y casos de prueba a partir de la funcionalidad del software, como es el caso de la Derivación o Extracción de Casos de Prueba a partir del Caso de Uso (ó Historia de Usuario).

Las técnicas de caja negra también pueden ser utilizadas para diseñar pruebas de software no funcionales.

1.6.5.2. Técnicas de Pruebas de Caja Negra

- Partición de equivalencias
- Análisis de valores borde
- Tablas de decisión
- Transición entre estados
- Pruebas de casos de uso

1.6.6. Técnicas de recopilación de datos

Entrevistas

Junto con las mediciones, éste es quizá el método de acopio de datos más importante. En la práctica, las entrevistas pueden ayudar a obtener información espacial sobre cualquier función de producción.

Este método funciona mejor si se encuentran expertos locales que puedan evaluar la fiabilidad de las respuestas y establecer una relación con las personas entrevistadas. Las entrevistas resultan particularmente valiosas cuando consiguen sacar a relucir aspectos de la información espacial que no estaban previstos originalmente, pero que terminan siendo importantes para el resultado del estudio.

Cuestionarios

Si bien los cuestionarios sólo ayudan a evaluar un número limitado de funciones de producción, a veces constituyen el único medio para averiguar ciertos datos importantes. Las preferencias alimentarias locales se pueden establecer de esta manera, los usuarios del hospital pueden entregar sus opiniones sobre la importancia relativa de cada función del sistema para el éxito del software.

Observaciones

La observación es otra técnica útil para el analista en su proceso de investigación, consiste en observar a las personas cuando efectúan su trabajo.

La observación es una técnica que consiste en observar hechos durante la cual, el analista participa activamente, actúa como espectador de las actividades llevadas a cabo por una persona para conocer mejor su sistema.

El propósito de la observación es múltiple, permite al analista determinar que se está haciendo, como se está haciendo, quien lo hace, cuando se lleva a cabo, cuánto tiempo toma, donde se hace y porque se hace.

1.7. LÍMITES Y ALCANCES

1.7.1. Límites

Los límites del sistema web para la administración de historias clínicas en el Hospital de la Mujer son:

- Las interfaces del sistema estarán restringidas a los tipos de usuarios autorizados en el sistema.
- Otra limitante abocada más a los servicios de web de cada navegador, el sistema web no controlará la compatibilidad o ausencia de plugins o complementos necesarios para la funcionalidad que proporcione el sistema.
- El sistema web no podrá ser accedido por los pacientes a través de internet por las características que se plantea obtener.
- No se tomará en cuenta la parte contable, ya que esta se centrará solo en el registro del paciente.

1.7.2. Alcances

Es necesario delimitar los alcances del proyecto, para lo cual se debe especificar el ámbito de acción del mismo.

- El trabajo implica en desarrollar un sistema web de administración de historias clínicas, en el cual se tendrá almacenado los datos generales, estudios realizados a los pacientes.
- Los módulos que se implementarán son:
 - Módulo de registro del paciente
 - Módulo de hoja de control de enfermería
 - Módulo registro médico
 - Módulo de reporte
 - Módulo de internación.

- El sistema manejará una arquitectura cliente-servidor, la cual requerirá configuración de equipos servidores y clientes dentro de la institución.
- La obtención de la información la realizara mediante la autorización del personal a cargo.

1.8. APORTES

1.8.1. Institucional

El sistema web de gestión de información en cuanto a la administración de historias clínicas de los pacientes, logrará aportes al Hospital de la Mujer en términos de:

- Agilización de obtención de información en cuanto al control médico de los pacientes, tratamiento y cita médica.
- Brindar información actual tanto en los servicios que brinda el Hospital de la Mujer, como la de los pacientes.
- Evitará pérdida de tiempo al buscar historiales clínicos por paciente, y de esta forma el hospital conservará a los pacientes antiguos como a los nuevos.

1.8.2. Académico

Se realiza el aporte con un perfil de proyecto de grado, el cual va a tener como objetivo el de mostrar el conocimiento adquirido en la Carrera de Ingeniería de Sistemas, el uso de lenguajes de programación para la creación y desarrollo de un Sistema Web para la Administración de Historias Clínicas.

MARCO TEORICO

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

En este capítulo se expondrá un conjunto de conceptos y definiciones que se tomarán en cuenta para lograr los objetivos propuestos para el desarrollo del proyecto, donde los elementos teóricos están extraídos de varias fuentes por lo tanto constituyen la base para la descripción y explicación del problema planteado.

2.1.1. Dato

Son el conjunto básico de hechos referentes a una persona, cosa o transacción. Incluyen cosas como: tamaño, cantidad, descripción, volumen, tasa, nombre o lugar. (Murdick: pág 157).

Se puede definir los datos como aquella información extraída de la realidad que tiene que ser registrada en algún soporte físico o simbólico, que implica una elaboración conceptual y además que se pueda expresar a través de alguna forma de lenguaje. (Gil Flores, 1994).

Un dato es la simple representación de una variable que indica un valor que se le asigna a las cosas. No pueden demostrar demasiado, siempre se evalúa el conjunto organizado y tabulado para poder examinar los resultados.

Podemos mencionar dos tipos de datos:

- **Datos cuantitativos:** son aquellos que se pueden contar o medir
- **Datos cualitativos:** son aquellos que únicamente pueden describirse

2.1.1.1. Operaciones básicas sobre los datos

- **Captura (Captación):** Es la obtención de datos antes de ser procesados o almacenados. La captura puede ser manual (formatos, documentos) o mecanizada a través de dispositivos electrónicos como: teclado, Mouse, lápiz óptico, dispositivo de reconocimiento de voz.

Según Burch John Esta operación se refiere al registro de datos hecho a partir de un evento o acontecimiento, en forma de notas de ventas, nóminas de personal, órdenes de compra, etc.

Segun Murdick: Es el proceso de recolección de datos, puede tener lugar en virtud de una transacción interna o de un evento externo a la compañía.

- **Validación (Verificación):** Proceso de verificación y corrección de datos durante la captura o después, con la finalidad de minimizar el número de errores.
- **Almacenamiento:** Consiste en guardar los datos previamente capturados en un medio de almacenamiento como: disco duro o diskette.
- **Recuperación:** proceso mediante el cual se logra acceder, escoger y extraer datos almacenados.
- **Reproducción:** Duplicación de los datos o información para el traslado de los mismos de un lugar a otro.

2.1.1.2. Procesamiento de datos

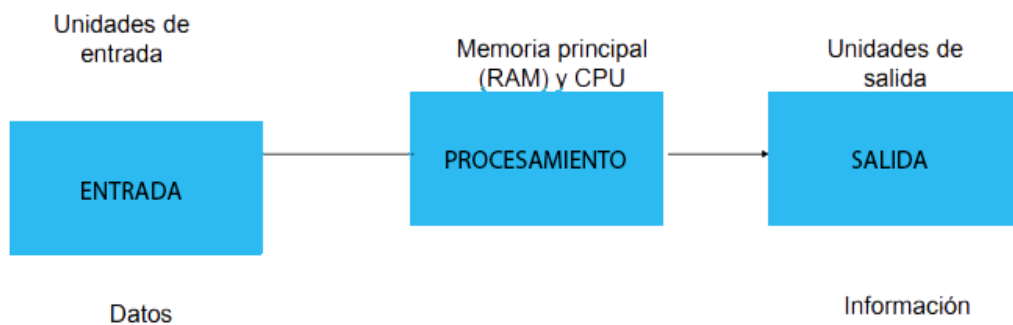
Son operaciones que se realizan con determinados datos para producir información según grupos específicos de reglas.

- **TIPOS**
 - **En línea:** ocurre de manera inmediata. El tiempo de respuesta es mínimo.
 - **En lote:** los datos y transacciones se codifican y reúnen en grupos (lotes) antes de ser procesados.

2.1.1.3. Ciclo de procesamiento de datos

- **Entradas:** Acto de transferir datos desde una unidad de entrada (dispositivo periférico de entrada) hacia la memoria.
- **Procesos:** Manipulación o manejo de la información.
 - Clasificación: Establecer una secuencia lógica con los datos, ascendente o descendentemente, numérica, alfabética o alfanumérica.
 - Agrupación (ordenamiento): Separar los datos según categorías mutuamente excluyentes.
 - Cálculos: Operaciones aritméticas o lógicas que se efectúan sobre los datos.
 - Síntesis: Reducir grandes volúmenes de datos.
 - Comparación
 - Suma
- **Salidas:** presentación de la información en un formato previamente definido, es decir, la computadora despliega información, no datos sin procesar.

Figura 2. 1 Ciclo de Procesamiento de Datos
Fuente: Elaboración Propia



2.1.2. Información

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

Alvin y Heidi Toffler, en su libro "La Revolución de la Riqueza" nos brindan la siguiente diferencia, entre lo que son los datos y lo que es información: "Los datos suelen ser descritos como elementos discretos, huérfanos de contexto: por ejemplo, (300 acciones). Cuando los datos son contextualizados, se convierten en información: por ejemplo, (tenemos 300 acciones de la empresa farmacéutica X)".

Sin embargo, conceptualizamos a la información como un conjunto de datos acerca de algún suceso, hecho o fenómeno, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo.

2.1.2.1. Características de la información

La información tiene las siguientes características:

- La información está compuesta por una serie de datos organizados y cargados de contenido.
- Puede generar conocimiento.
- Tener mayor acceso a la información no garantiza mayor conocimiento, eso dependerá del valor, la vigencia y lo confiable que sea su contenido.
- Contiene un significado que, una vez analizado de manera crítica es capaz de modificar una conducta o toma de decisión. Por tanto, conlleva a un cambio de la interpretación de aquello que se conocía anteriormente.

- Debe ser precisa, es decir, exponer todos los detalles necesarios para comprender su significado según la naturaleza del contenido.
- Debe estar al alcance de las personas en el momento oportuno a fin de poder aclarar una duda o responder a una situación en específico.
- Debe ser de utilidad, ya que a partir de la misma serán formuladas respuestas o tomas de acciones por lo que el individuo cambiará su relación con el entorno.
- Su contenido debe ser vigente, es decir, estar actualizada en el tiempo y el espacio que atañe, de lo contrario puede que no ofrezca ningún aporte a nuestros conocimientos previos.
- Debe ser confiable o válida, de esta manera el contenido de los datos tendrá mayor valor para el receptor de la información. En los casos en los que se obtiene información falsa, esta pierde toda su utilidad y valor.
- Existen diferentes maneras de procesar la información y su contenido dependiendo del área.

El uso de la información, cualquiera que sea, debe hacerse de manera responsable, ya que puede derivar en consecuencias positivas o negativas según su interpretación, análisis y veracidad del contenido, lo que implica un gran riesgo porque podemos modificar nuestras conductas o tomar decisiones de manera errónea.

En este sentido, es importante destacar cuáles son los objetivos que debemos seguir cuando tenemos entre manos una información. En principio, este contenido debe aportar conocimiento o complementar aquello que ya se sabe, reducir nuestras dudas en relación a un tema en particular y aportar soluciones o respuestas a un problema si fuese el caso.

En la actualidad existe un apetito insaciable por tener información a través de las nuevas tecnologías y canales de comunicación. Sin embargo, cantidad no es calidad, no porque se posea mayor cantidad de información tenemos más

conocimiento, y esto se debe a que nos podemos encontrar con información repetida, carente de bases, noticias falsas, entre otros.

2.1.2.2. Tipos de información

Existen distintos tipos de información, las cuales citaremos a continuación:

a) Información privilegiada

Se refiere a la información que solo conoce un grupo de personas y que todavía no se ha hecho pública, lo que se puede deber a que su contenido es de acceso restringido y por eso no se debe compartir o divulgar. Por ejemplo, los datos compartidos en una reunión de médicos.

b) Información pública

Es un tipo de información que se comparte al público en general y a la que todos tienen acceso, por tanto, se basa en la libertad de expresión. Sin embargo, su publicación dura un tiempo determinado. Por ejemplo, anuncios de la televisión, informes de organismos públicos, entre otros.

c) Información privada

Es el tipo de información que no se divulga de manera pública según lo estipulado en la ley, ya que puede afectar la seguridad personal, empresarial, nacional, entre otros. Por ejemplo, datos bancarios personales.

d) Información externa

Es el tipo de información que entra a una organización o empresa por diferentes vías externas, y que se emplea en aquellos casos en los que

sea necesario resolver alguna situación en particular. Por ejemplo, datos que una empresa necesite para sobrepasar alguna recaída de producción.

e) Información interna

Se trata de aquella información que conoce un selectivo grupo de personas en una empresa u organización, sin que esta tenga necesariamente una mayor relevancia sobre un tema público. Simplemente se trata de una información que no es del interés general. Por ejemplo, las pautas de organización interna en una empresa.

2.1.2.3. Información en informática

La informática es una ciencia que se encarga del procesamiento, almacenamiento y transmisión automática de una serie de datos a través del uso de diversos sistemas de programación que contienen los ordenadores.

En este sentido, la función de la informática es codificar la información recibida a través de los datos y diversos mecanismos a fin de ayudar a las personas a resolver o gestionar distintas tareas diarias como.

En informática el concepto de dato se refiere a las magnitudes numéricas, conjunto de símbolos, frases, valores cualitativos, imágenes, sonidos, entre otros, que son procesados por diversos sistemas o mecanismos informáticos a fin de generar una información de manera automática y eficiente.

2.1.3. Sistema

Van Gigch define sistema como la reunión o conjunto de elementos relacionados, los cuales pueden ser concepto, objetos, sujetos, o puede estructurarse de conceptos, objetos y sujetos como un sistema hombre-máquina que comprende las tres clases de elementos. En definitiva, según este autor, un sistema es la unión de partes o componentes, conectados en una forma organizada.

Según Hugo O. Cajizo Pérez, en la Teoría General de Sistemas define un Sistema como un conjunto ordenado de componentes o elementos interrelacionados, interdependientes e interactuantes, que tienen por finalidad el logro de objetivos determinados en un plan.

Un sistema es un conjunto de elementos relacionados entre sí que funciona como un todo. Si bien cada uno de los elementos de un sistema puede funcionar de manera independiente, siempre formará parte de una estructura mayor. Del mismo modo, un sistema puede ser, a su vez, un componente de otro sistema.

2.1.3.1. Teoría general de los sistemas

La teoría de sistemas (también conocida con el nombre de teoría general de sistemas, abreviado con la sigla TGS) consiste en un enfoque multidisciplinario que hace foco en las particularidades comunes a diversas entidades. El biólogo de origen austriaco Ludwig von Bertalanffy (1901–1972), cuentan los historiadores, fue quien se encargó de introducir este concepto a mediados del siglo XX.

2.1.3.2. Premisas de la teoría general de los sistemas

La Teoría General de Sistemas se basa en las siguientes premisas:

- Los sistemas existen dentro de sistemas. Las moléculas existen dentro de las células, las células dentro de tejidos, los tejidos dentro de órganos, los órganos dentro de organismos, y así sucesivamente.
- Los sistemas son abiertos. Ésta es consecuencia de la premisa anterior. Cada sistema que se examine, excepto el menor o el mayor, recibe y descarga algo en los otros sistemas, generalmente en los contiguos. Los sistemas abiertos se caracterizan por un proceso de intercambio infinito con su ambiente, constituido por los demás sistemas.

- Las funciones de un sistema dependen de su estructura. Por ejemplo, los tejidos musculares se contraen porque están constituidos por una estructura celular que permite contracciones para funcionar.

También se pueden considerar otros principios como son: sinergia, homeostasis, entropía, organicidad y recursividad, por ellos podemos entender que:

- **Sinergia:** Se entiende cuando la suma de las partes es más que el todo, es decir, cuando un objeto cumple con este principio o requisito decimos que posee o existe sinergia.
- **Homeostasis:** Posibilidad del sistema de mantenerse en equilibrio en cada una de sus partes.
- **Entropía:** Todo sistema tiende al desgaste o posibilidad de destruir al sistema.
- **Organicidad:** Proceso de evolución que tiende a aumentar el grado de organización que poseen los sistemas. Por ejemplo, cuando intentan operar en el sistema las fuerzas opuestas de la homeostasis y la entropía.
- **Recursividad:** El hecho de que un objeto sinérgico, un sistema, esté compuesto de partes con características tales que son a su vez objetos sinérgicos. Hablamos entonces de sistemas y subsistemas.

O, si queremos ser más extensos, de suprasistemas, sistemas y subsistemas. Lo importante del caso, y que es lo esencial de la recursividad, es que cada uno de estos objetos, no importando su tamaño, tiene propiedades que lo convierten en una totalidad, es decir, en elemento independiente.

2.1.3.3. Principios de los sistemas

Dentro de los principios que enmarcan los sistemas, la integración o coherencia no son dos propiedades separadas, puesto que son los extremos de una misma propiedad y prestan cierto grado de integración, equifinalidad y jerarquización

para la unificación de los diferentes campos en los que se mueve la empresa tal como se describe a continuación.

- **Principios de Integración**

La teoría general de los sistemas es análoga al principio o de las partes integrantes dentro de un todo. Por lo tanto, el principio de integración es vital en el concepto de sistemas.

Entre los principios se integraron según Jhonson, Kast y Rosenwing se pueden mencionar:

- a) El todo es primario y las partes secundarias.
- b) La integración es la condición de la interrelacionalidad de las muchas partes dentro de un todo.
- c) Las partes así constituidas forman un todo indisoluble en el cual todo o ninguna parte puede ser afectada sin afectar todas las otras partes.
- d) El papel que juegan las partes depende del propósito para la cual existe un todo.

- **Principio de Equifinalidad**

El fundamento de este principio, sugiere que el administrador no necesariamente debe buscar la solución óptima sino diversas soluciones satisfactorias; buscar la solución óptima rígidamente, es propiedad de los sistemas cerrados, que conciben a los organismos sociales como una simple relación causa-efecto.

La equifinalidad implica el logro de resultados finales con diferentes condiciones iniciales y de múltiples maneras, con diferentes insumos y finalidades; en síntesis, disponer de una variedad de alternativas satisfactorias que las caracterizan como sistema abierto.

- **Principio de Jerarquización**

Consiste en que todo sistema, es dependiente de otro y a la vez otros dependen de él. En general, todos los sistemas físicos, biológicos y sociales pueden ser considerados en un sentido jerárquico. Un sistema este compuesto de orden mayor. Por tanto, existe jerarquía entre los elementos o componentes de todo sistema.

2.1.3.4. Clasificación de los sistemas

Todo sistema existe y funciona en un ambiente. Ambiente es todo lo que rodea a un sistema y sirve para proporcionar los recursos que requiere la existencia del sistema, además, es en donde el sistema arroja sus resultados y aunque es una fuente de recursos e insumos, también lo es de contingencias y amenazas para el sistema.

- **Según el modo como interactúan con el ambiente**

a) **El Sistema cerrado:** Tiene pocas entradas y salidas en relación con el ambiente externo, que son bien conocidas y guardan entre sí una razón de causa y efecto: a una entrada determinada (causa) sigue una salida determinada (efecto); por esta razón, el sistema cerrado también se denomina mecánico o determinista. El mejor ejemplo de sistema cerrado lo constituyen los motores, las máquinas y la mayor parte de la tecnología inventada por el hombre.

b) **El Sistema abierto:** Posee numerosas entradas y salidas para relacionarse con el ambiente externo, las cuales no están bien definidas; sus relaciones de causa y efecto son indeterminadas. Por esta razón, el sistema abierto también se llama orgánico. Son ejemplos de este sistema las organizaciones en general y las empresas en particular, todos los sistemas vivos y especialmente el hombre.

- **En cuanto a su constitución**
 - a) **Sistemas Físicos o Concretos:** cuando están compuestos por equipos, por maquinaria y por objetos y cosas reales. Pueden ser descritos en términos cuantitativos de desempeño.
 - b) **Sistemas Abstractos:** cuando están compuestos por conceptos, planes, hipótesis e ideas. Aquí, los símbolos representan atributos y objetos, que muchas veces sólo existen en el pensamiento de las personas. En realidad, en ciertos casos, el sistema físico (hardware) opera en consonancia con el sistema abstracto (software).
Es el ejemplo de una secuencia con sus consultorios, equipos de computación, escritorios, iluminación, (sistema físico) para desarrollar un programa de registro de pacientes (sistema abstracto); o un centro de procesamiento de datos, en el que el equipo y los circuitos procesan programas de instrucciones al computador.
- **Con relación a su origen**
 - a) **Sistemas Naturales o Artificiales:** Están compuestos por sistemas creados o hechos por el hombre y que las organizaciones e instituciones la utilizan para la realización de diferentes actividades, distinción que apunta a destacar la dependencia o no en su estructuración por parte de otros sistemas.

2.1.3.5. Elementos de los sistemas

Los sistemas constan de cuatro elementos fundamentales y esenciales para la realización y unificación de los resultados, los cuales ayudan a un mejor análisis y procesamiento de la información desde su inicio hasta la finalización de este, tal como se describe a continuación.

- a) **Entradas:** Todo sistema recibe entradas o insumos del ambiente circundante. Mediante las entradas (inputs) el sistema consigue los recursos e insumos necesarios para su alimentación y nutrición.
- b) **Procesamiento:** Es el núcleo del sistema; transforma las entradas en salidas o resultados. Regularmente, lo constituyen subsistemas especializados en procesar cada clase de recurso o insumo recibido por el sistema.
- c) **Salidas:** Son el resultado de la operación del sistema. Mediante éstas (outputs) el sistema envía el producto resultante al ambiente externo.
- d) **Retroalimentación:** Es la acción que las salidas ejercen sobre las entradas para mantener el equilibrio del sistema. La retroalimentación constituye, por lo tanto, una acción de retorno. La retroalimentación es positiva cuando la salida estimula y amplía la entrada para incrementar el funcionamiento del sistema; es negativa cuando la salida restringe y reduce la entrada para disminuir la marcha del sistema.

2.1.4. Sistema de información

Un sistema de información es un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos (Laudon & Laudon, 2012).

Un sistema de Información (SI) no requiere necesariamente de un computador. Sin embargo, en la actualidad se asocia el concepto de SI a “Sistema de Información Computarizado” (Cohen & Asin, 2000).

Cuando se habla de un sistema de información (SI) se refiere a un conjunto ordenado de mecanismos que tienen como fin la administración de datos y de información, de manera que puedan ser recuperados y procesados fácil y rápidamente.

Todo sistema de información se compone de una serie de recursos interconectados y en interacción, dispuestos del modo más conveniente en base al propósito informativo trazado, como puede ser recabar información personal, procesar estadísticas, organizar archivo, etc. Estos recursos pueden ser:

- **Recursos humanos:** Personal de variada índole y destrezas.
- **Datos:** Cualquier tipo de información masiva que precisa de organizarse.
- **Actividades:** Procedimientos, pasos a seguir, estaciones de trabajo, etc.
- **Recursos informáticos:** Aquellos determinados por la tecnología.

Se debe destacar que no es lo mismo un sistema de información que un sistema informático, si bien estos últimos constituyen a menudo el grueso de los recursos de un SI. Pero existen muchos otros métodos para los sistemas de información, que no necesariamente pasan por la informática

2.1.4.1. Tipos de sistemas de información

Desde un punto de vista organizativo, los sistemas de información pueden clasificarse en:

- **Sistemas de Procesamiento de Transacciones (TPS):** También conocidos como sistemas de gestión operativa, recopilan la información pertinente a las transacciones de la organización, es decir, de su funcionamiento.

- **Sistemas de Información Ejecutiva (EIS):** Monitoriza las variables gerenciales de un área específica de la organización, a partir de la información interna y externa de la misma.
- **Sistema de Información Gerencial (MIS):** Contemplan la información general de la organización y la comprenden como un todo.
- **Sistema de soporte de decisiones (DSS):** Orientados al procesamiento de información intra y extra organizacional, para el apoyo en la conducción de la empresa.

Existen otras formas especializadas o aplicadas de SI, dependiendo del campo puntual y de las funciones específicas que se esperan de cada uno. Sería demasiado extenso enlistarlas a todas.

2.1.4.2. Elementos de un sistema de información

Por lo general se considera que todos los SI contienen una diversidad de elementos clasificables en cinco grandes categorías:

- a) **Elementos financieros:** Aquellos vinculados con el capital y con los activos disponibles de la organización.
- b) **Elementos tecnológicos:** Aquellos que tienen que ver con la maquinaria especializada y la capacidad de procesamiento automatizado de la información.
- c) **Elementos humanos:** Básicamente, personal, tanto especializado y directivo, como no especializado o común.
- d) **Elementos materiales:** Se refiere al emplazamiento del sistema, a su soporte físico y ubicación.
- e) **Elementos administrativos:** Aquellos relacionados con los procesos, la mecánica de conducción, los permisos, informes, transacciones, etc.

2.1.4.3. Ejemplos de sistema de información

Algunos ejemplos de SI pueden ser:

- Los sistemas de control de calidad, en los que se pide una retroalimentación al cliente y se evalúan los resultados estadísticamente para elaborar resultados interpretables por la gerencia.
- Las bases de datos de una biblioteca, en donde está contenido el grueso volumen de documentos (libros, revistas, tesis, etc.) de la biblioteca, en función de ubicar y recuperar cada uno lo más rápida y precisamente posible.
- Las hojas de cálculo, en las que se ingresa información en bruto y se la organiza de manera cuantificable para obtener directrices de conducción financiera.

2.1.4.4. Funciones de un sistema de información

Las funciones de un sistema de información son:

- **Recolección**

Esta función implica la captura y el registro de datos. Actúa como el órgano sensorio de la organización. Es la función más expuesta a generación de errores puesto que interactúa con el usuario, aunque este último aspecto está siendo atenuado con la aplicación de nuevas tecnologías de captura de datos, como la lectura de código de barras y código QR (Saroka, 2002).

- **Clasificación**

Esta función consiste en identificar los datos, agruparlos en conjuntos homogéneos, y ordenarlos teniendo en cuenta la manera en que será necesario recuperarlos. Vale decir que los datos se agrupan en estructuras diseñadas conforme a las necesidades del uso que se hará de ellos. (Saroka, 2002)

El almacenamiento de datos en archivos computadorizados dispone de técnicas que han permitido alcanzar un elevado nivel de refinamiento. Sin embargo, ya que el diseño del sistema de clasificación debe hacerse de acuerdo con la forma en que el usuario recuperará la información, tal diseño no puede ser adecuadamente definido si no se posee una clara comprensión de los procesos de decisión (Saroka, 2002).

- **Comprensión**

La compresión es la función por la cual se reduce el volumen de los datos sin disminuir necesariamente la información que suministrarán a su destinatario; muy por el contrario, la compresión generalmente aumenta o hace más expresivo el contenido informativo de los datos. (Saroka, 2002).

- **Almacenamiento**

Esta función se vincula con la conservación física de los datos y con su adecuada protección. Aunque no todos los datos que procesa un sistema de información se conservan en dispositivos de computación, éstos constituyen el soporte prácticamente obligado del banco de datos de las organizaciones. Aun en las empresas de mayor envergadura en el mundo, la tecnología de computación disponible permite una capacidad virtualmente ilimitada para mantener este banco de datos en condiciones de ser consultado en forma inmediata. En materia de archivos computadorizados, la teoría y la práctica del diseño, la generación, el mantenimiento, la reorganización y la consulta de las estructuras de datos han alcanzado un alto grado de sofisticación y eficiencia. Como una definición general, puede decirse que se denomina “base de datos” a un conjunto de archivos que responde a la aplicación de herramientas lógicas orientadas específicamente al logro de esa eficiencia (Saroka, 2002).

- **Recuperación**

Esta función tiene el propósito de suministrar el acceso a la base de datos. Como se dijo más arriba, depende de un apropiado sistema de clasificación. Cada día están más difundidas las aplicaciones de computación en las que la recuperación de los datos (y, muchas veces, su actualización) debe hacerse en tiempo real, es decir, en el mismo momento en que sucede el hecho que genera la necesidad de la recuperación o la actualización. En estos casos, la computadora interviene en alguna parte de la ejecución de la propia transacción que demanda el uso o actualización de los datos (Saroka, 2002).

- **Procesamiento**

El sistema de información (como todo sistema) es un transformador de entradas en salidas a través de un proceso. Esta transformación se realiza mediante cómputos, clasificaciones, cálculos, agregaciones, relaciones, transcripciones y, en general, operaciones que, no importa qué recursos humanos o tecnológicos empleen, persiguen el objetivo de convertir datos en información, es decir, en datos que habrán de tener valor y significado para un usuario. La función de procesamiento implica, principalmente, la modificación de la base de datos para mantenerla actualizada (Saroka, 2002).

- **Transmisión**

Esta función comporta la comunicación entre puntos geográficos distantes, sea por el traslado físico del sostén de los datos (papeles, dispositivos de archivos, cintas de audio o video, microfichas, etc.) o por la transmisión de señales (comunicación entre equipos de computación, transmisión de facsímiles, teléfono, etc.).

Este aspecto del sistema de información se vincula con la tecnología de comunicaciones, la que se halla tan asociada con la de la computación, e

igualmente tan desarrollada, que resulta muy difícil trazar una línea de separación entre ellas. De ahí que suele aplicarse la denominación de telemática a la disciplina o ambiente tecnológico que surge de la combinación de las telecomunicaciones y la informática (Saroka, 2002).

- **Exhibición**

Mediante esta función, se proporciona una salida de información preparada de modo tal que resulte legible y útil a su destinatario. En un sistema de información basado en el uso de computadoras, esta función es la que implica la interfaz con el ser humano. Todas las funciones descritas hasta aquí realizan diversos tratamientos de la información, pero no producen resultados visibles para el usuario. De ello se encarga esta función de exhibición, la que expone la información en forma impresa, en una pantalla de representación visual o en otros dispositivos (Saroka, 2002).

2.1.5. Ingeniería web

Según (López J., 2010) la ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.

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

Desde que esto empezó a suceder el Internet se volvió más que una diversión y empezó a ser tomado más en serio, ya que el aumento de publicaciones y de informaciones hizo que la Web se volviera como un desafío para los (Ingeniería del software) ingenieros del software, a raíz de esto se crearon enfoques disciplinados, sistemáticos y metodologías donde tuvieron en cuenta aspectos

específicos de este nuevo medio. Uno de los aspectos más tenidos en cuenta, en el desarrollo de sitios web es sin duda alguna el diseño gráfico y la organización estructural del contenido.

En la actualidad la web está sufriendo grandes cambios, que han obligado a expertos en el tema a utilizar herramientas y técnicas basadas en la ingeniería del software, para poder garantizar el buen funcionamiento y administración de los sitios web.

