

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



PROYECTO DE GRADO

“SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS”

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

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCION INGENIERIA DE SISTEMAS INFORMÁTICOS**

POSTULANTE: UNIV. NEIL ANGEL GRANEROS FLORES
TUTOR METODOLÓGICO: M.SC. GROVER RODRIGUEZ RAMÍREZ
TUTOR: PH.D. MARISOL TELLEZ RAMIREZ
ASESOR ADJUNTO: DR. RAFAEL MONTAÑO

La Paz - Bolivia
2022

Dedicatoria

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

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

A mi hermana Vania Graneros Flores por ser un modelo a seguir y por ser una hermana ejemplar.

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

Agradecimiento

Agradezco a la Universidad Mayor de San Andrés, Carrera de Informática

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

A todos los docentes, quienes me concedieron sus conocimientos. Al M. Sc. Grover Rodríguez, por dar su tiempo y paciencia en la revisión del proyecto

A la Ph.D. Marisol Téllez por la colaboración prestada en la elaboración del proyecto.

Al Lic. Florencio Antonio por el recibimiento en la unidad RED-UMSALUD, así como a mis compañeros de trabajo Galo y Joel por compartir sus conocimientos hacia mi persona.

Al Doctor Rafael Montaña del Instituto de Genética, quien me brindo su tiempo y cooperación en la elaboración del proyecto y a todo el plantel médico.

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

A mis amigos y compañeros por darme consejos y aliento para salir adelante, mencionando a algunos: Gary, Kamil, Laura, Pilar, Rudy, Priscila, Boris, David, Kenji, Iván, Judith, Mel, entre otros muchas gracias.

A los grupos de estudio GNU Linux y a GDCS por el conocimiento dado y compartido entre todos.

Muchas Gracias

I. ÍNDICE

CAPITULO I MARCO REFERENCIAL	1
1.1 INTRODUCCIÓN	1
1.2 PROBLEMA.....	3
1.2.1 Antecedentes al Problema.....	3
1.2.2 PLEANTEAMIENTO DEL PROBLEMA.....	11
1.2.3 FORMULACIÓN DEL PROBLEMA	11
1.3 OBJETIVOS	12
1.3.1 OBJETIVO GENERAL.....	12
1.3.2 OBJETIVOS ESPECÍFICOS	12
1.4 JUSTIFICACIÓN	13
1.4.1 JUSTIFICACIÓN SOCIAL.....	13
1.4.2 JUSTIFICACIÓN ECONÓMICA.....	13
1.4.3 JUSTIFICACIÓN TECNOLÓGICA.....	13
1.5 ALCANCES Y LIMITES	14
1.5.1 ALCANCE TEMÁTICO.....	14
1.5.2 ALCANCE TEMPORAL	15
1.5.3 ALCANCE GEOGRÁFICO.....	15
1.5.4 LÍMITES.....	15

1.6	METODOLOGÍA	16
1.6.1	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN.....	16
1.6.2	METODOLOGÍA DE DESARROLLO	16
	CAPITULO II MARCO TEÓRICO	20
2.1	MARCO INSTITUCIONAL	20
2.2	METODOLOGIA	26
2.2.1	Metodologías Ágiles del Desarrollo.	26
2.2.2	Proceso Unificado Ágil (AUP).....	27
2.2.3	Lenguaje Unificado De Modelado (UML).	34
2.2.4	UML-Based Web Engineering.	35
2.3	INFORMACIÓN	37
2.3.1	Características de la información.....	38
2.3.2	Información en el contexto Tecnológico.	39
2.4	TEORÍA GENERAL DE LOS SISTEMAS	40
2.5	SISTEMA.....	41
2.5.1.1	Características De Los Sistemas	42
2.5.2	Principios de los Sistemas.	43
2.5.3	Elementos de los Sistemas.....	44

2.6	SISTEMA DE INFORMACIÓN	45
2.6.1	Tipos de Sistemas de Información.....	46
2.6.2	Funciones de un Sistema de Información.....	48
2.7	HISTORIA CLINICA.....	49
2.7.1	Características De Una Historia Clínica.	50
2.7.2	Gestión de Historias Clínicas.....	52
2.7.3	Historias Clínicas Electrónicas.	53
2.8	MARCO TECNOLÓGICO.....	54
2.8.1	El Lenguaje JavaScript.	54
2.8.2	Node Js.....	55
2.8.3	React.	56
2.8.4	PostgreSQL.....	57
2.8.5	Object-Relational Mapping (ORM).....	58
2.8.6	Orm Sequelize.	58
2.9	METRICAS DE CALIDAD SEGURIDAD DE SOFTWARE	60
2.9.1	Metodología WebQem.....	60
2.9.2	Fases de WebQem.	60
2.9.3	Características de WebQem.....	61
2.10	MODELO DE COSTOS DE SOFTWARE	66

2.10.1	Modelo COCOMO II.....	66
REFERENCIAS BIBLIOGRÁFICAS		69

II. INDICÉ DE FIGURAS

Figura 1.1.1 <i>Iniciativa de Digitalización de los Sistemas de Salud de la Oficina Regional de la OMS para Europa. Fuente: (OMS, 2019)</i>	1
Figura 1.2.1 <i>Tabla de aplicación de TIC's en el subsector público. Fuente: Gil López, E., & Medinaceli Díaz, K. (2019)</i>	3
Figura 1.2.2 <i>Tabla de aplicación de TIC's en el subsector de la Seguridad Social. Fuente: Gil López, E., & Medinaceli Díaz, K. (2019)</i>	4
Figura 1.2.3 <i>Tabla de aplicación de TIC's en el subsector privado y actores Fuente: Gil López, E., & Medinaceli Díaz, K. (2019)</i>	4
Figura 1.2.4 <i>Estructura Orgánica del Instituto de Genética. Fuente: Instituto de Genética</i>	10
Figura 1.6.1 <i>Tablero Kanban Básico Fuente: Elaboración Propia.</i>	18
Figura 1.6.2 <i>Gráfica de patrones de arquitectura MVC. Fuente: FreeCodeCamp, Hernandez (2021)</i>	19
Figura 2.1.1 <i>Historia Clínica. Fuente: Instituto de Genética</i>	22
Figura 2.1.2 <i>Historia Clínica para paciente con Síndrome de Down. Fuente: Instituto de Genética</i>	23
Figura 2.1.3 <i>Historia Clínica para pacientes con Infertilidad. Fuente: Instituto de Genética</i>	24
Figura 2.1.4 <i>Historia Clínica en ACCESS. Fuente: Instituto de Genética</i>	25
Figura 2.2.1 <i>Fases e hitos de un proyecto Fuente: JACOBSON (2000)</i>	33
Figura 2.2.2 <i>Actividades de la metodología AUP Fuente: JACOBSON (2000)</i>	33

Figura 2.2.3 <i>Visión general de los modelos UML. Fuente: UWE – UML-based Web Engineering (2016)</i>	36
Figura 2.4.1 <i>Mapa Conceptual Finalidad de la TGS. Fuente: Gutiérrez Gómez et al., 2013</i>	41
Figura 2.9.1 <i>Características WebQem, Usabilidad. Fuente: Olsina, 1999</i>	63

CAPITULO I MARCO REFERENCIAL

1.1 INTRODUCCIÓN

En el ámbito médico, la historia clínica es un documento fundamental que recoge toda la información relevante sobre la salud de un paciente, incluyendo sus antecedentes médicos, diagnósticos, tratamientos y evolución clínica. Actualmente, la transformación digital está teniendo un gran impacto en este sector, permitiendo mejorar los procesos y el trabajo del personal médico. La implementación de tecnología en la atención sanitaria trae beneficios y ventajas, como la mejora de tratamientos, la optimización de los recursos, la atención más flexible a los pacientes y la mejora de la eficiencia en los procesos médicos.

A pesar de los avances tecnológicos, muchos actores en el sector médico no están completamente conscientes del impacto que la tecnología genera, lo que puede llevar a percances y problemas en el uso de los sistemas digitales. Para abordar este problema, es necesario que los gobiernos y los entes encargados de la salud establezcan normativas que prioricen la salud digital.



Figura 1.1.1 *Iniciativa de Digitalización de los Sistemas de Salud de la Oficina Regional de la OMS para Europa.*

Fuente: (OMS, 2019)

Según el simposio de la OMS (Organización Mundial de la Salud) sobre el futuro de sistemas de salud digitales en la región europea (2019) indicó:

La adopción de tecnologías digitales en el ámbito de la salud está ampliamente reconocida como crucial para el buen funcionamiento de los sistemas en salud y para capacitar a las personas en el marco de la transición hacia una atención integrada y centrada en la persona.

En este ámbito, los centros de salud, en su mayoría, han ido automatizando algunas tareas y procesos para mejorar la atención a los pacientes y controlar sus recursos. Además, la telemedicina se ha vuelto cada vez más común, lo que permite brindar servicios de salud a la ciudadanía con ayuda de la tecnología.

En este contexto, el proyecto de grado “Sistema Web de Administración de Historias Clínicas. Caso: Facultad de Medicina, Instituto de Genética, Unidad de Genética Médica” se presenta como una solución para administrar las historias clínicas de los pacientes de manera más eficiente y accesible. El objetivo principal de este proyecto es desarrollar un sistema web mediante una metodología ágil (AUP) con el propósito de facilitar el acceso a la información de historias clínicas de los pacientes, lo que permitirá al personal médico agilizar su trabajo y brindar una mejor atención al paciente. El proyecto se enfoca en la unidad de Genética Médica del Instituto de Genética de la Facultad de Medicina de la Universidad Mayor de San Andrés, reconocida por tener investigadores y logros importantes en el área de genética médica, citogenética y genética toxicológica.

1.2 PROBLEMA

1.2.1 Antecedentes al Problema

1.2.1.1 Estado del Arte.

Actualmente el proceso de registro de historias clínicas se realiza de la digitalización de historias clínicas todavía de realiza de forma tradicional, usando medios físicos para guardar la información de un paciente, y esto tal vez se deba a falta de presupuesto para el equipamiento necesario y la adaptación a los medios tecnológicos.

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

La deficiencia general que se presenta en Bolivia es la dificultad en el acceso a la información ya que ésta no se encuentra disponible en los sitios web oficiales, y algunas instituciones del ámbito público o privado no cuentan con memorias institucionales anuales para poder acceder a la información.

Ellos también realizaron un estudio de campo basado en entrevistas donde se generó las siguientes tablas, identificando la aplicación, uso y desarrollo de TIC's en centros de salud públicos y privados en Bolivia.

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

Figura 1.2.1 *Tabla de aplicación de TIC's en el subsector público.*

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

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

Figura 1.2.2 *Tabla de aplicación de TIC's en el subsector de la Seguridad Social.*

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

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

Figura 1.2.3 *Tabla de aplicación de TIC's en el subsector privado y actores*

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

Sacando las conclusiones de dicho artículo (Gil López, E., & Medinaceli Díaz, K. 2019) sobre la situación de la aplicación de la historia clínica electrónica en los establecimientos de salud del Sistema Nacional de Salud de Bolivia se verifico que:

- La Caja de Salud de la Banca Privada desde la gestión de 2005, los profesionales de esta caja han desarrollado su propio software médico y sistema administrativo médico (SAMI). El cual cuenta con módulos con funciones específicas para cada usuario, que pude generar reportes de exámenes, laboratorios, médicos y de seguimiento.
- La Corporación del Seguro Social Militar (Cossmil) 2004 una empresa boliviana desarrolló el Sistema de Gestión Hospitalaria (SIGEH). Desde la gestión de 2014 se ha desarrollado el Sistema de Información Integrado de Control y Seguimiento Hospitalario (SISHAP). El SISHAP se ha implementado también en las ciudades de Sucre y Puerto Suárez
- El Hospital Arco Iris (HAI) ha implementado hace dos años el sistema openHAI, que contiene la HCE (historia clínica electrónica) de hospitalización, terapia intensiva (UTI), emergencias y consulta externa (desde septiembre de 2015), el cual tiene un módulo específico para guardar imágenes de resonancias, tomografías, rayos X y cualquier otra imagen radiológica.

1.2.1.2 Trabajos Similares.

1.2.1.2.1 Contexto Internacional

[Jácome, F. (2015)].

“Desarrollo del Sistema de Administración de Historias Clínicas del Departamento Médico de la Empresa Imprenta Mariscal Cia.Ltda”.

La problemática principal descrita en el proyecto, fue que no se contaba con herramientas tecnológicas que ayuden con las actividades de la organización, por lo que las tareas realizadas en dicha institución son realizadas manualmente que genera

desorganización, pérdida de información, duplicidad de historias clínicas y demoras al buscar historias clínicas. Su objetivo es desarrollar una aplicación que permita la administración y gestión de la información generada en la institución según los requerimientos que esta necesita ser cubiertas y resueltas. Utilizando la metodología RUP (Rational Unified Process) y UML (Lenguaje Unificado de Modelado) para el modelado del sistema, las herramientas que se usaron en este proyecto fueron; Filemaker (Front End) y FileMaker Server (Back End y BD). Realizado en la Carrera de Ingeniería en Sistemas Informáticos de la Universidad Israel, Quito Ecuador.

[Ortiz, O. L. E. (2022)].

“Desarrollo de un sistema web de gestión de historias clínicas en un consultorio privado de medicina general”.

Tuvo como problemática principal fue hacia la atención y acceso a la información de los pacientes, en la obtención de los datos de los pacientes se tuvo problemas ya que estos se guardaban de forma física y el llenado de formularios poco legibles generaba información errónea. Su objetivo es estandarizar, organizar y automatizar los procesos de diagnóstico y manipulación de la historia clínica de los pacientes del Área de Consulta Externa, facilitando el acceso a la información, disminuir errores al digitalizar la información y estandarizar los formularios de registro. Utilizando la metodología RUP (Rational Unified Process) y UML (Lenguaje Unificado de Modelado) para el modelado del sistema, las herramientas que se usaron en este proyecto fueron; MySQL 5.0.45 como base de datos, JDK 1.6.0 como framework de desarrollo (Front End) y GlassFish V 2.0 update 2 como servidor de aplicaciones. Realizado en la Facultad De Ingeniería De Sistemas de la Escuela Politécnica Nacional, Quito Ecuador.

[Padilla, P. J. (2018)].

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

Observo problemas por los cuales algunos centros de salud no implementan un sistema de historia clínica electrónica (hce), como ser el crecimiento exponencial de la información, compatibilidad de los archivos, costos, entre otros. Su objetivo es diseñar e implementar un sistema de administración y consulta de historias clínicas electrónicas (HCE) mediante el uso de tecnología webservices, desarrollando tecnologías webservice de acceso y administración a la información y con el diseño de pruebas de funcionamiento. Utilizando la metodología RUP, las herramientas que se usaron en este proyecto son; los lenguajes como C, C++ en el entorno Visual Studio, además de HTML 5, JavaScript y CSS. Realizado en la Carrera de Ingeniería de Sistemas, Facultad de Ingenierías de la Universidad de Córdoba, Colombia.

1.2.1.2.2 Contexto Nacional

[Paz, G. C. A. (2020)].

“SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS”.

Tuvo como problemática principal que el manejo de historias clínicas de forma física ocasiona molestias en el personal médico y a los pacientes, generando expedientes clínicos desactualizados, pérdida de información, expedientes duplicados y demora en la atención a los pacientes. Su objetivo es desarrollar un sistema web de administración de historias clínicas. Utilizando la metodología SCRUM y UWE-UML para el desarrollo y modelado del sistema. Para el desarrollo del sistema, se utilizó el lenguaje de programación PHP (Versión 7.4.6), para la base de datos MariaDB (Versión 10.1.21) y además de Bootstrap (Versión 4.5) para la parte del Front End. Realizado en la Carrera Ingeniería de Sistemas de la Universidad Pública de El Alto.

[Flores, G. P. (2018)].

“Sistema web de administración de historias clínicas Caso: centro médico quirúrgico Erzengel”

Los problemas que identifiqué en el centro de salud fueron, la redacción manual de historias clínicas, duplicidad de historias clínicas, deterioro de los documentos físicos, retraso en la atención de los pacientes debido al retraso al buscar y registrar información de forma manual y además de expedientes clínicos desactualizados. Su objetivo es desarrollar un sistema web de administración de Historias Clínicas, que permita mejorar las tareas de los especialistas en salud y dando una atención más optima a los pacientes de dicha institución. Utilizando la metodología SCRUM, las herramientas que se usaron fueron; PHP como lenguaje de programación para el desarrollo del sistema, Bootstrap para la parte del Front End, PostgreSQL como gestor de base de datos. Realizado en la carrera de Informática de la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés.

[Ponce, Y. Y. A. (2016)].

“Software como servicio para la administración de historias”.

Tuvo como problemática central que el control y seguimiento de los pacientes era ineficaz para los procesos que se requieren en centros de salud. Su objetivo es desarrollar un software como servicio, para la administración de historias clínicas facilitando el control y seguimiento de la información de los pacientes. Utilizando la metodología UWE aplicando los modelos de diseño web. Para el desarrollo del sistema, se utilizó el framework Django mediante el lenguaje de programación Python para la creación del webservice y Heroku como servidor en la nube. Realizado en la carrera de Informática de la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés.

1.2.1.3 Antecedentes Institucionales

El Instituto de Genética dependiente de Facultad de Medicina, Enfermería, Nutrición y Tecnología Médica de la Universidad Mayor de San Andrés ubicada en la ciudad de La Paz Av. Saavedra N° 2246, Miraflores, Edificio de la Facultad de Medicina Piso 9. Dirigida por la M. Sc. Dra. Ximena Aguilar Mercado, realiza sus funciones desde

el 15 de junio de 1972, siendo el único centro de referencia a nivel nacional en el campo de la genética. Actualmente el instituto de Genética está conformado por tres áreas, el área de interacción social, área de docencia y el área de investigación, con las unidades de Genética Medica, Citogenética y Genética Toxicológica, brindando servicios a la ciudadanía y a la comunidad estudiantil.

Misión:

Es un Instituto de investigación científica en salud genética a nivel nacional, con capacidad para planificar, ejecutar y evaluar proyectos de investigación y desarrollo tecnológico, socialmente comprometida, que presta servicios docentes, asistenciales y de laboratorio de alta especialidad y tecnología avanzada

Visión:

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

El siguiente organigrama muestra la estructura del instituto en las áreas ya descritas.

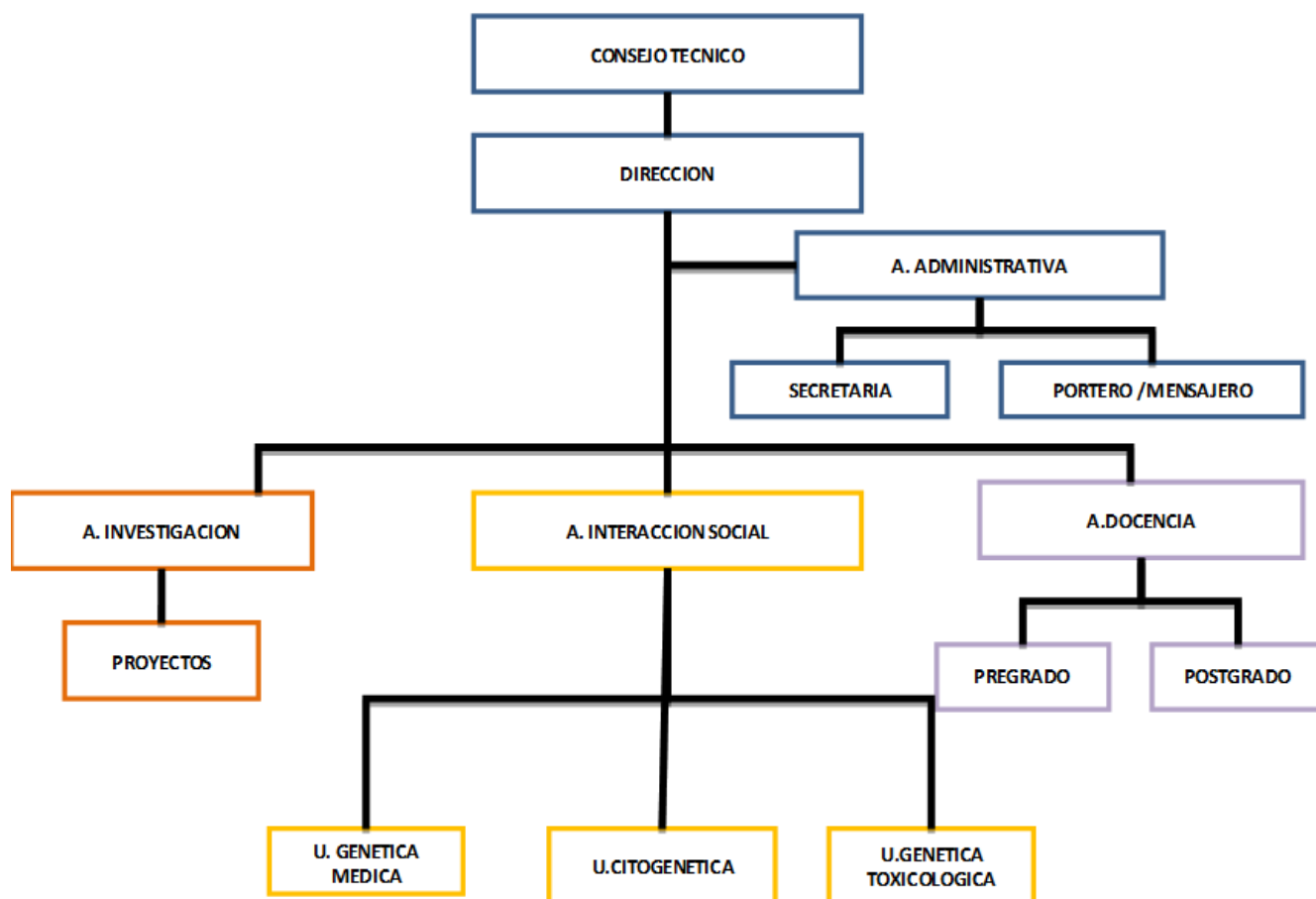


Figura 1.2.4 Estructura Orgánica del Instituto de Genética.

Fuente: Instituto de Genética

La unidad de genética medica se encarga de la atención y seguimiento a los pacientes, siendo esta la unidad que tiene la necesidad de poder acceder a la información de las historias clínicas. Dicha unidad está dentro del proyecto de Telemedicina, el cual actualmente está muy activo realizando proyectos que ayuden a la población.

1.2.1.4 Lista de Problemas

- Por la falta de conocimientos informáticos y la falta de equipamiento de computación, llevan a que el registro de los datos del paciente sea de una forma manual y en archivos Excel.
- La ausencia de un formato estandarizado para cada historia clínica, generan dificultades al acceder a la información de un paciente.
- La mala manipulación de la información genera pérdidas y duplicidad de las historias clínicas, ocasionando conflictos en las unidades, atrasando la obtención de datos de los pacientes.
- La ausencia de una base centralizada genera retraso en el registro y acceso de la información de los datos de los pacientes.

1.2.2 PLEANTEAMIENTO DEL PROBLEMA

El problema en que cuenta el Instituto de Genética radica en el proceso de administración de Historias Clínicas, actualmente el proceso es deficiente, los datos son vulnerables a perdidas y el acceso a la información se encuentra limitado. Las historias clínicas actualmente son llenadas por medios físicos y archivos Excel, lo que genera conflictos internos y confusiones a la hora de brindar un servicio la paciente.

1.2.3 FORMULACIÓN DEL PROBLEMA

Teniendo en cuenta lo descrito anteriormente surge la pregunta problemática:

¿Cómo facilitar el acceso a la información de historias clínicas de pacientes para la unidad de genética medica del instituto de genética?

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar un sistema web de administración de historias clínicas para la unidad de genética medica del Instituto de Genética que facilite el acceso a la información de los pacientes.

1.3.2 OBJETIVOS ESPECÍFICOS

- Proporcionar acceso a la información del paciente de forma eficiente.
- Crear una base de datos relacional centralizada según los requerimientos.
- Desarrollar e implementar de manera integral los módulos necesarios para el sistema de gestión de historias clínicas, incluyendo el registro de usuarios, pacientes, historias clínicas y generación de reportes, con el fin de garantizar un manejo eficiente y seguro de la información clínica de los pacientes y una gestión óptima del personal y de recursos de la unidad de genética médica.
- Definir la autenticación de usuarios, asignando roles de acceso, según los requerimientos del usuario dentro del sistema.
- Desarrollar interfaces de usuario intuitivas y amigables que permitan la visualización eficiente y fácil uso de los datos de historias clínicas por parte del personal de la unidad de genética médica, mejorando la accesibilidad y eficiencia en la gestión de la información.

1.4 JUSTIFICACIÓN

1.4.1 JUSTIFICACIÓN SOCIAL

Automatizar los procesos de registro, actualización y búsqueda de datos con un sistema de administración de historias clínicas para el instituto de genética, facilitará el acceso a la información del paciente, reduciendo del tiempo que lleva realizar dichos procesos y brindando una mejor atención mediante un manejo eficiente de la información. El personal tendrá una mejor información del paciente, evitando conflictos por datos anexos de cada unidad al historial clínico de un paciente, simplificando el trabajo del personal.

1.4.2 JUSTIFICACIÓN ECONÓMICA

El desarrollo del Sistema de Administración de historias clínicas es económicamente justificable, ya que se reducirá gastos de material (hojas de papel, folders, formularios para las historias clínicas, muebles de oficina, etc.) y ahorrar espacios en los ambientes por la cantidad de archivos almacenados de forma física. Asimismo, no se realizó ningún cobro al Instituto de Genética, obviando un gran monto económico que usualmente llega a costar sistemas afines.

Se utilizó tecnologías y frameworks de software libre (open source) para el desarrollo del sistema y no se necesitó la contratación de personal para administrar el sistema dado que el Instituto y la facultad tienen personal en el área de sistemas que colaborarán al Instituto de Genética en esta función.

1.4.3 JUSTIFICACIÓN TECNOLÓGICA

El proyecto es justificable tecnológicamente, ya que el instituto cuenta con recursos informáticos para el acceso, administración y mantenimiento del sistema, en sus instalaciones se tiene computadoras con las siguientes características:

- Procesador de 5ta. generación, con 8 a 16 GB de RAM.

- Sistema operativo Windows 10 de 64 bits.
- Disco duro HDD de 1Tb de almacenamiento.
- Red interna mediante conexión DHCP.

Además, también cuentan con impresoras y escáneres. En cuanto al acceso al internet, el instituto cuenta con el servicio que es proporcionado por la Facultad de Medicina, poseen un router de la marca CISCO, el cual provee internet a todo el instituto, los cables que actualmente están instalados usan una categoría y una norma antigua que generan baja velocidad de conexión y breves instantes de latencia. Pero actualmente la dirección del instituto a cargo de la Dr. Aguilar están en proceso de actualización del cableado de red del Instituto de Genética, mejorando la velocidad del internet.

1.5 ALCANCES Y LIMITES

1.5.1 ALCANCE TEMÁTICO

- El trabajo implica desarrollar un sistema web de administración de historias clínicas, en el cual tendrá almacenado las historias clínicas de los pacientes.
- Se creará una base de datos relacional que centralizará los datos, existen historias clínicas almacenadas en una base de datos en ACCESS que servirán para realizar pruebas del sistema.
- Los módulos que se implementarán son: módulo de registro de usuarios, módulo registro de pacientes, módulo de historia clínica, módulo consulta médica, módulo de reporte, módulos de interfaces y de autenticación.
- El sistema se desarrollará con la arquitectura modelo-vista-controlador, utilizando frameworks de JavaScript, Node para el backend, React para el frontend.
- Se harán reportes de cada historia clínica.

1.5.2 ALCANCE TEMPORAL

El Instituto de Genética cuenta con una base de datos en Access, el cual almacena información de gestiones pasadas, a su vez almacenan datos en archivos Excel y en documentos físicos. El sistema se desarrollará los meses de enero, febrero, marzo y abril de 2023, a partir del mes de mayo se hará la importación de la información para hacer pruebas en el sistema.

1.5.3 ALCANCE GEOGRÁFICO

El sistema de administración de historias clínicas, gestionará información que se genere en el Instituto de Genética dependiente de la Facultad de Medicina de la Universidad Mayor de San Andrés, ubicada en la ciudad de La Paz Bolivia.

El proyecto no considera información, estudios y requerimientos que se realicen fuera del mismo.

1.5.4 LÍMITES

Los límites del presente proyecto se indican a continuación:

- El proyecto realizará el análisis, diseño y creación de módulos según los requerimientos de la institución.
- No se tomará en cuenta actividades económicas y estadísticas, ya que solo se toma en cuenta la información del paciente y procesos relacionados a la historia clínica.
- Solo el personal autorizado podrá acceder al sistema y no así los pacientes.
- El sistema será dependiente de la unidad de Genética Medica, y no así para las demás unidades.

1.6 METODOLOGÍA

1.6.1 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN

Las técnicas que se utilizaron fueron entrevistas, análisis documental, observación y grupos de discusión que permitieron la recolección de información para obtener una comprensión completa de las necesidades y requerimientos de los usuarios.

1.6.2 METODOLOGÍA DE DESARROLLO

1.6.2.1 Metodología Agile Unified Process (AUP).

La metodología a emplear en este proyecto es la de AUP, el Proceso Unificado Ágil (AUP, del inglés Agile Unified Process) es una versión simplificada del Rational Unified Process (RUP) que describe un enfoque simple y fácil de entender para desarrollar software de aplicaciones comerciales utilizando técnicas y conceptos ágiles y, al mismo tiempo, mantenerse fiel a RUP (The Agile Unified Process (AUP), Ambler, 2022).

La metodología AUP cuenta con cuatro fases que transcurren de manera consecutiva y son las siguientes:

- Concepción (Inception)
- Elaboración (Elaboration)
- Construcción (Construction)
- Transición (Transition)

La metodología cuenta con disciplinas que son ejecutadas por los miembros del equipo de desarrollo, mediante actividades que ayudan a alcanzar el producto según las necesidades del cliente. Las disciplinas son las siguientes:

- Modelo (Model)
- Aplicación (Implementation)
- Prueba (Test)

- Despliegue (Deployment)
- Gestión de configuración (Configuration Management)
- Gestión de proyectos (Project Management)
- Entorno (Environment)

1.6.2.2 Lenguaje De Modelado Unificado (UML)

El lenguaje unificado de modelado (UML) es el lenguaje gráfico destinado al modelado de sistemas de software y procesos orientado a objetos.