Entonces la ingeniería de la Web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web. En este sentido, la ingeniería de la Web hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones Web complejas y de gran dimensión en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones.

2.1.6. Aplicación web

En la ingeniería software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web, y en la que se confía la ejecución de la aplicación al navegador.

Las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el

navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

2.1.7. Información clínica

Información relevante de la salud de un paciente que los profesionales de la salud generan y requieren conocer y utilizar en el ámbito de la atención de salud que brindan al paciente. (ley 30024, 2013)

2.1.8. Historia clínica

Según (Gesmed, 2011) antiguamente, cuando un único médico atendía de forma individual todas las necesidades del paciente, las historias clínicas que se realizaban eran como un cuaderno de notas en donde se registraban los datos más importantes según su criterio. Más adelante cuando un mismo paciente era llevado por diferentes médicos debido a la aparición de las especializaciones, la historia clínica pasaba a ser responsabilidad compartida por un grupo de profesionales, lo cual obligó a estructurar la información de una manera más coordinada.

La Historia Clínica como tal ha tenido dificultades en las Instituciones de Salud por ser un documento legal que a veces es ambiguo, no claro de leer, con riesgo de perder información contenida en ella por los aspectos inherentes al manejo del papel, el variado acceso de personal, forma y espacio de archivar; con la historia clínica electrónica se pretende que muchas de estas dificultades tiendan a desaparecer.

2.1.8.1. Definición de historia clínica

Una historia clínica comprende un conjunto de los documentos relativos a los procesos asistenciales de cada paciente, con la identificación de los médicos y de los demás profesionales que han intervenido en ellos como ser (residentes,

enfermeras, auxiliares, etc.), con el objeto de obtener la máxima integración posible de la documentación clínica de cada paciente.

Se define una historia clínica como “un documento médico legal, que surge del contacto entre el profesional de salud (medico, enfermero, psicólogo, etc.) y el paciente, donde se recoge la información necesaria para la correcta atención de los pacientes. La historia clínica es un documento válido desde el punto de vista clínico y legal, que recoge información de tipo asistencial, preventivo y social”. (Pablo, 2011)

El origen de la una historia clínica “nace con el primer contacto que se establece entre el médico-paciente para la asistencia médica por medio del levantamiento de datos de forma interrogativa exploratoria”, como resume Fombella y Cereijo (2012). La concepción actual de lo que viene a ser una historia clínica fue consolidada por primera vez por Hipócrates (460 a. C. - 370 a. C.).

2.1.8.2. Características de la historia clínica

En el trabajo realizado por Giménez 2014, menciona que: Un historial clínico es un documento privado entre un paciente y su médico por lo cual debe cumplir las siguientes características:

- **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.

- **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.
- **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, siendo uno de los principios reflejados en el artículo 61 de la Ley General de Sanidad.
- **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.”

2.1.9. Atención de salud

Conjunto de acciones de salud que se brinda al paciente, las cuales tienen como objetivo la promoción, prevención, recuperación y rehabilitación en salud, y son efectuadas por los profesionales de salud. (NTS 022, 2008)

2.1.10. Paciente o usuario en salud

Beneficiario directo de la atención de salud (ley 30024, 2013)

2.1.11. Control

Según (Chiavenato, 2004) el control es la función dentro de un proceso que permite planificar los hechos planeados, mide y evalúa el desempeño y toma la

decisión correctiva cuando lo necesita. De este modo el control es un proceso esencialmente regulador.

2.1.11.1. Tipos de control

Existen distintos tipos de control, los cuáles son:

- a) Control Preliminar:** Centrado en prevenir posibles desvíos de calidad y la cantidad de los recursos empleados; los procedimientos preliminares de control incluyen todas las actividades de gestión encaminadas a acrecentar la probabilidad de los resultados obtenidos se comparen favorablemente con los resultados planeados.
- b) Control concurrente:** Consiste en el seguimiento de las operaciones en curso para asegurar que se procura alcanzar los objetivos.
- c) Control de Retroalimentación:** Se centra en los resultados finales. La acción correctiva está orientada a la mejora del proceso de adquisición de recursos o de las operaciones en curso.

2.1.12. Autenticar

Controlar el acceso a un sistema mediante la validación de identidad de un usuario, otro sistema o dispositivo antes de autorizar su acceso. (ley 30024, 2013)

2.2. INGENIERIA DE SOFTWARE

Existen diferentes definiciones sobre el término ingeniería de software una de ellas es, por Ian Sommerville, quien define a la ingeniería de software como, “Una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después de que se utiliza”.

Es decir, que la ingeniería de software comprende el conocimiento amplio de teorías, métodos y herramientas que permite el desarrollo de software, para que

sea eficiente, de calidad y confiable, partiendo de tres fases definidas por Roger S. Pressman (2002, pag.15) conocidas como:

a) Fase de definición y/o planificación:

Es en esta fase que se analiza la viabilidad del software, lo cual permite que existan bases y fundamentos de costo beneficio para su desarrollo. Una de las acciones más importantes y primordiales de esta fase en el levantamiento o análisis de los requerimientos, es decir, conocer y analizar de antemano que información ha de ser procesada, que procesos se necesitan para que el sistema sea confiable, la información se obtendrá como resultado, las validaciones correspondientes y las interfaces que han de ser necesarias.

b) Fase de desarrollo:

Es donde se definen las estructuras de datos, la funcionalidad de procesos como solución a los requerimientos analizados en la fase de definición, el diseño de interfaces y la elección de lenguajes de programación o lenguajes no procedimentales; las tareas que describe Roger S. Pressman dentro de esta fase, básicamente son: diseño del software, generación de código y prueba del software; teniendo en cuenta que pueden cambiar de acuerdo a la metodología utilizada en el desarrollo del software como se presentara en los posteriores subtítulos.

c) Fase de mantenimiento:

Una vez finalizado el desarrollo de software, la fase de mantenimiento es en la cual se presentan cambios, como ser, corrección de errores (y prevención de los mismos), adaptaciones y mejoras; de acuerdo a nuevos requisitos o cambio de alguno de los que se planteó en la fase de definición, por parte del cliente.

2.2.1. Metodologías de desarrollo del software

Según el INTECO (Instituto Nacional de Tecnologías de Comunicación, 2009), el desarrollo de software requiere del uso de metodologías, que ayuden y guíen las actividades y procesos, para conseguir las metas u objetivos planteados al inicio de proyecto de desarrollo de software y conseguir un producto que sea de calidad y cumpla con el ciclo de vida del proyecto.

Las metodologías de desarrollo permiten administrar el ciclo de vida de un proyecto, para este caso existen tres tipos de metodologías, que actúan o se diferencian por el tiempo de desarrollo e iteraciones que se presenta para controlar el buen desarrollo del software.

Las metodologías de Desarrollo de Software según (Carrillo I., Pérez R., Rodríguez A., 2008) surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software.

Se podrían clasificar en dos grandes grupos:

- **Las metodologías orientadas al control de los procesos**

Estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas Metodologías Pesadas.

- **Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software**

Mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas Metodologías ligeras/ágiles. En el desarrollo de software, una metodología hace cierto

énfasis al entorno en el cual se plantea y estructura el desarrollo de un sistema, por esta razón es importante que dependiendo del tipo de software que se vaya a desarrollar, se identifique la metodología para el diseño de software idóneo.

2.2.1.1. Métodos ágiles

Las metodologías ágiles, en resumen, las describe INTECO (2009) haciendo referencia a que el desarrollo de software ágil, se basa en el desarrollo incremental, ya que se caracteriza por entregas pequeñas de software, el tiempo o vida de los ciclos de vida concluyen más rápido. Otra característica de estos métodos es la interacción constante y cooperativa entre desarrolladores y clientes.

La ventaja de los métodos ágiles, es la facilidad de realizar cambios que intervienen en el desarrollo.

Entre los métodos conocidos como ágiles se puede mencionar:

- Extreme Programming (XP)
- Scrum
- Familia de Metodologías Crystal
- Feature Driven Development
- Proceso Unificado Rational (RUP)
- OpenUP

2.2.1.2. Metodología scrum

Scrum es una metodología ágil de gestión de proyectos de desarrollo de software, que está basada en un proceso de trabajo constante, iterativo e incremental.

El modelo Scrum fue creado por Jeff Sutherland en 1993, en su trabajo (Bahit, Eugenia- 2011), define a la metodología como: “una de las más utilizadas, la propuesta de Scrum consiste en realizar entregas potencialmente utilizables de

forma iterativa incremental, en periodos de 2 a 4 semanas denominados ‘Sprints’.

Ken Schwaber y Jeff Sutherland, 2013, describen que:” El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso “.

El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable.

Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective), durante el Sprint:

- No se realizan cambios que puedan afectar al Objetivo del Sprint (Sprint Goal).
- Los objetivos de calidad no disminuyen.
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de desarrollo a medida que se va aprendiendo más.

“Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para lograr algo. Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante” (Ken Schwaber y Jeff Sutherland, 2013).

2.2.1.2.1. Proceso y roles de scrum

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado Requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada.

Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes. El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio.

Según (Schwaber, 2010) Scrum comprende las siguientes fases como se muestra en la Figura 2.1

:

- **Product backlog:** Conjunto de requisitos denominados historias descritos en un lenguaje no técnico y priorizados por valor de negocio, o lo que es lo mismo, por retorno de inversión considerando su beneficio y coste. Los requisitos y prioridades se revisan y ajustan durante el curso del proyecto a intervalos regulares.
- **Sprint planning:** Reunión durante la cual el Product Owner presenta las historias del backlog por orden de prioridad. El equipo determina la cantidad de historias que puede comprometerse a completar en ese sprint, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir.

- **Sprint:** Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del Product Backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo.
- **Sprint backlog:** Lista de las tareas necesarias para llevar a cabo las historias del sprint.

2.2.1.2.2. FASES DE SCRUM

a) Pre-Game (Pre Juego)

Esta fase incluye dos sub fases: la de Planning y Architecture.

- **Planning:** Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. La Product Backlog List es actualizada constantemente con ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems.
- **Architecture/High level Design:** El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la Product Backlog List. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifican los cambios necesarios para implementar los elementos que aparecen en la lista Product Backlog y el impacto que pueden tener estos cambios. Se sostiene una Design Review Meeting para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión. Se preparan planes preliminares sobre el contenido de cada release.

b) Game (Juego)

En esta fase se espera que ocurran cosas impredecibles.

Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar. Este control se realiza durante los Sprints. Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación. En lugar de tenerlas en consideración al comienzo del desarrollo, Scrum propone controlarlas constantemente para poder adaptarse a los cambios en forma flexible.

c) Post Game (Post Juego)

Contiene el cierre del realce. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno por ejemplo que los requerimientos fueron completados. El sistema está listo para ser liberado y es en esta etapa en la que se realiza integración, pruebas del sistema y documentación.

2.2.2. Arquitectura de software

La arquitectura de software es el diseño de más alto nivel de la estructura de un sistema. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad (wikipedia, 2018).

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.

Algunas características que poseen, son:

- Una arquitectura de software se selecciona y diseña con base en objetivos (requisitos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también

otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.

- La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

2.2.2.1. Modelo vista controlador

El Modelo Vista Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. Es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo.

A partir del uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores (Fernandez Romero & Diaz Gonzales, 2012).

Algunas de sus ventajas son:

- La implementación se realiza de forma modular.

- Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
- MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad única comparadas con otras aplicaciones basadas en otros patrones.

El MVC tiene los siguientes componentes.

- **MODELO:** Es un conjunto de clases que representan la información del mundo real que el sistema debe reflejar. Es la parte encargada de representar la lógica de negocio de una aplicación. Es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado (Andalucía, 2013).
- **VISTA:** Son las interfaces de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con éste. (Alicante, 2018)
La vista solo necesita la información requerida del modelo para realizar un despliegue. Cada vez que se realiza una actuación, que implica una modificación del modelo de dominio, la vista cambia a través de notificaciones generadas por el modelo de la aplicación. Sencillamente, es

la representación visual del modelo que redibuja las partes necesarias cuando se produce una modificación del mismo (Andalucía, 2013).

- **CONTROLADOR:** El controlador es el encargado de interpretar y dar sentido a las instrucciones que realiza el usuario, realizando actuaciones sobre el modelo. Si se realiza algún cambio, comienza a actuar, tanto si la modificación se produce en una vista o en el modelo.

Interactúa con el Modelo a través de una referencia al propio Modelo. Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información y otros. En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo (Andalucía, 2013).

2.2.2.2. Metodología uwe

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web.

UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML. Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado UML, pero adaptados a la web. En requisitos separa las fases de captura,

definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web.

Entre los principales modelos de UWE podemos citar:

- Modelo de Requerimientos
- Modelo Lógico-Conceptual
- Modelo de Usuario
- Modelo de Navegación
- Modelo de Presentación
- Modelo de Adaptación

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además, UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos.

Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML.

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

2.2.2.3. Lenguaje unificado de modelado (UML)

UML (Unified Modeling Language), es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Se ha convertido en el estándar de facto, debido a que ha sido impulsado por los autores de los tres métodos más usados de la orientación a objetos.

UML fue representado por Gready Bootch, Jim Rumbahugt, Ivar Jacobson en 1997, para definir un lenguaje y una notación estándar del lenguaje de construcción de modelos Orientado a objetos. Cabe distinguir que UML no es un método de desarrollo de sistemas, es un lenguaje de construcción de modelos, en general no interviene en modelos ni procesos de desarrollo para la creación de software. Debido a su estandarización y su definición completa no ambigua y aunque no sea un lenguaje d programación, UML se puede conectar de manera directa a lenguajes de programación como java, C++ o Visual Basic, esta correspondencia permite lo que se denomina como ingeniería directa (obtener el código fuente partiendo de los modelos) pero además es posible reconstruir un modelo en UML partiendo de la implementación o sea la ingeniería inversa.

También se emplea en el diseño de sistemas hipermedia ya sean con el modelado de clases como en el modelado navegacional, donde interviene en la definición de nodos.

2.2.2.3.1. Diagrama de casos de uso

Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular.

Un caso de uso se utiliza para organizar los aspectos del comportamiento en un modelo. Se representa el sistema como una caja rectangular con el nombre en

su interior, los casos de uso están en el interior de la caja del sistema y los actores fuera, cada actor está unido a los casos de uso en los que participa mediante una línea, ejemplo:

Las partes que conforman el diagrama de casos de uso son:

- a) Elementos:** Los elementos que pueden aparecer en un diagrama de casos de uso son actores y sus relaciones entre ellos.
- b) Actores:** Un actor es algo con comportamiento como una persona (identificada por un rol), un sistema informatizado u organización y que realiza algún tipo de interacción con el sistema. Se representa mediante una figura humana dibujada con palotes.
- c) Casos de uso:** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad y se representa en el diagrama de casos de uso mediante una elipse. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.
- d) Relaciones entre casos de uso:** Un caso de uso en principio debería describir una tarea que tiene un sentido completo para el usuario, hay ocasiones en las que es útil describir una interacción con un alcance menor. Para el caso de utilizar casos de uso más pequeños, las relaciones entre estos y los casos de uso ordinarios pueden ser de los siguientes tres tipos:
 - **Comunicación:**
Relación (asociación) entre un actor y un caso de uso.
El estereotipo de la relación de comunicación es: <<communicate>> aunque generalmente no se estipula ningún nombre, como podemos apreciar en el siguiente ejemplo de comunicación:

- **Inclusión:**

Un caso de uso base incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia.

La relación de inclusión sirve para enriquecer un caso de uso con otro y compartir una funcionalidad común entre varios casos de uso, también puede utilizarse para estructurar un caso de uso describiendo sus subfunciones. El caso de uso incluido existe únicamente con ese propósito, ya que no responde a un objetivo de un actor.

Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<include>>. Algunos casos de uso típicos de inclusión son: comprobar, verificar, buscar, validar, autenticar o login.

En principio, no deberíamos abusar de este tipo de relación, para no hacer una descomposición funcional del sistema. A partir de UML 1.3 la relación <<include>> reemplazó al denominado <<uses>>.

Ejemplo de inclusión entre casos de uso:

- **Extensión:**

Un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por este otro caso de uso. En el caso de uso base, la extensión se hace en una serie de puntos concretos y previstos en el momento del diseño, llamados puntos de extensión, los cuáles no son parte del flujo principal.

La relación de extensión sirve para modelar: la parte opcional del sistema, un subflujo que sólo se ejecuta bajo ciertas condiciones o varios flujos que se pueden insertar en un punto determinado. Este tipo de relación produce confusión y no debería utilizarse en exceso. Conviene su uso sólo para insertar un nuevo comportamiento no previsto en un caso de uso existente. Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<extend>>.

Ejemplo de extensión entre casos de uso:

- **Especialización y generalización de los casos de uso:**

Un caso de uso (sub caso) hereda el comportamiento y significado de otro, es decir las relaciones de comunicación, inclusión y extensión del super caso de uso. En muchas ocasiones este super caso de uso es abstracto y corresponde a un comportamiento parcial completado en el sub caso de uso.

O, dicho de otra manera, Los casos de uso “hijo” son una especialización del caso de uso “padre”. En la medida de lo posible debería evitarse puesto que produce cierta confusión en algunas ocasiones.

Ejemplo de especialización entre casos de uso:

Como podemos ver en este último ejemplo también pueden existir vínculos de generalización o herencia entre actores.

Además de estar considerado como una extensión del estándar UML, también se basa en otros estándares como, por ejemplo: XMI como modelo de intercambio de formato, MOF para la meta-modelado, los principios de modelado de MDA, el modelo de transformación del lenguaje QVT y XML.

2.2.2.4. Actividades de modelado de UWE

Las actividades base de modelado de UWE son:

- El análisis de requerimientos
- El modelo conceptual
- El modelo navegacional
- El modelo de presentación.

A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de Escenarios Web.

El modelado que propone UWE está compuesto por etapas o sub modelos:

- Modelo de Casos de Uso
- Modelo de Contenido
- Modelo de Usuario
- Modelo de estructura
- Modelo Abstracto
- Modelo de Adaptación
- Modelo de flujo de presentación.
- Modelo de ciclo de vida del objeto.

2.2.2.4.1. Modelo lógico-conceptual

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

2.2.2.4.2. Modelo de navegación

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

2.2.2.4.3. Modelo de presentación

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

2.2.2.4.4. Interacción temporal

Presenta los objetos que participan en la interacción y la secuencia de los mensajes enviados entre ellos.

2.2.2.4.5. Escenarios web

Permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. Junto con el modelo de interacción temporal, los escenarios Web proveen la representación funcional dinámica del modelo de navegación.

2.2.2.5. Diagramas

Los diagramas usados por UWE, son diagramas UML puro. Entre los más importantes tenemos:

- Diagramas de estado
- Diagramas de Secuencia
- Diagramas de colaboración
- Diagramas de Actividad.

2.2.2.6. Fases de la UWE

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

Las fases o etapas a utilizar son:

a) Captura, análisis y especificación de requisitos:

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

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales.

Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

b) Diseño del sistema:

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

c) Codificación del software:

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

d) Pruebas:

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

e) La instalación o Fase de Implementación:

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

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

f) El Mantenimiento:

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

Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes, opiniones de los usuarios, el comportamiento y uso del sitio.

2.2.2.7. Ventajas UWE

La especificación de UWE par a la especificación de aplicaciones adaptativas provee al usuario páginas más apropiadas ya que estas están descritas en función de las preferencias del usuario o de las características de contexto y se basan en técnicas orientadas a los aspectos. UWE permite un modelado de aplicaciones Web basado en las demandas de cada usuario en particular, separando requerimientos, enfoques, interfaces, adaptabilidad, aspectos y componentes para mayor flexibilidad, definiendo un conjunto de procesos adecuados durante todas las etapas del desarrollo, cual permite mantener la integridad del diseño y la funcionalidad del sistema.

2.2.3. Seguridad de software

El autor y experto en seguridad Gary McGraw define a la seguridad de software como “la relación por completo con la calidad, debe pensarse en seguridad, confiabilidad, disponibilidad y dependencia, en la fase inicial, en la de diseño, en la de arquitectura, pruebas y codificación, durante todo el ciclo de vida del software [proceso].”

En pocas palabras, el software que no tiene alta calidad es fácil de penetrar por parte de intrusos y, en consecuencia, el software de mala calidad

aumenta indirectamente el riesgo de la seguridad, con todos los costos y problemas que eso conlleva.

También plantea que la base de los problemas de seguridad son la conectividad, la complejidad y la extensibilidad de los sistemas actuales y su defunción está dada bajo 2 conceptos orientados dentro los objetivos de la Ingeniería de Software:

- La seguridad de un producto desarrollado se orienta a la búsqueda de que dicho producto continúe funcionando correctamente ante ataques maliciosos.
- La seguridad del Software en construcción se orienta a la resistencia proactiva de posibles ataques

2.2.3.1. Seguridad de software y estándares de calidad

En aras de garantizar la seguridad del software resulta imprescindible buscar un desarrollo casi perfecto, es por eso que deben tenerse en cuenta factores de calidad.

Uno de ellos no inventar lo que ya está hecho. Los sistemas se basan en patrones similares. Es preciso, entonces, estudiar para no reinventar soluciones a problemas que ya han sido resueltos, esto optimiza tiempo y recursos.

El mecanismo con el que se estructura la información es crucial. En este aspecto la obsesión nunca se queda corta, todo elemento debe estar documentado, espaciado sin fallas y debe ser de rápido entendimiento.

Por el último, el mismo software puede tener elementos de seguridad que identifiquen riesgos, por ejemplo, prevenir accesos no autorizados.

Como ya se ha expuesto, nadie está exento de un ataque, pero sí puede haber alguien enfocado en intentar prevenirlo.

2.2.4. Calidad de software

2.2.4.1. Calidad

La Real Academia Española, define la calidad como: “Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor”. Esta definición está orientada al mercado.

La International Standards Organization, ISO en la norma 8402:1994, la define como la “Totalidad de propiedades y características de un producto, proceso o servicio que le confiere su aptitud para satisfacer unas necesidades expresadas o implícitas.” En la actualización de la Norma ISO, la 9000:2000, la definición quedó “Grado en el que un conjunto de características inherentes cumple con los requisitos”. En esta definición se hace especial énfasis en cumplir los requerimientos de los consumidores.

2.2.4.2. Definición de calidad de software

Son características propias del software, aquellas que tú quieres controlar y asegurar. El software es un producto inmaterial que no se fabrica, tampoco se degrada físicamente, pero sí se desarrolla. El software puede tener errores e incidencias, pero no son similares a las de cualquier equipo de carácter físico (Wikipedia, 2013)

Si hablamos de la calidad del software, una de las primeras definiciones aseguraba que “la calidad de un programa o sistema se evaluaba de acuerdo al número de defectos por cada mil líneas de código. (KLOC: Kilo Lines Of Code)”.

En el libro Ingeniería del Software de Pressman, 1998, se afirma que la calidad es la “Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario”

Podemos concluir que la calidad de Software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de

eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

2.2.4.3. Pruebas de software

Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder. Es una actividad más en el proceso de control de calidad (Wikipedia,2018)

Según Kaner, profesor de ingeniería de Software en el instituto tecnológico de Florida es uno de los principales defensores de las pruebas de software, y su definición de la misma es la siguiente: “Las pruebas de software son la investigación empírica y técnica realizada para facilitar a los interesados información sobre la calidad del producto o servicio bajo pruebas”.

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo.

2.2.4.3.1. Tipos de prueba

Hay diferentes tipos de prueba de software. Las que buscan probar una funcionalidad del software, las que buscan probar una característica no funcional, como puede ser la fiabilidad, y las que buscan probar la estructura del software. Teniendo en cuenta esto, vamos a diferenciar los tipos de prueba en tres puntos principales:

- **Pruebas Funcionales**

Las características de funcionalidad según las establece la ISO 25010 son idoneidad, exactitud, interoperabilidad y seguridad, indicando a la funcionalidad como, “la representación de la capacidad del producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones específicas”.

Estas pruebas se dividen en las siguientes características:

- Complejidad Funcional: El grado en el que las funcionalidades cubren todas las tareas y objetivos del usuario especificados.
- Corrección Funcional: Capacidad del producto o sistema para proveer resultados correctos en el nivel de precisión requerido.
- Pertenencia Funcional: Capacidad del producto de software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

- **Pruebas No Funcionales**

Este tipo de pruebas tienen en cuenta el comportamiento externo del software, es decir cómo funciona el sistema, y se suelen utilizar técnicas de diseño de caja negra.

La ISO 25010 también define las características que han de tener estas pruebas, que son, fiabilidad, facilidad de uso, eficiencia, compatibilidad y seguridad.

Estas pruebas se dividen en las siguientes características:

- Pruebas de carga: Consisten en la medición del comportamiento del sistema para aumentar la carga del mismo, ya sea mediante el número de peticiones que se realizan a una WEB al mismo tiempo, el número de usuarios que trabajan simultáneamente, etc.

- Pruebas de rendimiento: En estas pruebas se medirán la velocidad de procesamiento y el tiempo de respuesta del sistema.
- Pruebas de volumen: Se mide la capacidad del sistema para procesar gran cantidad de datos, como procesar archivos con tamaños muy grandes.
- Pruebas de esfuerzo: Se realizan pruebas donde se sobrecarga el sistema y se analiza la capacidad de recuperación.
- Pruebas de seguridad: Se realizan diferentes pruebas de accesos no autorizados, ataque de denegación de servicio, etc.
- Pruebas de estabilidad, eficiencia y robustez: Se realiza una medición de la respuesta del sistema a los errores de funcionamiento.
- Pruebas de compatibilidad: Son pruebas del funcionamiento del sistema con los diferentes sistemas operativos, plataformas de hardware, etc., con los que puede interactuar el programa.
- Pruebas de usabilidad: Se mide la facilidad de uso, efectividad y satisfacción, siempre dentro de un grupo específico de usuarios.
- **Pruebas Estructurales**
Las pruebas estructurales permiten medir la totalidad de las pruebas mediante la evaluación de tipo estructura. En estas pruebas se aplican las técnicas de diseño de caja blanca y el ISTQB utiliza el término “prueba estructural” para las pruebas de caja blanca.

2.2.4.3.2. Técnicas de prueba

Para conseguir el objetivo de que el producto tenga la calidad deseada vamos a ver diferentes técnicas de prueba que se pueden aplicar a la hora de realizar las pruebas. Estas técnicas tienen el objetivo de identificar condiciones de la prueba, casos de prueba y datos de la prueba.

Se identifican tres tipos de pruebas:

- **Técnicas estáticas.**

Este tipo de técnicas son aquellas que no ejecutan la aplicación. Se llevan a cabo a nivel de especificaciones. No ejecutan código, pero si realizarán un análisis estático del código.

- **Técnicas dinámicas.**

Este tipo de técnicas son las realizadas ejecutando la aplicación y son las utilizadas para el diseño de los casos de prueba.

La mayoría del software puede probarse de dos maneras diferentes. Conociendo el funcionamiento interno, podemos probar que todos los módulos encajan unos con otros, es decir, desde una visión interna. Estas pruebas son las pruebas de caja blanca.

Al conocer las funciones específicas del producto se pueden llevar a cabo pruebas que demuestren que estas funciones son operativas y la búsqueda de errores en dichas funciones. Estas pruebas se realizan desde una visión externa, mediante las pruebas de caja negra.

- **Técnicas basadas en la experiencia.**

El ISTQB define también las técnicas basadas en la experiencia y las define como, “aquellas en las que las pruebas se derivan de la habilidad e intuición del probador y de su experiencia con aplicaciones y tecnologías similares”.

En este proyecto adaptamos la prueba de caja negra debido a que es más factible evaluar el comportamiento del sistema web.

2.2.4.3.3. Prueba de caja negra

Las técnicas de diseño de caja negra, también llamadas pruebas de comportamiento, son las que utilizan el análisis de la especificación, tanto funcional como no funcional, sin tener en cuenta la estructura interna del

programa para diseñar casos de prueba y, a diferencia de las pruebas de caja blanca, estas pruebas se suelen realizar durante las últimas etapas de la prueba.

Como los métodos de caja negra se intenta encontrar los errores de:

- Funciones incorrectas o faltantes.
- Errores de inicialización y terminación.
- Errores de interfaz.
- Errores en las estructuras.

Hay varios métodos que se pueden aplicar a la hora de escoger la técnica de caja negra como modelo para las pruebas.

- Partición de equivalencia.
- Análisis del valor límite.
- Pruebas de tabla de decisión.
- Pruebas de transición de estado.
- Pruebas de caso de uso.

2.2.4.3.4. Pruebas de tabla de decisión

- Identificar todas las posibles “condiciones” (entradas) y posibles “acciones” (salidas).
- Usadas para representar reglas complejas de negocio.
- Hay una regla para cada posible combinación de “condiciones”.
- Es una técnica documental porque nos permite recolectar requerimientos del sistema.

Fig. 2.2 Condiciones y Acciones

Fuente Elaboración propia

	Regla 1	Regla 2	...	Regla N
Condiciones				
Condición 1				
Condición 2				
...				
Condición N				
Acciones				
Acción 1				
Acción 2				
...				
Acción N				

Donde:

Condiciones: Muestran las entradas.

Acciones: Son los resultados que sucederán, basados en la combinación de condiciones.

Reglas: Define un único conjunto de condiciones que terminan en un conjunto de acciones.

PASOS:

- Identificar CONDICIONES y los posibles valores.
- Identificar ACCIONES de salida
- Realizar las combinaciones “viables” de CONDICIONES
- Definir los valores ACCIONES de acuerdo con CONDICIONES
- Identificar cada columna como una regla diferente
- Diseñar los casos de prueba.

CASOS DE PRUEBA

- Cada columna, cada regla es un caso de prueba
- Elimine las reglas que son “imposibles”

¿Son reales?

¿Cómo puedo generarlas?

- Combinar con otras técnicas como casos de equivalencia y valores límite.
- ¿Cuál de las reglas de la Tabla de Decisión puede eliminarse? ¿O unificarse?

2.3. HERRAMIENTAS

- **Visual Studio Code (Versión 1.45)**

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

- **Microsoft Windows 10 (Pro)**

Es el vigente sistema operativo desarrollado por Microsoft como parte de la familia de sistemas operativos Windows NT.

Fue dado a conocer oficialmente en septiembre de 2014, seguido por una breve presentación de demostración en la conferencia Build 2014.

Entró en fase beta de prueba en octubre de 2014 y fue lanzado al público en general el 29 de julio de 2015.

- **PHP (Versión 7.4.6)**

PHP son las siglas en inglés de “Hypertext Pre-Processor” que significa “Lenguaje de Programación Interpretado”. Este lenguaje es al que le debemos la visualización de contenido dinámico en las páginas web.