Los principales objetivos de UML es visualizar, especificar, construir y documentar los procesos de desarrollo y creación de un proyecto. Y es importante tener en cuenta que UML es un lenguaje de modelado estándar y no así un proceso o método de desarrollo de software, es independiente por lo que puede aplicarse en diferentes procesos.

Contiene elementos gráficos (nodos) conectados con aristas (arcos) que representan elementos UML del sistema diseñado. También pueden contener otro tipo de documentación de forma escrita, para facilitar el proceso de documentación.

UML no impide mezclar diferentes tipos de diagramas, por ejemplo, para combinar elementos estructurales y de comportamiento para mostrar un proceso de un caso de uso. Por lo tanto, el uso de estas puede generar una documentación más amplia.

Así mismo, algunas herramientas UML limitan algunos elementos gráficos disponibles que pueden utilizarse cuando se trabaja con un tipo específico de diagrama.

1.6.2.3 Marco De Trabajo Kanban

Kanban es el nombre que se le da al método de gestión de flujo de trabajo, donde se trabaja sobre los roles y áreas de trabajo, promoviendo el liderazgo y compromiso del equipo, facilitando la visibilidad de los procesos del proyecto.

Los procesos son visualizados mediante tableros “Kanban”, dividiendo sus etapas en columnas donde se hace uso de etiquetas con tareas que son demandadas por el proyecto. Permitiendo administrar el flujo de trabajo y el límite de trabajo, según lo requiera el equipo para solucionar o completar situaciones o tareas que aparezcan a lo largo del flujo.

El tablero Kanban básico consta de 3 a 4 columnas:

- Por hacer (To Do)
- En curso (In Progress/doing):
 - Trabajando (Working)
- En espera (Waiting)
- Finalizado (Done)



Figura 1.6.1 Tablero Kanban Básico

Fuente: Elaboración Propia.

1.6.2.4 Arquitectura Modelo Vista Controlador (MVC)

La arquitectura de software Modelo-Vista-Controlador (MVC), es un patrón que separa los datos de una aplicación, facilitando la administración y cambios del desarrollo del software.

La estructura MVC ("Model-View-Controller") es un paradigma utilizado en el desarrollo de diversos softwares, a través de este patrón se logra una división de las diferentes partes que conforman una aplicación, permitiendo la actualización y mantenimiento del software de una forma sencilla y en un reducido espacio de tiempo (González, 2012).

Actualmente existen varios frameworks basados en el patrón MVC, permitiendo separar componentes de la aplicación y reutilizar porciones de código o elementos del proyecto, agilizando el trabajo de los desarrolladores y diseñadores.

Patrones de Arquitectura MVC

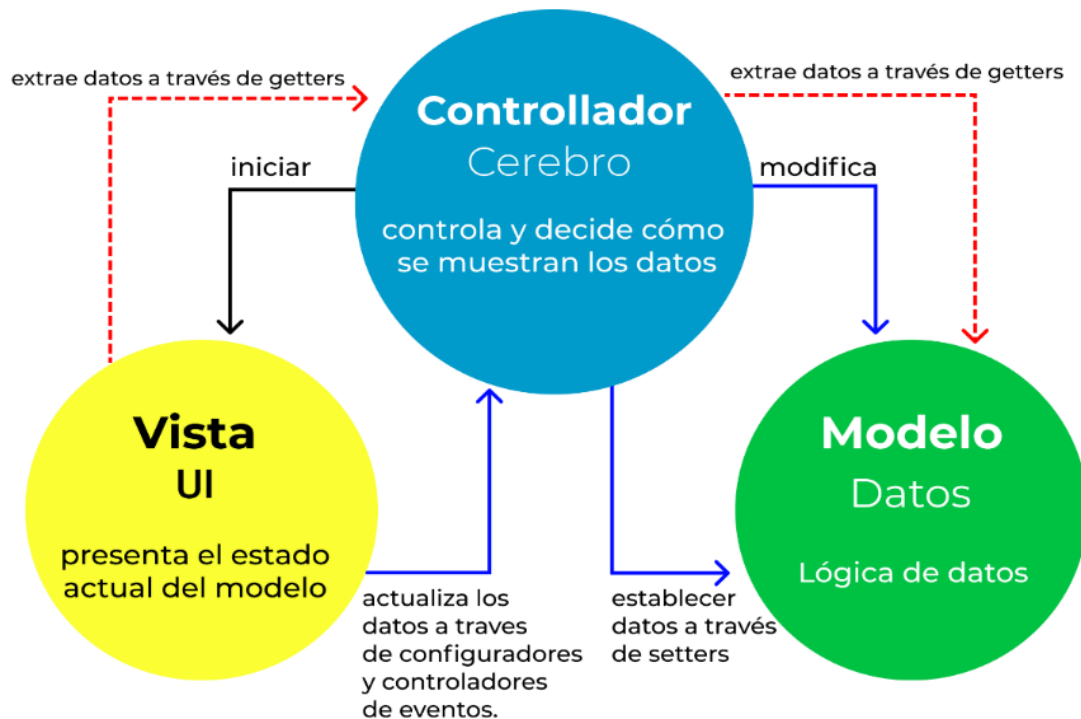


Figura 1.6.2 Gráfica de patrones de arquitectura MVC.

Fuente: FreeCodeCamp, Hernandez (2021)

CAPITULO II MARCO TEÓRICO

2.1 MARCO INSTITUCIONAL

El Instituto de Genética dependiente de la Facultad de Medicina de la Universidad Mayor de San Andrés ofrece sus servicios en el área de genética médica, genética toxicológica y citogenética. El equipo de profesionales que cuentan valora y realizan consultas a personas con problemas genéticos además de personas que necesiten exámenes en el área genética. El instituto en cuanto a historias clínicas trabajo con un formato dado por el ministerio de salud desde 1993 al 2003. A partir de esa fecha se fue trabajando en un formato que consideraba los campos requeridos para la filiación y de consulta (ver figura 1.7.1).

El flujo del sistema organizativo que llevan actualmente para atender a un paciente empieza por el registro de datos de pacientes que deseen una cita, el personal administrativo realiza la filiación de los pacientes en un formulario físico que a lo largo de los años se fue modificando, afectando el proceso de registro, ya que los campos dentro del formulario se alteraban según el director del instituto lo veía conveniente (ver figura.1.7.1), en el mencionado formulario se almacena información del paciente.

Posteriormente, al paciente se le indica la fecha y hora para su consulta. El día de la consulta el medico a cargo realiza la valoración y el diagnóstico del paciente, llenando los campos del formulario del historial clínico, si el doctor en la consulta ve que el paciente necesita abrir una historia clínica adicional, lo hace (ver figuras 1.7.2 y 1.7.3).

Dichas historias adicionales son independientes a la general, son historias que se abren cuando el paciente sufre de otras enfermedades y problemas de salud. El doctor al terminar su consulta genera estas historias que son guardadas en formato físico y en archivos Excel. Las historias clínicas son usadas por otras unidades que dan el servicio de exámenes, añadiendo información a la historia clínica.

Todo este proceso actual es muy ineficiente ya que la pérdida de información, el tiempo de espera para que se obtenga la historia clínica de una unidad a otra y la cantidad de archivos afecta en la obtención de información y a la atención de los pacientes.

La dificultad para acceder y administrar la información del historial clínico de cada paciente en la unidad de genética medica del instituto de genética, genera un manejo inadecuado por la cantidad de información, limitando el acceso y registro de la información de los pacientes.

El instituto en el año de 2012, consiguieron desarrollar una base de datos en ACCESS (ver figura) que almacenaba los datos e información de historias clínicas de los pacientes. Pero el acceso a la base de datos era ineficiente, no se contaba con una interfaz para visualizar los datos, el registro era complicado y algunos doctores no se adaptaban a la tecnología, de modo que la base de datos dejó de ser utilizada y se volvió al sistema antiguo. Utilizando formularios impresos, implementando archivos Word y Excel para un registro digital.

HISTORIA CLÍNICA: PACIENTES AMBULATORIOS DE GENÉTICA MÉDICA**I. FILIACIÓN:****R.G.:**Nombre Luz María Martí F.Nac. / / Sexo: Raza F. de Consulta / / Procedencia Residencia Dirección Tel. **II. ANTECEDENTES FAMILIARES:**Padre F.Nac. / / R. Prof. Madre F.Nac. / / R. Prof. Edad materna cuando nació el propósito del padre Edad de la abuela cuando nació la madre **III. MOTIVO DE CONSULTA:****IV. ANTECEDENTES PERSONALES:**1. Gestación - a término ☐ pre-término ☐ post-término ☐ Obs. 2. Patologías en el embarazo - Fiebre ☐ Enf.Infec. ☐ Diabetes ☐ Epilepsia ☐
otras enfermedades crónicas 3. Factores físicos durante el embarazo - Rayos X: Si ☐ No ☐ Ecografía: Si ☐ No ☐
Lugar donde se realizó N° de veces

4. Factores químicos durante la gestación

Fármacos anticonvulsivantes: otros Anticonceptivos orales Gestágenos para evitar AB. Exposición Profesional Enolismo 5. Gesta Para N° Nativos Malformados N° Natimortos N° AB. Exp. Ind. de uso de anticonceptivos: Si ☐ No ☐ Tipo: Período entre 1ª y 2ª gestación: 2ª y 3ª: 3ª y 4ª: Período de uso 6. Parto eutócico ☐ Cesárea ☐ Forceps ☐ ¿Por que? 7. Datos del nacimiento: Peso: Talla PC Apgar Llanto - Inmediato ☐ Tardío ☐ No sabe ☐ Cianosis - Si ☐ No ☐ No sabe ☐Oxígeno terapia - Si ☐ No ☐ No sabe ☐ Incubadora - Si ☐ No ☐ No sabe ☐Ictericia Neonatal - Si ☐ No ☐ No sabe ☐ Fototerapia - Si ☐ No ☐Exsanguíneo - Si ☐ No ☐ Fiebre ☐ Convulsiones ☐Hemorragias - Si ☐ No ☐ No sabe ☐ Inicio: Duración: Otras alteraciones: Criptorquidea ☐ Cardiopatía Congénita ☐ FLAP ☐ Atresia anal ☐Defectos del tubo Neural ☐ Obs. **Figura 2.1.1 Historia Clínica.**

Fuente: Instituto de Genética

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

I. FILIACIÓN:

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

II. MOTIVO DE CONSULTA:

III. ANTECEDENTES FAMILIARES:

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

IV. ANTECEDENTES PRENATALES

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

Observaciones: _____


Ecografías prenatales: Si ☐ No ☐ Semana (mes, trimestre) de realización: _____

Translucencia nucal: _____ Hueso nasal: _____ Otros hallazgos: _____

Figura 2.1.2 Historia Clínica para paciente con Síndrome de Down.

Fuente: Instituto de Genética

Maltrato fetal 6



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

HISTORIA CLÍNICA REPRODUCCIÓN Y MATERNOFETAL (HC 3/IG)

I. FILIACIÓN: ☒
 Nombre: _____ Fecha de Nacimiento: ____/____/____ Edad: ____
 Sexo: M ☐ F ☐ Dirección: _____ Teléfono: _____ Fecha: ____/____/____
 Ocupación: _____ REGISTRO IG: _____

II. MOTIVO DE CONSULTA: diagn

III. GENEALOGÍA: ☒

IV. ANTECEDENTES PERSONALES PATOLÓGICOS (APP): _____

V. ANTECEDENTES DE LA PAREJA:
 Nombre: _____ Fecha de Nacimiento: ____/____/____ Edad: ____
 Sexo: M ☐ F ☐ Teléfono: _____ Ocupación: _____
 APP: _____

VI. ANTECEDENTES GINECO-OBSTÉTRICOS. G: ____ A: ____ P: ____ C: ____
 Menarca: ____ Ritmo: ____ FUM: ____
 Uso de Anticonceptivos: _____
 Descripción de abortos u óbitos (edad gestacional, causa, sexo del producto, orden gestacional, etc.)

1

Figura 2.1.3 Historia Clínica para pacientes con Infertilidad.

Fuente: Instituto de Genética

Personalizado

Grupo personalizado 1

Objetos no asignados

antece_familiar

antece_personal

composicion_familia

craneo

down

examen_fisico

examen_neurológico

fotos

fotos1

genitales_externos

Lista de direcciones

localidad

localidad1

maxilar_boca_cuello

miembros

orejas_ojos_nariz

persona

Raza

Registros de servicio

resumen

sexo

tejido_musculatura_piel

tiempos

torax_columna_abdomen

C_D_sexof

C_D_sexom

CDown

Cedades_0A10

Cedades_10A20

Cedades_20A30

Cedades_30A40

Cedades_40A50

Consulta1

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

DATOS PERSONALES

Registro

4127 B1 C1

Nombre

Gladis

Apellido Paterno

MONTANO

Apellido Materno

López

Sexo

Masculino

Raza

Blanca

Fecha Nacimiento

24-dic-62

Edad

43

Año (s)

Días

Meses

Años

Procedencia

Oruro

Residencia

La Paz

Dirección

Av. 6 de agosto #2549 Edif El Carmán 4B

Teléfono

2434022

Enfermedad de Down

☒

Agregar nuevo registro

Eliminar registro

Primer

Anterior

Siguiente

Último

Buscar

Registro: 1 de 648

Sin filtrar

Buscar

Foto

Figura 2.1.4 *Historia Clínica en ACCESS.*

Fuente: Instituto de Genética

2.2 METODOLOGIA

2.2.1 Metodologías Ágiles del Desarrollo.

Las metodologías ágiles de desarrollo han surgido como una alternativa al enfoque tradicional de gestión de proyectos de software. A diferencia del modelo de cascada, donde se sigue un enfoque secuencial y se planifica todo el proyecto antes de comenzar el desarrollo, las metodologías ágiles promueven un enfoque iterativo e incremental que se adapta a los cambios en los requisitos del proyecto. En este marco teórico se discutirán los conceptos clave de las metodologías ágiles de desarrollo, así como las ventajas y desventajas de su implementación.

2.2.1.1 Conceptos clave de metodologías ágiles de desarrollo.

La metodología ágil se centra en el trabajo en equipo, la comunicación constante y la entrega temprana de software. El Manifiesto Ágil es una declaración de valores y principios fundamentales para el desarrollo ágil de software. El manifiesto establece que los desarrolladores deben priorizar "individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación exhaustiva, colaboración con el cliente sobre negociación contractual y responder al cambio sobre seguir un plan" (Beck et al., 2001).

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

La comunicación constante entre los miembros del equipo es crucial para el éxito de las metodologías ágiles. Las reuniones diarias de pie, conocidas como reuniones de scrum, son una forma efectiva de mantener a todos informados sobre el progreso del proyecto y cualquier

problema que surja. Además, las metodologías ágiles fomentan la colaboración entre el equipo de desarrollo y el cliente a lo largo del proyecto.

2.2.1.2 Ventajas y desventajas de la implementación de metodologías ágiles.

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

Sin embargo, la implementación de las metodologías ágiles también puede tener desventajas. Una de las principales preocupaciones es la falta de documentación exhaustiva. A diferencia del enfoque tradicional de cascada, las metodologías ágiles no requieren una documentación detallada de cada fase del proyecto. Esto puede ser un problema si los miembros del equipo cambian o si se requiere una auditoría posterior del proyecto.

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

2.2.2 *Proceso Unificado Ágil (AUP).*

El Proceso Unificado Ágil (AUP, por sus siglas en inglés) es una metodología de desarrollo de software que combina elementos del Proceso Unificado de Rational (RUP) y de las metodologías ágiles. El AUP es una metodología iterativa e incremental que se enfoca en la entrega temprana de software funcional a través de iteraciones cortas. En este marco teórico se discutirán los conceptos clave del AUP, así como las ventajas y desventajas de su implementación.

2.2.2.1 Ventajas y desventajas de la implementación del Proceso Unificado Ágil

(AUP).

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

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

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

2.2.2.2 ¿Por qué usar la Metodología de Desarrollo AUP?

El Proceso Unificado Ágil (AUP) es un marco de trabajo de desarrollo de software que se enfoca en la entrega rápida y frecuente de software funcional. El AUP combina los principios del desarrollo ágil de software con la estructura disciplinada y organizada del Proceso Unificado de Rational (RUP). El objetivo del AUP es proporcionar un enfoque flexible y adaptable para el desarrollo de software, que permita a los equipos de desarrollo ajustar el proceso según sus necesidades y contextos específicos.

El proceso AUP establece un modelo más simple que el que aparece en RUP, reuniendo en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño.

Esto permite una mayor flexibilidad y adaptabilidad en el proceso de desarrollo, permitiendo a los equipos centrarse en las necesidades del negocio y en la entrega rápida de software funcional.

Las disciplinas restantes del AUP (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las del RUP. Estas disciplinas son esenciales para garantizar la calidad, la estabilidad y la escalabilidad del software desarrollado, así como para asegurar que el software sea desplegado y mantenido correctamente.

Los equipos de AUP suelen ofrecer versiones de desarrollo al final de cada iteración en preproducción área(s). Una versión de desarrollo de una aplicación es algo que podría ser liberado en la producción si se ponen a través de su pre-producción de garantía de calidad (QA), las pruebas y los procesos de despliegue. La primera producción de liberación a menudo toma más tiempo para entregar versiones posteriores.

Según Cockburn y Highsmith (2001), los stakeholders son aquellos que están interesados en el desarrollo del sistema, y que tienen algún tipo de influencia en él. El proceso AUP establece que el equipo de desarrollo debe estar en constante comunicación con los stakeholders, a fin de asegurarse de que el sistema esté cumpliendo con sus expectativas y necesidades. Además, se enfatiza en la importancia de mantener un diálogo continuo con los stakeholders a lo largo de todo el proceso de desarrollo, para poder hacer ajustes en el sistema de manera oportuna.

El proceso AUP también promueve la implementación de pruebas continuas durante todo el proceso de desarrollo, para asegurarse de que el sistema esté funcionando correctamente en todo momento. De esta manera, se pueden detectar errores y problemas de manera temprana, y hacer ajustes en el sistema de manera oportuna.

Existen diversas razones por las cuales una organización podría decidir utilizar la metodología de desarrollo AUP (Proceso Unificado Ágil), tomando en cuenta lo anteriormente descrito se tienen las siguientes razones:

- **Flexibilidad y adaptabilidad:** La metodología AUP es iterativa e incremental, lo que significa que el desarrollo del software se realiza en pequeñas etapas, permitiendo al equipo de desarrollo adaptarse a los cambios y requerimientos del cliente de manera más fácil y rápida.
- **Mejora de la calidad:** A través de la implementación de pruebas continuas durante todo el proceso de desarrollo, se pueden detectar y corregir errores de manera temprana, lo que se traduce en una mejora en la calidad del software.
- **Mayor satisfacción del cliente:** Al enfocarse en la entrega de valor al cliente en cada iteración, la metodología AUP permite una mayor satisfacción del cliente, ya que se asegura de que sus necesidades y expectativas sean cumplidas de manera oportuna.
- **Colaboración y comunicación constante:** La metodología AUP enfatiza la colaboración y comunicación constante entre el equipo de desarrollo y los stakeholders, lo que permite asegurarse de que el sistema esté cumpliendo con las necesidades y expectativas de todos los involucrados.
- **Gestión de riesgos:** La metodología AUP incluye un proceso de gestión de riesgos, lo que permite al equipo de desarrollo anticipar y mitigar posibles riesgos y problemas en el proceso de desarrollo.

En resumen, el proceso AUP es una metodología ágil de desarrollo de software que se enfoca en la simplicidad, la colaboración y la implementación de pruebas continuas durante todo el proceso de desarrollo. Y a través de sus fases iterativas y enfocadas en la entrega de valor al cliente, permite un desarrollo más flexible y adaptable a los cambios, y una mayor satisfacción de los stakeholders. Es una opción atractiva para aquellas organizaciones que buscan una metodología flexible, adaptable y centrada en el valor al cliente, que les permita mejorar la calidad del software y la satisfacción del cliente, y que enfatice la colaboración y la comunicación constante entre todos los involucrados en el proceso de desarrollo.

2.2.2.3 Principios De La Metodología AUP.

AUP se basa en los siguientes principios, por eso es considerada ágil:

- a) La colaboración del equipo es fundamental en todas las etapas del proyecto (Jacobson, Booch y Rumbaugh, 1999).
- b) El desarrollo de software es un esfuerzo iterativo e incremental, en el que se realizan pruebas continuas y se recibe retroalimentación de los usuarios (Kruchten, 2003).
- c) Se deben enfocar los esfuerzos en los requisitos esenciales del proyecto, evitando los requisitos "deseables" que pueden no ser necesarios para el éxito del proyecto (Cockburn, 2001).
- d) El trabajo se debe organizar en fases definidas y claramente definidas, con entregables específicos al final de cada fase (Booch, Rumbaugh y Jacobson, 2005).
- e) La arquitectura del sistema debe ser considerada cuidadosamente al inicio del proyecto, ya que puede tener un impacto significativo en la viabilidad del proyecto (Larman, 2003).

Estos conceptos se resumen en:

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

2.2.2.4 Ciclo De Vida Del Proceso Unificado Ágil (AUP).

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del sistema.

2.2.2.4.1 Fases del Ciclo de Vida del Proceso Unificado Ágil (AUP).

El AUP consta de cuatro fases principales:

- **Inicio:** En esta fase se establecen los objetivos del proyecto y se identifican los principales interesados. También se realiza un análisis preliminar de riesgos y se define la visión y el alcance del proyecto.
- **Elaboración:** En esta fase se lleva a cabo una planificación detallada y se crea un modelo arquitectónico preliminar del sistema. También se identifican los requisitos del sistema y se realiza una evaluación detallada de riesgos.
- **Construcción:** En esta fase se desarrolla y prueba el software. Se realizan iteraciones cortas y se lleva a cabo una evaluación continua del progreso y de los riesgos del proyecto.
- **Transición:** En esta fase se realiza la implementación del software y se entregan los productos finales al cliente. También se lleva a cabo una evaluación del proyecto y se identifican las lecciones aprendidas para mejorar futuros proyectos.

Cada fase se subdivide en iteraciones. En cada iteración se desarrolla en secuencia un conjunto de disciplinas o flujos de trabajos.

Fases e Hitos de un Proyecto

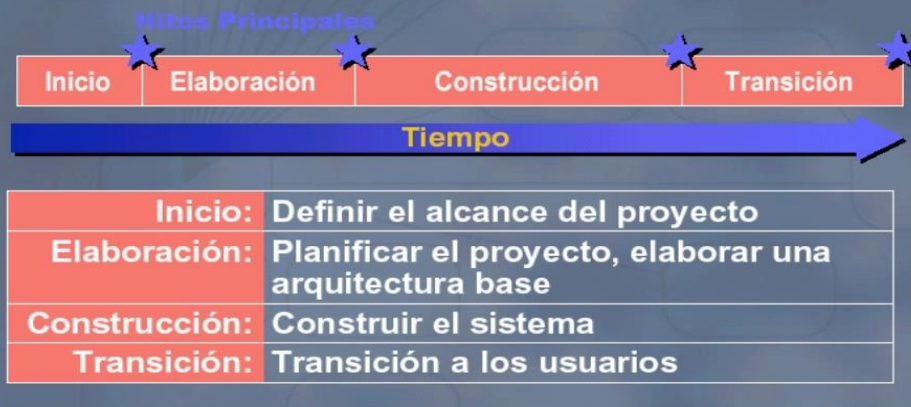


Figura 2.2.1 Fases e hitos de un proyecto

Fuente: JACOBSON (2000)

Disciplinas Básicas

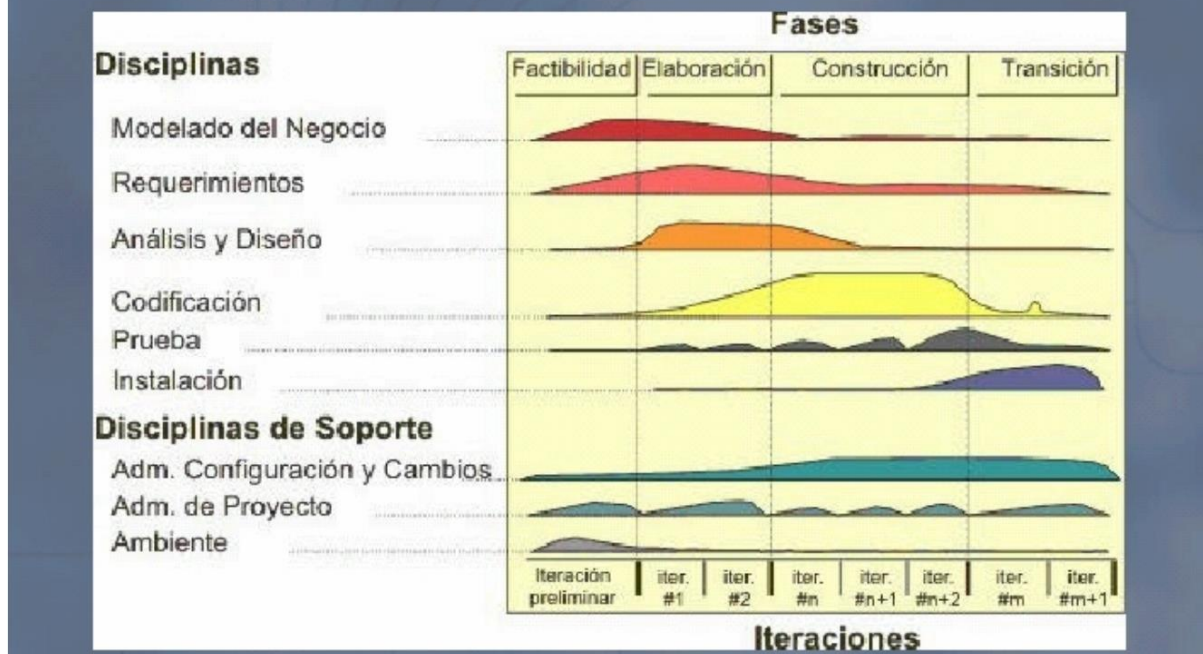


Figura 2.2.2 Actividades de la metodología AUP

Fuente: JACOBSON (2000)

AUP se establecen cuatro fases de ciclo de desarrollo que transcurren de manera consecutiva y que acaban con partes funcionales del proyecto:

- **Fase de Inicio:** La fase de Inicio del AUP se enfoca en establecer los objetivos del proyecto, identificar los principales interesados y definir la visión y el alcance del proyecto. Según Pressman y Maxim (2015), en esta fase se realizan actividades tales como la identificación de los requisitos del sistema, la evaluación de riesgos, y la definición de los criterios de éxito del proyecto.
- **Fase de Elaboración:** En la fase de Elaboración del AUP se lleva a cabo una planificación detallada y se crea un modelo arquitectónico preliminar del sistema. También se identifican los requisitos del sistema y se realiza una evaluación detallada de riesgos. De acuerdo con Cockburn (2008), en esta fase se definen los casos de uso, se establecen los objetivos de calidad y se planifican las iteraciones.
- **Fase de Construcción:** La fase de Construcción del AUP se enfoca en el desarrollo y prueba del software. Se llevan a cabo iteraciones cortas y se realiza una evaluación continua del progreso y de los riesgos del proyecto. Según Boehm y Turner (2005), en esta fase se lleva a cabo la implementación de los casos de uso, se realiza la integración y prueba del software, y se realizan actividades de aseguramiento de la calidad.
- **Fase de Transición:** En la fase de Transición del AUP se realiza la implementación del software y se entregan los productos finales al cliente. También se lleva a cabo una evaluación del proyecto y se identifican las lecciones aprendidas para mejorar futuros proyectos. De acuerdo con Larman (2004), en esta fase se llevan a cabo actividades tales como la formación del usuario, la aceptación del sistema por parte del cliente, y la identificación de las lecciones aprendidas.

2.2.3 Lenguaje Unificado De Modelado (UML).

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group) (OMG, 2017). Es un

lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar. UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas (Rumbaugh, Jacobson, & Booch, 2004; Fowler, 2004).

2.2.4 UML-Based Web Engineering.

UML-Based Web Engineering es una metodología de desarrollo de software para aplicaciones web basada en el uso de UML como lenguaje de modelado y diseño. Esta metodología se enfoca en el desarrollo de sistemas web a través de la creación de modelos visuales y la documentación detallada de los requisitos, funcionalidades y arquitectura del sistema.

El uso de UML en la ingeniería web ha sido ampliamente adoptado debido a su capacidad para modelar diferentes aspectos de las aplicaciones web, incluyendo la navegación, la interacción con el usuario, la lógica de negocio, la persistencia de datos, entre otros. UML

proporciona una notación visual para describir estos aspectos de una manera clara y comprensible para todas las partes involucradas en el proceso de desarrollo.

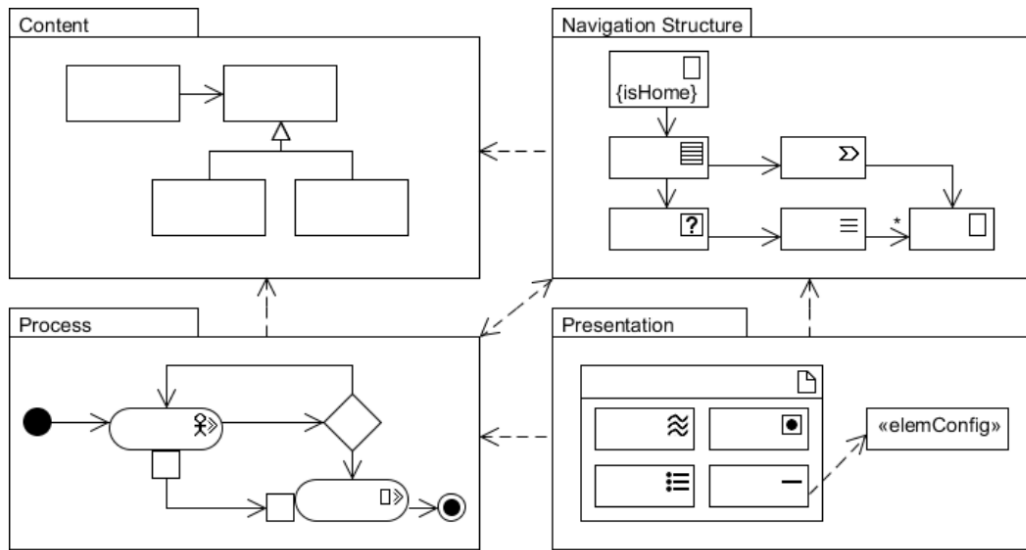


Figura 2.2.3 Visión general de los modelos UML.