Todo el código PHP es invisible para el usuario, porque todas las interacciones que se desarrollan en este lenguaje son por completo

transformadas para que se puedan ver imágenes, variedad de multimedia y los formatos con los que somos capaces de interactuar añadiendo o descargando información de ellos.

- **Apache (Versión estable 2.4.43)**

Apache es un acrónimo de «a patchy server», es un servicio de páginas web HTTP de código abierto que sirve para colocar varias plataformas como Unix, BSD, GNU/Linux, Windows, Macintosh entre otros que implementan el protocolo HTTP y el conocimiento o conceptos de sitios virtual y se basó inicialmente en el código NCSA HTTP.

- **Ajax (Versión 4.0.0)**

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor.

- **MS Project 2010**

Microsoft Project es una herramienta de trabajo para administradores y jefes de proyectos, la cual sirve para organizar y realizar un seguimiento de las tareas de forma eficaz evitando retrasos en las tareas y desfases en el presupuesto. Básicamente se encarga de la gestión de proyectos y evaluar su progreso. Existen muchos tipos de herramientas dentro de este software, de forma que el análisis de los datos es mucho más sencillo.

- **MariaDB (Versión 10.1.21)**

MariaDB es un sistema de gestión de bases de datos. Se deriva de MySQL, una de las bases de datos más importantes que ha existido en el mercado, utilizada para manejar grandes cantidades de información.

Para que se tenga una idea de la enorme capacidad para mover grandes cantidades de información, MySQL ha sido la base de datos utilizada por proyectos de internet de la índole de Facebook, Twitter y Wikipedia.

La simplicidad de la sintaxis permite crear bases de datos simples o complejos con mucha facilidad; es compatible con múltiples plataformas informáticas y está provista de una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias de la gestión de base de datos.

Además, permite a los desarrolladores y diseñadores realizar cambios en los sitios web con sólo cambiar un archivo, (sin necesidad de modificar todo el código web) para que se ejecuten en toda la estructura de datos que se comparte en la red.

- **Rational Rose 2013.**

Rational Rose es una herramienta de diseño orientada a objetos, que da soporte al modelado visual, es decir, que permite representar gráficamente el sistema, permitiendo hacer énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un software de mayor calidad, empleando un lenguaje estándar común que facilita la comunicación.

Proporciona mecanismos para realizar la Ingeniería Inversa, es decir, que a partir del código se pueda obtener información sobre su diseño; adicionalmente permite generar código en diferentes lenguajes a partir de un diseño en UML, brinda la posibilidad de que varias personas trabajen a la vez, permitiendo que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y permite que tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

- **Bootstrap (Versión 4.5)**

Es un framework CSS y Javascript diseñado para la creación de interfaces limpias y con un diseño responsive. Además, ofrece un amplio abanico de herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de los mismos.

2.4. MÉTRICAS DE CALIDAD

Es claro como hoy en día los ordenadores y el software que se ejecuta en éstos, son utilizados cada vez para una más amplia variedad de campos y aplicaciones. Es por este motivo, que el desarrollo y selección de productos software de alta calidad resulta relevante, teniendo en cuenta que su correcto desarrollo e implementación implican el éxito o fracaso de los procesos que se soporten sobre estas herramientas.

Así, la especificación y la evaluación extensiva de la calidad de los productos de software es un factor clave para asegurar una calidad adecuada y el éxito de las tareas que se basen en éste.

La Organización Internacional de Normalización ISO y la Comisión Electrotécnica Internacional IEC, constituyen el núcleo para la normalización a nivel internacional.

Estas entidades trabajan en colaboración con otras organizaciones internacionales, gubernamentales y privadas, conformando comités técnicos, a través de los cuales se han desarrollado estándares como ISO/IEC 9126 e ISO/IEC 14598.

La ISO/IEC 9126 fue creada para la especificación y evaluación extensiva de la calidad de los productos software teniendo en cuenta métricas, especificando características relevantes de calidad y describiendo un modelo del proceso de producción de los productos software desde el punto de vista interno, externo y de uso.

2.4.1. Familia de estándares ISO/IEC 9126

La ISO/IEC 9126 es una familia de estándares que regulan la calidad de los productos software teniendo en cuenta:

- Los modelos que le componen
- Las características intrínsecas y externas
- La forma en que se miden este tipo de características y funcionalidad del modelo propuesto

La norma se compone de cuatro partes que comparten el mismo título general: Tecnologías de información-Calidad de los productos software.

Tabla 2. 1 Familia del estándar ISO/IEC

Fuente: Management Service

Estándar	Objetivo
ISO/IEC 9126-1	Modelo de calidad
ISO/IEC 9126-2	Métricas externas
ISO/IEC 9126-3	Métricas internas
ISO/IEC 9126-4	Métricas de calidad de uso

2.4.2. Estándar ISO/IEC 9126 – 1

El estándar ISO/IEC 9126–1 describe un modelo de calidad para productos software, dividido en dos partes:

- Calidad Interna y Externa:

La primera parte del modelo especifica seis características para la calidad interna y externa, que se subdividen posteriormente en subcaracterísticas. Estas subcaracterísticas se manifiestan externamente cuando el software se usa como parte de un sistema de computación, y son el resultado del

análisis de los atributos internos del software, que son los elementos a tener en cuenta para verificar las funciones del paquete a evaluar.

- **Calidad en Uso**

Esta parte del modelo especifica cuatro características de calidad en uso, pero no elabora el modelo más allá de este nivel. La calidad en uso es el efecto combinado para el usuario de las seis características de calidad de software. ISO/IEC 9126–1 permite especificar y evaluar la calidad de los productos software desde diferentes perspectivas por parte de aquellos involucrados en la adquisición, los requisitos, el desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría del software.

2.4.3. Modelos de referencia

El estándar ISO/IEC 9126–1 tiene un modelo conformado por características y subcaracterísticas.

Además, se incluye la subcaracterística conformidad para cada una de las características. Estos elementos son la base del modelo propuesto.

Tabla 2. 2 Características y Subcaracterísticas

Fuente: Elaboración Propia

Características	Subcaracterísticas
Funcionalidad	Aplicabilidad, exactitud, seguridad Inter operatividad
Fiabilidad	Madurez, tolerancia a fallos, capacidad de recuperación
Usabilidad	Capacidad para ser entendido, capacidad para ser operado, capacidad para ser aprendido, capacidad de atracción
Eficiencia	Comportamiento temporal, utilización de recursos
Mantenibilidad	Capacidad para ser analizado, capacidad para ser cambiado, estabilidad, capacidad para ser probado
Portabilidad	Adaptabilidad, facilidad de instalación, coexistencia, capacidad para ser remplazado, cumplimiento de la portabilidad

2.4.4. Modelos de calidad

La obtención de un software con calidad implica la utilización de modelos o procedimientos estándares para el análisis, diseño, desarrollo y prueba de software que permitan uniformar la filosofía de trabajo, para lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo, como para el control de la calidad del software.

En todos los diferentes modelos para conseguir una certificación, no solo es necesario que la metodología o la documentación de los procesos cumpla con los requisitos del modelo, sino que es necesario, además, que existan suficientes

evidencias que demuestren el uso consistente y sistemático de las prácticas definidas en la organización.

Por esta razón, el objetivo principal de las acciones de mejora, no reside tanto en obtener la certificación en alguno de los niveles del modelo, si no en implantar unos procesos que, independientemente del reconocimiento de cara a clientes y proveedores, mejoran sustancialmente la calidad y el desempeño de los resultados y del propio proceso en estudio.

Los modelos de calidad se dividen en tres tipos:

- Modelos de calidad a nivel del producto.
- Estándares de calidad a nivel del producto.
- Modelos y estándares de calidad a nivel de procesos.

En este proyecto utilizaremos un modelo de calidad a nivel de productos, debido a que las características de medición se la realizarán en la culminación del mismo.

2.4.4.1. Metodología WEBQEM

Emplea un modelo de calidad que proporciona un enfoque cuantitativo y sistemático para evaluar, y compara productos web en la fase operativa y en la fase de desarrollo de ciclo de vida del producto

2.4.4.2. Características de WEBQEM

La metodología WebQem toma las métricas del modelo de calidad ISO 9126-1, la cual da referencia a las siguientes características; Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad.

- **FUNCIONALIDAD**

Funcionalidad es una métrica orientada a la función del sistema y al proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa.

- **Entradas del usuario**, se toma en cuenta cada entrada del usuario que el sistema proporciona a medida que ingresa al sistema.
- **Salidas del usuario**, se refleja las salidas que tiene el sistema tanto reportes como estadísticas que tiene el sistema.
- **Número de peticiones del usuario**, una petición se define como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida.
- **Número de archivos**, se define cada archivo lógico.
- **Número de interfaces externas**, se definen todas aquellas interfaces legibles por el ordenador que solicitan transmitir información a otro sistema.

Para calcular los puntos de función se usó las siguientes fórmulas:

Ecuación 2.1

$$PF = Cuenta\ Total * (confiabilidad\ proyecto + error\ min * \sum F_i)$$

Ecuación 2.2

$$Funcionalidad = \frac{PF}{PF\ máximo}$$

Donde:

PF: Medida de funcionalidad

PF máximo: Medida de funcionalidad con su valor máximo

Cuenta Total: Es la suma de los siguientes datos:

- Número de entradas
- Número de salidas
- Número de peticiones
- Número de archivos
- Número de interfaces externas

Confiabilidad del proyecto: La confiabilidad de un proyecto varía de 1% al 100% (0a1)

Error min: Error mínimo aceptable de complejidad

$\sum F_i$: Son los valores de ajuste de complejidad, donde ($1 \leq i \leq 14$)

- **CONFIABILIDAD**

Es la probabilidad de operación libre de fallos de un programa en un entorno determinado y durante un tiempo específico, se toma en cuenta:

A) Enlaces

- Enlaces Rotos
- Enlaces Inválidos
- Enlaces no implementados

B) Páginas

- Páginas Muertas
- Páginas bajo Construcción
- Errores de ortografía

Para calcular los puntos función se usa la siguiente fórmula:

Ecuación 2.3

$$\text{Porcentaje de enlaces rotos} = \frac{CERI + CERE}{CTE} * 100$$

Donde:

CERI: Cantidad de enlaces rotos internos

CERE: Cantidad de enlaces rotos externos

CTE: Cantidad total de enlaces

Porcentaje de presencia de propiedad:

Ecuación 2.4

$$\text{Porcentaje presencia ALT} = \frac{\text{Cantidad_Imagenes_ALT}}{\text{Cantidad_Total_Imagenes}} * 100$$

Y la confiabilidad se calcularía con la siguiente fórmula:

Ecuación 2.5

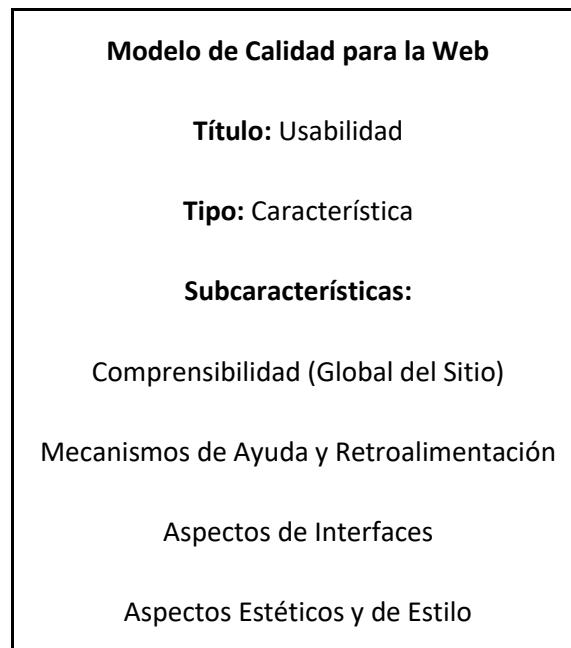
$$\text{Confiabilidad} = \text{Porcentaje de enlaces rotos} + \text{Porcentaje presencia ALT}$$

- **USABILIDAD**

Toma en cuenta la capacidad del software para ser comprendido, utilizado y atractivo para el usuario en determinadas condiciones.

Figura 2. 2 WebQem Usabilidad

Fuente: Elaboración Propia



Se realiza una tabla que toma en cuenta los siguientes puntos:

A) Comprensión Global del sitio

- i) Esquema de Organización Global
 - Tabla de Contenidos
 - Mapa del Sitio

- Índices (Alfabéticos, Temáticos, Híbridos ...)
- ii) Visita Guiada (convencional y/o virtual)
- iii) Mapa de Imagen

B) Aspectos de Interfaces y Estéticos

- i) Permanencia y Estabilidad en la Presentación de los Controles Principales
 - Controles Directos
 - Controles Indirectos
 - Estabilidad

C) Mantenimiento de Color de los Enlaces

Se toma en cuenta el diseño que tiene el sistema.

- **EFICIENCIA**

Singularidad que permite calificar el grado en el que el software hace óptimo el funcionamiento del software con respecto a la cantidad de recursos usados.

- **MANTENIBILIDAD**

La mantenibilidad es la facilidad con que una modificación puede ser realizada. Está indicada por los siguientes sub atributos.

- Facilidad de análisis
- Facilidad de cambio
- Estabilidad
- Facilidad de prueba

El índice de madurez del software se calcula con la siguiente fórmula:

Ecuación 2.6

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

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

Fc: Número de módulos en la versión anterior que se han borrado en la versión Actual.

- **PORTABILIDAD**

La portabilidad es la capacidad que tiene el sistema para ser trasladado de un entorno a otro.

Para poder medir la portabilidad del sistema usaremos la siguiente fórmula que indica el grado de portabilidad que tiene un software.

Ecuación 2.7

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

Donde:

ET: Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

ER: Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

Si **GP > 0**, la portabilidad es más rentable que el re-desarrollo

Si **GP = 1**, la portabilidad es perfecta

Si **GP < 0**, el re-desarrollo es más rentable que la portabilidad

2.4.4.3. Fases de WEBQEM

A continuación, se describe las fases de la metodología WebQem que se utilizará para realizar los cálculos que ayudará a comprobar que el sistema es confiable, eficiente y de calidad.

a) Definición de las Metas de Evaluación y Selección del perfil de Usuario

En esta fase se consideran dos pasos primordiales y comunes en toda evaluación de calidad siguiendo la metodología WebQem:

- Metas de Evaluación

En esta fase se define las metas de evaluación y selección del perfil de usuario, los evaluadores deben definir las metas y establecer el alcance del proyecto.

Las metas a llegar a cumplir con la evaluación del sistema son:

- Conocer la percepción de calidad que tienen los usuarios finales sobre el sistema.
- Lograr que la calidad del sistema esté por encima de las expectativas del usuario final.

- Selección de Perfil de Usuario

Para el diseño e implementación de un caso de estudio de evaluación de calidad en uso, una meta muy utilizada es determinar el cumplimiento de requerimientos elementales, parciales y globales de calidad para una aplicación Web operativa, considerando el perfil de encargado de requerimiento de pedidos y compras.

Los encargados de realizar los registros y uso de información, son los usuarios que interactúan con el sistema de forma recurrente y a diario, estos usuarios conocen la lógica del negocio y tuvieron capacitación sobre el manejo del sistema.

b) Definición de los Requerimientos de calidad y/o Costo

En esta fase, teniendo en cuenta los aspectos definidos en la fase anterior respecto a metas de evaluación, selección de perfil de usuario, se deben establecer atributos y subconceptos (características) de calidad cuantificables que, agrupados jerárquicamente, representen un modelo de calidad apropiado para el perfil de usuario seleccionado.

Para esto se realizará un árbol de requisitos:

- **Usabilidad**
 - Comprensibilidad del Sistema
El sistema debe ser amigable para el usuario
 - Aspectos de Interfaces y Estéticos
Los datos deben estar ordenados de la forma correcta para una mejor comprensión.
Las interfaces deben ser claras
- **Funcionalidad**
 - Aspectos de Búsqueda
Mecanismo de Búsqueda
 - * Búsqueda parcial bajo diversos parámetros
 - * Mostrar todos los resultados coincidentes con los parámetros ingresados.
 - * Validar todos los parámetros de ingreso de búsqueda
 - Aspectos de Navegación y Exploración
Navegabilidad Local
 - * Navegación sólo por interfaces asignados al rol del usuario
 - Funciones de registro de compras y pedidos
Formulario comprensible
Indicar campos obligatorios
Validar campos del formulario
Ver estado actual de la historia clínica
 - Funciones de registro de movimiento de madera
Formulario de registro comprensible
Validar campos de formulario
- **Confiabilidad**
 - No deficiencia
Errores de Enlace

- * Enlaces rotos
- * Enlaces Inválidos
- * Enlaces no implementados

- **Eficiencia**

- Accesibilidad de Información

Visualización de la información solicitada de forma clara

Eficiencia a la hora de realizar un registro

- Performance

Páginas Rápidas

c) Definición de Criterios elementales y Procedimientos de Medición

En esta etapa se definirá una base de criterios para evaluación elemental y realizar el proceso de medición y puntuación de los valores que se calcule para medir la calidad del software en tal caso se definirá los siguientes criterios:

Tabla 2. 3 Criterios de Evaluación

Fuente: Elaboración Propia

Características de Calidad	Mala	Regular	Buena	Muy Buena
Funcionalidad	0 - 30 %	31 - 50 %	51 - 90 %	91 - 100 %
Confiabilidad	0 - 30 %	31 - 50 %	51 - 90 %	91 - 100 %
Usabilidad	0 - 30 %	31 - 50 %	51 - 90 %	91 - 100 %
Mantenibilidad	0 - 30 %	31 - 50 %	51 - 90 %	91 - 100 %
Portabilidad	0 - 30 %	31 - 50 %	51 - 90 %	91 - 100 %
TOTAL	0 - 30 %	31 - 50 %	51 - 90 %	91 - 100 %

Esta tabla nos indica que nuestras características de calidad serán evaluadas según al rango que se encuentren.

d) Definición de Estructuras de Agregación e Implementación de la Evaluación Global

Para esta fase se realizará un promedio de todos los parámetros encontrados para así poder sacar un resultado global y así el usuario pueda comprender de mejor manera la calidad del sistema.

e) Análisis de Resultados y Recomendaciones

En esta fase una vez calculado las características de calidad se darán las recomendaciones respectivas además de saber los resultados y ver si el sistema cumple con lo pronosticado.

2.5. MODELO DE COSTOS DE SOFTWARE

El proceso de estimación del coste de un producto software está formado por un conjunto de técnicas y procedimientos que se usan en la organización para poder llegar a una predicción fiable. Éste es un proceso continuo, que debe ser usado y consultado a lo largo de todo el ciclo de vida del proyecto. Se divide en los siguientes pasos:

- Estimación del tamaño.
- Estimación del costo y del esfuerzo.
- Estimación de la programación temporal.
- Estimación de la cantidad de recursos computacionales.
- Asunción de riesgos.
- Inspección y aprobación.
- Redacción de informes de estimación.

2.5.1. Modelo COCOMO II

COCOMO II es un modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste). El modelo COCOMO original se ha

convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria.

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.

2.5.1.1. Estimación de esfuerzo

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

2.5.1.1.1. Modelo Composición de Aplicación

La fórmula propuesta en este modelo es la siguiente:

Ecuación 2.8

$$PM = \frac{NOP}{PROD}$$

Donde:

NOP (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

Ecuación 2.8

$$NOP = OP * (100 - \%reuso)/100$$

OP (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objetos

%reuso: Porcentaje de reuso que se espera lograr en el proyecto.

PROD: Es la productividad promedio determinada a partir del análisis de datos de proyectos en [Banker 1994],

Tabla 2. 4 Productividad Modelo

Fuente: Boehm 1995/2

Experiencia y capacidad de los desarrolladores	Muy bajo	Bajo	Normal	Alto	Muy alto
Madurez y Capacidad del ICASE	Muy bajo	Bajo	Normal	Alto	Muy alto
PROD	4	7	13	25	50

2.5.1.1.2. Modelo Diseño Temprano

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura. El modelo de Diseño Temprano ajusta el esfuerzo nominal usando siete factores de costo.

La fórmula para el cálculo del esfuerzo es la siguiente:

Ecuación 2.9

$$PM_{estimado} = PM_{nominal} * \prod_{i=1}^7 E M_i$$

Ecuación 2.10

$$PM_{nominal} = A * (KSLOC)^B$$

$$B = 1.01 + 0.01 * \sum_{j=1}^5 W_j$$

Donde:

$PM_{estimado}$: es el esfuerzo Nominal ajustado por 7 factores, que reflejan otros aspectos propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo.

KSLOC: es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.

A: es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, (**A=2.94**).

B: es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías y deseconomías de escala producidas cuando un proyecto de software incrementa su tamaño.

EM_i : corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados Multiplicadores de Esfuerzo (Effort Multipliers). Cada factor se puede clasificar en seis niveles diferentes que expresan el impacto del multiplicador sobre el esfuerzo de desarrollo. Esta escala varía desde un nivel Extra Bajo hasta un nivel Extra Alto. Cada nivel tiene un peso asociado. El peso promedio o nominal es 1.0.

Si el factor provoca un efecto nocivo en el esfuerzo de un proyecto, el valor del multiplicador correspondiente será mayor que 1.0, caso contrario el multiplicador será inferior a 1.0. La Figura 4 muestra una pantalla del software COCOMO II.1999.0, donde se aprecian los valores de los factores de acuerdo a cada nivel, según la calibración efectuada para el año 1999.

Clasificados en categorías, los 7 Multiplicadores de Esfuerzo son:

Del Producto

RCPX: Confiabilidad y Complejidad del producto

RUSE: Reusabilidad Requerida

De la Plataforma

PDIF: Dificultad de la Plataforma

Del Personal

PERS: Aptitud del Personal

PREX: Experiencia del Personal

Del Proyecto

FCIL: Facilidades

SCED: Cronograma de Desarrollo Requerido

2.5.1.1.3. Modelo Post-Arquitectura

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida.

Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y Generadores de Aplicaciones.

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

Ecuación 2.11

$$PM_{estimado} = PM_{nominal} * \prod_{i=1}^{17} EM_i$$

Los 17 factores de costo correspondientes a este modelo

2.5.1.2. Estimación del Cronograma

La versión inicial de COCOMO II provee un modelo de estimación del cronograma similar al presentado en COCOMO' 81 y ADA COCOMO.

La ecuación inicial para los tres modelos de COCOMO II es:

Ecuación 2.11

$$TDEV = \left[3.0 * PM_*^{(0.33+0.2*(B-1.01))} \right] * \frac{SCED\%}{100}$$

Donde:

TDEV: es el tiempo calendario en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad que certifique que el producto cumple con las especificaciones.

PM*: es el esfuerzo expresado en meses personas, calculado sin tener en cuenta el multiplicador de esfuerzo SCED.

B: es el factor de escala

SCED%: es el porcentaje de compresión/expansión del cronograma.

2.5.1.3. Métricas del Software

En la estimación del tamaño de software COCOMO II utiliza tres técnicas:

- Puntos Objeto
- Puntos Función No Ajustados
- Líneas de Código Fuente.

Además, se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reuso, reingeniería, conversión, y mantenimiento.

Es necesario unificar criterios de medición de tamaño, tanto para poder planificar y controlar proyectos, como para realizar estudios y análisis entre proyectos en pro de la mejora de procesos [Park 1992].

2.5.1.3.1. Puntos Objeto

A pesar de que la estimación a través de Puntos Objeto es un enfoque de medición de tamaño de software relativamente nuevo, es apropiado para las aplicaciones con componentes y para estimar esfuerzos en las etapas de prototipación.

A continuación, se describe el procedimiento para determinar Puntos Objeto en un proyecto de software:

Primero: Determinar Cantidad de Objetos: Estimar la cantidad de pantallas, reportes, componentes de 3GL que contendrá la aplicación.

Segundo: Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil).

Tercero: Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.

Cuarto: Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

Tabla 2. 5 Clasificación de puntos Objeto

Fuente: Boehm 1995/2

Para Pantallas			
Cantidad de vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3 - 5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
< 3	Simple	Simple	Media
3 - 7	Simple	Media	Difícil
> 8	Media	Difícil	Difícil

Para Reportes			
Cantidad de vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3 - 5 cliente)	Total 8 + (> 3 servidor < 5 cliente)
0 o 1	Simple	Simple	Media
2 o 3	Simple	Media	Difícil
4 +	Media	Difícil	Difícil

Tabla 2. 6 Peso de un Punto Objeto

Fuente: Boehm 1995/2

Tipo de Objeto	Complejidad – Peso		
	Simple	Media	Difícil
Pantalla	1	2	3
Reporte	2	5	8
Componente 3GL			10

2.5.1.3.2. Puntos función

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura.

Las métricas para puntos función están basadas en las guías proporcionadas por el "International Function Point User Group" IFPUG [IFPUG 1994][Behrens 1983][Kunkler 1985].

Los Puntos Función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema.

La fórmula de Albretch [Albretch 1979] para calcular los puntos función, es la siguiente:

Ecuación 2.12

$$FP = UFP * TCF$$

Donde

UFP: Puntos Función no Ajustados

TCF: Factor de Complejidad Técnica

Para calcular los UFP, se deben identificar los siguientes tipos de ítems:

- Entradas Externas (Inputs)
- Salidas Externas (Outputs)
- Archivo Lógicos Internos (Archivos)
- Archivos Externos de Interfase (Interfases)
- Solicitudes Externas (Queries)

Una vez identificados los ítems se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto. Se asigna un peso a cada ítem según el tipo y el grado de complejidad correspondiente. Finalmente, los UFP son calculados mediante la sumatoria de los pesos de todos los ítems identificados.

Ecuación 2.13

$$UFP = \sum_{i=1}^{15} (Cantidad_items_tipo_i) * (Peso_i)$$

Para el cálculo del Factor de Complejidad Técnica, TCF, se considera la siguiente fórmula:

Ecuación 2.14

$$TCF = 0.65 + 0.01 * \sum_{i=1}^{14} F_i$$

Donde los F_i corresponden a los pesos asignados a los siguientes factores:

F1: Mecanismos de recuperación y back-up confiables

F2: Comunicación de Datos

F3: Funciones de Procesamiento Distribuido

F4: Performance

F5: Configuración usada rigurosamente

F6: Entrada de datos on-line

F7: Factibilidad Operativa

F8: Actualización de archivos on-line

F9: Interfaces Complejas

F10: Procesamiento Interno Complejo

F11: Reusabilidad

F12: Fácil Instalación

F13: Soporte de múltiples instalaciones

F14: Facilidad de cambios y amigabilidad

Los pesos se consideran dentro de una escala de 0 a 5, descrita a continuación:

0: Sin influencia

1: Incidental

2: Moderado

3: Medio

4: Significativo

5: Esencial

Estas 14 características consideran aspectos como reusabilidad, performance, complejidad, confiabilidad, etc., contemplados por COCOMO II a través de los factores de costo. Es por ello que este modelo utiliza los UFP como métrica de determinación de tamaño.

2.5.1.3.3. Líneas de código fuente

COCOMO II considera a la sentencia fuente lógica como línea standard de código. Ahora bien, definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes. El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa.

Para minimizar esos problemas, se usa el checklist de definición desarrollado por el SEI, que permite unificar criterios en la definición de una línea de código fuente [Park 1992], [Goethert et al.1992].

Existen herramientas automatizadas para medir la cantidad de líneas de código fuente, como por ejemplo Amadeus [Amadeus 1994]. Para realizar un análisis de mayor especificidad, Amadeus automáticamente recolecta medidas adicionales como total de líneas fuente, de comentarios, declaraciones, interfases, anidamientos, sentencias ejecutables y otras. Esta herramienta provee varias medidas de tamaño, incluyendo métricas aplicables a tecnologías de objetos de [Chidamber and Kemerer 1994].

2.5.1.3.4. Conversión de Puntos Función a Líneas de Código Fuente (SLOC)

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a líneas de código fuente considerando el lenguaje de implementación (assembler, lenguajes de alto nivel, lenguajes de cuarta generación, etc.).

2.5.1.3.5. Desperdicio de Código (Breakage)

Se considera como Desperdicio al porcentaje de código que se debe eliminar debido a la volatilidad de los requerimientos. Éste se usa para ajustar el tamaño efectivo del software a ser desarrollado a los efectos del proceso de estimación. De este modo la ecuación del esfuerzo nominal modificada, para contemplar este aspecto, es la siguiente:

Ecuación 2.15

$$PM_{nominal} = A * \left[\left(1 + \frac{BRAK}{100} \right) * KSLOC \right]^B$$

2.5.1.3.6. Modelo de Reuso

COCOMO II usa un modelo no lineal para estimar el tamaño del software cuando éste incluye componentes reusables.

El modelo COCOMO II permite tener en cuenta si un proyecto de software va a ser construido a partir de componentes existentes. Para ello, reemplaza en la ecuación de estimación de esfuerzo el parámetro KSLOC por el KESLOC, que representa la cantidad equivalente de nuevas líneas de código a desarrollar.

ESLOC se calcula de la siguiente forma:

Ecuación 2.16

$$AAF = 0.4(DM) + 0.3(CM) + 0.3(IM)$$

Ecuación 2.17

$$ESLOC = \frac{ASLOC[AA + AAF(1 + 0.02(SU)(UNFM))]}{100}, AAF \leq 0.5$$

Ecuación 2.18

$$ESLOC = \frac{ASLOC[AA + AAF + (SU)(UNFM)]}{100}, AAF > 0.5$$

Donde:

ASLOC: Cantidad de líneas de código fuente del software existente usadas para desarrollar el nuevo producto.

DM: Porcentaje del diseño del software que requiere modificación para alcanzar los objetivos del nuevo software a desarrollar.

CM: Porcentaje del código del software que requiere modificación para lograr los objetivos del nuevo software a desarrollar

IM: Porcentaje del esfuerzo requerido para integrar y testear el software adaptado al producto global.

SU: Porcentaje de comprensibilidad del software existente. Se determina en función a tres características: estructura, claridad y descriptividad.

AA: Grado de Evaluación y Asimilación. Porcentaje de esfuerzo necesario para determinar si un módulo de software a adaptar es apropiado a la aplicación, como así también para integrar su descripción a la descripción total del producto.

UNFM: Nivel de familiaridad del programador con el software.

2.5.1.3.7. Reingeniería y Conversión

El modelo de Reuso de COCOMO II necesita un refinamiento adicional para estimar el costo de reingeniería y de conversión. La principal diferencia entre reingeniería y conversión está dada por la eficiencia de las herramientas automatizadas utilizadas para reestructurar el software.