Fuente: UWE – UML-based Web Engineering (2016)

2.2.4.1 Ventajas de UML-Based Web Engineering.

El uso de UML-Based Web Engineering tiene varias ventajas, incluyendo:

- Proporciona una notación gráfica fácil de entender para describir la estructura, el comportamiento y las relaciones entre los componentes de una aplicación web.
- Facilita la comunicación entre los miembros del equipo de desarrollo y con los clientes, ya que todos pueden entender los diagramas UML.
- Permite la detección temprana de problemas y errores en el diseño, lo que reduce los costos y el tiempo de desarrollo.
- Proporciona una documentación detallada del sistema que puede ser utilizada para futuras actualizaciones o mejoras.
- Facilita la reutilización de componentes y módulos de software.

2.2.4.2 Fases de UML-Based Web Engineering.

UML-Based Web Engineering consta de varias fases que deben ser seguidas para desarrollar una aplicación web exitosa. Estas fases son:

- **Fase de Análisis:** En esta fase se definen los requisitos del sistema web y se realiza un análisis detallado de los mismos. Los requisitos son documentados utilizando diferentes tipos de diagramas UML, como diagramas de casos de uso y diagramas de secuencia. También se identifican los actores del sistema y se define la interacción entre ellos.
- **Fase de Diseño:** En esta fase se utiliza la información obtenida en la fase de análisis para diseñar la arquitectura del sistema web. Se crean diferentes diagramas UML para describir la estructura y el comportamiento de la aplicación web, como diagramas de clases, diagramas de componentes y diagramas de secuencia. También se definen los patrones de diseño que se utilizarán en el desarrollo.
- **Fase de Implementación:** En esta fase se desarrolla el sistema web utilizando la arquitectura y el diseño definidos en la fase anterior. Se crean los componentes y se integran en el sistema. En esta fase se utilizan diferentes lenguajes de programación y herramientas de desarrollo para implementar el sistema.
- **Fase de Pruebas:** En esta fase se realizan diferentes pruebas para asegurarse de que el sistema web funciona correctamente y cumple con los requisitos definidos en la fase de análisis. Se realizan pruebas de unidad, integración y aceptación para garantizar que el sistema cumpla con los estándares de calidad.
- **Fase de Despliegue:** Durante esta fase, es importante tener en cuenta los requisitos no funcionales del sistema, como la seguridad y el rendimiento, para garantizar que el sistema se despliega de manera efectiva y cumple con los requisitos del usuario.

2.3 INFORMACIÓN

Según (Chiavenato, 2006) “La Información es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En

verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones”.

Conceptualizando lo descrito por Chiavenato, la información es un conjunto de datos organizados coherentemente que permiten adquirir conocimiento, por tanto, ser una herramienta que pueda relacionar individuos.

2.3.1 Características de la información.

La información es un recurso valioso en la era actual de la tecnología de la información. Las organizaciones y las personas utilizan la información para tomar decisiones, realizar operaciones y resolver problemas. Es esencial que la información sea precisa, confiable, oportuna y relevante para su uso efectivo. Las características de la información que son importantes para asegurar su calidad y utilidad en la toma de decisiones y el desempeño de las tareas son:

- **Precisión:** La precisión es la medida en que la información es correcta y se ajusta a la realidad. La información inexacta o incorrecta puede llevar a decisiones equivocadas y, en última instancia, a pérdidas financieras y de reputación. Por lo tanto, es esencial que la información sea precisa y que se verifique su precisión antes de su uso.
- **Confidencialidad:** La confidencialidad es la medida en que la información está protegida contra el acceso no autorizado. La información confidencial puede ser sensible y puede incluir información personal o comercial. Es importante que la información confidencial se proteja para evitar su divulgación no autorizada y el uso inapropiado.
- **Disponibilidad:** La disponibilidad se refiere a la medida en que la información está disponible cuando se necesita. La información no disponible puede llevar a retrasos y a la toma de decisiones equivocadas. Es importante que la información esté disponible cuando se necesite para su uso efectivo.
- **Integridad:** La integridad se refiere a la medida en que la información es completa y exacta. La información incompleta o inexacta puede llevar a decisiones equivocadas y a pérdidas

financieras y de reputación. Por lo tanto, es esencial que la información sea completa y exacta y que se verifique su integridad antes de su uso.

- **Relevancia:** La relevancia se refiere a la medida en que la información es pertinente y útil para el propósito para el que se utiliza. La información irrelevante puede distraer y confundir, y puede llevar a la toma de decisiones equivocadas. Es importante que la información sea relevante y que se utilice para su propósito previsto.

2.3.2 Información en el contexto Tecnológico.

La información es un concepto fundamental en la informática, y se refiere a los datos que se han procesado y organizado para su uso en una determinada tarea o propósito. La información puede tomar muchas formas, incluyendo texto, imágenes, audio y video. A continuación, se describen algunos de los conceptos clave relacionados con la información en el contexto de la informática.

- **Datos:** Los datos son elementos básicos e indivisibles que no tienen significado en sí mismos, pero que pueden ser procesados para obtener información. Según Turban et al. (2008), "los datos se pueden considerar como los bloques de construcción de la información" (p. 17).
- **Información:** La información es el resultado del procesamiento de datos, y se refiere a los datos que se han organizado y presentado de manera significativa para su uso en una tarea o propósito específico. Según Laudon y Laudon (2012), "la información es el resultado de procesar datos para revelar su significado" (p. 10).
- **Conocimiento:** El conocimiento se refiere a la comprensión y la interpretación de la información, y a su aplicación en una determinada tarea o contexto. Según Davenport y Prusak (1998), "el conocimiento es la capacidad de usar información para resolver problemas o tomar decisiones" (p. 5).
- **Calidad de la información:** La calidad de la información se refiere a la medida en que la información es precisa, completa, relevante y confiable. Según Wang y Strong (1996), "la

calidad de la información se puede definir en términos de su idoneidad para el uso previsto" (p. 8).

- **Seguridad de la información:** La seguridad de la información se refiere a la protección de la información contra el acceso no autorizado, la divulgación, la alteración o la destrucción. Según Pfleeger y Pfleeger (2003), "la seguridad de la información se puede considerar como la preservación de la confidencialidad, la integridad y la disponibilidad de la información" (p. 14).

2.4 TEORÍA GENERAL DE LOS SISTEMAS

La teoría general de los sistemas (TGS) es un método de investigación, una forma de pensar, que enfatiza el sistema total en vez de sistemas componentes, se esfuerza por optimizar la eficacia del sistema total en lugar de mejorar la eficacia de sistemas cerrados. Se basa principalmente en la visión de no ser reduccionista en su análisis ya que es un medio fundamental para solucionar problemas de cualquier tipo (ITSON | Enfoque de Sistemas | Inicio, s. f.).

- Las premisas básicas de la TGS, son:
- Los sistemas existen dentro de sistemas.
- Los sistemas son abiertos.
- Las funciones de un sistema dependen de su estructura.

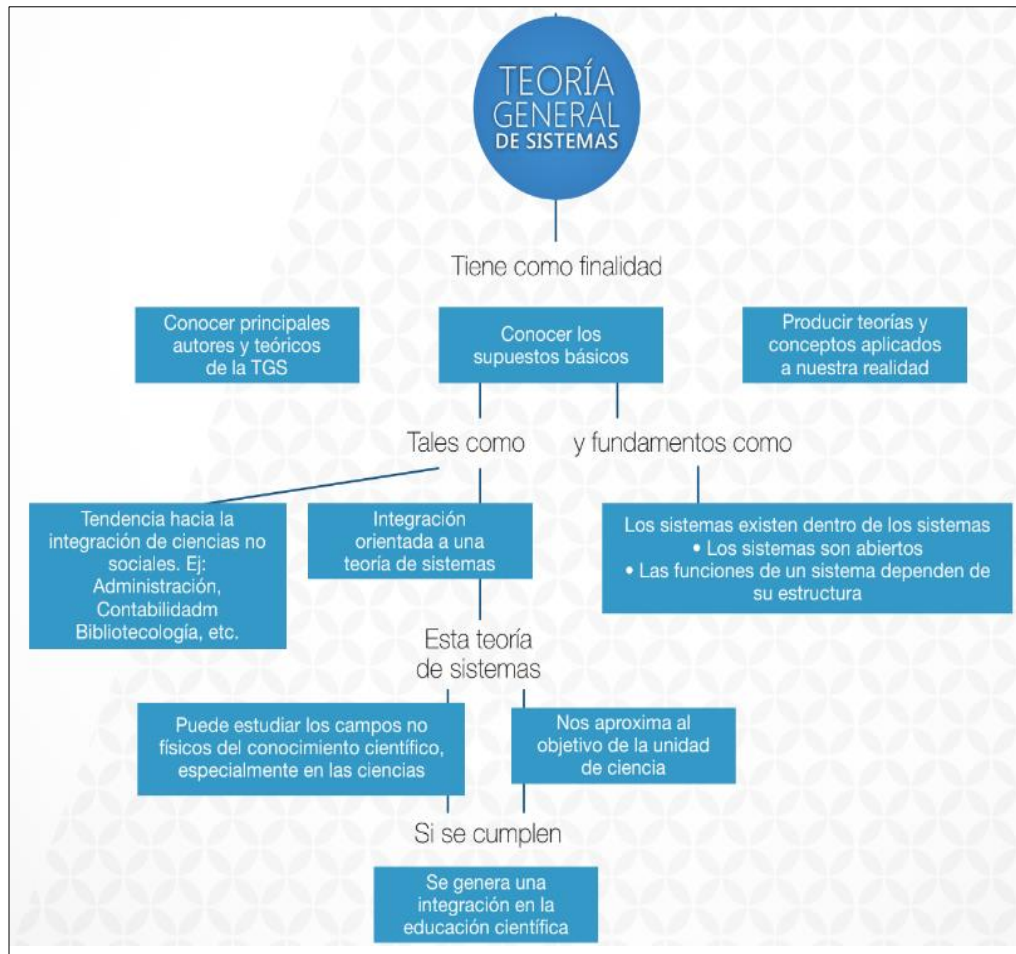


Figura 2.4.1 Mapa Conceptual Finalidad de la TGS.

Fuente: Gutiérrez Gómez et al., 2013

2.5 SISTEMA

Hall, define un sistema como un conjunto de objetos y sus relaciones, y las relaciones entre los objetos y sus atributos. Los objetos son simplemente las partes o componentes de un sistema y estas partes pueden poseer una variedad limitada. Los atributos son las propiedades de los objetos (Bertoglio, 2013. Como se citó en Hall, 1964).

Según (Gutiérrez Gómez et al., 2013) indica las siguientes definiciones:

- Es una reunión o conjunto de elementos interrelacionados con un objetivo común.
- Es una unión de partes o componentes conectados de una forma organizada.
- Un conjunto de elementos dinámicamente relacionados, formando una actividad para alcanzar un objetivo, operando sobre datos/energía/materia para proveer información/energía/materia.
- Sistema es un todo organizado y complejo; un conjunto o combinación de cosas o partes que forman un todo complejo o unitario.
- Es un conjunto de objetos unidos por alguna forma de interacción o interdependencia.
- Es un conjunto de elementos que ordenadamente interactúan entre sí, contribuyendo a lograr un objetivo.

Tomando en cuenta las definiciones citadas, podemos conceptualizar al sistema como un conjunto de elementos coherentemente relacionados entre sí, que contribuyen a un mismo objetivo

2.5.1.1 Características De Los Sistemas.

Según (Gutiérrez Gómez et al., 2013) se deducen dos conceptos troncales: propósito (u objetivo) y globalismo (o totalidad). Y, además de las siguientes características:

- Propósito u objetivo: todo sistema tiene uno o algunos propósitos.
- Globalismo o totalidad: un cambio en una de las unidades del sistema, con probabilidad producirá cambios en las otras. Hay una relación de causa/efecto. De estos cambios y ajustes, se derivan dos fenómenos: entropía y homeostasia.
- Entropía: es la tendencia de los sistemas a desgastarse, a desintegrarse, para el relajamiento de los estándares y un aumento de la aleatoriedad.
- Homeostasia: es el equilibrio dinámico entre las partes del sistema. Los sistemas tienen una tendencia a adaptarse con el fin de alcanzar un equilibrio interno frente a los cambios externos del entorno.

2.5.2 Principios de los Sistemas.

Los sistemas son estructuras complejas que se componen de elementos interdependientes, y cuyo comportamiento y funcionamiento son el resultado de la interacción de dichos elementos. Los principios de los sistemas son los conceptos fundamentales que describen el comportamiento de los sistemas y que permiten su análisis y diseño. A continuación, se describen algunos de los principales principios de los sistemas.

- **Holismo:** El holismo es un principio que establece que un sistema debe ser considerado como un todo, y no como la suma de sus partes. Según este principio, el comportamiento del sistema no puede ser comprendido a partir del estudio de cada uno de sus elementos de manera aislada, sino que debe ser analizado en su totalidad.
- **Interdependencia:** La interdependencia es un principio que establece que los elementos de un sistema están interconectados y se influyen mutuamente. La interdependencia es fundamental para el funcionamiento de un sistema, ya que cada elemento depende de los demás para poder cumplir su función.
- **Retroalimentación:** La retroalimentación es un principio que establece que la salida de un sistema se retroalimenta a su entrada, lo que permite que el sistema se adapte y modifique su comportamiento en función de las condiciones del entorno. La retroalimentación es esencial para el control y la regulación de los sistemas.
- **Entropía:** La entropía es un principio que establece que los sistemas tienden a la desorganización y la pérdida de energía a lo largo del tiempo, a menos que se realice un trabajo para mantenerlos ordenados y con energía. La entropía es una medida de la cantidad de desorden en un sistema y es importante para entender la tendencia natural de los sistemas a la degradación.
- **Homeostasis:** La homeostasis es un principio que establece que los sistemas tienen una tendencia a mantener un estado estable y constante, a pesar de las fluctuaciones en el entorno. La homeostasis es fundamental para la supervivencia de los sistemas, ya que les permite mantener su funcionamiento en un entorno cambiante.

- **Equifinalidad:** La equifinalidad es un principio que establece que un mismo resultado puede ser alcanzado a través de diferentes caminos o procesos. Este principio es importante porque muestra que no existe una única manera de diseñar o implementar un sistema, y que diferentes soluciones pueden ser igualmente válidas.
- **Jerarquía:** La jerarquía es un principio que establece que los sistemas están organizados en niveles o niveles de abstracción. La jerarquía es importante para entender la estructura y el funcionamiento de los sistemas, ya que permite descomponerlos en sus componentes más básicos y comprender su complejidad.