MARCO APLICATIVO

3. MARCO APLICATIVO

3.1. INTRODUCCION

La aplicación del proyecto se realizará siguiendo la metodología ágil de SCRUM, cuyo objetivo es el determinar el análisis, diseño y desarrollo de Sistema Web de Administración de Historias Clínicas para mostrar las ventajas que otorga el sistema de información.

3.2. PRE-GAME

3.2.1. Concepción

La tarea de Ingeniería de requisitos es fundamental para que un sistema de información sea exitoso, en dicho sentido para la realización del presente proyecto se realizaron las siguientes actividades que se detallan a continuación para la obtención de requisitos.

Tabla 3. 1 Tareas para la obtención de requisitos

Fuente: Elaboración Propia

Entrevistas Personales	Se realizaron entrevistas al personal del Hospital de la Mujer, como ser el Lic. responsable del área de sistemas, Doctores, trabajadores de salud. Con la finalidad de obtener información sobre la entidad y su funcionamiento.
Observación	Se observaron los procesos que realiza el personal del Hospital, como ser: Registro de Pacientes Nuevos, Apertura de Historias Clínicas de un paciente en cada Consulta y Guardar dicha información para su posterior Registro y Uso.

Documentación	Fue Posible Tener copias de respaldo de un Historial Clínico, el formato de recetas y petición de exámenes de Laboratorio. El formato de reporte de resultados en base a requerimientos y el reporte de Historias Clínicas que tiene actualmente el Hospital de la Mujer
----------------------	--

3.2.2. Indagación y elaboración - PRODUCTO BACKLOG

En primera instancia se identifican los principales requisitos los cuáles son:

- Verificar Paciente Nuevo
- Registrar Paciente Nuevo
- Registrar Servicio y Costo
- Ingresar información Inicial del Paciente en el área de Enfermería.
- Registrar Información Principal del paciente por parte del Especialista
- Emitir informes de Historial Clínico y Receta respectiva de cada paciente.
- Controlar el Inventario de Historiales Clínicos

La obtención de los principales requisitos funcionales permite la construcción de una lista de requerimientos detallados que vienen a construir el Product Backlog. Tal como se muestra en la Tabla 3.2.

Tabla 3. 2 Requisitos Funcionales

Fuente: Elaboración Propia

Ref.	Requisitos Funcionales
R1	Implementar Seguridad a Sistema
R2	Crear Usuarios
R3	Registrar al Personal del Hospital
R4	Verificar Pacientes ya Registrados
R5	Registrar Pacientes Nuevos
R6	Registrar Información en Enfermería

R7	Registrar Información en Consultorio
R8	Almacenar Información
R9	Emitir reportes de Historias Clínicas
R10	Emitir reportes de Recetas Clínicas
R11	Facilitar el control de pagos y pacientes atendidos
R12	Registrar y Estructurar los Servicios del Hospital
R13	Realizar reportes de atención por día, mes y año
R14	Facilitar la administración de Usuarios
R15	Permitir la administración del perfil y contraseña a cada usuario

Tabla 3. 3 Requisitos No Funcionales

Fuente: Elaboración Propia

Referencia	Requisitos no funcionales
Interfaz de Usuario	El sistema debe ser ejecutado en un navegador web
Interfaz de Respuesta	La demora no debe exceder los 10 segundos
Acceso	El acceso al sistema debe ser restringido y controlado
Portabilidad	El sistema funciona en cualquier sistema operativo, siempre y cuando cuente con un navegador

A partir de los requerimientos funcionales encontrados en la Tabla 3.2. se genera una lista de Sprints ordenados por prioridad tal como se ve en la Tabla 3.3.

Tabla 3. 4 Tabla de Iteraciones

Fuente: Elaboración Propia

Iteración	Sprint	Prioridad	Detalle descripción
1	Acceso al Sistema	Muy Alta	Generar un menú dinámico a partir de los roles que tengan los usuarios de la base de datos
2	Registro	Muy Alta	Registrar a usuarios y pacientes cada uno dependiendo de los privilegios de usuario
3	Orden de atención	Muy Alta	Crear nueva orden y atender una orden planificada
4	Historias Clínicas	Muy Alta	Crear un reporte de Historial Clínico para su resguardo
5	Resultados	Muy Alta	Administrar los resultados facilitando el acceso a uno determinado
6	Gestión de Información	Media	Facilidad e agregar, modificar y eliminar información de Hospital
7	Gestión de Servicios	Media	Facilidad e agregar, modificar y eliminar información de Servicios ofrecidos
8	Reportes	Media	Generar reportes de Historias Clínicas y Recetas de los pacientes atendidos
9	Gestión de Pacientes	Media	Administración de Pacientes, cambiar información de pacientes
10	Gestión de usuarios	Media	Administración de Usuarios, dar de baja a Usuarios, Agregar y Modificar usuarios.

3.2.3. Negociación

La obtención de requerimientos genera discrepancias entre los usuarios del sistema, principalmente entre el médico y el administrador. Por ello se realiza una negociación donde se puntualiza algunas condiciones de trabajo. En este proyecto son los siguientes:

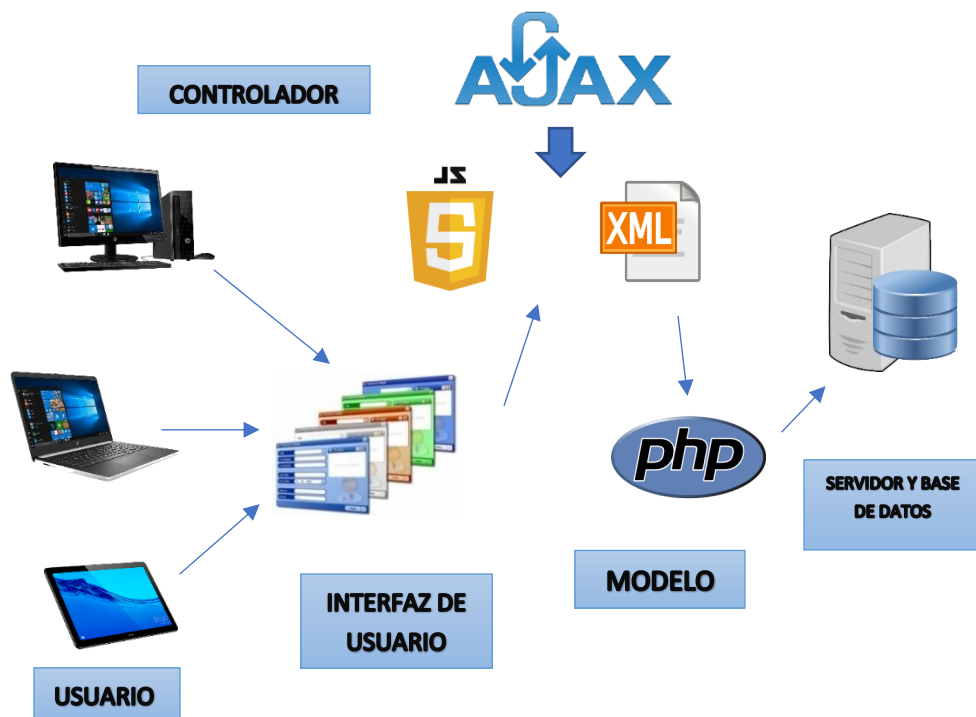
- Se registrará solo a los pacientes nuevos
- Se crearán usuarios con diferentes roles y privilegios.
- El usuario administrador tendrá privilegios de usar las funciones de médico.

3.2.4. Especificación

En base al MVC las vistas vendrían a ser las paginas php, el modelo está conformado por los Scripts PHP que manipulan los datos e interactúan con la base de datos y el controlador está constituido por los Scripts de JavaScript juntamente con las acciones del usuario. (Ver Figura 3.1).

Figura 3. 1 Arquitectura del Sistema

Fuente: Elaboración Propia



3.2.4.1. Actores del Sistema

Los actores que se identifican en el Hospital de la Mujer y en su entorno de administración son los presentados en la Tabla 3.4.

Tabla 3. 5 Actores del Sistema

Fuente: Elaboración Propia

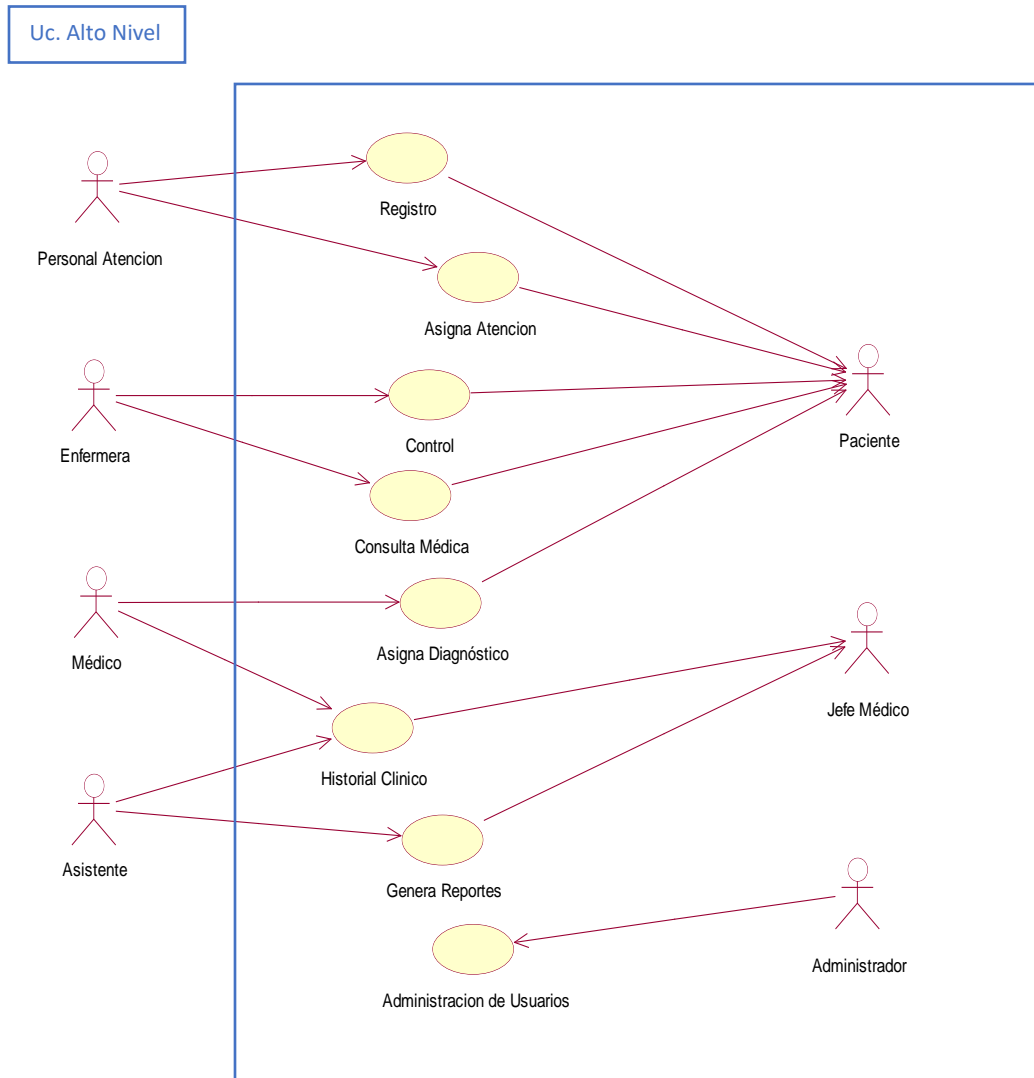
Actor	Objetivo
Personal de Atención	Registra pacientes, toma petición de servicio.
Enfermera	Registra información general de pacientes, previo a la atención de especialista
Médico	Accede a información general y específica de pacientes.
Asistente	Registra información general y específica de pacientes además de llevar el registro de historiales clínicos.
Administrador	Administra usuarios, administra acciones en el sistema, tiene acceso a todo el sistema.

3.2.4.2. Diagramas de caso de uso

La planificación de los Sprints de desarrollo presentados en la tabla 3.4 permite modelar los procesos del contexto mediante un diagrama de casos de uso representado en la Figura 3.2.

Figura 3. 2 Diagrama Casos de uso Principal

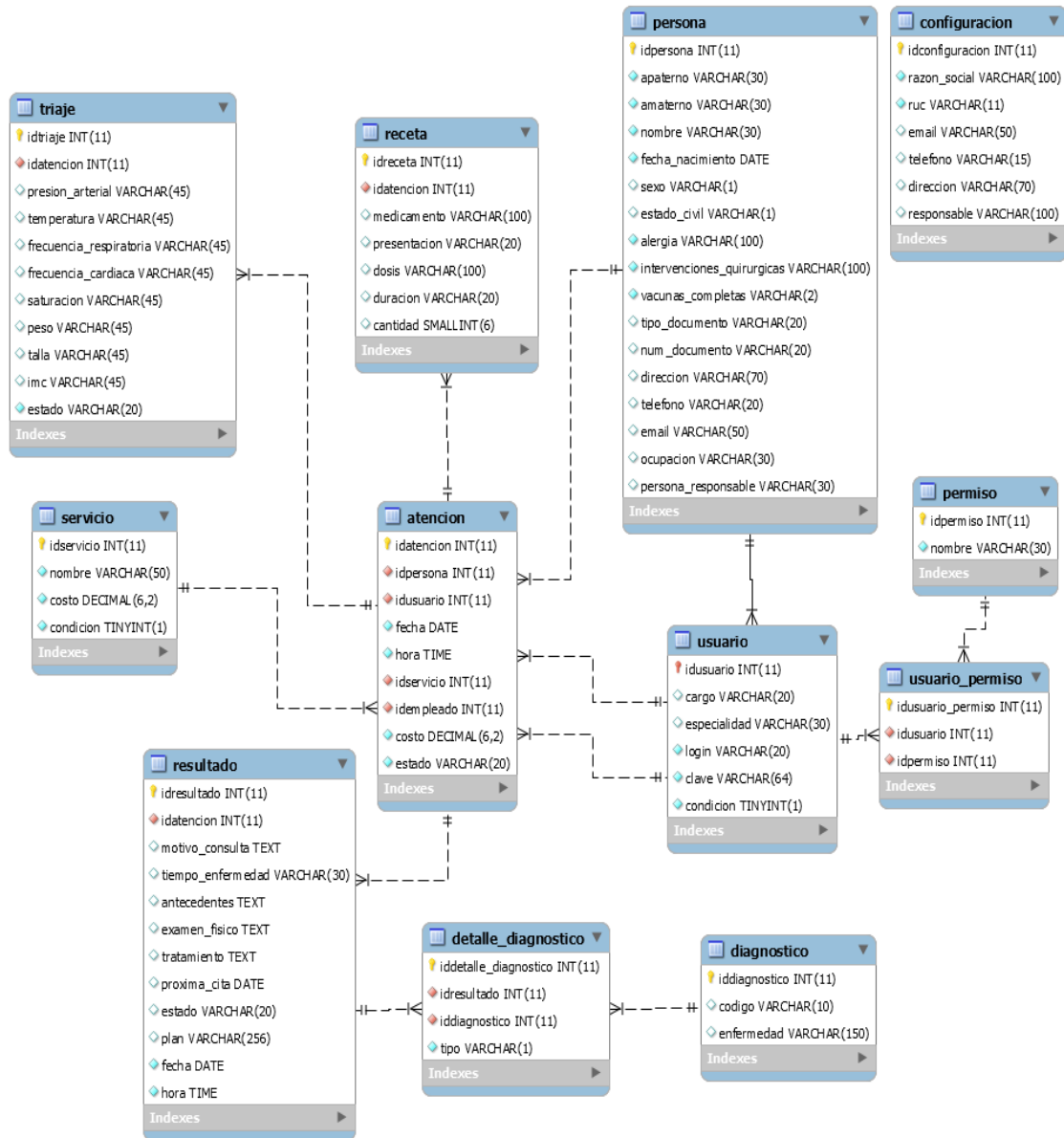
Fuente: Elaboración Propia



3.2.4.3. Base de Datos

Figura 3. 3 Modelo de Base de Datos del Sistema

Fuente: Elaboración Propia



3.3. GAME

3.3.1. Sprint 1: Acceso al Sistema

- Planeación

En la Tabla 3.3 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 6 Planificación del Sprint 1

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		Desde	Hasta	Estado
		02/03/2020	05/03/2020	4 días
Logueo de Usuarios al Sistema	Diseño de Diagramas UML	02/03/2020	03/03/2020	Completado
	Codificación	03/03/2020	04/03/2020	Completado
	Pruebas	05/03/2020	05/03/2020	Completado

A. Diagrama de casos de uso

El diagrama de casos de uso Acceso al sistema abarca los procesos que realizan los usuarios administradores, médico y asistente para ingresar al sistema (Ver figura 3.5).

Figura 3. 4 Diagrama UC Acceso al sistema

Fuente: Elaboración Propia

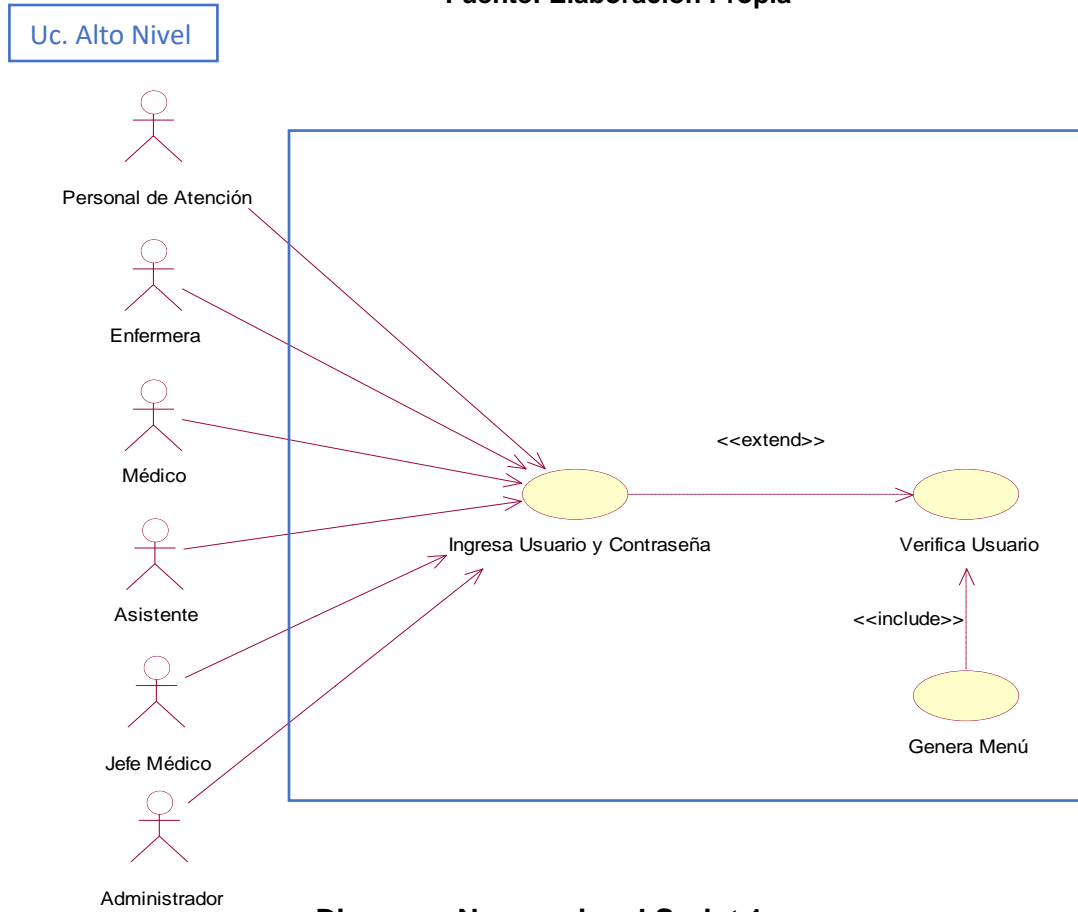
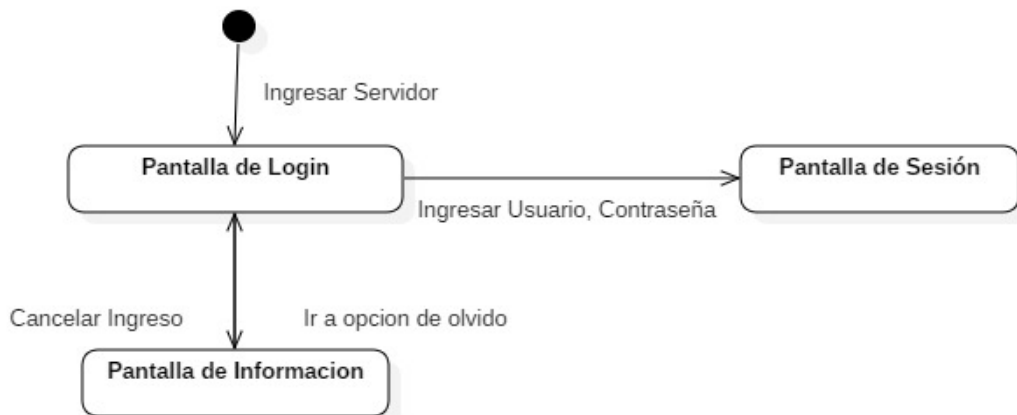


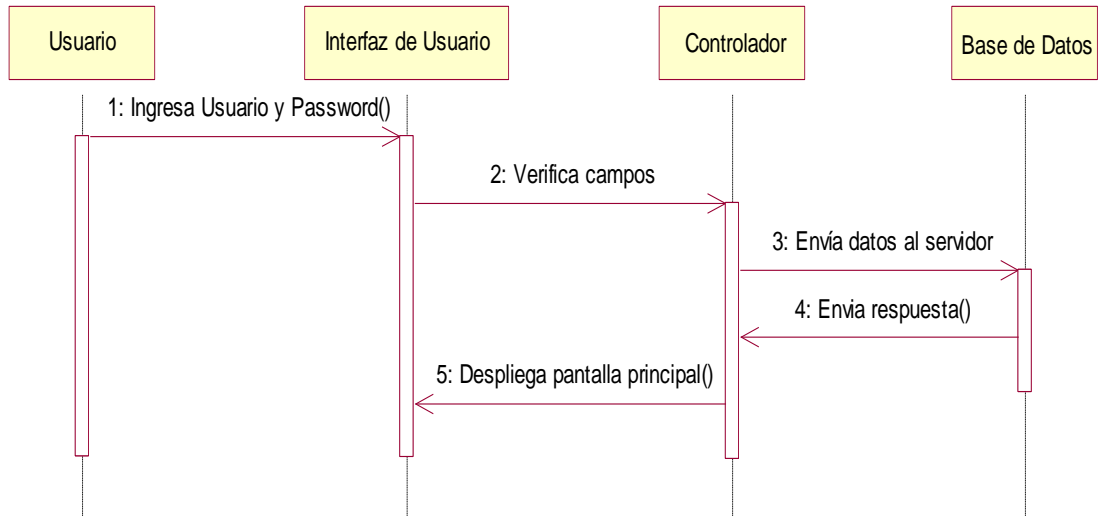
Diagrama Navegacional Sprint 1



B. DIAGRAMA DE SECUENCIA

Figura 3. 5 Diagrama de Secuencia Acceso al Sistema

Fuente: Elaboración Propia



(from Actores)

C. Sprint Backlog

En base a los requisitos R1 y R2 presentados en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación. (Ver Tabla 3.6)

Tabla 3. 7 Sprint Backlog Acceso al Sistema

Fuente: Elaboración Propia

SPRINT-BACKLOG1-ACCESO AL SISTEMA				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
Ingreso al Sistema	Validar información de usuario	Muy Alta	Desarrollo	Terminado
	Generar menú según rol	Muy Alta	Desarrollo	Terminado

- Desarrollo: El Sprint Backlog presentado en la tabla 3.6. permite generar historias de usuarios que describen puntualmente los subprocesos: Validar información del usuario (Ver tabla 3.7), Generar menú según rol (Ver tabla 3.8).

Tabla 3. 8 Validar Información del Usuario

Fuente: Elaboración Propia

Nombre de historia de Usuario: Validar información de usuario	
Usuario: Administrador, Jefe Médico, Asistente, Enfermera, Personal de Atención	Iteración Asignada: 1
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Verificar el Usuario y la contraseña ingresados para luego crear una sesión	
Observaciones: Ninguna	

Tabla 3. 9 Generar menú según Rol

Fuente: Elaboración Propia

Nombre de historia de Usuario: Generar menú según rol	
Usuario: Administrador, Jefe Médico, Asistente, Enfermera, Personal de Atención	Iteración Asignada: 1
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Desplegar una lista de opciones o páginas a las que puede acceder el usuario.	
Observaciones: El menú cambia según usuario	

- **Revisión:**

El desarrollo del Sprint en base a las historias de usuario presentadas en las tablas 3.7 y 3.8 genera una interfaz gráfica o pantalla para la administración e interpretación de las órdenes. Mismo que es presentada a los usuarios para su revisión (Ver Figura 3.8).

Figura 3. 6 Interfaz Gráfica Sprint 1
Fuente: Elaboración Propia



3.3.2. Sprint 2: Registro

- **Planeación:**

En la Tabla 3.10 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 10 Planificación del Sprint 2

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		09/03/2020	17/03/2020	7 días
		Desde	Hasta	Estado

Registro de Pacientes y Usuarios	Diseño de Diagramas UML	09/03/2020	11/03/2020	Completado
	Codificación	12/03/2020	16/03/2020	Completado
	Pruebas	17/03/2020	17/03/2020	Completado

A. Diagrama de Casos de Uso

El diagrama de casos de uso Registro abarca los procesos que realizan los usuarios administradores, médico y asistente para registrar usuarios y pacientes (ver figura 3.7).

Figura 3. 7 Diagrama de Casos de uso Registro

Fuente: Elaboración Propia

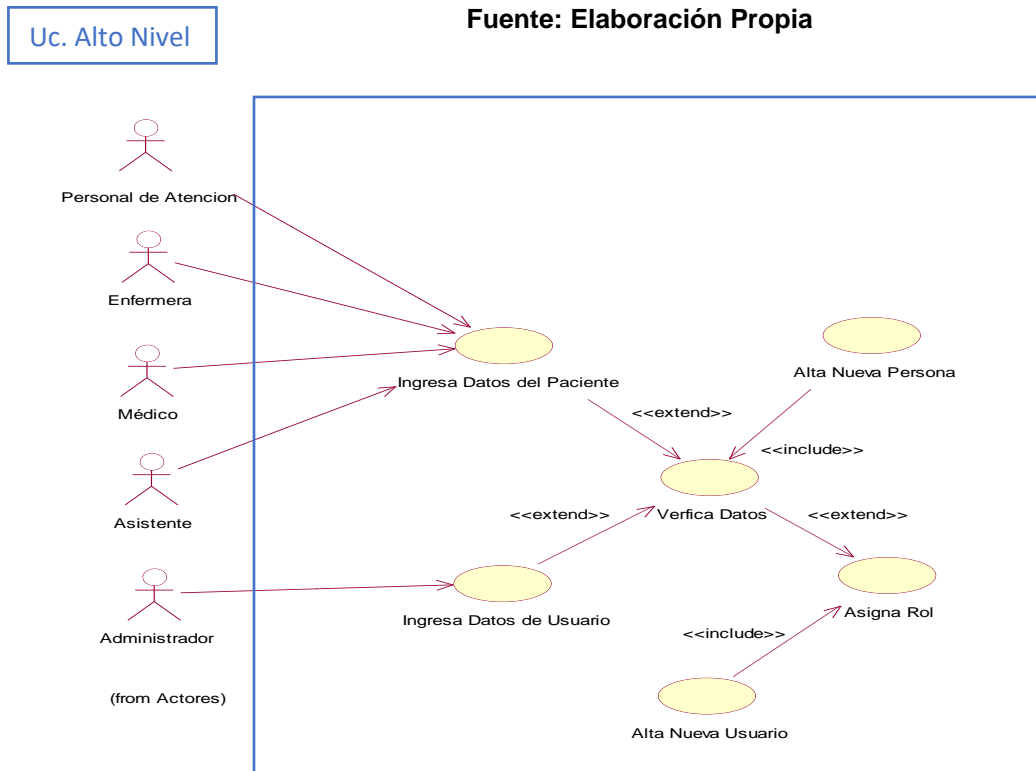
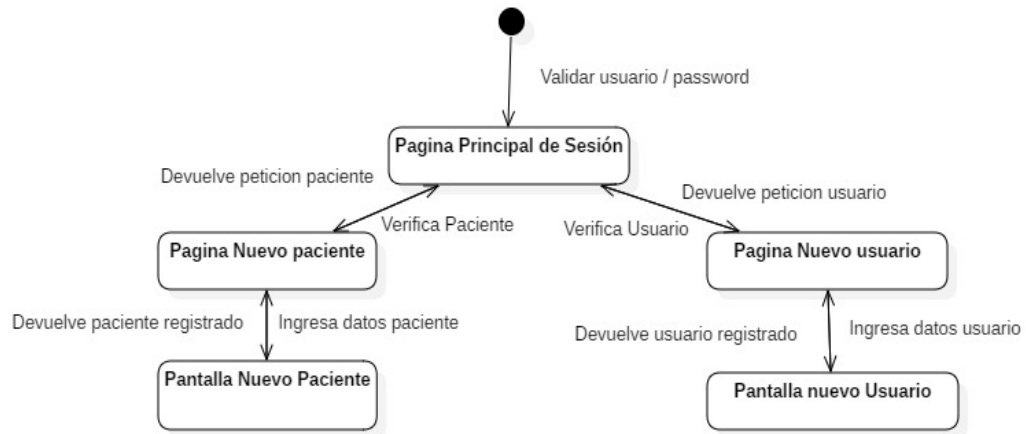


Diagrama Navegacional Sprint 2

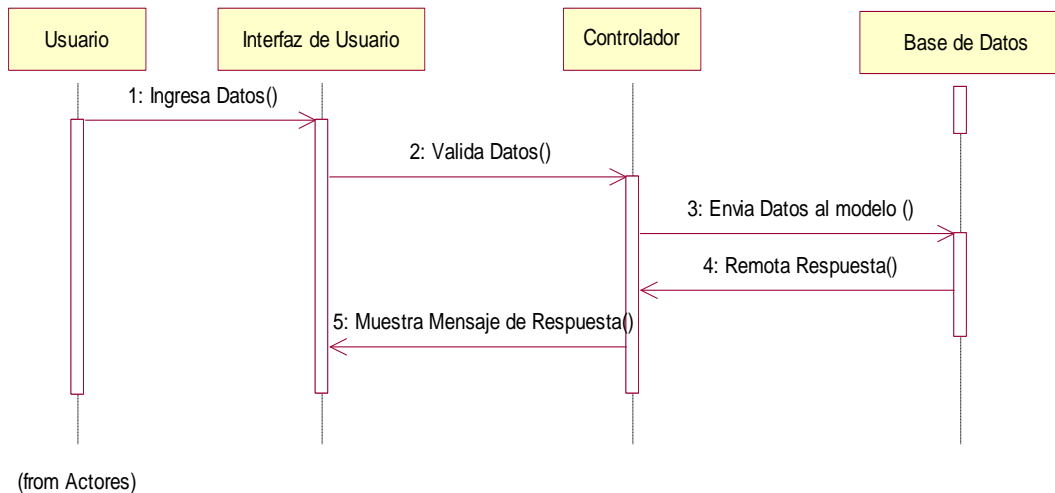


B. Diagrama de Secuencia

El diagrama de secuencia Registro refleja los procesos necesarios para llevar a cabo el registro de un paciente o de un nuevo usuario (Ver figura 3.8).

Figura 3. 8 Diagrama de Secuencia Registro

Fuente: Elaboración Propia



En base a la los requisitos R3 y R4 presentados en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación. (Ver Tabla 3.11)

Tabla 3. 11 Sprint Backlog Registro

Fuente: Elaboración Propia

SPRINT-BACKLOG2-REGISTRO				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
Registro de Pacientes	Altas de Pacientes	Muy Alta	Desarrollo	Terminado
	Crear Historial de Visitas al Hospital	Muy Alta	Desarrollo	Terminado
Registro de Usuarios	Altas de Usuarios	Alta	Desarrollo	Terminado
	Asignar un Rol a un usuario	Alta	Desarrollo	Terminado

- **Desarrollo:**

El Sprint Backlog presentado en la Tabla 3.11 permite generar historias de usuarios que describen puntualmente los procesos: Registro de pacientes (Ver Tabla 3.12), Registro de usuarios (Ver Tabla 3.13).

Tabla 3. 12 Registro de Pacientes

Fuente: Elaboración Propia

Nombre de historia de Usuario: Registro de Pacientes	
Usuario: Jefe Médico, Asistente, Enfermera, Personal de Atención	Iteración Asignada: 2
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Dar de Alta un nuevo paciente.	

Observaciones: Se registra al paciente una única vez

Tabla 3. 13 Generar menú según Rol

Fuente: Elaboración Propia

Nombre de historia de Usuario: Registro de Usuarios

Usuario: Administrador

Iteración Asignada: 4

Prioridad: Alta

Riesgo en desarrollo: Media

Descripción: Registrar un nuevo usuario asignando un rol para que pueda acceder a diferentes módulos

Observaciones: El usuario y contraseña serán arbitrarios

- **REVISION**

El desarrollo del Sprint en base a las historias de usuario presentadas en las tablas 3.12 y 3.13 generan interfaces gráficas o pantallas para administrar el registro de pacientes y de usuarios. Mismas que son presentadas a los usuarios para su revisión (Ver Figura 3.9 y 3.10).

Figura 3. 9 Interfaz gráfica Registro Paciente

Fuente: Elaboración Propia

Paciente Nuevo

Apellido Paterno (*): <input type="text" value="apellido Paterno"/>	Apellido Materno (*): <input type="text" value="Apellido Materno"/>	Nombre (*): <input type="text" value="nombre"/>
Fecha de Nacimiento (*): <input type="text" value="dd/mm/aaaa"/>	Sexo: <input type="text" value="Femenino"/>	Estado Civil: <input type="text" value="Soltero"/>
Tipo Documento: <input type="text" value="DNI"/>	Número Documento: <input type="text" value="Documento"/>	Dirección: <input type="text" value="Dirección"/>
Teléfono: <input type="text" value="Teléfono"/>	Email: <input type="text" value="Email"/>	
Ocupación: <input type="text" value="Ocupacion"/>	Persona Responsable: <input type="text" value="Persona responsable"/>	
Alergias: <input type="text" value="Alergia"/>	Intervenciones quirúrgicas: <input type="text" value="Intervenciones quirúrgicas"/>	Vacunas completas: <input type="text" value="SI"/>

Figura 3. 10 Interfaz de Usuario Registro

Fuente: Elaboración Propia

Usuario

Nombre(*): <input type="text" value="Nombre"/>	Apellido Paterno (*): <input type="text" value="apellido Paterno"/>	Apellido Materno (*): <input type="text" value="apellido Materno"/>
Fecha de Nacimiento (*): <input type="text" value="dd/mm/aaaa"/>	Sexo (*): <input type="text"/>	Estado civil (*): <input type="text"/>
Tipo Documento(*): <input type="text"/>	Número(*): <input type="text" value="Documento"/>	Dirección: <input type="text" value="Dirección"/>
Teléfono: <input type="text" value="Teléfono"/>	Email: <input type="text" value="Email"/>	Ocupación: <input type="text" value="Ocupacion"/>
Cargo: <input type="text" value="Cargo"/>	Especialidad: <input type="text" value="Especialidad"/>	Logín (*): <input type="text" value="Login"/>

Clave (*):

Permisos:

- escritorio
- pacientes
- clinica
- atencion
- triaje
- resultado
- consultas

Guardar
Cancelar

3.3.3. Sprint 3: Orden de Atención

- **PLANEACION:**

En la Tabla 3.14 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 14 Planificación del Sprint 3

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		23/03/2020	31/03/2020	7 días
		Desde	Hasta	Estado
Registro de Nueva orden de Atención de Paciente	Diseño de Diagramas UML	23/03/2020	25/03/2020	Completado
	Codificación	26/03/2020	30/03/2020	Completado
	Pruebas	31/03/2020	31/03/2020	Completado

A. Diagramas de Caso de Uso

El diagrama de casos de uso Orden de Atención de Hospital, abarca los procesos que realizan el personal de atención, médico, enfermeras y asistentes para crear y atender una orden de servicio (Ver figura 3.11).

Figura 3. 11 Diagrama UC Orden de atención
Fuente: Elaboración Propia

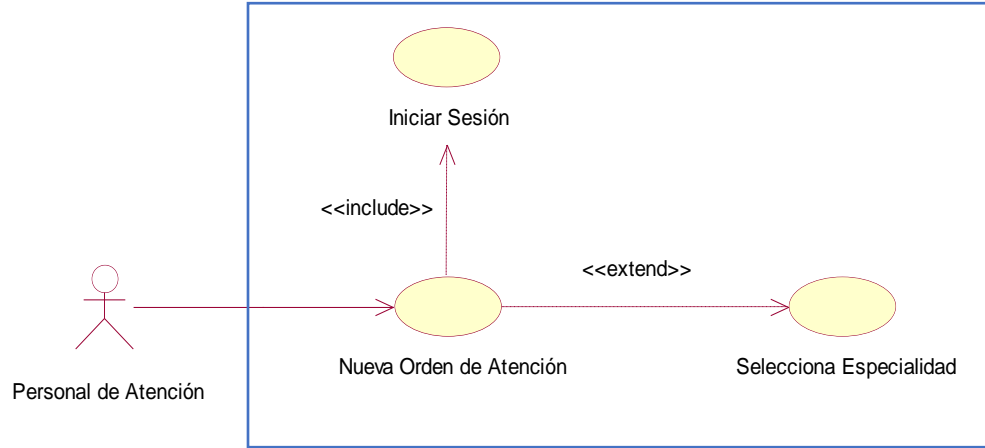
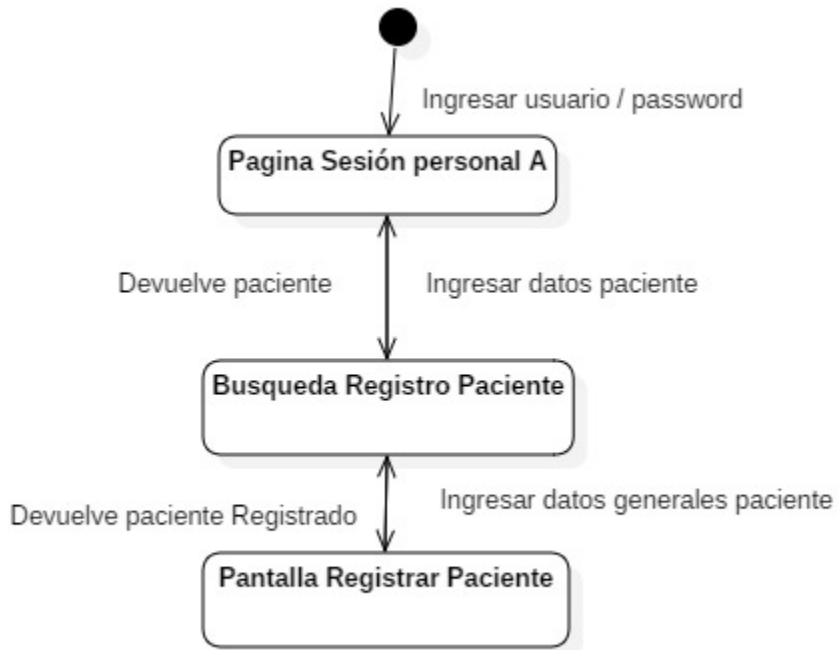
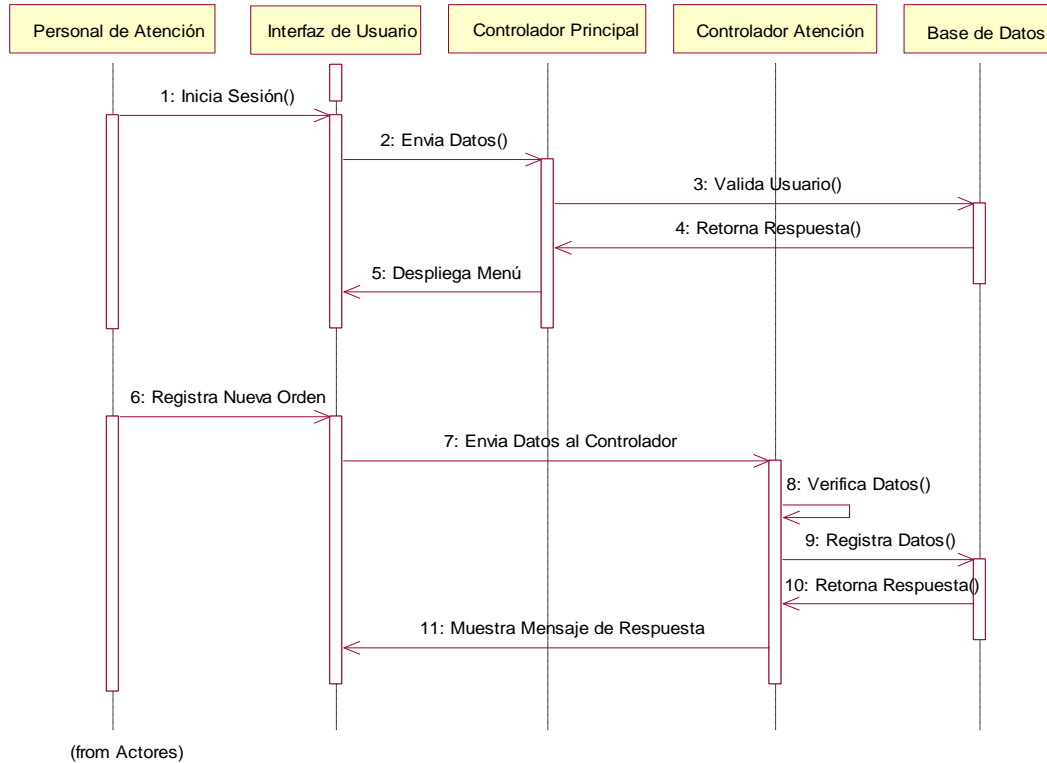


Diagrama Navegacional Sprint 3



B. Diagrama de Secuencia

Figura 3. 12 Diagrama de secuencia Orden de Laboratorio
Fuente: Elaboración Propia



C. Sprint Backlog

En base a la los requisitos R5 y R6 presentados en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación. (Ver Tabla 3.15).

Tabla 3. 15 Sprint Backlog Atención

Fuente: Elaboración Propia

SPRINT-BACKLOG3-REGISTRO DE ATENCION				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
	Ingreso Datos para Atención	Muy Alta	Desarrollo	Terminado

Nueva Orden de Atención	Selecciona Especialista Requerido	Muy Alta	Desarrollo Terminado
Atención Pendiente	Visualiza Atención pendiente	Muy Alta	Desarrollo Terminado

- **Desarrollo:**

El Sprint Backlog presentado en la Tabla 3.15 permite generar historias de usuarios que describen puntualmente los procesos: Nueva atención (Ver tabla 3.16).

Tabla 3. 16 Nueva Orden

Fuente: Elaboración Propia

Nombre de historia de Usuario: Nueva Orden de Atención	
Usuario: Personal de Atención	Iteración Asignada: 5
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Registrar una orden de atención de servicio nueva seleccionando especialista requerido por el paciente y Visualizar atención pendiente.	
Observaciones: El paciente debe encontrarse registrado antes de solicitar una orden de atención de servicio.	

- **Revisión:**

El desarrollo del Sprint en base a las historias de usuario presentada en la tabla 3.16 genera una interfaz gráfica o pantalla para administrar las órdenes de atención de servicio.

Figura 3. 13 Interfaz Gráfica Nueva Orden
Fuente: Elaboración Propia

Atención Nueva






Número Documento (*): Paciente (*):

Servicio (*): Especialista (*): Costo:

Figura 3. 14 Interfaz de Usuario Orden Pendiente
Fuente: Elaboración Propia

Atención Nueva

Copy Excel CSV PDF Buscar:

Opciones ^	Fecha	Registrador	Servicio	Especialista	Paciente	DNI	Edad	Costo	Estado
 	2020-06-25 - 21:35:30	Gonzales Paz Cesar	Ginecología	A A A	A A A	12345	0 años, 0 meses, 24 días	0.00	<input type="button" value="Registrado"/>
 	2020-06-24 - 10:36:47	Gonzales Paz Cesar	Ginecología	A A A	pet pet pet	23232323	5 años, 3 meses, 20 días	90.00	<input type="button" value="Triage"/>
	2020-06-24 - 02:37:06	Gonzales Paz Cesar	Ginecología	Gonzales Paz Cesar	Gonzales Paz Cesar	11111111	3 años, 3 meses, 17 días	60.00	<input type="button" value="Atendido"/>
Opciones	Fecha	Registrador	Servicio	Especialista	Paciente	DNI	Edad	Costo	Estado

Mostrando 1 a 3 de 3 registros Anterior Siguiente

3.3.4. Sprint 4: Historias Clínicas

- **Planeación:**

En la Tabla 3.17 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 17 Planificación del Sprint 4

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		06/04/2020	15/04/2020	8 días
Registro de Información en Historial Clínico	Diseño de Diagramas UML	Desde	Hasta	Estado
		06/04/2020	08/04/2020	Completado
	Codificación	09/04/2020	14/04/2020	Completado
	Pruebas	15/04/2020	15/04/2020	Completado

A. Diagramas de casos de uso

El diagrama de casos de uso Historias Clínicas abarca los procesos que realizan los usuarios médico y enfermera para administrar la información de los pacientes (Ver Figura 3.15).

Figura 3. 15 Diagrama UC Historial Clínico

Fuente: Elaboración Propia

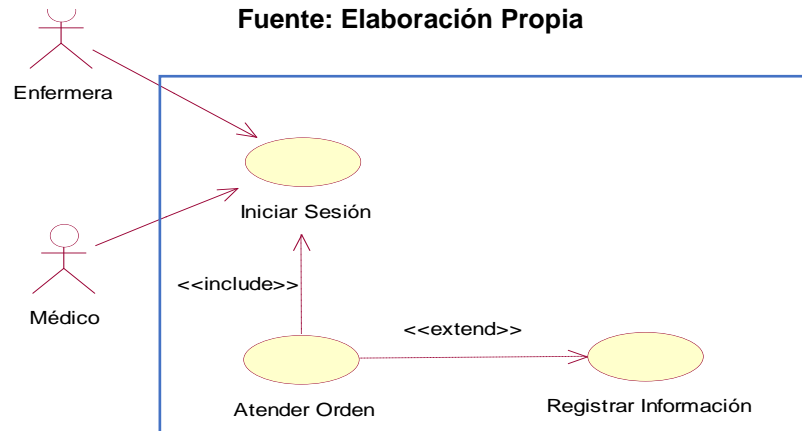
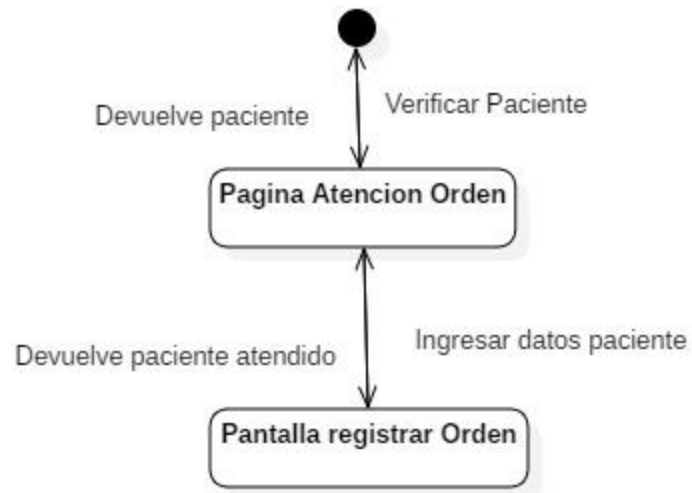


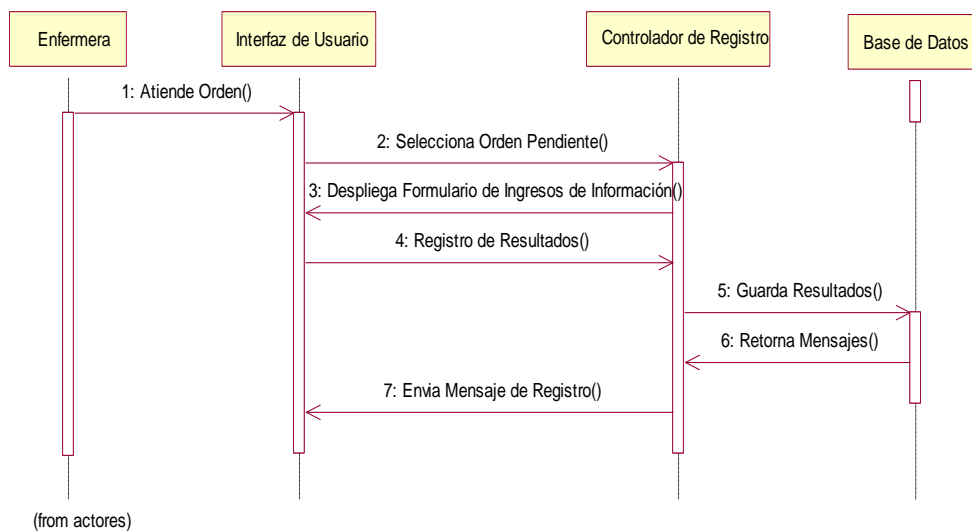
Diagrama Navegacional Sprint 4

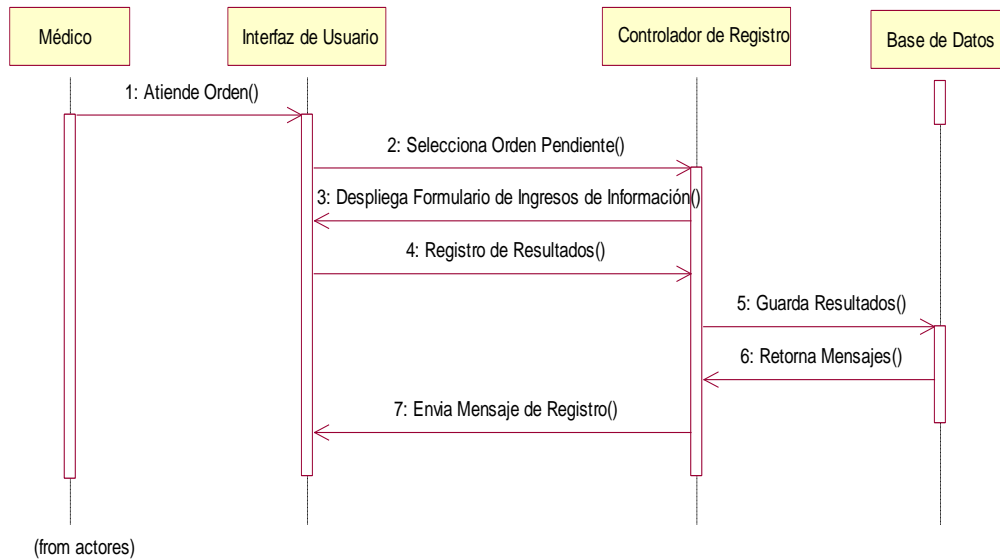


B. Diagrama de Secuencia

El diagrama de secuencia Historia Clínica, refleja los procesos necesarios para llevar a cabo el registro de información de los pacientes (Ver figura 3.16).

Figura 3. 16 Diagrama de secuencia de uso Historial Clínico
Fuente: Elaboración Propia





C. Sprint Backlog

En base a los requisitos R5 y R6 presentados en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación. (Ver Tabla 3.18).

Tabla 3. 18 Sprint Backlog Registro Historial

Fuente: Elaboración Propia

SPRINT-BACKLOG4-REGISTRO DE HISTORIAL CLINICO				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
Registro de Información de Paciente	Ingresa Información de Paciente	Muy Alta	Desarrollo	Terminado
	Guarda la información clínica de cada Paciente	Muy Alta	Desarrollo	Terminado
Atención Pendiente	Visualiza Paciente Pendiente	Muy Alta	Desarrollo	Terminado

- **Desarrollo:**

El Sprint Backlog presentado en la Tabla 3.16 permite generar historias de usuarios que describen puntualmente La Información (Ver tabla 3.19).

Tabla 3. 19 Registro Historial Clínico

Fuente: Elaboración Propia

Nombre de historia de Usuario: Registro Historial Clínico	
Usuario: Médico, Enfermera	Iteración Asignada: 6
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Registrar información de cada paciente.	
Observaciones: El paciente debe encontrarse en espera para la atención de servicio.	

- **Revisión:**

El desarrollo del Sprint en base a las historias de usuario presentadas en la tabla 3.17 generan interfaces gráficas o pantallas para administrar la información de cada paciente. Mismas son guardadas en la base de datos para su revisión.

Figura 3. 17 Interfaz Gráfica Historia Clínica Enfermería

Fuente: Elaboración Propia

Cola Enfermeria

Paciente:	DNI:	Edad:	
AAA	12345	0 años, 0 meses, 24 días	
Servicio:	Especialista:		
Ginecología	AAA		
Presion Arterial (mmHg):	Temperatura (C°):		
Presion Arterial	temperatura		
Frecuencia Respiratoria (x Minuto):	Frecuencia Cardiaca (x Minuto):	Saturacion O2:	
Frecuencia Respiratoria	Frecuencia Cardiaca	Saturacion	
Peso: (Kg) utilice punto para decimales	Talla: (cm)	Imc:	Estado:
Peso	Talla	Imc	estado

Figura 3. 18 Interfaz Gráfica Historia Clínica Médico Especialista
Fuente: Elaboración Propia

Motivo de la Consulta:

TE:

Antecedentes:

Alergias:

Intervenciones Quirúrgicas:

Vacunas completas:

Examen Físico:

Código (Nombre)	Diagnósticos	Aplicar
<input type="text"/>		<input type="button" value="Borrar"/> <input type="button" value="Tipo"/> <input type="button" value="Enfermedad"/>
<input type="button" value="Consultar"/>		

Tratamiento:

Plan:

Estudios:

Próxima Cita:
 dd/mm/aaaa 📅

Receta
+

Borrar	Medicamento	Presentación	Dosis	Duración	Cantidad
🗑	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>

↶ Cancelar

3.3.5. Sprint 5: Resultados

- **Planeación:**

En la Tabla 3.19 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 20 Planificación del Sprint 5

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		20/04/2020	29/04/2020	8 días
Registro de Información Previa en Historial Clínico	Diseño de Diagramas UML	20/04/2020	23/04/2020	Completado
	Codificación	24/04/2020	28/04/2020	Completado
	Pruebas	29/04/2020	29/04/2020	Completado

A. Diagramas de casos de Uso

El diagrama de casos de uso Resultados abarca los procesos que realizan los usuarios médico y asistente para administrar los resultados de los pacientes (Ver Figura 3.16).

Figura 3. 19 Diagrama de Casos de Uso Resultados

Fuente: Elaboración Propia

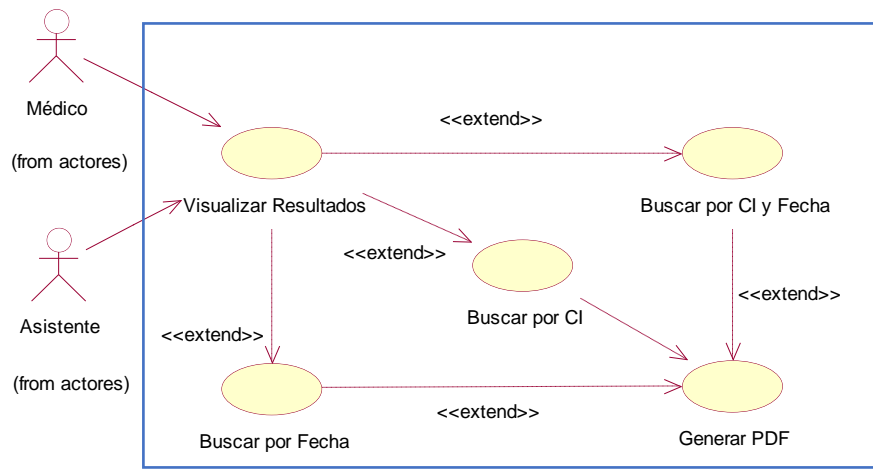


Diagrama Navegacional Sprint 5

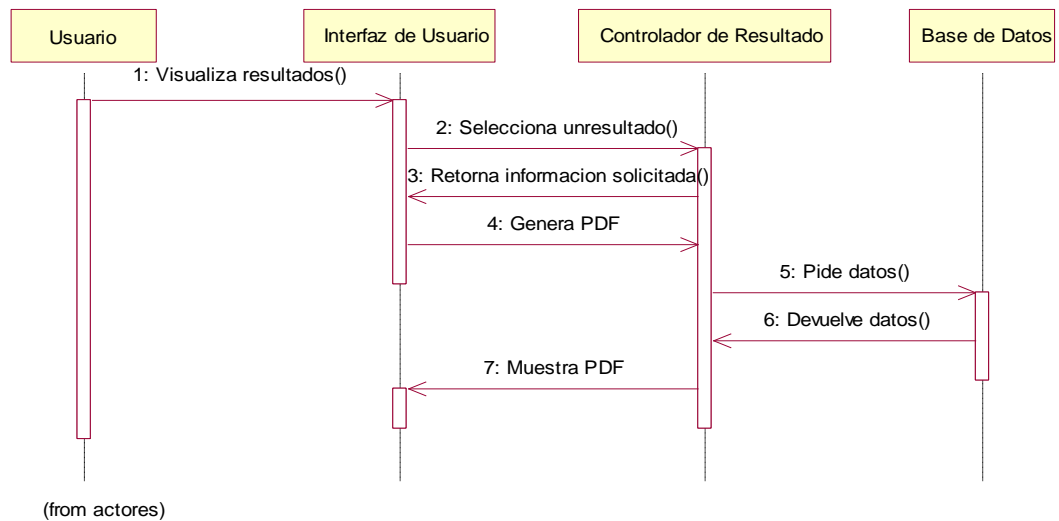


B. Diagrama de Secuencia

El diagrama de secuencia Resultados refleja los procesos necesarios para llevar a cabo la administración de los resultados de los pacientes (Ver figura 3.17).

Figura 3. 20 Diagrama de Secuencia Resultados

Fuente: Elaboración Propia



C. Sprint Backlog

En base a la los requisitos R7 y R8 presentados en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación.

Tabla 3. 21 Sprint Backlog Resultados

Fuente: Elaboración Propia

SPRINT-BACKLOG5-GESTION DE RESULTADOS				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
	Visualizar, buscar resultado	Muy Alta	Desarrollo	Terminado

Gestión de Resultados	Generar PDF	Muy Alta	Desarrollo Terminado
-----------------------	-------------	----------	----------------------

- **Desarrollo:**

El Sprint Backlog presentado en la Tabla 3.19 permite generar historias de usuarios que describen puntualmente los subprocesos: Buscar resultado (Ver tabla 3.20), Generar pdf (ver tabla 3.21).

Tabla 3. 22 Buscar Resultados

Fuente: Elaboración Propia

Nombre de historia de Usuario: Buscar Resultado	
Usuario: Médico, Asistente	Iteración Asignada: 7
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Seleccionar resultados según criterios del usuario.	
Observaciones: Ninguna	

Tabla 3. 23 Generar PDF de Resultado

Fuente: Elaboración Propia

Nombre de historia de Usuario: Generar PDF de Resultado	
Usuario: Médico, Asistente	Iteración Asignada: 7
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Generar un documento PDF en el que se detalla la información de paciente y sus resultados.	
Observaciones: El documento generado presenta la información pertinente de cada paciente y la fecha de próxima cita.	

- **Revisión:**

El desarrollo del Sprint en base a las historias de usuario presentadas en las tablas 3.20 y 3.21 generan interfaces gráficas o pantallas para la administración de los resultados. Mismas que son presentadas a los usuarios para su revisión (Ver Figura 3.17 y 3.18).