2.5.3 Elementos de los Sistemas.

Los elementos de un sistema son los componentes que conforman el sistema y que, al interactuar, permiten que éste cumpla su función. Según Senge (1990), los elementos principales de un sistema son los siguientes:

- **Objetivo o propósito:** es el fin que se busca alcanzar con el sistema. Es lo que da sentido a su existencia y orienta todas las acciones hacia ese fin.
- **Entradas:** son los elementos que ingresan al sistema y que son procesados para cumplir con el objetivo. Pueden ser datos, materiales, energía, entre otros.
- **Procesamiento:** es el conjunto de actividades y operaciones que se realizan en el sistema para transformar las entradas en salidas. Incluye el uso de recursos y la aplicación de reglas y procedimientos.
- **Salidas:** son los resultados o productos del procesamiento que se obtienen al cumplir con el objetivo del sistema. Pueden ser información, productos, servicios, entre otros.
- **Retroalimentación:** es el mecanismo mediante el cual el sistema obtiene información sobre su desempeño y la utiliza para mejorar su funcionamiento. La retroalimentación permite detectar errores y desviaciones, y corregirlos a tiempo.
- **Entorno:** es el contexto en el que se desenvuelve el sistema y que influye en su funcionamiento. El entorno puede ser físico, social, político, entre otros.

Es importante destacar que estos elementos no son estáticos, sino que están en constante interacción y cambio. El procesamiento de las entradas puede generar nuevos datos que retroalimenten al sistema y que modifiquen su objetivo o su entorno.

2.6 SISTEMA DE INFORMACIÓN

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Está compuesto por el equipo computacional, el hardware necesario para que el sistema de información pueda operar, y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema (Programas de Estudio a Distancia, s. f.).

Un sistema de información busca que la información esté disponible para satisfacer las necesidades requeridas por una organización, logrando acceder a la información mucho más eficiente. Además, cuenta con recursos que interactúan entre ellos, estos son:

- Entrada de información: proceso en el cual el sistema toma los datos que requiere.
- Almacenamiento de información: puede hacerse por computadora o archivos físicos para conservar la información.
- Procesamiento de la información: permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones
- Salida de información: es la capacidad del sistema para producir la información procesada o sacar los datos de entrada al exterior.

Un actor dentro que un sistema de información tiene diferente grado de participación según su jerarquía dentro del sistema, de esta manera se puede definir

usuarios primarios quienes alimentan el sistema, usuarios indirectos que se benefician de los resultados pero que no interactúan con el sistema, usuarios gerenciales y directivos quienes tienen responsabilidad administrativa y de toma de decisiones con base a la información que produce el sistema.

2.6.1 Tipos de Sistemas de Información.

Los sistemas de Información se clasifican en:

- **Sistemas Transaccionales:** Son sistemas de información que registran transacciones comerciales y financieras, tales como ventas, compras, pagos, entre otras. Estos sistemas son utilizados para automatizar procesos de negocio y garantizar la precisión y consistencia de la información. (O'Brien, 2008).
- **Sistemas de Soporte a la Decisión (DSS):** Son sistemas que proporcionan información para ayudar a la toma de decisiones en una organización. Estos sistemas permiten analizar datos complejos y presentar información en formatos fáciles de entender para los usuarios. Los DSS se utilizan en áreas como finanzas, marketing, producción y recursos humanos. (Laudon & Laudon, 2017).
- **Sistemas de Información Gerencial (MIS):** Son sistemas que proporcionan información para la gestión de una organización. Estos sistemas permiten a los gerentes obtener información sobre el rendimiento de la organización y tomar decisiones estratégicas basadas en datos. Los MIS se utilizan en áreas como finanzas, contabilidad, ventas y recursos humanos. (O'Brien, 2008).
- **Sistemas de Información para la Toma de Decisiones (EIS):** Son sistemas que proporcionan información de alta calidad para la toma de decisiones estratégicas en una organización. Los EIS se utilizan principalmente por los altos directivos y proporcionan información en tiempo real sobre el rendimiento de la organización, así como datos del entorno externo, como tendencias del mercado y cambios en la regulación. (Laudon & Laudon, 2017).
- **Sistemas de Automatización de Oficinas (OAS):** Son sistemas que automatizan tareas administrativas y operativas en una organización, tales como gestión de documentos, correo electrónico, calendarios y agendas. Los OAS mejoran la eficiencia y productividad de los empleados al reducir el tiempo y el esfuerzo requerido para realizar tareas repetitivas y rutinarias. (O'Brien, 2008).

- **Sistemas de Información para la Cadena de Suministro (SCM):** Son sistemas que gestionan y coordinan los procesos de la cadena de suministro, desde la adquisición de materias primas hasta la entrega de productos terminados a los clientes. Los SCM permiten una mejor colaboración y coordinación entre los diferentes actores de la cadena de suministro, lo que se traduce en una mayor eficiencia y reducción de costos. (Laudon & Laudon, 2017).
- **Según su alcance:** los sistemas de información pueden ser clasificados en sistemas de información de gestión (también conocidos como sistemas de información empresarial), sistemas de información de soporte a la toma de decisiones, sistemas de información de control de procesos, sistemas de información de colaboración y sistemas de información de conocimiento. Cada uno de estos tipos de sistemas tiene un alcance y una finalidad específica en la organización.
- **Según su arquitectura:** los sistemas de información pueden ser clasificados en sistemas centralizados, distribuidos o en red, dependiendo de la manera en que están estructurados y organizados sus componentes y recursos. Cada tipo de arquitectura tiene ventajas y desventajas en términos de rendimiento, seguridad, escalabilidad y costos.
- **Según su grado de estructuración:** los sistemas de información pueden ser clasificados en sistemas estructurados, semiestructurados o no estructurados, según el nivel de formalización y organización de la información que manejan. Los sistemas estructurados son aquellos que operan con datos altamente organizados y normalizados, mientras que los sistemas no estructurados manejan información no organizada, como texto libre o imágenes.
- **Según su ciclo de vida:** los sistemas de información pueden ser clasificados en sistemas en desarrollo, sistemas en operación y sistemas en fase de mantenimiento. Cada fase del ciclo de vida de un sistema de información tiene sus propias características y requerimientos, y requiere diferentes habilidades y recursos para su gestión y mantenimiento.

Es importante tener en cuenta que estos enfoques de clasificación no son excluyentes, y que un sistema de información puede ser clasificado en función de varios criterios al mismo

tiempo. Además, la clasificación de los sistemas de información puede variar en función de las necesidades y objetivos específicos de la organización.

2.6.2 Funciones de un Sistema de Información.

Las funciones de un sistema de información son:

- **Captura de datos:** Esta función implica la recopilación de datos e información de diferentes fuentes, incluyendo bases de datos, sensores, formularios en línea, entre otros. Según Laudon y Laudon (2016), la captura de datos es una de las funciones más importantes de un sistema de información, ya que permite recopilar información precisa y actualizada para su posterior procesamiento y análisis.
- **Almacenamiento de datos:** Una vez que los datos son capturados, estos deben ser almacenados en un sistema que permita su fácil acceso y búsqueda. Según O'Brien y Marakas (2011), el almacenamiento de datos es una función crítica de los sistemas de información, ya que permite a los usuarios acceder a la información de manera rápida y eficiente.
- **Procesamiento de datos:** Los datos capturados deben ser procesados para que sean útiles y relevantes para los usuarios. El procesamiento de datos incluye la limpieza, integración y transformación de los datos en información significativa. Según Stair y Reynolds (2013), el procesamiento de datos es una función clave de los sistemas de información, ya que permite convertir datos en información útil para la toma de decisiones.
- **Análisis de datos:** El análisis de datos implica la identificación de patrones, tendencias y relaciones en la información procesada. Según Sharda et al. (2014), el análisis de datos es una función esencial de los sistemas de información, ya que permite a los usuarios tomar decisiones informadas y basadas en datos.
- **Presentación de información:** La presentación de información es la función que permite a los usuarios acceder a la información procesada y analizada de manera fácil y comprensible. Según O'Brien y Marakas (2011), la presentación de información es una

función crítica de los sistemas de información, ya que permite a los usuarios tomar decisiones informadas y basadas en datos.

- **Almacenamiento y gestión de documentos:** Los sistemas de información también pueden ser utilizados para almacenar y gestionar documentos, tales como contratos, informes, facturas, entre otros. Según Laudon y Laudon (2016), la gestión de documentos es una función importante de los sistemas de información, ya que permite a las organizaciones administrar y acceder fácilmente a la información en formato de documentos.

2.7 HISTORIA CLINICA

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

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

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

La gestión de la información en las historias clínicas es crucial para garantizar la calidad y seguridad de la atención médica. La información debe ser precisa, actualizada, accesible y protegida de manera adecuada para garantizar la privacidad y la confidencialidad de los

pacientes (González et al., 2015). La adopción de sistemas electrónicos de gestión de historias clínicas ha permitido mejorar la gestión de la información y la atención al paciente en entornos clínicos.

En cuanto a la estructura de la historia clínica, existen diferentes modelos y formatos utilizados en todo el mundo. Sin embargo, en general, la historia clínica se compone de varios elementos fundamentales, como los antecedentes personales, antecedentes familiares, exploración física, diagnóstico, tratamiento y evolución (Tirado et al., 2017).

2.7.1 Características De Una Historia Clínica.

Según lo descrito por (Guzmán & Arias, 2012) “**Su práctica es obligatoria.** Ningún acto médico hospitalario o de consultorio debe efectuarse sin su correspondiente registro en la historia clínica. En las instituciones de salud se exige la historia clínica como elemento indispensable para ejercer una medicina de calidad. Por otro lado, en caso de complicaciones (salvo en algunos casos de extrema urgencia y corto tiempo disponible), su ausencia no tiene excusa.

Es irremplazable. La escritura de la historia no puede ser reemplazada por la memoria del médico. Es lógico que no se puedan conocer detalles de cada paciente, ni por el número ni por la complejidad individual de cada ser humano.

Es privada y pertenece al paciente. Aquí se integran los conceptos de confidencialidad, secreto profesional e información “.

A su vez, (Cuenca GK, Rodríguez LML, Soto CAD, et al) en su artículo menciona “La historia clínica, el secreto médico y la confidencialidad e intimidad constituyen una triada y se relacionan recíprocamente. La historia clínica es considerada como el soporte documental biográfico de la asistencia médico-sanitaria-administrativa de un paciente. Es el documento más privado que existe de una persona.

Una de sus características es la seguridad de los datos, no solo del paciente, sino de los facultativos y personal sanitario que intervengan en el proceso asistencial. Por razones económicas y gerenciales, la historia clínica estomatológica, es el documento más importante para respaldar, por escrito, procedimientos practicados, complicaciones aparecidas, entre otras.”

Tomando en cuenta lo anterior podemos describir a grandes rasgos algunas de las características más importantes que debe cumplir una historia clínica:

- **Completa:** La historia clínica debe incluir toda la información necesaria para conocer la situación actual del paciente, así como su historial médico completo.
- **Precisa:** La información registrada en la historia clínica debe ser precisa y estar respaldada por pruebas y diagnósticos médicos.
- **Actualizada:** La información debe ser actualizada regularmente para reflejar cualquier cambio en el estado de salud del paciente.
- **Confidencial:** La historia clínica es un documento confidencial que solo puede ser compartido con personas autorizadas por el paciente o por la ley.
- **Accesible:** A pesar de su carácter confidencial, la historia clínica debe ser accesible por el personal médico autorizado que la requiera para brindar una atención adecuada al paciente.
- **Legible:** La información registrada en la historia clínica debe ser legible y comprensible para cualquier persona autorizada que la lea.
- **Organizada:** La información debe estar organizada de forma lógica y coherente para que sea fácil de entender y utilizar.
- **Resumida:** La información debe ser resumida en un formato claro y conciso que permita una rápida revisión del historial médico del paciente.
- **Auditada:** La historia clínica debe ser auditada periódicamente para verificar la calidad de la información registrada y la adecuada utilización del documento.

Es importante tener en cuenta que estas características son fundamentales para garantizar una adecuada atención médica al paciente, así como para cumplir con las regulaciones y estándares legales y éticos establecidos en relación con la documentación médica.

2.7.2 Gestión de Historias Clínicas.

Hace mención a la gestión de toda la información relacionada con la atención médica que se brinda a un paciente en un entorno clínico. El objetivo principal de la gestión de historias clínicas es proporcionar una visión completa y actualizada del historial médico de un paciente para apoyar la toma de decisiones clínicas informadas. Además, la gestión de historias clínicas también se centra en la optimización de los procesos de atención médica y la reducción de errores médicos.

A continuación, se describen algunos aspectos importantes de la gestión de historias clínicas, que según American Health Information Management Association. (2018) son las más importantes:

- **Acceso y seguridad:** La gestión de historias clínicas debe garantizar que la información del paciente solo esté disponible para el personal médico autorizado y que se respeten las normas de privacidad y confidencialidad del paciente. La seguridad de la información del paciente también debe ser un aspecto importante para garantizar que la información no se modifique, se destruya o se acceda sin autorización.
- **Registro y documentación:** La gestión de historias clínicas debe garantizar que toda la información relevante sobre la atención médica del paciente se documente de manera oportuna, precisa y completa. Esto incluye información sobre las visitas médicas, los diagnósticos, los tratamientos, los medicamentos y los resultados de las pruebas.
- **Accesibilidad y portabilidad:** La gestión de historias clínicas debe garantizar que la información del paciente esté disponible para el personal médico autorizado en todo momento y en cualquier lugar donde se brinde atención médica. Además, la información del paciente debe ser fácilmente transportable entre diferentes sistemas de información y organizaciones de atención médica.

- **Análisis y toma de decisiones:** La gestión de historias clínicas debe permitir la realización de análisis de datos para mejorar la calidad de la atención médica y la toma de decisiones clínicas. Esto incluye el uso de análisis de datos para identificar patrones en los resultados de las pruebas, la eficacia de los tratamientos y la atención al paciente en general.
- **Integración e interoperabilidad:** La gestión de historias clínicas debe integrarse con otros sistemas de información y ser interoperable para permitir el intercambio de información entre diferentes organizaciones de atención médica y sistemas de información.
- **Mejora continua:** La gestión de historias clínicas debe estar en constante evolución para adaptarse a los cambios en la atención médica y las nuevas tecnologías. Además, debe haber una cultura de mejora continua para garantizar que la gestión de historias clínicas se adapte a las necesidades cambiantes de los pacientes y los proveedores de atención médica.

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

2.7.3 Historias Clínicas Electrónicas.

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

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

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

La gestión de HCE implica la implementación y el mantenimiento de sistemas informáticos para su almacenamiento, recuperación y uso. Según Ammenwerth et al. (2009), la gestión de HCE debe garantizar la confidencialidad, integridad, disponibilidad y accesibilidad de los datos del paciente.

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

2.8 MARCO TECNOLÓGICO

2.8.1 El Lenguaje JavaScript.

JavaScript es un lenguaje de programación ligero, interpretado y compilado, conocido mayormente como un lenguaje de secuencia de comandos para páginas web, surgió como una tecnología del lado del navegador permitiendo que las aplicaciones web fueran más dinámicas. Fue creado por Brendan Eich en 1995 mientras trabajaba en Netscape Communications Corporation y ha evolucionado a lo largo de los años evolucionando (Flanagan, 2011), estableciendo bibliotecas y librerías que se usan fuera de un navegador web, convirtiéndose en un lenguaje basado en prototipos, dinámico, orientado a objetos y declarativa. Permitiendo desarrollar tanto del lado del cliente como del lado del servidor.