Figura 3. 21 Interfaz Gráfica 1 Sprint 4

Fuente: Elaboración Propia

Opciones	Apellido Paterno	Apellido Materno	Nombre	Fecha de Nacimiento	Sexo	Estado Civil	Numero de Documento	Direccion	Teléfono
	A	A	A	2020-06-01	M	S	12345	Avenida Kollasuyo	12121212
	Gonzales	Paz	Cesar	2017-03-08	M	S	11111111	Ceja	12345678
	pet	pet	pet	2015-03-05	M	C	23232323	Avenida Kollasuyo	2121212

Figura 3. 22 Reportes de Resultados

Fuente: Elaboración Propia

 HOSPITAL DE LA MUJER Av. Saavedra N° 2273 La Paz - Bolivia			
HISTORIA CLINICA			
APELLIDOS Y NOMBRES: Gonzales Paz Cesar			
FECHA NACIMIENTO	2017-03-08	DNI	11111111
EDAD	3 años, 3 meses, 17 días	OCUPACIÓN	Doctor Especialista
SEXO	M	PERSONA RESPONSABLE	Saori Kido
ESTADO CIVIL	S	FECHA ATENCIÓN	2020-06-24
DOMICILIO	Ceja	HORA ATENCIÓN	03:14:21
MOTIVO DE LA CONSULTA			T.E
Dolor Cabeza			Habilitado
ANTECEDENTES			
Migraña			
ALERGIAS	VACUNAS COMPLETAS	INTERVENCIONES QUIRÚRGICAS	
Ninguna	SI	si	
SIGNOS VITALES			
PRESIÓN ARTERIAL (mmhg)	654	SATURACIÓN (%)	89
TEMPERATURA (°C)	43	PESO ACTUAL (Kg)	89
FRECUENCIA RESPIRATORIA (x')	78	TALLA (cm)	135
FRECUENCIA CARDIACA (x')	34	IMC	48.83
EXAMEN FISICO			
Bueno			
DIAGNÓSTICO			PLAN
Fiebre Tifoidea			TIPO: P Mensual
TRATAMIENTO			
Paracetamol			
PROXIMA CITA			
2020-07-12			

3.3.6. Sprint 6: Gestión de Servicios

- **Planeación:**

En la Tabla 3.22 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 24 Planificación del Sprint 6

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		04/05/2020	13/05/2020	8 días
Gestión de Servicios	Diseño de Diagramas UML	Desde	Hasta	Estado
		04/05/2020	06/05/2020	Completado
	Codificación	07/05/2020	12/05/2020	Completado
	Pruebas	13/05/2020	13/05/2020	Completado

A. Diagrama de casos de uso

El diagrama de casos de uso Gestión de servicios abarca los procesos que realiza el usuario administrador para gestionar los servicios ofrecidos por el Hospital de la Mujer. Ver figura 3.28.

Figura 3. 23 Diagramas UC Gestión de Servicios

Fuente: Elaboración Propia

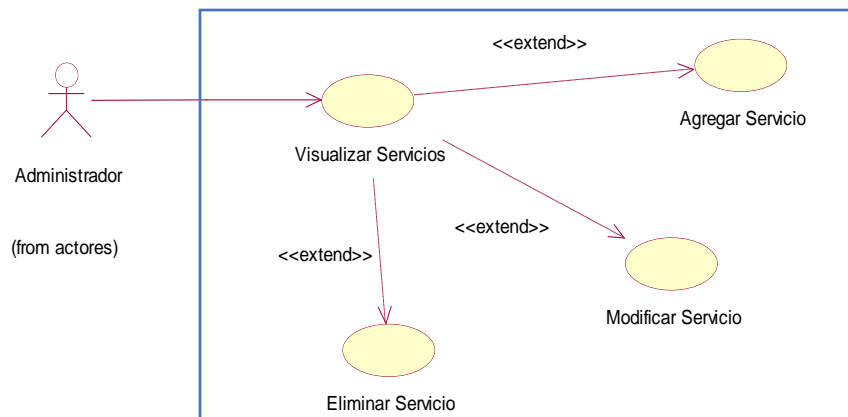
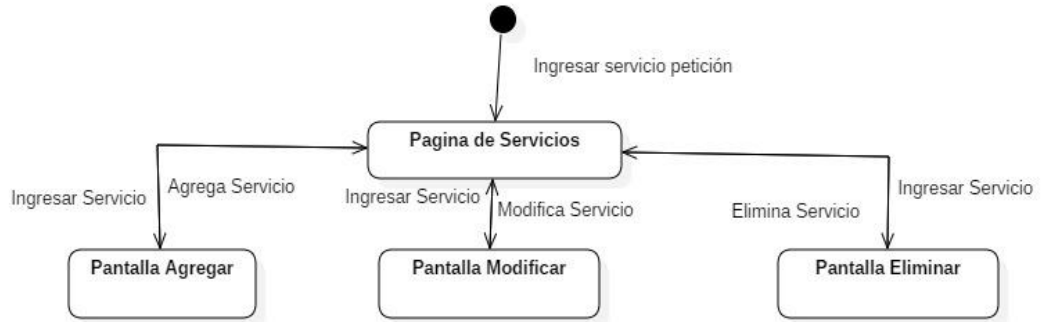


Diagrama Navegacional Sprint 6

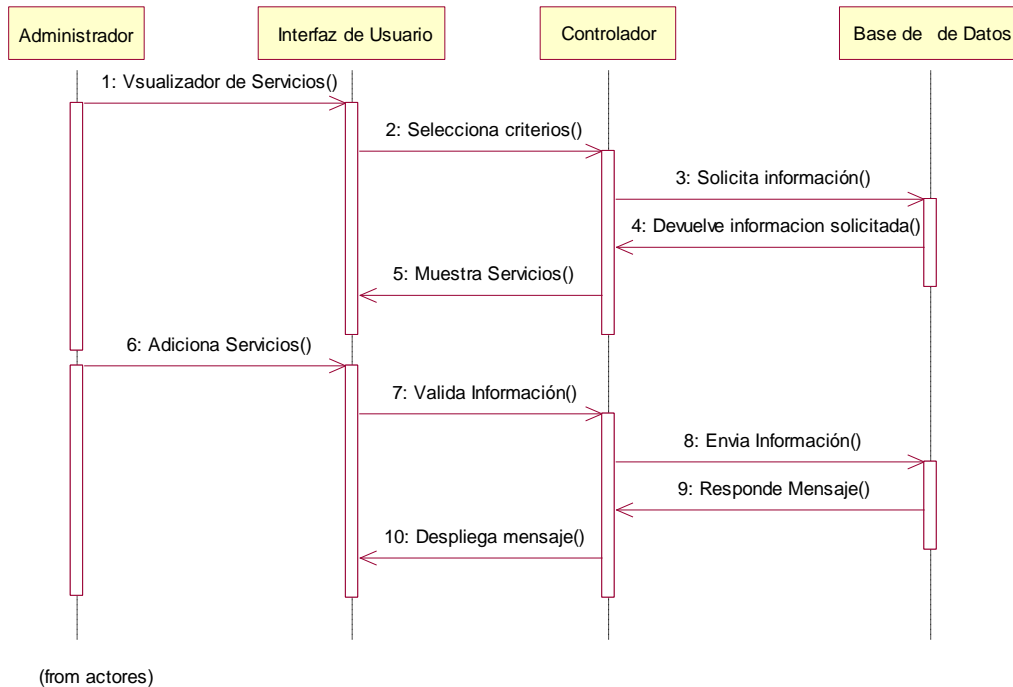


B. Diagrama de secuencia

El diagrama de secuencia Gestión de servicios refleja la planeación de los procesos por los que atraviesa el usuario médico para gestionar los servicios brindados por el laboratorio (Ver figura 3.29).

Figura 3. 24 Diagrama de secuencia Gestión de Servicios

Fuente: Elaboración Propia



C. Sprint Backlog

En base al requisito R11 presentados en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación. (Ver Tabla 3.23).

Tabla 3. 25 Sprint Backlog Servicios

Fuente: Elaboración Propia

SPRINT-BACKLOG6-GESTION DE SERVICIOS				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
Gestión de Servicios	Adicionar Servicio	Muy Alta	Desarrollo	Terminado
	Modificar Servicio	Muy Alta	Desarrollo	Terminado
	Eliminar Servicio	Media	Desarrollo	Terminado

- **Desarrollo:**

El Sprint Backlog presentado en la tabla 3.23 permite generar historias de usuarios que describen puntualmente los subprocesos: Adicionar servicio (ver tabla 3.24), Modificar servicio (Ver tabla 3.25), Eliminar servicio (Ver tabla 3.26).

Tabla 3. 26 Adicionar Servicio

Fuente: Elaboración Propia

Nombre de historia de Usuario: Adicionar Servicio	
Usuario: Usuario Administrador	Iteración Asignada: 6
Prioridad: Muy Alta	Riesgo en desarrollo: Media
Descripción: Agregar un servicio de cualquier tipo, creando y agregando opciones.	
Observaciones: Ninguna	

Tabla 3. 27 Modificar Servicio

Fuente: Elaboración Propia

Nombre de historia de Usuario: Modificar Servicio

Usuario: Usuario Administrador	Iteración Asignada: 6
Prioridad: Muy Alta	Riesgo en desarrollo: Media

Descripción: Modificar algún valor de determinado servicio.

Observaciones: Ninguna.

Tabla 3. 28 Eliminar Servicio

Fuente: Elaboración Propia

Nombre de historia de Usuario: Generar PDF de Resultado

Usuario: Usuario Administrador	Iteración Asignada: 6
Prioridad: Muy Alta	Riesgo en desarrollo: Media

Descripción: Eliminar de la base de datos servicios que ya no se brindaran.

Observaciones: No se puede eliminar un servicio que ya fue atendido alguna vez.

- **Revisión:**

El desarrollo del Sprint en base a las historias de usuario presentadas en las tablas 3.24, 3.25 y 3.26 genera una interfaz gráfica o pantalla de administración de los servicios ofrecidos por el laboratorio. Misma que es presentada a los usuarios para su revisión (Ver Figura 3.20).

Figura 3. 25 Interfaz Gráfica Sprint 6

Fuente: Elaboración Propia

The screenshot shows a web interface for 'Servicios'. At the top left, there is a header 'Servicios' and a green button labeled 'Agregar'. Below the header, there are four buttons: 'Copy', 'Excel', 'CSV', and 'PDF'. To the right of these buttons is a search box labeled 'Buscar:'. Below the search box is a table with the following columns: 'Opciones', 'Nombre', 'Costo', and 'Estado'. The table contains one row with the following data: 'Ginecología', '60.00', and 'Activado'. Below the table, there is a footer that says 'Mostrando 1 a 1 de 1 registros' and navigation buttons for 'Anterior', '1', and 'Siguiete'.

Figura 3. 26 Interfaz Gráfica Sprint 6

Fuente: Elaboración Propia

The screenshot shows a web interface for 'Servicios'. At the top left, there is a header 'Servicios'. Below the header, there are two input fields: 'Nombre (*)' and 'Costo (*)'. Below the input fields, there are two buttons: 'Guardar' and 'Cancelar'.

3.3.7. Sprint 7: Reportes

- **Planeación:**

En la Tabla 3.27 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 29 Planificación del Sprint 7

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		18/05/2020	27/05/2020	8 días
Reportes	Diseño de Diagramas UML	Desde	Hasta	Estado
		18/05/2020	21/05/2020	Completado
	Codificación	22/05/2020	25/05/2020	Completado
	Pruebas	25/05/2020	27/05/2020	Completado

A. Diagramas de Casos de uso

El diagrama de casos de uso Reportes abarca los procesos que realiza el usuario médico para generar reportes de inventarios y de gestión del Laboratorio Clínico (Ver figura 3.33).

Figura 3. 27 Diagrama UC Reportes

Fuente: Elaboración Propia

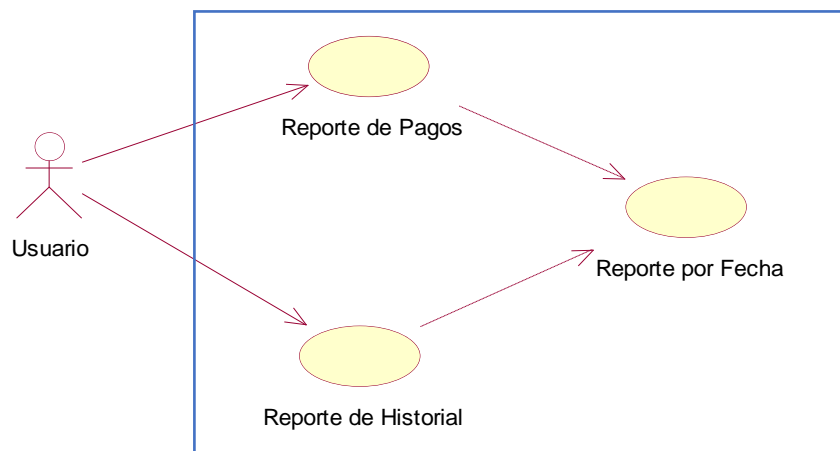
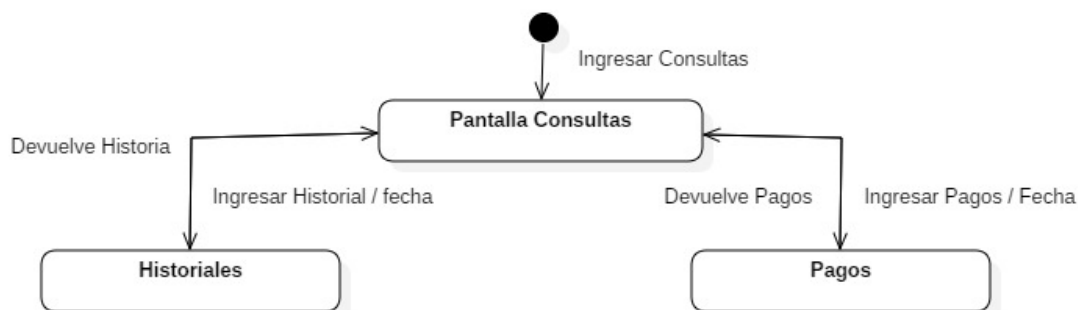


Diagrama Navegacional Sprint 7

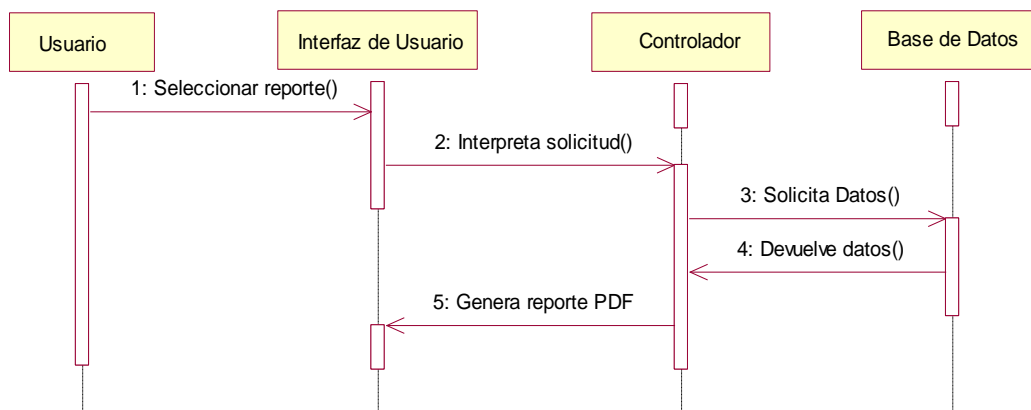


B. Diagrama de Secuencia

El diagrama de secuencia Reportes refleja la planeación de los procesos necesarios para llevar a cabo la generación de reportes de inventarios y de gestión. (Ver figura 3.34).

Figura 3. 28 Diagrama de Secuencia Reportes

Fuente: Elaboración Propia



C. Sprint Backlog

En base a la los requisitos R12 y R13 presentados en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación (Ver Tabla 3.28).

Tabla 3. 30 Sprint Backlog Reportes

Fuente: Elaboración Propia

SPRINT-BACKLOG7-REPORTES				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
	Generar reportes PDF de la lista requerida USUARIOS	Muy Alta	Desarrollo	Terminado
Reportes	Generar reportes PDF de la lista requerida pacientes	Muy Alta	Desarrollo	Terminado
	Generar reportes PDF de la lista requerida SERVICIOS	Media	Desarrollo	Terminado

- **Desarrollo:**

El Sprint Backlog presentado en la Tabla 3.38 permite generar lista de usuarios, pacientes y servicios que describen puntualmente los subprocesos: Generar lista de usuarios (Ver tabla 3.39) y Generar lista de pacientes (Ver tabla 3.40) y Generar lista de servicios (Ver tabla 3.41).

Tabla 3. 31 Lista de Usuarios

Fuente: Elaboración Propia

Nombre de historia de Usuario: Generar lista de usuarios	
Usuario: Usuario Administrador	Iteración Asignada: 8
Prioridad: Media	Riesgo en desarrollo: Baja
Descripción: Generar reportes pdf del estado actual de usuarios en el sistema.	
Observaciones: Ninguna.	

Tabla 3. 32 Lista de Pacientes

Fuente: Elaboración Propia

Nombre de historia de Usuario: Generar lista de pacientes

Usuario: Usuario Administrador	Iteración Asignada : 8
Prioridad: Media	Riesgo en desarrollo: Baja

Descripción: Generar reportes pdf del estado actual de pacientes en el sistema.

Observaciones: Ninguna.

Tabla 3. 33 Lista de Servicios

Fuente: Elaboración Propia

Nombre de historia de Usuario: Generar lista de servicios

Usuario: Usuario Administrador	Iteración Asignada: 8
Prioridad: Media	Riesgo en desarrollo: Baja

Descripción: Generar reportes pdf del estado actual de servicios en el sistema.

Observaciones: Ninguna.

- **Revisión:**

El desarrollo del Sprint en base a las listas de usuarios, pacientes y servicios presentadas en las tablas 3.39, 3.40, 3.41 generan interfaces gráficas o pantallas de administración de los reportes.

Mismas que son presentadas a los usuarios para su revisión (Ver Figura 3.35, 3.36 y 3.37).

Figura 3. 29 Reportes de Usuarios

Fuente: Elaboración Propia

GESTION CLINICA HDM									
Opciones	Apellidos	Nombre	Fecha de Nacimiento	Sexo	Estado Civil	Documento	Número	Cargo	Especialidad
	Gonzales Paz	Cesar	2017-03-08	M	S	DNI	11111111	Gerente General	Medicinal General
A A	A	A	2020-06-01	M	S	CEDULA	12345	Enfermera	Ginecologia

Figura 3. 30 Reportes de Pacientes

Fuente: Elaboración Propia

GESTION CLINICA HDM									
Opciones	Apellido Paterno	Apellido Materno	Nombre	Fecha de Nacimiento	Sexo	Estado Civil	Numero de Documento	Direccion	Teléfono
	A	A	A	2020-06-01	M	S	12345	Avenida Kollasuyo	12121212
	Gonzales	Paz	Cesar	2017-03-08	M	S	11111111	Ceja	12345678
	Miranda	Perez	Marcos	2020-06-12	M	S	8313131	Avenida Kollasuyo	234567
	pet	pet	pet	2015-03-05	M	C	23232323	Avenida Kollasuyo	2121212

Figura 3. 31 Reporte de Servicios

Fuente: Elaboración Propia

GESTION CLINICA HDM			
Opciones	Nombre	Costo	Estado
	Ginecología	60.00	Activado

3.3.8. Sprint 8: Gestión de Usuarios

- **Planeación:**

En la Tabla 3.41 se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 3. 34 Planificación del Sprint 8

Fuente: Elaboración Propia

REQUISITO	TAREAS	Inicio	Fin	Duración
		04/06/2020	12/06/2020	7 días
Gestión de Usuarios	Diseño de Diagramas UML	Desde	Hasta	Estado
		04/06/2020	06/06/2020	Completado
		07/06/2020	10/06/2020	Completado
		10/06/2020	12/06/2020	Completado

A. Diagrama de Casos de Uso

El diagrama de casos de uso Gestión de usuarios abarca los procesos que realiza el usuario administrador para administrar los usuarios del Hospital (Ver figura 3.38).

Figura 3. 32 Diagrama UC Gestión de Usuarios

Fuente: Elaboración Propia

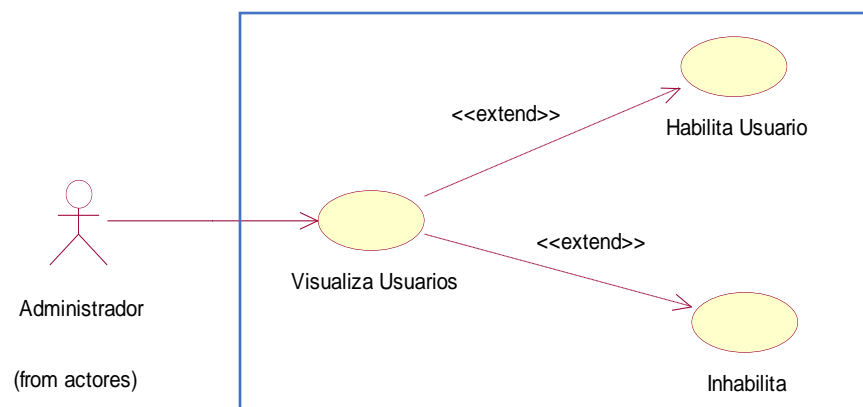
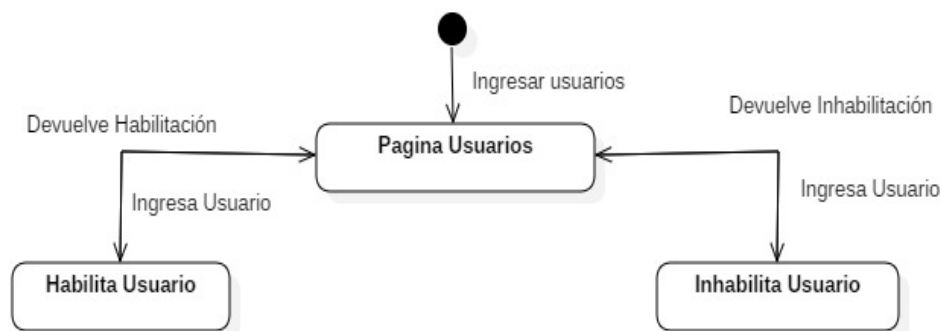


Diagrama Navegacional Sprint 8

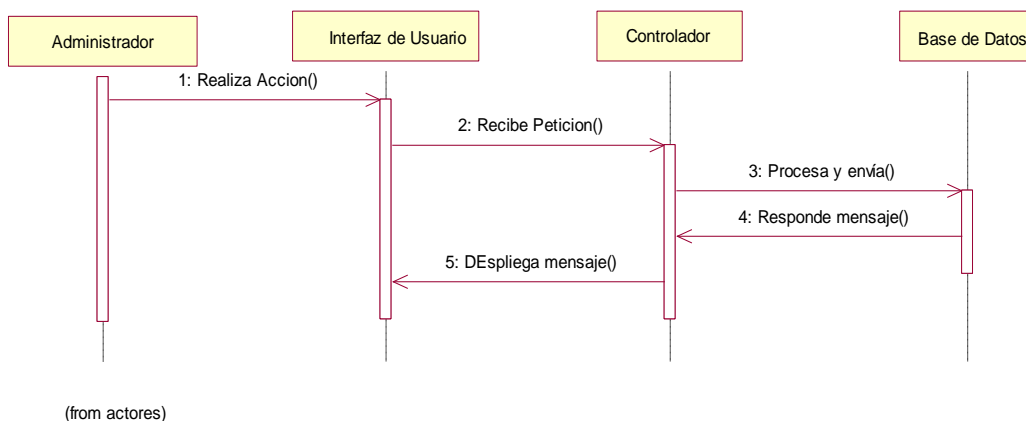


B. Diagrama de secuencia

El diagrama de secuencia Gestión de usuarios refleja los procesos por los cuales atraviesa la información para llevar a cabo la gestión de usuarios (Ver figura 3.39).

Figura 3. 33 Diagrama de secuencia Gestión de Usuarios

Fuente: Elaboración Propia



C. Sprint Backlog

En base al requisito R14 presentado en la Tabla 3.2 se elabora el Sprint Backlog detallando los procesos necesarios para su culminación (Ver Tabla 3.42).

Tabla 3. 35 Sprint Backlog Usuarios

Fuente: Elaboración Propia

SPRINT-BACKLOG8-GESTION DE USUARIOS				
PROCESO	SUBPROCESO	PRIORIDAD	TIPO	ESTADO
Gestión de Usuarios	Visualizar usuarios activos e inactivos	Muy Alta	Desarrollo	Terminado
	Habilitar o deshabilitar Usuarios	Muy Alta	Desarrollo	Terminado

- **Desarrollo:**

El Sprint Backlog presentado en la tabla 3.42 permite generar historias de usuarios que describen puntualmente los subprocesos: Visualizar información de usuarios (Ver tabla 3.43) y Habilitar o deshabilitar usuarios (Ver tabla 3.44).

Tabla 3. 36 Visualizar información de Usuarios

Fuente: Elaboración Propia

Nombre de historia de Usuario: Visualizar información de usuarios	
Usuario: Usuario Administrador	Iteración Asignada: 8
Prioridad: Alta	Riesgo en desarrollo: Media
Descripción: Listar todos los usuarios activos e inactivos	
Observaciones: Ninguna.	

Tabla 3. 37 Habilitar o deshabilitar Usuarios

Fuente: Elaboración Propia

Nombre de historia de Usuario: Habilitar o deshabilitar usuarios

Usuario: Usuario Administrador	Iteración Asignada: 8
Prioridad: Alta	Riesgo en desarrollo: Media
Descripción: Deshabilitar un usuario o volver a habilitar uno deshabilitado	
Observaciones: Ninguna.	





- **Revisión**

El desarrollo del Sprint en base a las historias de usuario presentadas en las tablas 3.43 y 3.44 genera una interfaz gráfica o pantalla de administración para la gestión de usuarios. Misma que es presentada a los usuarios para su revisión (Ver Figura 3.40)

Figura 3. 34 Interfaz Gráfica Sprint 8

Fuente: Elaboración Propia

The screenshot shows a user management interface. At the top left, there is a 'Usuario' label and a green 'Agregar' button. Below this, there are buttons for 'Copy', 'Excel', 'CSV', and 'PDF', and a search box labeled 'Buscar:'. The main part of the interface is a table with the following columns: Opciones, Apellidos, Nombre, Fecha de Nacimiento, Sexo, Estado Civil, Documento, Número, Cargo, Especialidad, Login, and Estado. Two rows of user data are visible, each with edit and delete icons in the 'Opciones' column and an 'Activado' status in the 'Estado' column.

Opciones	Apellidos	Nombre	Fecha de Nacimiento	Sexo	Estado Civil	Documento	Número	Cargo	Especialidad	Login	Estado
 	Gonzales Paz	Cesar	2017-03-08	M	S	DNI	11111111	Gerente General	Medicinal General	admin	Activado
 	AA	A	2020-06-01	M	S	CEDULA	12345	Enfermera	Ginecologia	A	Activado

3.4. POST-GAME

3.4.1. Pruebas

Se estableció la utilización de las pruebas de aceptación para cada módulo (Sprint) del sistema. Las pruebas de aceptación consisten en verificar la eficacia del sistema frente a un requerimiento del usuario. Esta verificación se realiza de acuerdo a ciertos criterios que el usuario establece inicialmente y el sistema debe mostrar la información esperada de acuerdo a dichos criterios. Este tipo de pruebas es importante debido a la necesidad de verificar que se está realizando los procesos en forma correcta. En la tabla 3.49, se muestran las pruebas de registro de pacientes, en la tabla 3.50 las pruebas de visualización de datos estadísticos y en la tabla 3.51 las pruebas de generación de reportes.

Tabla 3. 38 Prueba de Registro de Pacientes

Fuente: Elaboración Propia

Identificador	Numero de prueba: 1
Descripción:	Registro de Pacientes
Componente a validar:	Verificación del registro de un paciente
Parámetros iniciales:	Se establece el usuario y la contraseña del médico o asistente
Instrucciones:	* Se ingresa datos personales del paciente. * Se toma nueva orden de atención. * Se verifica el orden.
Resultados esperados:	Obtener mensaje de satisfacción al realizar las operaciones.

Tabla 3. 39 Prueba de aceptación de Información

Fuente: Elaboración Propia

Identificador	Numero de prueba: 2
Descripción:	Datos clínicos
Componente a validar:	Verificar la correcta inserción y devolución de datos clínicos
Parámetros iniciales:	Se establece la información obtenida del paciente por parte del médico, enfermera o asistente
Instrucciones:	* Se toma nueva orden de atención. * Se registran la información del paciente. * Se verifica la información clínica guardada
Resultados esperados:	Obtener mensaje de satisfacción al realizar las operaciones.

Tabla 3. 40 Prueba de aceptación de Reportes

Fuente: Elaboración Propia

Identificador	Numero de prueba: 3
Descripción:	Generación de reportes
Componente a validar:	Verificación de la correcta generación de reportes
Parámetros iniciales:	Se dispone de la información clínica guardada
Instrucciones:	* Se busca el paciente, usuario o información que se requiere. * Ingresar a la opción de generar reportes * Se verifica la información clínica guardada en un reporte pdf.
Resultados esperados:	Obtener reporte en una ventana nueva al realizar las peticiones.

METRICAS DE CALIDAD

4. METRICAS DE CALIDAD

4.1. INTRODUCCION

Los puntos determinados para determinar la calidad de producto de software se encuentran enmarcada en el ISO/IEC 9126. Estas características están determinadas en función de entrevistas a los usuarios, pruebas aplicadas y uso del producto de software.

4.2. CARACTERISTICAS PROPUESTAS POR LA ISO 9126

4.2.1. Funcionalidad

El punto función es una métrica orientada a la función del software y el proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa, los puntos función se calculan realizando una serie de actividades comenzando por determinar los siguientes puntos:

- Numero de entrada de usuarios, se cuenta cada entrada de usuario que proporciona el software diferentes datos orientados a la aplicación.
- Número de salidas de usuarios, estas se refieren a informes, mensajes de error y toda forma de interacción con el usuario.
- Número de peticiones de usuarios, una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva.
- Numero de archivos, se cuenta cada archivo maestro lógico.
- Numero de interfaces externas, se cuenta todas las interfaces legibles por el ordenador que son solicitados para transmitir información a otro sistema.

De acuerdo a lo mencionado se tiene la siguiente tabla.

Tabla 4. 1 Síntesis para Hallar el punto de función

Fuente: Elaboración Propia

Número	Cantidad
Entradas de usuario	9
Salidas de usuario	9
Consultas de usuario	9
Número de archivos	10
Interfaces externos	0

Los puntos función se calculan llenando la siguiente tabla:

Tabla 4. 2 Datos - Ponderación Media

Fuente: Elaboración Propia

Número	Cantidad	Simple	Medio	Complejo	Valor Obtenido
Número Entradas de usuario	9		* 3		27
Número Salidas de usuario	9		* 3		27
Número Consultas de usuario	9		* 3		27
Número de archivos	10		* 4		40
Número Interfaces externos	0	* 1			0
Total					121

La relación que permite calcular los puntos función es la siguiente:

$$Pf = cuenta_{total} * (grado_de_confiabilidad + tasa_de_error * \sum F_i)$$

Donde:

Pf = Medida de funcionalidad

Cuenta total = Suma del valor de las entradas, salidas, peticiones, interfaces externas y archivos.