2.8.1.1 Características y sintaxis de JavaScript.

JavaScript es un lenguaje de programación orientado a objetos y basado en prototipos. Esto significa que no tiene clases, como otros lenguajes de programación orientados a objetos, sino que se basa en la creación de objetos a partir de otros objetos existentes. La sintaxis de JavaScript es similar a la de otros lenguajes de programación como C y Java, pero tiene algunas diferencias importantes, como la falta de declaraciones de tipos de datos. (W3Schools, 2022)

2.8.1.2 Uso de JavaScript en la creación de Aplicaciones Web.

JavaScript se utiliza ampliamente en la creación de aplicaciones web, tanto en el lado del cliente como en el lado del servidor. En el lado del cliente, se utiliza principalmente para la interactividad y la manipulación del DOM (Document Object Model) de una página web. En el lado del servidor, se utiliza para la creación de aplicaciones web en tiempo real y para la interacción con bases de datos y otros servicios web. (MDN Web Docs, 2022)

2.8.1.3 Frameworks y Librerías de JavaScript.

JavaScript cuenta con una gran cantidad de frameworks y librerías que facilitan el desarrollo de aplicaciones web. Algunos de los frameworks más populares son Angular, React y Vue, mientras que algunas de las librerías más utilizadas son jQuery, Lodash y Moment.js. Estas herramientas permiten a los desarrolladores de aplicaciones web crear aplicaciones más rápidas y eficientes. (JavaScript.com, 2022)

2.8.2 Node Js.

Node.js es un entorno de tiempo de ejecución de JavaScript que permite ejecutar código JavaScript en el servidor. Node.js utiliza el motor de JavaScript V8 de Google para compilar y ejecutar el código JavaScript. Node.js es conocido por su capacidad para manejar grandes cantidades de conexiones simultáneas, lo que lo hace ideal para aplicaciones web en tiempo real. (Node.js, 2022).

Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente, dichos eventos permiten la comunicación con el servidor, además funcionan como intermediarios entre el servidor y el usuario.

2.8.2.1 Npm.

NPM (Node Package Manager) es el administrador de paquetes oficial de Node.js. NPM permite a los desarrolladores de Node.js compartir y reutilizar código fácilmente. NPM cuenta con una amplia variedad de paquetes que pueden ser utilizados para agregar funcionalidades a las aplicaciones de Node.js. (NPM, 2022).

El manejador de paquetes de node NPM, es la herramienta de instalación y gestión de paquetes de JavaScript. Este se compone de paquetes y librerías de software libre que juntos forman módulos y son utilizados en proyectos de node, facilitando proceso de desarrollo de un proyecto.

2.8.2.2 Express.

Express es un framework de Node.js para la creación de aplicaciones web. Express es conocido por su facilidad de uso y su enfoque minimalista, lo que lo hace ideal para la creación de aplicaciones web rápidas y sencillas. Express cuenta con una amplia variedad de middleware que permite a los desarrolladores agregar funcionalidades a sus aplicaciones web de manera rápida y fácil. (Express, 2022)

2.8.3 *React.*

React es una biblioteca de JavaScript para la creación de interfaces de usuario (UI). React se utiliza para construir componentes reutilizables que pueden ser utilizados para construir aplicaciones web y móviles. React utiliza un enfoque basado en componentes que hace que el desarrollo de aplicaciones sea más rápido y fácil. (React, 2022).

React es una biblioteca de JavaScript de código abierto para la creación de interfaces de usuario (UI). React fue desarrollado por Facebook y se lanzó por primera vez en 2013. Desde entonces, React se ha convertido en una de las bibliotecas de JavaScript más populares y utilizadas, especialmente para la creación de aplicaciones web y móviles.

El elemento más importante de React es el componente, que es un fragmento de interfaz de usuario. Cada componente es independiente y reusable para, crear interfaces de usuario más complejas.

2.8.4 PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto y gratuito. PostgreSQL es conocido por su confiabilidad, escalabilidad y capacidad de manejar grandes volúmenes de datos. PostgreSQL es compatible con una amplia variedad de plataformas, incluyendo Windows, Linux, MacOS, Solaris y FreeBSD. (PostgreSQL, 2022).

2.8.4.1 Características de PostgreSQL.

PostgreSQL tiene una serie de características que lo hacen único entre los sistemas de gestión de bases de datos relacionales. Algunas de las características más destacadas de PostgreSQL incluyen:

- Soporte para transacciones ACID
- Soporte para llaves foráneas
- Soporte para disparadores (triggers)
- Soporte para procedimientos almacenados
- Soporte para vistas materializadas
- Soporte para índices y consultas de texto completo
- Soporte para replicación y alta disponibilidad (PostgreSQL, 2022)

2.8.4.2 Ventajas de PostgreSQL.

PostgreSQL tiene varias ventajas sobre otros sistemas de gestión de bases de datos relacionales. Algunas de las ventajas más destacadas incluyen:

- Es de código abierto y gratuito
- Es altamente escalable y confiable
- Tiene una comunidad de desarrolladores activa y una amplia gama de herramientas de soporte
- Es compatible con una amplia variedad de plataformas
- Tiene un rendimiento rápido y eficiente (PostgreSQL, 2022)

2.8.5 *Object-Relational Mapping (ORM).*

Un ORM es un patrón de diseño de software que permite a los desarrolladores interactuar con una base de datos relacional utilizando objetos y métodos en lugar de SQL. Los ORM se utilizan para simplificar el proceso de acceso a la base de datos y reducir la cantidad de código que los desarrolladores tienen que escribir para interactuar con la base de datos. Los ORM también proporcionan una capa de abstracción entre el código de la aplicación y la base de datos, lo que hace que la aplicación sea más independiente de la base de datos subyacente. (Hibernate, 2022)

2.8.6 *Orm Sequelize.*

Sequelize es un ORM para Node.js que admite bases de datos relacionales. Sequelize proporciona una API orientada a objetos para interactuar con la base de datos, lo que permite a los desarrolladores escribir consultas de base de datos utilizando objetos y métodos JavaScript. Sequelize también proporciona soporte para migraciones de bases de datos, validaciones de modelos y asociaciones entre modelos. (Sequelize, 2022).

2.8.6.1 Características de Sequelize.

Sequelize tiene una serie de características que lo hacen único entre los ORM para Node.js. Algunas de las características más destacadas de Sequelize incluyen:

- Soporte para múltiples bases de datos relacionales, incluyendo PostgreSQL, MySQL, SQLite y MSSQL
- Soporte para migraciones de bases de datos
- Soporte para validaciones de modelos
- Soporte para asociaciones entre modelos
- Soporte para consultas de base de datos utilizando objetos y métodos JavaScript
- Soporte para transacciones
- Soporte para consultas en crudo (raw queries)
- Soporte para hooks (ganchos) para ejecutar código antes o después de la ejecución de una consulta de base de datos (Sequelize, 2022)

2.8.6.2 Ventajas de Sequelize.

Sequelize tiene varias ventajas sobre otros ORM para Node.js. Algunas de las ventajas más destacadas incluyen:

- Es fácil de aprender y usar, especialmente para desarrolladores de JavaScript
- Es altamente personalizable y flexible
- Proporciona soporte para múltiples bases de datos relacionales
- Proporciona soporte para migraciones de bases de datos y validaciones de modelos
- Proporciona soporte para consultas de base de datos utilizando objetos y métodos JavaScript
- Proporciona soporte para transacciones y consultas en crudo (Sequelize, 2022)

2.9 METRICAS DE CALIDAD Y SEGURIDAD DE SOFTWARE

2.9.1 Metodología WebQem.

La metodología de Evaluación de Calidad de Sitios y Aplicaciones Web (WebQEM) fue introducido por Olsina, Covella y Rossi, desarrollada desde mediados del 98, con el propósito de aportar una estrategia eficaz para evaluar y analizar la calidad de sitios y aplicaciones Web en general. (Covella & Olsina, 2002).

Está basada en un modelo jerárquico de requerimientos de calidad, partiendo de las características de las normas ISO 9126-1 y ISO 2001: Usabilidad, Funcionalidad, Confiabilidad y Eficiencia. También se basa en la norma ISO/IEC 27001: Esta norma establece los requisitos para un sistema de gestión de seguridad de la información. La metodología WebQEM se enfoca en la seguridad lógica y física de las aplicaciones web, y esto incluye la implementación de medidas de seguridad de la información y la gestión de riesgos de seguridad.

2.9.2 Fases de WebQem.

WebQem utiliza fases para realizar los cálculos que ayudaran a comprobar que el sistema cumple sus características, las fases son:

- 1) Definición de las metas de evaluación y selección del perfil de usuario
- 2) Definición de los requerimientos de calidad (y/o costo).
- 3) Definición de criterios de preferencia elementales y procedimientos de medición.
- 4) Definición de estructuras de agregación e implementación de la evaluación global.
- 5) Análisis de resultados y recomendaciones

Estas fases nos ayudaran a comprobar que el sistema es confiable, eficiente y de calidad.

2.9.3 Características de WebQem.

2.9.3.1.1 Funcionalidad.

Se aprecia evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global. (Gonzalez, 2001).

- 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 función se usa las siguientes formulas (Gonzalez, 2001):

$$PF = \text{Cuenta Total} * (\text{confiabilidad proyecto} + \text{error min} * \sum Fi) \text{ Ecuación 1.}$$

$$\text{Funcionalidad} = \left(\frac{PF}{PF_{\text{máximo}}} \right) \text{ Ecuación 2.}$$

Donde:

- **PF:** Medida de funcionalidad.
- **PFmáximo:** 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 y número de interfaces externas. **Confiabilidad proyecto:** Confiabilidad del proyecto,

varia de 1% al 100% (0 a 1). **Error min:** Error mínimo aceptable de complejidad.

- ΣFi : Son los valores de ajuste de complejidad, donde ($1 \leq i \leq 14$).

2.9.3.2 Confiabilidad.

(Cueva, 2005) Nos dice. Es la probabilidad de operación libre de fallos en 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) Paginas

- Páginas Muertas
- Páginas bajo Construcción
- Errores de Ortografía

Para calcular los puntos función se usa la siguiente formula:

$$\text{Porcentaje de enlaces rotos} = \frac{\text{CERI} + \text{CERE}}{\text{CTE}} * 100 \text{ *Ecuación 3.*}$$

Donde:

CERI: Cantidad de enlaces rotos internos.

CERE: Cantidad de enlaces rotos externos.

CTE: Cantidad total de enlaces.

Para hallar el porcentaje de presencia de propiedad se realiza mediante la Ecuación:

$$\text{Porcentaje presencia ALT} = \frac{\text{CantidadImágenesALT}}{\text{CantidadTotalImágenes}} * 100 \text{ **Ecuación 4.**}$$

Y la confiabilidad se calcularía con la siguiente formula:

$$\text{Confiabilidad} = \text{PorcentajeEnlaces} + \text{PorcentajePresenciaALT} \text{ **Ecuación 5.**}$$

2.9.3.3 Usabilidad.

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

Modelo de Calidad para la Web
<u>Título:</u> Usabilidad
<u>Tipo:</u> Característica
<u>Sub-característica/s:</u>
Comprensibilidad (Global del Sitio)
Mecanismo de Ayuda y Retroalimentación
Aspectos de Interfaces
Aspectos Estéticos y de Estilo

Figura 2.9.1 Características WebQem,
Usabilidad. Fuente: Olsina, 1999

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 del Color de Los Enlaces:

Se toma en cuenta el diseño que tiene el sistema.

2.9.3.4 Mantenibilidad.

Facilidad con que una modificación puede ser realizada. Está indicada por los siguientes sub atributos (Sicilia, 2009):

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

(Gonzalez, 2001) Nos indica que el índice de madurez del software se calcula con la Ecuación 6:

$$IMS = \frac{[M_T - (F_a + F_b + F_c)]}{M_T} \text{ Ecuación 6.}$$

Donde:

- M_t : Número de módulos en la versión actual
- F_a : Número de módulos en la versión actual que se han cambiado
- F_b : Número de módulos en la versión actual que se han añadido
- F_c : Número de módulos en la versión anterior que se han borrado en la versión actual.

2.9.3.5 Portabilidad.

La portabilidad se refiere a la facilidad con que un sistema o componente del mismo puede ser transferido a diferentes ambientes de hardware o software (Bruegge, Dutoit, 2004, como se citó en Galicia, 2004).

Para poder medir la portabilidad del sistema haremos uso de la Ecuación 7 (Galicia, 2004), que indica el grado de portabilidad que tiene un software.

$$GP = 1 - \frac{CPS}{CRS} \quad \text{Ecuación 7.}$$

Donde:

- CPS: Es la medida o el costo de portar el software, además de los recursos necesarios para llevar el sistema a otro entorno.
- CRS: costo de redesarrollar el software, 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.

2.10 MODELO DE COSTOS DE SOFTWARE

El modelo de costos de software es una herramienta que permite a los desarrolladores y empresas estimar el costo y los recursos necesarios para desarrollar un proyecto de software. El modelo de costos se basa en una serie de factores, como la complejidad del proyecto, el tiempo requerido, los recursos necesarios, el personal involucrado y el tipo de tecnología utilizada. (Boehm, 1981).

Algunos de los factores clave que se deben tener en cuenta al desarrollar un modelo de costos de software incluyen:

- Complejidad del proyecto
- Tiempo requerido
- Recursos necesarios
- Personal involucrado
- Tipo de tecnología utilizada (Boehm, 1981)

Existen varios métodos para desarrollar un modelo de costos de software, como el método de punto de función y el método de línea de código. El método de punto de función se basa en el número de entradas, salidas, consultas, archivos lógicos y archivos de interfaz de usuario de un proyecto de software para estimar su tamaño y complejidad. El método de línea de código se basa en el número de líneas de código necesarias para desarrollar el proyecto de software. (Boehm, 1981)

2.10.1 Modelo COCOMO II.

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

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.

2.10.1.1 Factores clave del Modelo COCOMO II.

El Modelo COCOMO II se basa en varios factores clave que influyen en el costo y el esfuerzo necesarios para desarrollar un proyecto de software, como el tamaño del proyecto, la complejidad, la experiencia del equipo de desarrollo, el lenguaje de programación utilizado y el entorno de desarrollo. (Boehm, 2000)

2.10.1.2 Ecuaciones del Modelo COCOMO II.

El Modelo COCOMO II se compone de tres ecuaciones que se utilizan para estimar el costo y el esfuerzo necesarios para desarrollar un proyecto de software:

- Ecuación de tamaño: se utiliza para estimar el tamaño del proyecto en líneas de código fuente (LOC).
- Ecuación de costo: se utiliza para estimar el costo total del proyecto en dólares.
- Ecuación de esfuerzo: se utiliza para estimar el esfuerzo total del proyecto en persona-mes.

Las ecuaciones se presentan a continuación:

Ecuación de tamaño: $Size = A * (KLOC)^B$

Ecuación de costo: $Cost = E * (Size)^F * (M)^G$

Ecuación de esfuerzo: $Effort = D * (Size)^E * (M)^H$

Donde:

- A, B, D, E, F, G, H son constantes que dependen del tipo de proyecto y del modelo de desarrollo utilizado.
- KLOC es el tamaño del proyecto en miles de líneas de código fuente.
- E es el costo de un esfuerzo de persona-mes en dólares.
- M es el factor de complejidad del proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- World Health Organization. Regional Office for Europe. (2019, 10 octubre). *Future of digital health systems: report on the WHO symposium on the future of digital health systems in the European region: Copenhagen, Denmark, 6–8 February 2019*. <https://apps.who.int/iris/handle/10665/329032>
- Gil López, E., & Medinaceli Díaz, K. (2019). La protección del derecho a la intimidad en la historia clínica electrónica en Bolivia: una perspectiva para Latinoamérica. *Boletín mexicano de derecho comparado*, 52(154), 489-511
- Jácome, F. (2015, 2 septiembre). Repositorio Digital Universidad Israel: Desarrollo del Sistema de Administración de Historias Clínicas del Departamento Médico de la Empresa Imprenta Mariscal Cia.Ltda. <http://repositorio.uisrael.edu.ec/handle/47000/584>
- Ortiz, O. L. E. (2022, 6 febrero). Repositorio Digital - EPN: Desarrollo de un sistema web de gestion de historias clinicas en un consultorio privado de medicina general. <https://bibdigital.epn.edu.ec:443/handle/15000/22122>
- Padilla, P. J. (2018, 23 enero). Repositorio Institucional Unicordoba: Diseño e implementación de un sistema de administración y consulta de historias clinicas electronicas (HCE) mediante el uso de tecnologia Webservices en diversos entes de salud del municipio de Santa Cruz de Lorica - Córdoba. <https://repositorio.unicordoba.edu.co/handle/ucordoba/482>
- Paz, G. C. A. (2020, 4 noviembre). *Repositorio Institucional UPEA: SISTEMA WEB DE ADMINISTRACIÓN DE HISTORIAS CLÍNICAS*. <http://repositorio.upea.bo/handle/123456789/67>
- Flores, G. P. (2018, 26 junio). Sistema web de administración de historias clínicas Caso: “centro médico quirúrgico Erzengel”. <https://repositorio.umsa.bo/xmlui/handle/123456789/16755>
- Ponce, Y. Y. A. (2016, 5 octubre). Software como servicio para la administración de historias clínicas. <https://repositorio.umsa.bo/xmlui/handle/123456789/8179>
- The Agile Unified Process (AUP) Home Page. (s. f.). Ambysoft Inc. <http://www.ambysoft.com/unifiedprocess/agileUP.html>
- Todos los diagramas UML. Teoría y ejemplos. (2022, 16 agosto). DiagramasUML.com. <https://diagramasuml.com>
- Fakhroutdinov, K. (2013, 25 noviembre). UML 2.5 Diagrams Overview. <https://www.uml-diagrams.org/uml-25-diagrams.html>

- Debrauwer, L., & Van der Heyde, F. (2016). UML 2.5: iniciación, ejemplos y ejercicios corregidos. Ediciones ENI.
- González, Y. D. (2012, 12 junio). Patrón Modelo-Vista-Controlador. | Telemática. <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/0>
- Hernandez, R. D. (2021, 28 junio). Patrones de arquitectura MVC. freeCodeCamp. <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>
- Alonso-Aranda, Carlos. (2019). MODELO-VISTA-CONTROLADOR. LENGUAJE UML [Repositorio de Trabajos Académicos de la Universidad de Jaén]. Jaén: Universidad de Jaén. <https://hdl.handle.net/10953.1/11437>
- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for agile software development. Agile Alliance.
- Schwaber, K., & Sutherland, J. (2017). The Scrum guide. Scrum.org.
- Cockburn, A. (2002). Agile software development. Addison-Wesley Professional.
- Cockburn, A., & Highsmith, J. (2001). Agile software development: the people factor. Computer, 34(11), 131-133.
- Scott W. Ambler. Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons, 2002.
- Unified Process. (n.d.). In Rational Edge. IBM Rational. Recuperado el 22 de febrero de 2023, de https://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep00/f_UnifiedProcess_db.html.
- Ambler, S. W. (2009). Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons.
- Boehm, B. (2002). Get ready for agile methods, with care. IEEE computer society.
- Martin, R. C. (2008). Agile software development: principles, patterns, and practices. Prentice Hall.
- Ali, M., Abood, A. H., y Rashid, A. N. (2012). A comprehensive survey on UML-based web engineering. Journal of Software Engineering and Applications, 5(9), 658-671.
- Conradi, R., y Westfechtel, B. (2014). Object-Oriented Technology. Springer.

- Mellado, D., García-Sánchez, F., Ruiz-Cortés, A., y Piattini, M. (2007). UML-based web engineering: a systematic mapping study. *Software & Systems Modeling*, 6(1), 1-20.
- Wang, Q., y Song, Y. (2008). Web engineering based on UML: a survey. *Journal of Software*, 3(5), 1-9.
- Larman, C. (2003). *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional.
- Sommerville, I. (2011). *Software engineering*. Pearson Education.
- Highsmith, J. (2002). *Agile software development ecosystems*. Addison-Wesley Professional.
- Kruchten, P. (2004). *The rational unified process: an introduction*. Addison-Wesley Professional.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The unified modeling language user guide*. Addison-Wesley Professional.
- Ambler, S. W. (2005). *Agile modeling: effective practices for extreme programming and the unified process*. John Wiley &
- OMG. (2017). *Unified Modeling Language (UML) - OMG*. Recuperado el 26 de febrero de 2023, de <https://www.omg.org/spec/UML/2.5.1/>
- Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The unified modeling language reference manual*. Pearson Education.
- Fowler, M. (2004). *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.
- Laudon, K. C., & Laudon, J. P. (2016). *Management Information Systems: Managing the Digital Firm*. Pearson.
- O'Brien, J. A., & Marakas, G. M. (2011). *Management Information Systems (10th ed.)*. McGraw-Hill.
- Stair, R. M., & Reynolds, G. W. (2013). *Principles of Information Systems (11th ed.)*. Cengage Learning.
- Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. George Braziller.
- Checkland, P. (1981). *Systems thinking, systems practice*. John Wiley & Sons.

- León, A. (2004). Ingeniería de sistemas. Pearson Education.
- Turban, E., Leidner, D., McLean, E., & Wetherbe, J. (2005). Information technology for management: transforming organizations in the digital economy. John Wiley & Sons.
- Davis, G. B., & Olson, M. H. (1984). Management information systems: conceptual foundations, structure, and development. McGraw-Hill.
- Senge, P. (1990). La quinta disciplina. Cómo impulsar el aprendizaje en la organización inteligente. Granica.
- American Medical Association. (1997). Medical Records and Documentation. Retrieved from <https://www.ama-assn.org/practice-management/medical-records-documentation>
- World Health Organization. (2016). WHO guideline on good data and record management practices. Retrieved from https://www.who.int/medicines/technical_briefing/tbs/WHO_Guidelines_on_Good_Data_and_Record_Management_Practices_2016.pdf
- Gallegos Tapia, B. I. & Reina Velásquez, M. F. (2013, julio). DISEÑO DE UN MANUAL DE HIGIENE, SALUD Y SEGURIDAD OCUPACIONAL PARA EL PERSONAL DE LA COOPERATIVA DE AHORRO Y CRÉDITO ANDINA, PERÍODO 2012-2013”. <https://core.ac.uk/download/pdf/336839831.pdf>
- Ministerio de Salud y Deportes, Luna Orosco, Dr. J. & Carrasco, Dra. M. (2008). Norma técnica para el manejo del expediente clínico. https://www.minsalud.gob.bo/images/Documentacion/dgss/Area_de_Calidad/64%20Norma%20expediente%20clinico.pdf
- Galindo, L. M., & Martínez, J. G. G. (2012). Fundamentos de administración. Trillas.
- Guzmán, F. & Arias, C. (2012). La historia clínica: elemento fundamental del acto médico. Revista Colombiana de Cirugía, vol.27(no.1), http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=s2011-75822012000100002.
- González, A. M., Rodríguez, J. J. R., Ojeda, R. A. V., & González, A. M. (2015). Sistema de información y gestión de historias clínicas en el contexto de la atención primaria. Revista Habanera de Ciencias Médicas, 14(2), 304-316.
- Tirado, A. R., Durán, E. C., & Blanco, R. A. (2017). Historia clínica electrónica y su uso en la atención primaria de la salud. Medisur, 15(1), 103-113.
- Vega, J. S. (2016). La historia clínica: concepto, evolución y situación actual. Acta Colombiana de Cuidado Intensivo, 16(2), 83-89.

- Cuenca GK, Rodríguez LML, Soto CAD, et al. The clinical dental history as a tool in the clinical method and as medical-legal document. *Rev Cub Med Mil* . 2014;43(4):534-540.
- Chiavenato I. (2006). *Introducción a la Teoría General de la Administración*, Séptima Edición, McGraw-Hill Interamericana, <https://esmirusite.files.wordpress.com/2017/07/i-admon-chiavenato.pdf>
- Bertoglio, J. O. (2013). *INTRODUCCION A LA TEORIA GENERAL DE SISTEMAS* (1.a ed.). LIMUSA.
- Gutiérrez Gómez, G., Universidad Santo Tomás & Vicerrectoría General de Universidad Abierta y a Distancia. (2013). *Teoría General de Sistemas* (Universidad Santo Tomás). Ediciones USTA.
- ITSON | Enfoque de Sistemas | Inicio. (s. f.). http://biblioteca.itson.mx/oa/ciencias_administrativa/oa3/enfoque_sistemas/index.htm
- American Health Information Management Association. (2018). *Health Informatics*. AHIMA.
- Bowden, T., & Buchanan, S. (2015). Data governance for electronic health records. *American Journal of Managed Care*, 21(12), e631-e637.
- Davis, K., & Koch, S. (2018). EHRs and the digitalization
- Ammenwerth, E., Schnell-Inderst, P., & Hoerbst, A. (2009). The impact of electronic patient portals on patient care: A systematic review of controlled trials. *Journal of Medical Internet Research*, 11(4), e32. doi: 10.2196/jmir.1350
- Chaudhry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., . . . Shekelle, P. G. (2006). Systematic review: Impact of health information technology on quality, efficiency, and costs of medical care. *Annals of Internal Medicine*, 144(10), 742-752. doi: 10.7326/0003-4819-144-10-200605160-00125
- Organización Mundial de la Salud. (2013). *Sistemas de información de salud*. Recuperado el 18 de febrero de 2023, de https://www.who.int/health-topics/health-systems#tab=tab_2
- Zhou, L., Soran, C. S., Jenter, C. A., Volk, L. A., Orav, E. J., & Bates, D. W. (2015). The relationship
- Programas de Estudio a Distancia & UNIVERSIDAD DE PAMPLONA- Facultad de Estudios a Distancia. (s. f.). *Análisis y Diseño de Sistemas de Información* (UNIVERSIDAD DE PAMPLONA-Facultad de Estudios a Distancia) [<https://www.unipamplona.edu.co>]. UNIVERSIDAD DE PAMPLONA.

https://www.unipamplona.edu.co/unipamplona/portallIG/home_109/recursos/octubre2014/administraciondeempresas/semestre7/11092015/analisisydisenosistinformacion.pdf

- Campo, L. (s. f.). Sistema de Información. <http://www.incap.int/sisvan/index.php/es/acerca-de-san/conceptos/797-sin-categoria/501-sistema-de-informacion>
- JavaScript | MDN. (2022, 7 noviembre). <https://developer.mozilla.org/es/docs/Web/JavaScript>
- ¿Qué es JavaScript? - JavaScript explicado - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/javascript/>
- Flanagan, D. (2011). JavaScript: The Definitive Guide: Activate Your Web Pages (6th ed.). O'Reilly Media.
- W3Schools. (2022). JavaScript Tutorial. Recuperado de <https://www.w3schools.com/js/default.asp>
- MDN Web Docs. (2022). JavaScript. Recuperado de <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- JavaScript.com. (2022). Frameworks and Libraries. Recuperado de <https://www.javascript.com/resources/frameworks-libraries>
- Netguru. (s. f.). What Is Node.js? Complex Guide for 2022. <https://www.netguru.com/glossary/node-js>
- Node.js. (2022). About. Recuperado de <https://nodejs.org/en/about/>
- NPM. (2022). About npm. Recuperado de <https://docs.npmjs.com/about-npm/>
- Express. (2022). About Express. Recuperado de <https://expressjs.com/en/about.html>
- React. (2022). Getting Started. Recuperado de <https://reactjs.org/docs/getting-started.html>
- Qué es NodeJS y para qué sirve. (2022, 12 septiembre). OpenWebinars.net. <https://openwebinars.net/blog/que-es-nodejs/>
- Hernandez, M. (2021, 8 febrero). ¿Qué es NPM? freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/que-es-npm/>
- *React | Qué es, para qué sirve y cómo funciona | Descúbrelo todo.* (2021, 8 noviembre). Tribalbyte Technologies. <https://tech.tribalyte.eu/blog-que-es-react>

- PostgreSQL. (2022). About PostgreSQL. Recuperado de <https://www.postgresql.org/about/>
- PostgreSQL. (2022). Features. Recuperado de <https://www.postgresql.org/about/features/>
- PostgreSQL. (2022). Why PostgreSQL? Recuperado de <https://www.postgresql.org/why-postgresql/>
- Hibernate. (2022). Introduction to Hibernate ORM. Recuperado de <https://hibernate.org/orm/what-is-an-orm/>
- Sequelize. (2022). What is Sequelize? Recuperado de <https://sequelize.org/master/manual/getting-started.html>
- Sequelize. (2022). Features. Recuperado de <https://sequelize.org/master/manual/getting-started.html#features>
- Sequelize. (2022). Why Sequelize? Recuperado de <https://sequelize.org/master/manual/getting-started.html#why-sequelize>
- Covella, Guillermo & Olsina, Luis. (2002). *Evaluación de Calidad de Sitios Web con Funcionalidad E-Learning*.
- González, H. 2001. *CAPITULO 3. Métricas Técnicas*. Recuperado de: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo3.pdf
- González, H. 2001. *CAPITULO 4. Métricas en el Desarrollo del Software*. Recuperado de: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo4.pdf
- Cueva, J. M. 2005. *MÉTRICAS DE USABILIDAD EN LA WEB*. Recuperado de: <http://di002.edv.uniovi.es/~cueva/asignaturas/masters/2005/MetricasUsabilidad.pdf>
- Olsina, L. 1999. *Ingeniería de Software en la Web*. Recuperado de: <http://di002.edv.uniovi.es/~cueva/investigacion/tesis/WebsiteQEM.pdf>
- SICILIA, M. 2009. *ESTÁNDAR ISO 9126 DEL IEEE Y LA MANTENIBILIDAD*. Recuperado de: <http://cnx.org/exports/3d263044-60f5-4eda-a117-23660ce72819@3.pdf/est%C3%A1ndar-iso-9126-del-ieee-y-lamantenibilidad3.pdf>.
- GALICIA, K. M. 2004. *Capítulo 4: Prueba de Adaptabilidad*. Recuperado de: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/galicia_g_km/capitulo4.pdf
- Boehm, B. W. (1981). Software engineering economics. *IEEE Transactions on Software Engineering*, 7(3), 231-241.

- Pressman, R. S., & Maxim, B. R. (2000). Software engineering: A practitioner's approach. McGraw-Hill.
- Boehm, B. (2000). Software Engineering Economics. Prentice Hall.
- Boehm, B., Abts, C., Brown, A.W., Chulani, S., Clark, B.K., Horowitz, E., Madachy, R., Reifer, D.J., & Steece, B. (2000). COCOMO II Model Definition Manual. USC-CSE-TR-2000-TR005. University of Southern California, Center for Software Engineering.