Grado de confiabilidad = Confiabilidad estimada del sistema

Tasa de Error = Probabilidad subjetiva estimada del dominio de la información (1%).

Fi = Son los valores de ajustes de complejidad que toman los valores de la tabla y que dan respuesta a los siguientes datos de la tabla:

Tabla 4. 3 Niveles de Ponderación

Fuente: Elaboración Propia

Datos de Ajuste	Valor
Sin Importancia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Las valoraciones, el cual se basa en la relación empírica basada en medidas cuantitativas del dominio de información del software y valoraciones subjetivas acerca de la complejidad del software.

Tabla 4. 4 Factores Escala y Valores

Fuente: Elaboración Propia

	Factor Escala	Valor
1.	¿Requiere el sistema de copias de seguridad y de recuperación fiables?	5
2.	¿Se requiere de comunicación de datos?	5
3.	¿Existen funciones de procesos distribuidos?	3
4.	¿Es crítico el rendimiento?	4
5.	¿Será ejecutado en el SO existente?	4
6.	¿Requiere el sistema de entrada interactiva?	5
7.	¿Requiere el sistema entrada de datos interactiva sobre múltiples entradas?	3
8.	¿Se actualizan los archivos maestros de manera interactiva?	4
9.	¿Son complejas las entradas, salidas, los archivos y peticiones?	3
10.	¿Es complejo el procesamiento interno?	4
11.	¿Se ha diseñado el código para ser reutilizable?	4
12.	¿Están incluidas en el diseño la conversión y la instalación?	4
13.	¿Se ha diseñado el sistema para soportar múltiples instalaciones?	4
14.	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
Total		57

Con la obtención de los anteriores datos y considerando un grado de confiabilidad mínimo del es que a continuación calculamos el valor de Pf:

$$Pf = cuenta_{total} * (grado_{confiabilidad} + tasa_{error} * \sum F_i)$$

$$Pf = 121 * (0.81 + 0.01 * 57)$$

$$Pf = 166.98$$

Si consideramos que el máximo valor de ajuste de complejidad a $\sum F_i = 70$

$$Pf \text{ ideal} = 121 * (0.81 + 0.01 * 70)$$

$$Pf \text{ ideal} = 182.71$$

Entonces si $\sum F_i$ es considerada como el 100% la relación obtenida entre los puntos será:

$$x = \frac{(100\%)*Pf}{Pf \text{ ideal}} = \frac{(100\%)*166.98}{182.71}$$

$$x = 91.39$$

Por lo tanto, la funcionalidad del sistema es de 91.39 % en relación como un producto de software se desenvuelve satisfactoriamente.

4.2.2. Confiabilidad

En términos estadísticos la confiabilidad se define como la probabilidad de operación libre de fallos del sistema en un entorno determinado y durante un tiempo específico.

La confiabilidad del sistema se mide mediante la ley de distribución exponencial, cual se define como la confiabilidad $R(t)$ de un componente durante un periodo $[0,1]$ o que falle en un tiempo mayor que t , es decir se tiene la ecuación.

$$R(t) = P[T > t] = 1 - F(t)$$

Se toma en cuenta que el tiempo T para fallas es una variable aleatoria, exponencial, se tiene:

$$R(t) = 1 - [1 - e^{-\lambda t}]$$

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

λ = Tasa constante de fallo (λ = número de fallas/número de accesos al sistema).

Después de realizar un cálculo de prueba en cada uno de los subsistemas se obtienen los datos que se muestra en:

Tabla 4. 5 Componentes del Sistema

Fuente: Elaboración Propia

Ri(t)	Descripción	Λ	T	$e^{(\lambda t)}$
R1(t)	Registro de historias clínicas	1 / 100	5	0.95
R2(t)	Registro de asignación consultas médicas	2 / 100	5	0.90
R3(t)	Registro de asignación de citas médicas	2 / 100	5	0.90
R4(t)	Registro medico	2 / 100	5	0.90
R5(t)	Registro de hoja (enfermería/asistente)	1 / 100	5	0.95

La confiabilidad total del sistema está relacionada a dos casos detallados a continuación:

Caso 1: Sin componentes independientes, conectados en serie y el i-esimo componente tiene la confiabilidad R(t), del sistema completo es:

$$R(t) = R1(t) * R2(t) * \dots * Rn-1(t) * Rn(t)$$

$$R(t) = R1(t) * R4(t) * R5(t) = 0.95 * 0.90 * 0.95 = 0.81$$

Caso 2: Sin componentes independientes, conectados en serie y el i-esimo componente tiene la confiabilidad R(t), del sistema completo es:

$$R(t) = 1 - [1 - R1(t) * 1 - R2(t) * \dots * (1 - Rn-1(t) * (1 - Rn(t))]$$

$$R(t) = 1 - (1 - R3(t) * (1 - R2(t)) * (1 - R5(t))) = 1 - 0.1 * 0.1 * 0.05 = 0.999$$

La confiabilidad total del sistema R(t), considerando los datos de la tabla, está dado por la siguiente relación.

$$R(t) = 0.81 * 0.999 = 0.81$$

El resultado presenta una confiabilidad de 81%, esto indica que el sistema tiene una probabilidad de fallo de 19% en un tiempo de 15 días lo que significa que el sistema seguirá funcionando de manera correcta.

4.2.3. Usabilidad

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema, realizando las preguntas necesarias e interacción con el sistema se pudo obtener los siguientes resultados.

Tabla 4. 6 Resultados de medir la Usabilidad

Fuente: Elaboración Propia

Usuarios	Facilidad de Comprensión	Facilidad de Aprendizaje	Facilidad de Operación
Usuario 1	90%	95%	95%
Usuario 2	90%	95%	95%
Usuario 3	85%	80%	85%
Usuario 4	80%	80%	80%
Promedio	86%	88%	89%

Se puede concluir de acuerdo a los resultados obtenidos que la comprensión de la usabilidad del usuario en promedio de todos los usuarios es 88%.

4.2.4. Mantenibilidad

Para la mantenibilidad se utilizará la métrica de índice de madurez de software (IMS) que nos dice que cuanto más se acerque a 1 más mantenible es el sistema.

Fórmula

$$IMS = \frac{[Mt - (Fa + Fm + Fe)]}{Mt}$$

Donde:

M t= número de historias de usuario en la versión actual

Fm= número de historias de usuario en la versión actual que ha sido modificada

Fe = número de historias de usuario en la versión actual que han sido eliminados
en la versión actual

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

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

$$IMS = 0.9$$

En conclusión, el producto es estable y requiere de mantenimiento nulo.

4.2.5. Portabilidad

Dado que la plataforma fue desarrollada con herramientas multiplataforma, se garantiza el funcionamiento en servidores Linux y Windows. Ya que el lenguaje PHP se ejecuta en todos los lenguajes web y el gestor de base datos María DB existe todo tipo de servidores y sistemas operativos. También se realizaron pruebas, para ver su funcionamiento en todos los navegadores de mayor difusión en Bolivia, de donde se seleccionaron y ordenaron según el porcentaje de uso que se observa en la siguiente tabla.

Tabla 4. 7 Funcionamiento en Navegadores

Fuente: Elaboración Propia

Navegador	Porcentaje de Funcionamiento
Internet Explorer 8	90%
Mozilla Firefox 3.0 o 6	95%
Internet Explorer 9	90%
Opera	90%
Google Chrome	95%
Safari	90%

En conclusión, su portabilidad en navegadores es 92% lo que se desenvuelve correctamente en los navegadores.

4.3. METODOLOGIA WEB SITE QEM

En esta fase se definen un conjunto de propiedades (atributos y característica), de requerimientos de calidad los cuales deben responder a necesidades de un perfil usuario, viendo como base el árbol propuesto por la metodología Web Site Qem.

El árbol de requerimientos de calidad para el Sistema Web de Administración de Historias Clínicas para los consultorios del Hospital de la Mujer, se muestra en la tabla.

Tabla 4. 8 Árbol de Requerimientos para Calidad

Fuente: Elaboración Propia

1. Usabilidad	
1.1	Comprensibilidad global del sitio
1.1.1	Esquema de organización global
1.1.1.1	Mapa de sitio
1.1.1.2	Menú de contenido

- 1.2 Mecanismo de ayuda y retroalimentación en línea
 - 1.2.1 Calidad de ayuda
 - 1.2.1.1 Ayuda explicitaría orientado al usuario
 - 1.2.1.2 Ayuda de la Búsqueda
 - 1.2.2 Indicador de última actualización
 - 1.2.2.1 Global todo el sitio web
 - 1.2.2.2 Restringido por subsidio o página
 - 1.2.3 Retroalimentación
 - 1.2.3.1 Formulario de Entrada
 - 1.2.3.2 Reportes
 - 1.3 Permanencia y estabilidad en la presentación de los controles principales
 - 1.3.1 Permanencia de controles directos
 - 1.3.2 Permanencia de controles indirectos
 - 1.3.2.1 Estabilidad
 - 1.3.2.2 Aspecto de estilo
 - 1.3.2.3 Uniformidad en el color de enlaces
 - 1.3.3 Uniformidad en el estilo global
 - 1.3.3.1 Guía de estilo global
 - 1.3.4 Preferencia estética
 - 1.4 Misceláneos
-

2. Funcionalidad

- 2.1 Aspectos de búsqueda y recuperación
 - 2.1.1 Mecanismos de búsqueda en el sitio web
 - 2.1.1.1 Búsqueda restringida
 - 2.1.1.1.1 De pacientes y personal médico
 - 2.1.1.1.2 De Historiales clínicos
 - 2.1.1.1.3 De consultas y citas médicas
 - 2.1.1.2 Búsqueda global
 - 2.1.2 Mecanismos de recuperación
 - 2.1.2.1 Nivel de personalización
 - 2.1.2.2 Nivel de retroalimentación en la recuperación
- 2.2 Aspectos de navegación y exploración
 - 2.2.1 Navegabilidad
 - 2.2.1.1 Orientación
 - 2.2.1.1.1 Indicador de camino
 - 2.2.1.1.2 Etiqueta de posición actual

- 2.2.1.2 Promedio de enlaces por página
 - 2.2.2.1.1 Objetos de control de navegación
 - 2.2.2 Nivel de desplazamiento
 - 2.2.2.1 desplazamiento vertical
 - 2.2.2.2 desplazamiento horizontal
 - 2.2.3 Predicción navegacional
 - 2.2.3.1 Enlace del título
 - 2.2.3.2 Calidad de la frase de enlace
 - 2.3 Aspectos del dominio orientado al usuario
 - 2.3.1 Relevancia del contenido
 - 2.3.1.1 Información del historial clínico
 - 2.3.1.1.1 Información de la consulta médica
 - 2.3.1.1.2 Información de la cita médica
 - 2.3.1.2 Información del personal médico
 - 2.3.1.2.1 Información de la hoja de enfermería
 - 2.3.1.2.2 Información del reporte
-

3. Confiabilidad

- 3.1 No deficiencia
 - 3.1.1 Errores de enlaces
 - 3.1.1.1 Enlaces Rotos
 - 3.1.1.2 Enlaces inválidos
 - 3.1.1.3 Enlaces no implementados
 - 3.1.2 Errores o deficiencias varias
 - Deficiencias o cualidades ausentes debido a diferentes navegadores
 - 3.1.2.1 navegadores
 - 3.1.2.2 Deficiencias o resultados inesperados
 - 3.1.2.3 Nodos destino inesperadamente en construcción
 - 3.1.2.4 Nodos muertos sin en laces de retorno
-

4. Eficiencia

- 4.1 Performancia
 - 4.1.1 Páginas de acceso rápido
- 4.2 Accesibilidad
 - 4.2.1 Accesibilidad de la información
 - 4.2.1.1 Soporte versión sólo texto
 - 4.2.1.2 Legibilidad al desactivarse propiedades
 - 4.2.1.2.1 Descripción título

- 4.2.1.2.2 Legibilidad Global
 - 4.2.2 Accesibilidad de entorno
-

5. Portabilidad

- 5.1 Independencia de hardware
 - 5.2 Independencia de software
 - 5.3 Inestabilidad
 - 5.4 Conformidad de la transportabilidad
-

6. Mantenibilidad

- 6.1 Analizabilidad
 - 6.2 Expansibilidad
 - 6.3 Estabilidad
 - 6.4 Cambiabilidad
-

4.3.1. Criterio de preferencia de calidad elemental

Para esta fase se determinan los criterios de evaluación cuantificable, que se aplicara en los criterios de preferencia elemental del modelo Web Site Qem.

El tipo de criterio elemental que se utilizará es el criterio de preferencia elemental absoluta de variable discreta ya que se emplea para determinar la preferencia absoluta discreta. Se debe determinar los valores de las variables de preferencias de calidad elemental (IE_j) para cada atributo A_j (Hojas del árbol de requerimientos, es importante mencionar que cada atributo A_j tendrá asociada una variable X_i e R que tomara un valor real a partir de un proceso de medición el cual producirá un valor de IE_j que interpreta el porcentaje del requerimiento satisfecho.

4.3.2. Especificación de atributos

Debemos especificar los atributos del árbol existentes para hallar los índices de calidad elemental (IE_j); estos índices se utilizaron en los diferentes criterios de evaluación descritos a continuación.

CVN (Criterio de Variable Normalizada):

$$\text{Indicador Elemental (IE)} = (Xt/Y)*100\%$$

Donde: **X** = \sum Puntaje máximo

Y = \sum Puntaje obtenido

CN (Criterio Normalizado):

$$\text{Indicador Elemental (IE)} = (Xt/Y)*100\%$$

Donde: **X** = Cantidad total de datos para la variable

Y = Cantidad total de datos

CB (criterio binario)

Indicador elemental (IE) = 0 si no existe indicador elemental
(IE) = 1 si existe

CMN (criterio multinivel)

IE= 0 ausente

IE= 2 presente parcial

IE= 3 presente

CPD (sujeto a la objetividad del observador)

4.3.3. Definición e implementación de la evaluación elemental

Partiendo del árbol de requerimientos para cada uno de los atributos A_j , determinar la variable X_j , que tomara un valor real a partir del proceso de medición.

Tabla 4. 9 Resultados de Usabilidad

Fuente: Elaboración Propia

Código	Atributo	Criterio Elemental	IE%
1.1.1	Esquema de organización global	CVN	65
1.1.1.1	Mapa de sitio	CB	60
1.1.1.2	Menú de contenido	CB	0
1.2	Mecanismo de ayuda y retroalimentación en línea	CVN	80
1.2.1	Calidad de ayuda	CVN	70
1.2.1.1	Ayuda explicitaria orientado al usuario	CPD	70

1.2.1.2	Ayuda de la Búsqueda	CMN	70
1.2.1.3	Aspectos de interfaz y estéticos	CVN	90
1.2.1.4	Cohesividad al agrupar los objetos de control principal	CVN	65
1.3	Permanencia y estabilidad en la presentación de los controles principales	CPD	80
1.3.1	Permanencia de controles directos	CVN	80
1.3.2	Permanencia de controles indirectos	CPD	80
1.3.2.1	Estabilidad	CPD	80
1.3.2.2	Aspecto de estilo	CVN	80
1.3.2.3	Uniformidad en el color de enlaces	CMN	80
1.3.3	Uniformidad en el estilo global	CMN	80
1.3.4	Preferencia estética	CPD	90

Tabla 4. 10 Resultados de Funcionalidad

Fuente: Elaboración Propia

Código	Atributo	Criterio Elemental	IE%
2.2.2.1	Desplazamiento vertical	CB	100
2.2.2.2	Desplazamiento horizontal	CB	100
2.2.3.1	Enlace del título	CMN	100
2.2.3.2	Calidad de la frase de enlace	CMN	80
2.3.1	Relevancia del contenido	CB	100
2.3.1.1	Información del historial clínico	CB	100
2.3.1.1.1	Información de la consulta médica	CB	100
2.3.1.1.2	Información de la cita médica	CB	100
2.3.1.2	Información del personal médico	CB	100
2.3.1.2.1	Información de la hoja de enfermería	CB	100
2.3.1.2.2	Información del reporte	CB	100

Tabla 4. 11 Resultados Confiabilidad

Fuente: Elaboración Propia

Código	Atributo	Criterio Elemental	IE%
3.1	No deficiencia	CVN	96
3.1.1	Errores de enlaces	CVN	100
3.1.1.1	Enlaces Rotos	CMN	70

3.1.1.2	Enlaces inválidos	CMN	60
3.1.1.3	Enlaces no implementados	CMN	60
3.1.2.1	Deficiencias o cualidades ausentes debido a diferentes navegadores	CMN	95
3.1.2.2	Deficiencias o resultados inesperados	CMN	86
3.1.2.3	Nodos destino inesperadamente en construcción	CMN	75
3.1.2.4	Nodos muertos sin en laces de retorno	CMN	75

Tabla 4. 12 Resultados de Eficiencia

Fuente: Elaboración Propia

Código	Atributo	Criterio Elemental	IE%
4.1.1	Páginas de acceso rápido	CPD	70
4.2.1	Accesibilidad de la información	CB	60
4.2.1.2	Legibilidad al desactivarse propiedades	CB	100
4.2.1.2.2	Legibilidad Global	CB	70
4.2.2	Accesibilidad de entorno	CMN	90

Tabla 4. 13 Resultados de Portabilidad

Fuente: Elaboración Propia

Código	Atributo	Criterio Elemental	IE%
5.1	Independencia de hardware	CN	95
5.2	Independencia de software	CN	50
5.3	Instabilidad	CN	90
5.4	Conformidad de la transportabilidad	CN	98

Tabla 4. 14 Resultados de Mantenibilidad

Fuente: Elaboración Propia

Código	Atributo	Criterio Elemental	IE%
6.1	Analizabilidad	CN	80
6.2	Expansibilidad	CN	70
6.3	Estabilidad	CN	62
6.4	Cambiabilidad	CVN	70.5

En la tabla se muestra los resultados obtenidos a partir de la evaluación elemental, donde la calidad global es de 75%.

Tabla 4. 15 Resultados Globales

Fuente: Elaboración Propia

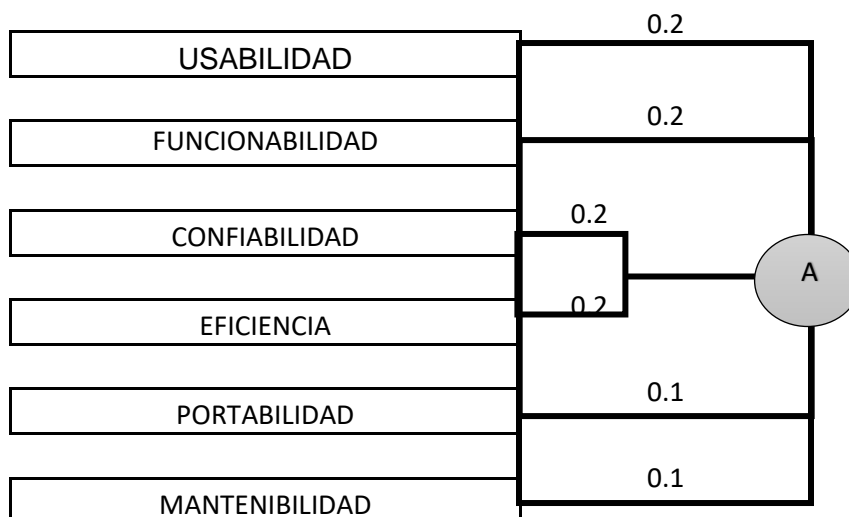
Nombre	Preferencia Elemental
Usabilidad	71.76
Funcionabilidad	98.18
Confiabilidad	80
Eficiencia	78
Portabilidad	83.25
Mantenibilidad	70.5
Calidad Global	75

4.3.4. Operadores de LSP para el modelado de relaciones lógicas

Muestra la estructura de agregación de preferencias parciales para las características de más alto nivel conforme a un visitante generador.

Figura 4. 1 Características de Alto Nivel

Fuente Elaboración propia



De acuerdo a la valoración de la calidad global del sitio web, aplicando la metodología Web Site Qem, está definido dentro de los márgenes de satisfacción (60% a 100%).

Observando los resultados obtenidos el nivel de disponibilidad de atributos y subcaracterísticas de calidad en general, como factores esenciales de los sitios web. Esquema de organización Global del sitio Mecanismos de Búsqueda, Mecanismos de orientación y Navegación, entre otros, superan en conjunto una disponibilidad, lo que confirmaría que el diseño en la web es suficiente, en consideración de un perfil de usuario y para el dominio específico.

EVALUACION DE COSTOS Y BENEFICIOS

5. EVALUACION DE COSTOS Y BENEFICIOS

5.1. INTRODUCCION

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

5.2. ANALISIS DE COSTOS

5.2.1. Estimación con el método COCOMO

Como mencionamos en el capítulo anterior el modelo constructivo de costos o COCOMO, es un es un modelo de estimación de costos de software, orientado a la magnitud del producto final, midiendo en tamaño del proyecto en líneas de código principalmente. El modelo provee tres niveles de aplicación: Básico, intermedio y avanzado, basándose en los factores considerados por el modelo.

Orgánico: proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.

Semiacoplados: proyectos intermedios en complejidad y tamaño (menores de 300 KDLC) donde la experiencia en este tipo de proyectos es variable y las restricciones intermedias.

Empotrado: proyectos bastantes complejos, en los que apenas se tienen experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Para mejorar esta estimación aplicamos el Modelo intermedio post-Arquitectura este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación seleccionamos nuestros calificadores para cada atributo. La ecuación del modelo básico es:

$$E = a * KLDC^b (\text{persona } x \text{ mes})$$

$$D = c * E^d$$

$$P = \frac{E}{D}$$

T= tiempo de duración del desarrollo = $c \text{ esfuerzo}^d$ (meses)

Donde E es el esfuerzo aplicado en hombre mes, D es el tiempo de desarrollo en meses y KLDC es el número de miles de líneas de código estimado para el proyecto. Los coeficientes a y c y los exponentes b y d se obtienen de la siguiente tabla:

Tabla 5. 1 Coeficiente COCOMO

Fuente: Elaboración Propia

TIPO DE PROYECTO	A	B	c	d
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Para el proyecto se considera el modo orgánico y se realizaron los siguientes cálculos:

$$a = 2.4 \quad b = 1.05 \quad c = 2.5 \quad d = 0.38 \quad KLDC = 16$$

$$E = a * KLDC^b$$

$$E = 2.4 * 16^{1.05} = 44.11 \text{ personas mes}$$

$$D = c * E^d$$

$$D = 2.5 * 44.11^{0.38} = 10.11 \text{ mes}$$

$$P = \frac{E}{D}$$

$$P = \frac{44.11}{10.11} = 4.36 \text{ personas}$$

Tabla 5. 2 Multiplicadores de Esfuerzo

Fuente: Elaboración Propia

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de Software						
Fiabilidad requerida del Software	0.75	0.88	1.00	1.15	1.40	-
Tamaño de Base de Datos	-	0.94	1.00	1.08	1.16	-
Complejidad del producto	0.70	0.85	1.00	1.15	1.30	1.65
Atributos de hardware						
Restricciones de tiempo de ejecución	-	-	1.00	1.11	1.30	1.66
Restricciones de memoria virtual	-	-	1.00	1.06	1.21	1.56
Volatilidad de la máquina virtual	-	0.87	1.00	1.15	1.30	-
Tiempo de respuesta del ordenador	-	0.87	1.00	1.07	1.15	-
Atributos de personal						
Capacidad del analista	1.46	1.19	1.00	0.86	1.15	-
Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82	-
Calidad de los programadores	1.42	1.17	1.00	0.86	0.70	-
Experiencia en la máquina virtual	1.21	1.10	1.00	0.90	-	-
Experiencia en S.O. utilizado	1.21	1.10	1.00	0.95	-	-
Experiencia en el lenguaje de programación	1.14	1.07	1.00	0.91	0.82	-
Atributos del proyecto						
Técnicas actualizadas de programación	1.24	1.10	1.00	0.91	0.82	-

Utilización de herramienta de software	1.24	1.10	1.00	0.91	0.83	-
Restricciones de tiempo de desarrollo	1.23	1.08	1.00	1.04	1.10	-

$$\mathbf{FAE} = 1.15 * 1.00 * 0.85 * 1.11 * 1.00 * 1.00 * 1.07 * 0.86 * 0.82 * 0.70 * 1.00 * 0.95 * 1.00 * 0.91$$

$$\mathbf{FAE} = 0.49$$

Y el cálculo de kilos líneas de código es:

$$\mathbf{KLDC} = (\text{Pf} * \text{líneas de código por cada Pf}) / 1000 = (121 * 16) / 1000 = 8.363$$

$$\mathbf{Pf} = \text{cuenta total} * (0.65 + 0.01) * \text{sum}(fi)$$

Cálculo del esfuerzo del desarrollo

Ahora aplicando estos resultados a la fórmula del esfuerzo tenemos:

$$\mathbf{E} = a \mathbf{KLDC} * \mathbf{FAE} = 3.2 * (16) ^{1.05 * 0}$$

Calculando tiempo de desarrollo

$$\mathbf{T} = c \text{Esfuerzo}^d = 2.5 * (15.91) ^{0.38} = 7.15 \text{ meses}$$

PRODUCTIVIDAD

$$\mathbf{PR} = \mathbf{LDC} / \mathbf{Esfuerzo} = 8363 / 15.91 = 525.64 \text{ LDC/ personas/mes}$$

Personal promedio

$$\mathbf{P} = \mathbf{E} / \mathbf{T} = 15.91 / 7.15 = 2.22 \text{ personas}$$

Costo

Suponiendo que el personal se les pague un promedio de 700 \$us/mes y se trabaje los 7.15 meses entonces el costo del proyecto será:

$$\mathbf{Costo} = 700 * 7.15 * 3 \text{ (salario * meses trabajo * \# de personas)}$$

$$\mathbf{Costo} = 15.015 \text{ $us} = 2.145 \text{ Bs.}$$

Como podemos ver el COCOMO nos da una aproximación del costo del sistema

Considerando que el sueldo de un desarrollador depende de la experiencia del mismo y es un valor muy subjetivo, se da un valor según la oferta de los programadores en el mercado de 1400 Bs. Realizando los costos necesarios para implantar el sistema se tiene:

Tabla 5. 3 Resumen Global

Fuente: Elaboración Propia

Descripción	Cantidad	Costo Mensual	Meses	total
Desarrolladores	2	1400	7	19600
Equipos	6	Existente		
Software		Gratuito		
Capacitación		1200	1	1200

El costo estimado es 20800 Bs.

5.2.2. Estimación método VAN Y TIR

Beneficios

El sistema proveerá acceso y transferencia de información en tiempo real, entre los diferentes usuarios, de los consultorios del Hospital de la Mujer, los beneficios tangibles que se pueden identificar son:

- Todos los datos estarán centralizados
- La información será accesible para todo el personal médico y pacientes
- Se reducirán los costos de operación y el tiempo de recuperación de datos
- Aplicando la tecnología se logrará una mejor atención a los pacientes.

Considerando que la institución requiere identificar los beneficios económicos que obtendrá de la implantación del sistema realizamos el siguiente análisis de costos. El estudio realizado anteriormente requiere de un monto de 20800 Bs.

Se estima que funcionamiento del sistema en su operación generara un ahorro de 1000 Bs. Mensual y de costos de operación de 700 Bs. El jefe médico percibe que el sistema informático le permitirá tomar decisiones oportunas, las que redundaran en un aumento en beneficios de 2800 Bs anuales. Dados estos datos, en consecuencia, los flujos de fondos para el sistema alcanzan a:

Tabla 5. 4 Inversión y Flujos Netos

Fuente: Elaboración Propia

Inversión 20800 Bs.		
Flujos netos de fondos		
Beneficios	Ahorro de costos	8700 Bs.
	Mejora de decisiones	2800 Bs.
Costos	costos de operación	5700 Bs.
Beneficios - costos		5800 Bs.

La evaluación del proyecto se llevará a cabo utilizando VAN y TIR y periodo de recupero, considerando periodos anuales y una vida útil del proyecto de cinco años, al cabo de los cuales no existe valor residual. El VAN (valor actual neto) está dado por la siguiente formula:

$$VAN = \sum Rt/(1+i)^t; t= 0..., n$$

$$VAN = Ro+(R1/(1+i)) +(R2/(1+i)^2) +.....+(Rn/(1+i)^n)$$

Donde Ro = es la inversión inicial Entonces Ro = 20800

Rt = flujo neto por periodo Rt = 5800

T = tiempo del periodo t = 5

I = interés residual i =3%

Tendremos:

$$VAN = 20800+(5800/ (1+0.03)) +(5800/ (1+0.03)^2) +(5800/ (1+0.03)^3) + (5800/ (1+0.03)^4) +(5800/ (1+0.03)^5)$$

VAN = 5762.30

Obtener para un proyecto un VAN positivo significa que es conveniente realizar el proyecto por lo cual el resultado da curso a la implementación del sistema. El TIR (tasa interna de retorno) llamada también tasa de actualización que hace el valor actual neto de un proyecto igual a cero, es decir, se trata de aquella tasa que iguala al valor actual de los egresos previstos con el valor actual de los ingresos previstos como tal como lo muestra la relación.

$$R_0 = \sum R_t / (1 + \text{tir})^t$$

Entonces $R_0 = 20800$; $R_t = 5800$; $t = 5$; $\text{TIR} = 4\%$

El TIR indica que el valor o margen de beneficio que se recibirá del proyecto es de 4%, considerando en este resultado que los beneficios no económicos son varios. El PPB (periodo de recupero) se define como el periodo de tiempo que requiere un proyecto para recuperar el monto de inversión inicial.

PPB = monto de la inversión inicial / flujo de caja anual

Donde

PPB = es el periodo en años

PPB = $20800/5800 = 3.58$

Lo que indica que en 5 años se recuperara el valor invertido en el proyecto.

**CONCLUSIONES
Y
RECOMENDACIONES**

6.1. CONCLUSIONES Y RECOMENDACIONES

Se ha logrado desarrollar e implementar un Sistema Web de Administración de Historias Clínicas de los pacientes, que permiten mejorar las tareas de admisión, consulta, búsqueda y elaboración de reportes o informes de manera más rápida y confiable, reduciendo de esta manera el tiempo en el registro de datos y manejo de la información de los Consultorios del Hospital de la Mujer.

Utilizando y apoyándose eficientemente en las herramientas y extensiones de modelado que nos brinda UML el Lenguaje de Modelado Unificado con el cual se especificó el análisis y diseño del sistema.

Con la solución automatizada, el sistema permite fortalecer el manejo adecuado y preciso de la base de datos.

Se ha cumplido con los objetivos planteados realizando los siguientes módulos: módulo de admisión del paciente, módulo de cita médica, módulo de consulta médica, módulo de historial clínico, módulo de enfermería, módulo de registro médico y módulo de reporte.

El sistema posee una interface gráfica amigable, que permite un manejo fácil e intuitivo para todo el personal médico de los consultorios.

6.2. RECOMENDACIONES

Es imprescindible señalar algunas recomendaciones que se cree que son de mucha importancia:

Se sugiere considerar, con respecto al software del sistema la implementación del módulo para la atención de reservas de fichas, que no se tomó en cuenta en este proyecto porque no abarcaba los puntos álgidos del proyecto.

El personal de la institución encargada de administrar el sistema debe incorporar normas y políticas de su uso.

BIBLIOGRAFIA

FUENTES DE INFORMACIÓN

BIBLIOGRAFIA

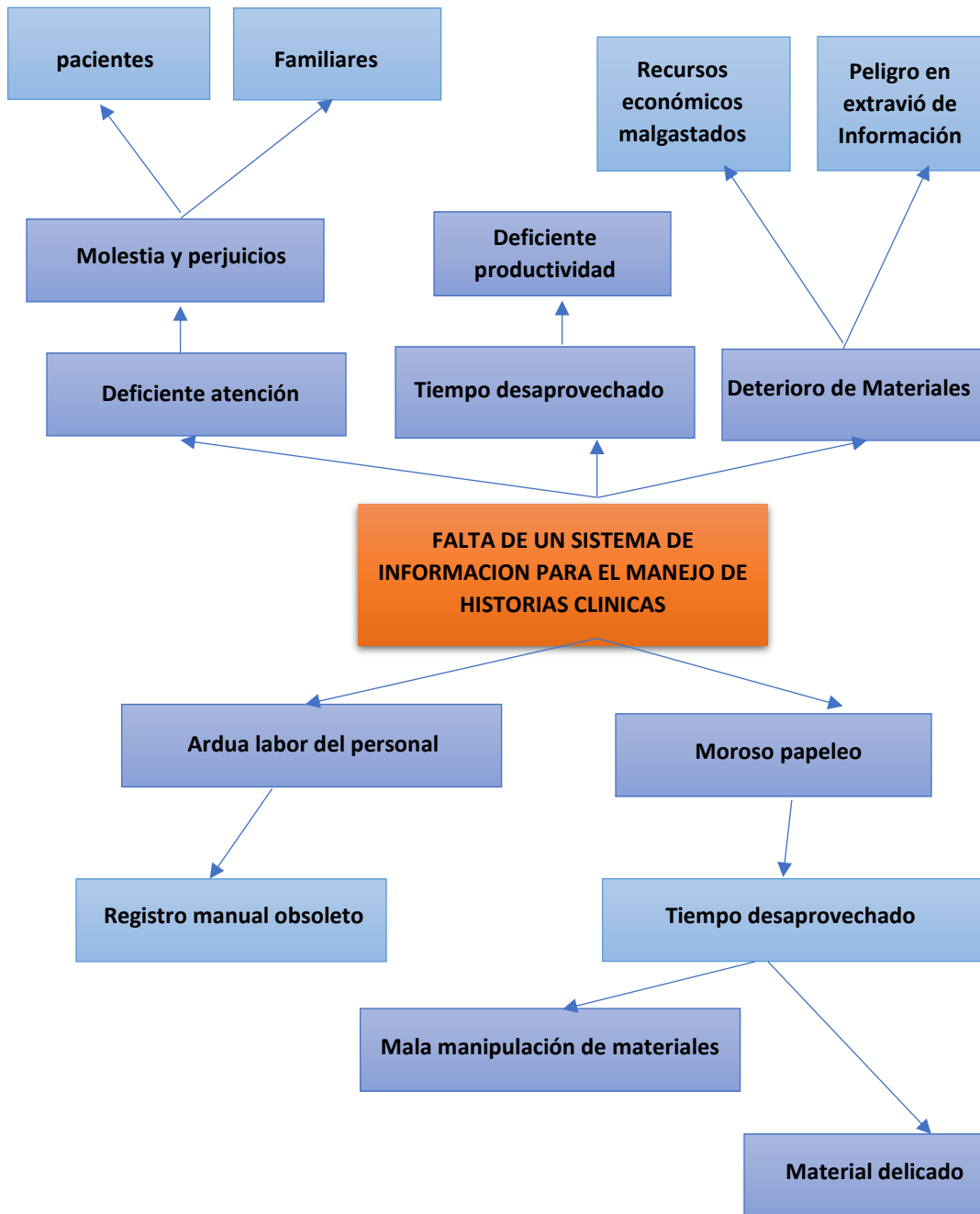
1. Schwabe Daniel & Rossi Gustavo, 2003; "The object oriented Hypermedia Design Model" (OOHDM)
2. Schwabe Daniel & Rossi Gustavo, 1995; "Building Hypermedia Applications as Navigational Views of Information Models"; Hawaii Vol. 38, #8, Pág. 45-46
3. Olsina Mag Luis Antonio, 1999; Ingeniería de Software en la Web; "Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web", Facultad de Ciencias Exactas, Universidad Nacional de La Plata; Argentina, noviembre
4. De la Vega Iván, 1998; "Módulo de Capacitación para la Recolección y el Análisis de Indicadores de Investigación y Desarrollo" (pag.1 La importancia de la información en la toma de decisiones), Banco Interamericano de Desarrollo Working Paper 6 Redes Bid
5. Pressman Roger S, 2002; Ingeniería de Software un Enfoque Práctico 5ta. Ed., España, McGraw Hill
6. Piattini Mario G., 2000; "Análisis y Diseño Detallado"
7. Kendall & Kendall, 1997; "Análisis y Diseño de Sistemas"
8. Kendall & Kendall, 2002; "Análisis y Diseño de Sistemas" 913 Tercera Edición Editorial Amy Kohen, Mexico
9. Larman, 1999; UML y Patrones
10. Fowler, 1999; UML Distilled M
11. Cerrini Maria Dolores, 2002, Ignacio Pablo 2002
12. Alvarez Rodrigues, 2006; "Desarrollo Ágil con J2EE con herramientas Open Source",

Artículo

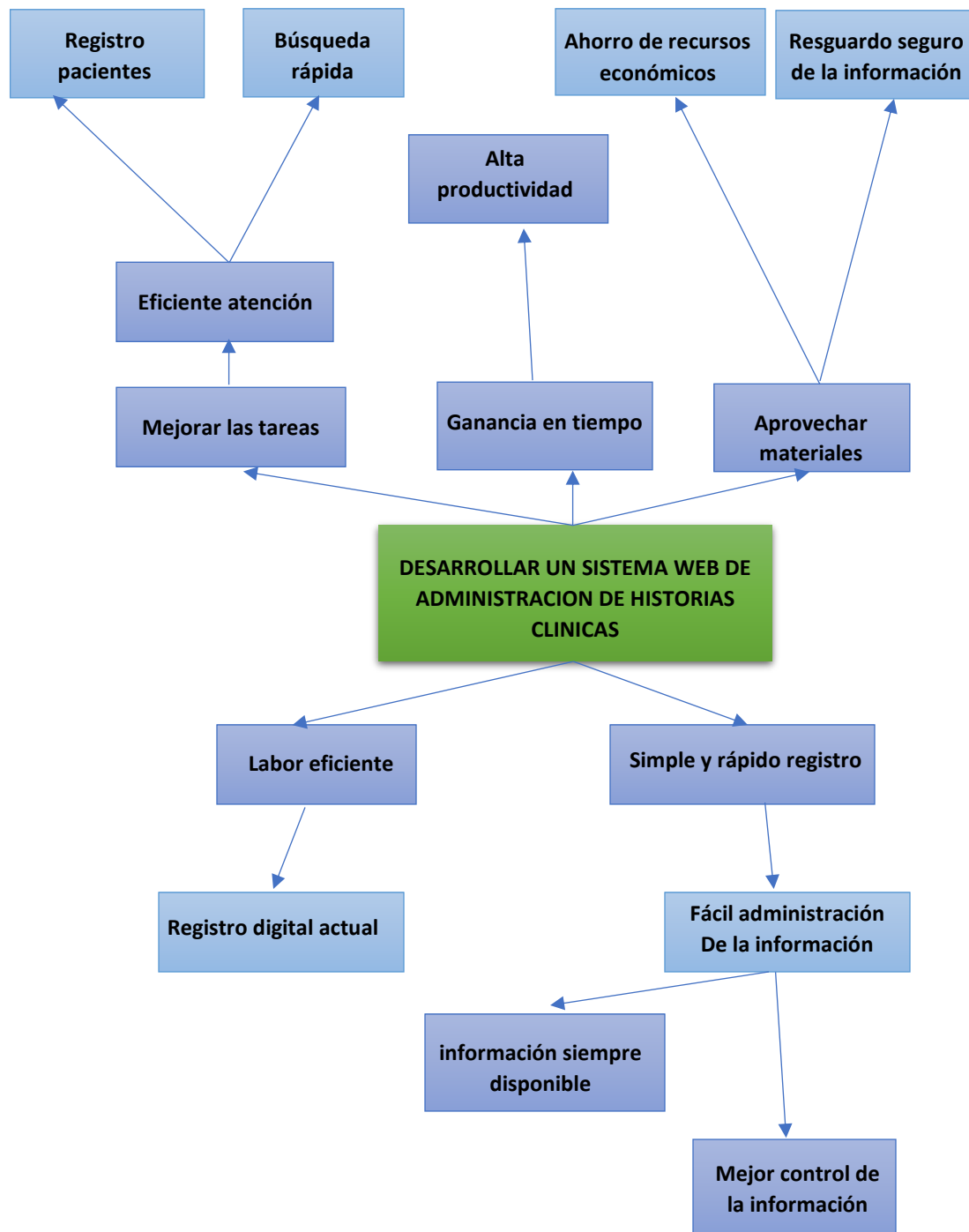
13. Schmuller Joseph, 2005; "Aprendiendo UML en 24 horas", Prentice Hall
14. Villate Jaime E., 2001; "Introducción al XML" Universidad de Oporto <http://www.oasisopen.org/cover/sgml-xml.html>
15. Martin James y Odell James ; "Análisis y Diseño Orientado a Objetos con UML"
16. Rumbaugh James & Blaha Michael & Premerlani William & Eddy, Frederick & Lorensen, William, 1995; "Modelado y Diseño Orientado a Objetos", España
17. Yourdan Edward, 1992; "Análisis y Diseño de Sistemas", Ed.
18. Antonio J. Martin Sierra, 2007; "Ajax en J2ee", Primera Edición, AlfaOmega Grupo Editor S-A de C.V. México ISBN: 978-970 RA-MA Editorial, Madrid, España
19. Carlos Jorge Navia Alanez, 1997; "Elaboración Científica del Perfil de Tesis", Primera Edición, EDCON Editores La Paz-Bolivia
20. Depto. Bienestar Social, 2010; "División de Salud y Servicios Psicológicos N° 055/2011" Universidad Mayor de San Andrés La Paz – Bolivia
21. Fuentes, 2003; <http://www.luiscifuentes.el/imm3100/clases/IMM3100-1b> Metodología Análisis de Riesgo
22. Luis Fernández , 2000, Mario G. Pattini José A. Calvo Manzano, Joaquín Cervera, "Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión"
23. Analista Programador Universitario JTP Jorge R. Mendoza, 2010, Universidad Nacional de Jujuy

ANEXOS

ARBOL DE PROBLEMAS



ARBOL DE OBJETIVOS



MANUAL DE USUARIO

Tenemos la ventana de acceso, a la cuál usted deberá ingresar con un usuario y password, ya asignados por el administrador del sistema



The image shows a login interface for a system titled "Gestión Clínica". The interface is contained within a light gray box. At the top, it says "Gestión Clínica". Below that, it prompts the user to "Ingrese sus datos de Acceso". There are two input fields: "Usuario" and "Password". Both fields are highlighted with red rectangular boxes. To the right of the "Usuario" field is a small person icon, and to the right of the "Password" field is a small eye icon. Below the "Password" field is a blue button labeled "Ingresar". At the bottom left of the form, there is a link that says "Olvidé mi password". Two red arrows point from the right side of the form towards the "Usuario" and "Password" fields.

Dependiendo de los privilegios que el usuario tenga. Puede acceder a distintos apartados del sistema.

- Administrador
- Personal (A)
- Personal (B)
- Enfermera
- Enfermera Pantalla
- Doctor Especialista
- Asistente

Ingresando como Personal “A”

Ingresamos al registro de Paciente (llenando los campos validados)

The screenshot shows the 'Paciente Nuevo' form with the following fields and values:

- Apellido Paterno (*): apellido Paterno
- Apellido Materno (*): Apellido Materno
- Nombre (*): nombre
- Fecha de Nacimiento (*): dd/mm/aaaa
- Sexo: Femenino
- Estado Civil: Soltero
- Tipo Documento: DNI
- Número Documento: Documento
- Dirección: Dirección
- Teléfono: Teléfono
- Email: Email
- Ocupación: Ocupacion
- Persona Responsable: Persona responsable
- Alergias: Alergia
- Intervenciones quirúrgicas: Intervenciones quirúrgicas
- Vacunas completas: SI

Buttons: Guardar, Cancelar

Presionar en “Guardar” para realizar el registro o en “Cancelar” para cancelar dicho registro.

Ingresando como Personal “B”

Accedemos a asignarle una atención al paciente, derivándole a un especialista.

The screenshot shows the 'Atención Nueva' form with the following fields and values:

- Número Documento (*): [Empty]
- Paciente (*): [Empty]
- Consultar: [Button]
- Servicio (*): Ginecología Bs.60.00
- Especialista (*): Gonzales Paz Cesar
- Costo: [Empty]






Buttons: Guardar, Cancelar

Pudiendo “Guardar” o “Cancelar” la atención requerida.

El usuario puede observar la atención registrada en la cola

Atención Nueva Agregar

Copy Excel CSV PDF Buscar:

Opciones ^	Fecha	Registrador	Servicio	Especialista	Paciente	DNI	Edad	Costo	Estado
 	2020-06-25 - 21:35:30	Gonzales Paz Cesar	Ginecología	A A A	A A A	12345	0 años, 0 meses, 24 días	0.00	Registrado
 	2020-06-24 - 10:36:47	Gonzales Paz Cesar	Ginecología	A A A	pet pet pet	23232323	5 años, 3 meses, 20 días	90.00	Triage
	2020-06-24 - 02:37:06	Gonzales Paz Cesar	Ginecología	Gonzales Paz Cesar	Gonzales Paz Cesar	11111111	3 años, 3 meses, 17 días	60.00	Atendido

Opciones Fecha Registrador Servicio Especialista Paciente DNI Edad Costo Estado

Mostrando 1 a 3 de 3 registros Anterior 1 Siguiente

Ingresando como “Enfermera”

Ingresando bajo este privilegio, se advierten dos pantallas, uno de registro y otro de control.

En la pantalla de registro, podemos proceder a registrar la información básica del paciente.

Pantalla de registro:

Cola Enfermería

Paciente: CCC CI: 1234 Edad: 0 años, 1 meses, 4 días

Servicio: Ginecología Especialista: B B B

Presion Arterial (mmHg): Presion Arterial Temperatura (C°): temperatura

Frecuencia Respiratoria (x Minuto): Frecuencia Respiratoria Frecuencia Cardiaca (x Minuto): Frecuencia Cardiaca Saturacion O2: Saturacion

Peso: (Kg) utilice punto para decimales Talla: (cm) Imc: Estado: estado

Guardar Cancelar

Pantalla de Control:

Donde podremos observar el estado de la cola

Pantalla Control			
#	Enfermeria	#	Consultorio
1	CCC		

Ingresando como “Médico Especialista” o “Asistente”

Luego de que el(la) paciente, pase por enfermería, Pasa a consultorio, donde el médico especialista o en todo caso la asistente, pasa a registrar información valiosa para el llenado de su Historial clínico, conociendo previamente la información básica de enfermería.

Pantalla de registro:

Motivo de la Consulta: **TE:**

Antecedentes:

Alergias: **Intervenciones Quirúrgicas:** **Vacunas completas:**

Examen Físico:

Código (Nombre) **Diagnósticos** **Aplicar**

Borrar	Tipo	Enfermedad
<input type="button" value="Consultar"/>		

Tratamiento:

Plan:

Estudios: **Próxima Cita:**

Receta

Borrar	Medicamento	Presentación	Dosis	Duración	Cantidad
<input type="button" value="Borrar"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

MANUAL DE ADMINISTRADOR

Este manual tiene como finalidad dar a conocer al encargado de administrar el sistema las características y las formas básicas de funcionamiento del “Sistema Web de Administración de Historias Clínicas”

Requerimientos Técnicos para el uso de “Sistema Web de Administración de Historias Clínicas”

- Navegador Firefox V. 77.0.1
- Navegador Google Chrome V. 83.0.4103.116
- Tener instalado Acrobat Reader
- Contar con acceso a internet o red

Acceso al sistema

El acceso al sistema debe estar puesta como usuario administrador, para tal caso el usuario y contraseña serán:

Usuario: admin

Password: admin



Gestión Clínica

Ingrese sus datos de Acceso

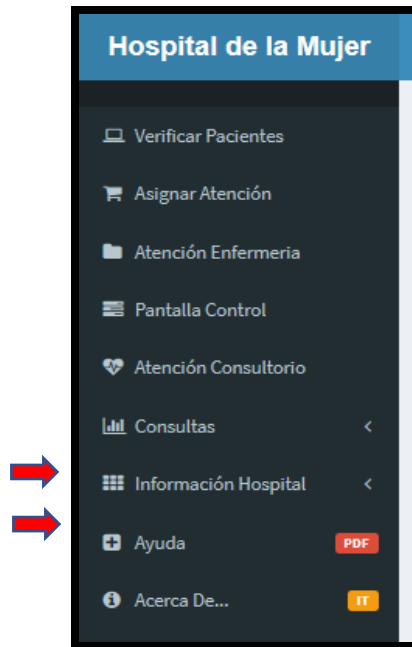
Usuario 

Password 

[Ingresar](#)

[Olvidé mi password](#)

Ingresando como administrador del sistema, tendremos control de todos los apartados anteriores, añadiendo otras importantes como “Información del Hospital” y “Consultas”.



En el apartado de Consultas, podremos realizar la consulta de pagos e historiales realizados en el día, y por fechas.

Pagos

Desde: 07/07/2020 Hasta: 07/07/2020

Copia Excel CSV PDF

Buscar (Número de Documento): CI, NIT

Fecha	Registrador	Servicio	Especialista	Paciente	Costo	Estado
2020-07-07 - 11:21:22	Gonzales Paz Cesar Ariel	Ginecología	B B B	C C C	70.00	Registrado

Mostrando 1 a 1 de 1 registros

Anterior 1 Siguiente

Historias del día

Desde: 07/07/2020 Hasta: 07/07/2020

Copia Excel CSV PDF

Buscar (Número de Documento): CI, NIT

Fecha	Registrador	Servicio	Especialista	Paciente	Costo	Imprimir
No existen datos						

Mostrando 0 a 0 de 0 registros

Anterior Siguiente

En el apartado de “Información Hospital” tendremos cinco opciones:

- Configuraciones
- Servicios
- Personal
- Permisos
- Diagnóstico

Partiendo de “Configuraciones”, podremos editar la información del sistema:

Opciones	Razon Social	Ruc	Email	Telefono	Direccion	Responsable
	Hospital de la Mujer	20987654321	hospitaldelamujer@hdm.com	123456789	Miraflores, Av. Saavedra No. 2273	Dr. Pascual

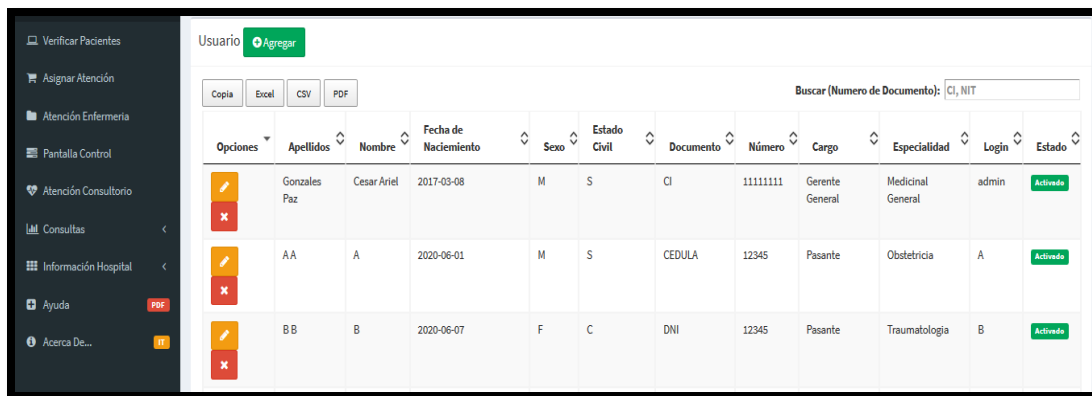
Siguiendo con servicios podemos visualizar y añadir nuevos servicios en el Hospital.

Opciones	Nombre	Costo	Estado
	Obstetricia	60.00	Activado
	Ginecología	70.00	Activado
	Ecografía	70.00	Activado







Nota Importante:

En el apartado de Edición, no podremos borrar un servicio que ya haya sido utilizado por un paciente, tan sólo podremos desactivarlo, para futuras referencias.

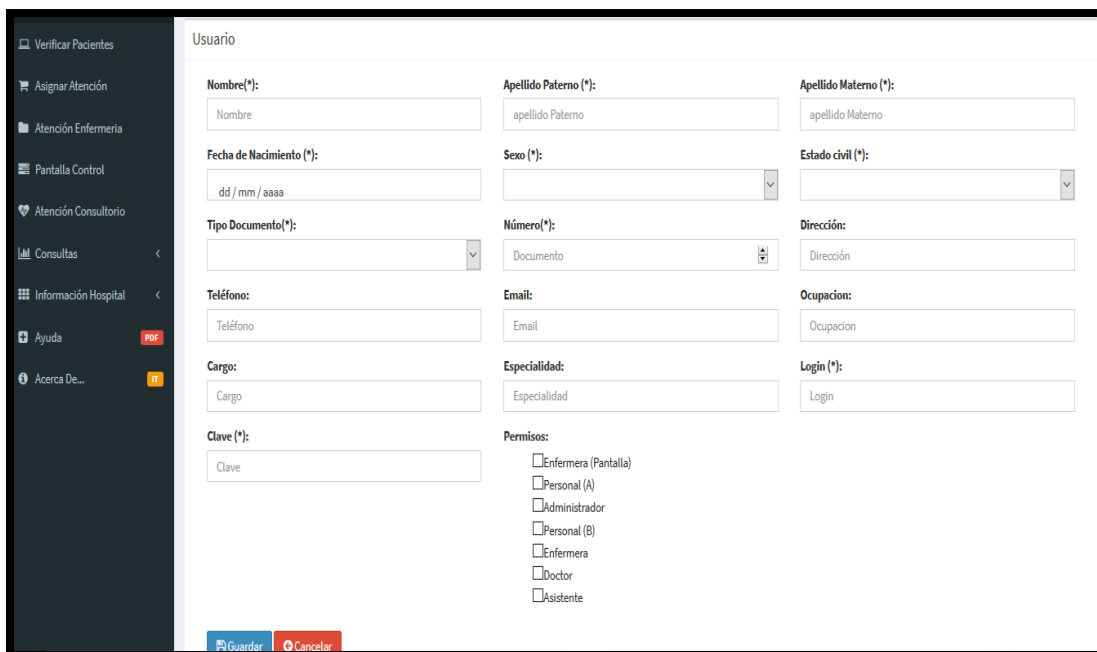
En el apartado de Personal, podremos administrar a todos los usuarios, y agregar nuevos.



The screenshot shows the 'Usuario' management interface. At the top, there is a search bar for 'Buscar (Número de Documento):' with 'CI, NIT' entered. Below the search bar are buttons for 'Copia', 'Excel', 'CSV', and 'PDF'. The main area contains a table with the following columns: Opciones, Apellidos, Nombre, Fecha de Nacimiento, Sexo, Estado Civil, Documento, Número, Cargo, Especialidad, Login, and Estado. The table lists three users:

Opciones	Apellidos	Nombre	Fecha de Nacimiento	Sexo	Estado Civil	Documento	Número	Cargo	Especialidad	Login	Estado
 	Gonzales Paz	Cesar Ariel	2017-03-08	M	S	CI	11111111	Gerente General	Medicinal General	admin	Activado
 	AA	A	2020-06-01	M	S	CEDULA	12345	Pasante	Obstetricia	A	Activado
 	BB	B	2020-06-07	F	C	DNI	12345	Pasante	Traumatologia	B	Activado

En caso de agregar nuevos usuarios, podremos asignarle un privilegio y contraseña correspondiente



The screenshot shows the 'Usuario' form for adding a new user. The form includes the following fields and options:

- Nombre (*):** Text input field.
- Apellido Paterno (*):** Text input field.
- Apellido Materno (*):** Text input field.
- Fecha de Nacimiento (*):** Date input field (dd / mm / aaaa).
- Sexo (*):** Dropdown menu.
- Estado civil (*):** Dropdown menu.
- Tipo Documento (*):** Dropdown menu.
- Número (*):** Text input field.
- Dirección:** Text input field.
- Teléfono:** Text input field.
- Email:** Text input field.
- Ocupación:** Text input field.
- Cargo:** Text input field.
- Especialidad:** Text input field.
- Login (*):** Text input field.
- Clave (*):** Text input field.
- Permisos:** List of checkboxes:
 - Enfermera (Pantalla)
 - Personal (A)
 - Administrador
 - Personal (B)
 - Enfermera
 - Doctor
 - Asistente

At the bottom of the form, there are buttons for 'Guardar' and 'Cancelar'.

En el apartado de Permisos, visualizamos los permisos habilitados en el sistema:

Permiso

Copia Excel CSV PDF

Nombre

Personal (B)

Personal (A)

Enfermera (Pantalla)

Enfermera

Doctor

Nombre

Mostrando 1 a 5 de 7 registros

En el apartado de Diagnóstico, podremos agregar y editar enfermedades, la lista actual esta tomada por la normativa CISE – 10.

Diagnóstico + Agregar

Enfermedad o Código: 🔍 Buscar

Copia Excel CSV PDF

Opciones	Código	Enfermedad
	M15.4	(Osteo)Artrosis Erosiva
	M15.0	(Osteo)Artrosis Primaria Generalizada
	Z38.2	A Terminio (Producto Unico en Lugar no Especificado)
	Z38.0	A Terminio (Producto Unico Nacido en Hospital)
	Z62.4	Abandono Emocional del Nino

Opciones **Codigo** **Enfermedad**

Mostrando 1 a 5 de 200 registros

Si requiere resolver alguna duda de carácter técnico, podrá comunicarse a la Dirección de Sistemas del Hospital de la Mujer, si tiene alguna duda de carácter académico, podrá ponerse en contacto con el responsable funcional.

AVAL DE CONFORMIDAD

La Paz, 10 de Julio de 2020

Señor:

Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II

Presente. -

REF.: Aval de conformidad

Distinguido Ingeniero.

Mediante la presente, tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS" CASO: HOSPITAL DE LA MUJER, que propone el postulante Univ. Cesar Ariel Gonzales Paz con cédula de identidad, 8313197 L.P. para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente.



Cesar Ríos Camacho
RESPONSABLE DE SISTEMAS
HOSPITAL DE LA MUJER

Ing. Cesar Ríos Camacho
TUTOR ESPECIALISTA

La Paz, 10 de Julio de 2020

Señor:

Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II

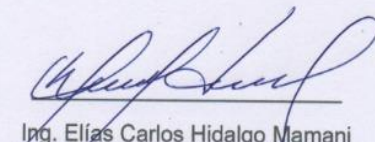
Presente. -

REF.: AVAL DE CONFORMIDAD

Distinguido Ingeniero.

Mediante la presente, tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS" CASO: HOSPITAL DE LA MUJER, que propone el postulante Cesar Ariel Gonzales Paz con cédula de identidad, 8313197 L.P. para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente.



Ing. Elías Carlos Hidalgo Mamani
TUTOR REVISOR



Hospital de la Mujer La Paz - Bolivia

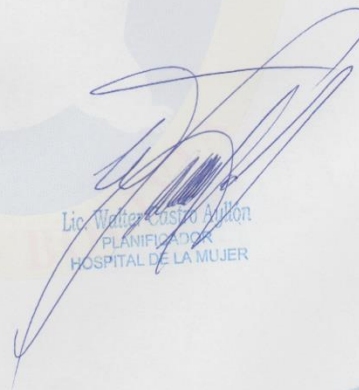
"Gobierno Autónomo Departamental de La Paz"
"Servicio Departamental de Salud La Paz"

CERTIFICACIÓN

Por medio de la presente, el Hospital de la Mujer, mediante la Unidad de Sistemas, Redes, en coordinación con la Unidad Gestora de Planificación y Proyectos de la Institución, se certifica que el proyecto de grado para la obtención del título en INGENIERÍA DE SISTEMAS titulado "**SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS**" fue desarrollado en su totalidad por el estudiante Sr. CESAR ARIEL GONZALES PAZ con CI: 8313197 L.P. , dicho Sistema desarrollado, cumple con los requerimientos de la institución y que está en etapa de implementación.

Atte.


Lic. Cesar Rizo Camacho
RESPONSABLE DE SISTEMAS
HOSPITAL DE LA MUJER


Lic. Walter Cosío Aullón
PLANIFICADOR
HOSPITAL DE LA MUJER

AVAL DE CONFORMIDAD

El Alto, 13 de Julio del 2020

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS

Presente. -

Ref.: Aval de Conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado, "SISTEMA WEB DE ADMINISTRACION DE HISTORIAS CLÍNICAS" CASO: HOSPITAL DE LA MUJER, que propone el postulante Universitario, **Cesar Ariel Gonzales Paz**, con cédula de identidad **8313197 L.P.**, para su defensa pública, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



MSc. Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO