**UNIVERSIDAD MAYOR DE SAN ANDRÉS**

**FACULTAD DE CIENCIAS PURAS Y NATURALES**

**CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

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

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

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMÁTICA

MENCION INGENIERIA DE SISTEMAS INFORMÁTICOS

**POSTULANTE: NEIL ANGEL GRANEROS FLORES**

**TUTOR: PH.D. MARISOL TELLEZ RAMIREZ**

La Paz - Bolivia   
2023

***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 concejos 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 Montaño 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***

**ÍNDICE**

[CAPÍTULO I MARCO REFERENCIAL 1](#_Toc150707037)

[1.1 Introducción 1](#_Toc150707038)

[1.2 Problema 2](#_Toc150707039)

[1.2.1 Antecedentes al Problema 2](#_Toc150707040)

[1.2.2 Planteamiento del Problema 12](#_Toc150707041)

[1.2.3 Formulación del Problema 12](#_Toc150707042)

[1.3 Objetivos 13](#_Toc150707043)

[1.3.1 Objetivo General 13](#_Toc150707044)

[1.3.2 Objetivos Específicos 13](#_Toc150707045)

[1.4 Justificación 14](#_Toc150707046)

[1.4.1 Justificación Social 14](#_Toc150707047)

[1.4.2 Justificación Económica 15](#_Toc150707048)

[1.4.3 Justificación Tecnológica 15](#_Toc150707049)

[1.5 Alcances Y Límites 15](#_Toc150707050)

[1.5.1 Alcance 15](#_Toc150707051)

[1.5.2 Límites 16](#_Toc150707052)

[1.6 Metodología 17](#_Toc150707053)

[1.6.1 Técnicas e Instrumentos de Recolección de Información 17](#_Toc150707054)

[1.6.2 Metodología de Desarrollo 17](#_Toc150707055)

[CAPÍTULO II MARCO TEÓRICO 20](#_Toc150707057)

[2.1 Ingeniería de Software 20](#_Toc150707058)

[2.1.1 El Proceso de Software 21](#_Toc150707059)

[2.1.2 Modelos de Desarrollo de Software 21](#_Toc150707060)

[2.1.3 Metodología Ágil en el Desarrollo de Software 25](#_Toc150707061)

[2.2 Proceso Unificado Ágil (AUP). 27](#_Toc150707062)

[2.2.1 Principios del Proceso Unificado Ágil (AUP). 27](#_Toc150707063)

[2.2.2 Ciclo de Vida del Proceso Unificado Ágil (AUP). 28](#_Toc150707064)

[2.2.3 Fases e Hitos del Ciclo de Vida. 29](#_Toc150707065)

[2.2.4 Disciplinas 32](#_Toc150707066)

[2.2.5 Hitos 33](#_Toc150707067)

[2.2.6 Ventajas y desventajas de la implementación. 35](#_Toc150707068)

[2.3 Ingeniería Web 36](#_Toc150707069)

[2.3.1 La Ingeniería Web Dirigida por Modelos 37](#_Toc150707070)

[2.3.2 Metodologías en Ingeniería Web 38](#_Toc150707071)

[2.4 UWE (UML-Based Web Engineering). 39](#_Toc150707072)

[2.4.1 Modelos - Fases de UWE. 41](#_Toc150707073)

[2.5 Historia Clínica 49](#_Toc150707074)

[2.5.1 Características de una Historia Clínica. 50](#_Toc150707075)

[2.5.2 Gestión de Historias Clínicas. 51](#_Toc150707076)

[2.5.3 Historia Clínica Electrónica (HCE). 53](#_Toc150707077)

[CAPITULO III MARCO APLICATIVO 55](#_Toc150707078)

[3.1 INTRODUCCIÓN 55](#_Toc150707079)

[3.2 FASE DE INICIO 56](#_Toc150707080)

[3.2.1 Especificación de Requerimientos 58](#_Toc150707081)

[ANEXOS 63](#_Toc150707082)

[REFERENCIAS BIBLIOGRÁFICAS 67](#_Toc150707083)

**ÌNDICE DE FIGURAS**

[**Figura 1.1** Estructura Orgánica del Instituto de Genética. 11](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149951816)

[**Figura 2.1** Capas de la ingeniería de software 20](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149951817)

[**Figura 2.2** Modelo en Cascada 22](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149951819)

[**Figura 2.3** Modelo en Espiral 23](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149951820)

[**Figura 2.4** Modelo de Prototipo 24](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149951821)

[**Figura** **2.5** Ciclo de vida de AUP 29](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149951822)

[**Figura 2.6** Visión general de los modelos UML 35](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149951823)

**ÍNDICE DE TABLAS**

[**Tabla 1.1** Aplicación de TIC's en el subsector público 3](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149952193)

[**Tabla 1.2** Aplicación de TIC's en el subsector de la Seguridad Social 4](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149952194)

[**Tabla 1.3** Aplicación de TIC's en el subsector privado y actores 5](file:///D:\NEIL\TALLER-II\borrador.docx#_Toc149952195)

[**Tabla 3.5.1** Tabla de Requerimientos Funcionales 47](#_Toc149952196)

# CAPÍTULO I MARCO REFERENCIAL

## Introducción

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

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

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

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

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

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

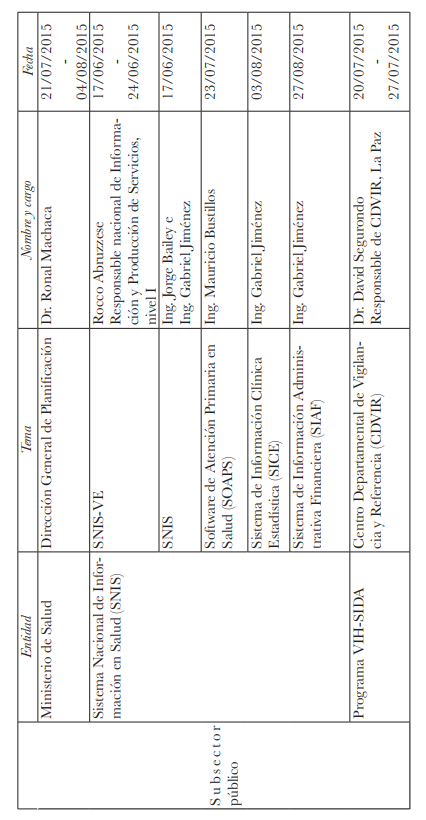
## Problema

### Antecedentes al Problema

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

#### Estado del Arte.

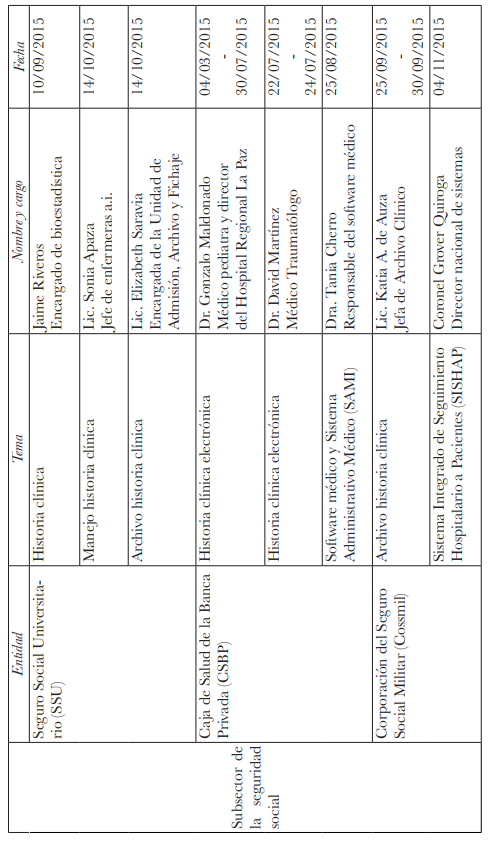
Según Gil López, E., y Medinaceli Díaz, K. (2019), en Bolivia se presenta una deficiencia general en el acceso a la información, ya que esta no se encuentra disponible en los sitios web oficiales. Además, algunas instituciones tanto públicas como privadas carecen de memorias institucionales anuales que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

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

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

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

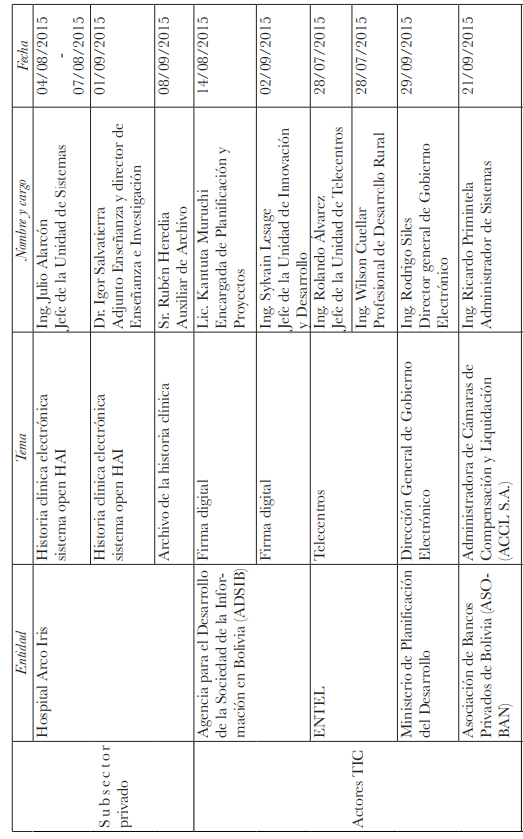
**Tabla 1.1** Aplicación de TIC's en el subsector público

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

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

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

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



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

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

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

* La Caja de Salud de la Banca Privada ha desarrollado su propio software médico y sistema administrativo médico (SAMI) desde el año 2005. Este sistema cuenta con módulos específicos para cada usuario, lo que permite la generación de informes de exámenes, laboratorios, médicos y seguimiento.
* La Corporación del Seguro Social Militar (Cossmil), una empresa boliviana, creó el Sistema de Gestión Hospitalaria (SIGEH) en 2004. Desde 2014, se ha implementado el Sistema de Información Integrado de Control y Seguimiento Hospitalario (SISHAP), que también se utiliza en las ciudades de Sucre y Puerto Suárez.
* El Hospital Arco Iris (HAI) adoptó el sistema openHAI hace dos años. Este sistema incluye la Historia Clínica Electrónica (HCE) de hospitalización, terapia intensiva (UTI), emergencias y consulta externa desde septiembre de 2015. Además, ofrece un módulo especializado para almacenar imágenes de resonancias, tomografías, radiografías y otros estudios radiológicos.

#### Trabajos Similares.

##### Contexto Internacional

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

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

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

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

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

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

* Este proyecto abordó desafíos significativos que afectan la implementación de sistemas de historia clínica electrónica (HCE) en centros de salud. Entre los obstáculos identificados se encontraban el crecimiento exponencial de la información, problemas de compatibilidad de archivos y consideraciones de costos, entre otros. El objetivo principal del proyecto fue diseñar e implementar un sistema de administración y consulta de historias clínicas electrónicas (HCE) mediante la aplicación de tecnología Webservices.
* Este enfoque involucró el desarrollo de tecnologías Webservices para acceder y administrar la información, además de la creación de pruebas de funcionamiento para garantizar su eficacia. Para llevar a cabo este proyecto, se siguió la metodología Rational Unified Process (RUP). Las herramientas utilizadas incluyeron lenguajes de programación como C y C++ en el entorno Visual Studio, así como tecnologías web como HTML 5, JavaScript y CSS. El proyecto se llevó a cabo en la Carrera de Ingeniería de Sistemas, Facultad de Ingenierías de la Universidad de Córdoba, Colombia.

##### Contexto Nacional

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

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

* La problemática central abordada en este proyecto estaba relacionada con la gestión de historias clínicas en formato físico, lo que ocasionaba molestias tanto al personal médico como a los pacientes. Esto generaba expedientes clínicos desactualizados, pérdida de información, duplicación de expedientes y retrasos en la atención médica. El objetivo principal de este proyecto fue desarrollar un sistema web de administración de historias clínicas. La metodología utilizada incluyó SCRUM y UWE-UML para el desarrollo y modelado del sistema. En cuanto a las herramientas, se empleó el lenguaje de programación PHP (Versión 7.4.6), la base de datos MariaDB (Versión 10.1.21) y Bootstrap (Versión 4.5) para la parte de Front End. Este proyecto se llevó a cabo en la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

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

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

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

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

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

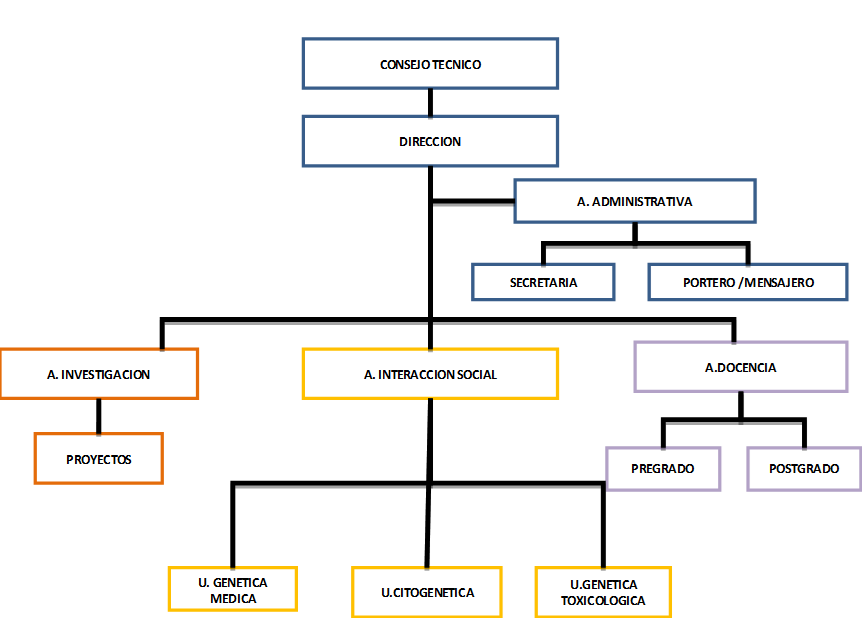
#### Antecedentes Institucionales

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

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

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

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



**Fuente:** Inst. Genética, 2022.

**Figura 1.0.1** Estructura Orgánica del Instituto de Genética.

La Unidad de Genética Médica tiene la responsabilidad de brindar atención y seguimiento a los pacientes, lo que la convierte en una parte fundamental que requiere acceso a la información contenida en las historias clínicas. Esta unidad está integrada en el marco del proyecto de Telemedicina, el cual se encuentra altamente activo en la actualidad, desarrollando proyectos destinados a beneficiar a la población.

#### Lista de Problemas

La gestión de historias clínicas en el Instituto de Genética de la Universidad Mayor de San Andrés se ha visto enfrentada a diversos problemas que afectan la eficiencia y la calidad del servicio. La falta de un sistema informático adecuado ha llevado a un proceso manual de registro de datos de pacientes (anexo 1 y anexo 2), almacenado en archivos Excel. Además, la ausencia de un formato estandarizado para las historias clínicas y la falta de un sistema centralizado han generado dificultades en el acceso y la gestión de la información de los pacientes. Incluso, en un intento previo de solucionar este problema, se implementó una base de datos en Access en 2012 (ver anexo 3), pero esta solución resultó ineficaz y se volvió al sistema anterior basado en formularios impresos y registros en archivos Word y Excel.

A continuación, presentamos los problemas identificados:

* 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 información de las historias clínicas tradicionales no está fácilmente accesible para el personal médico.
* 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.

### Planteamiento del Problema

El Instituto de Genética se enfrenta a un problema significativo relacionado con la gestión de Historias Clínicas. Actualmente, el proceso es altamente deficiente, ya que se basa en el uso de formatos físicos y archivos Excel para el registro de datos. Esto genera una serie de problemas, como la pérdida de información, conflictos internos y limitaciones en el acceso a los datos.

### Formulación del Problema

Teniendo en cuenta lo descrito anteriormente surge la siguiente pregunta:

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

## Objetivos

### Objetivo General

Desarrollar un sistema web de administración de historias clínicas para la unidad de genética medica del Instituto de Genética.

### Objetivos Específicos

* Analizar los procedimientos actuales de gestión de historias clínicas en la unidad de genética médica para identificar deficiencias y áreas críticas.
* Diseñar un sistema web para la gestión/administración de historias clínicas en la unidad de genética médica, centrándose en la accesibilidad, seguridad y confidencialidad de la información.
* Establecer un sistema de autentificación de usuarios con roles definidos que reflejen las necesidades específicas de acceso al sistema.
* Desarrollar de manera integral los módulos necesarios para el sistema de administración de historias clínicas, con el fin de garantizar un manejo eficiente y seguro de la información clínica de los pacientes.
* Implementar pruebas funcionales y de seguridad en el sistema de gestión de historias clínicas para garantizar su eficacia, confiabilidad y protección de datos.

## Justificación

### 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 implica beneficios significativos. En primer lugar, se optimiza el tiempo, lo que se traduce en una atención más ágil y precisa para los pacientes. La gestión eficaz de las historias clínicas mejora la atención al permitir un acceso más completo a la información de los pacientes. Esto también reduce los conflictos generados por datos anexos de diferentes unidades, garantizando una atención más coherente. Además, simplifica las tareas, aligerando la carga de trabajo del personal médico y administrativo y permitiéndoles concentrarse en brindar atención de calidad.

### Justificación Económica

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

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

### Justificación Tecnológica

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

## Alcances Y Límites

### Alcance

Es necesario delimitar los alcances del proyecto, para lo cual se debe especificar el área de acción que se cubrirá:

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

### Límites

Los límites del presente proyecto se indican a continuación:

* El proyecto llevará a cabo el análisis, diseño y desarrollo de los módulos de acuerdo con los requerimientos de la institución.
* Las actividades económicas y estadísticas no se considerarán, ya que el enfoque se centra únicamente en la información del paciente y los procesos relacionados con la historia clínica.
* El acceso al sistema estará restringido solo al personal autorizado, excluyendo a los pacientes.
* El sistema será desarrollado y utilizado por la unidad de Genética Medica, sin considerar su aplicación en otras unidades del Instituto.

.

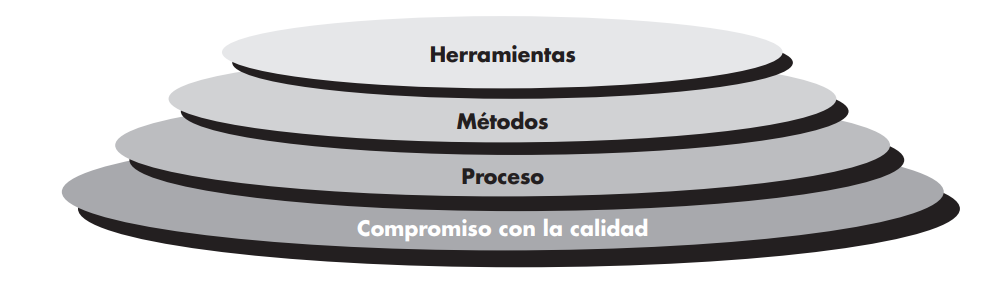
# CAPÍTULO II MARCO TEÓRICO



## Ingeniería de Software

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

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

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

**Figura 2.1** Capas de la ingeniería de software

**Fuente:** Pressman, 2010

### El Proceso de Software

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

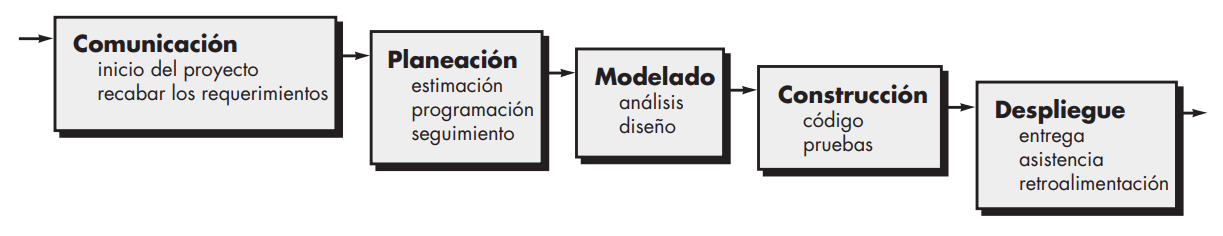
Pressman (2010) identifica una estructura de proceso general para la ingeniería de software, la cual consta de cinco actividades: **comunicación, planificación, modelado, construcción y despliegue**.

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

### Modelos de Desarrollo de Software

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

#### Modelo en Cascada

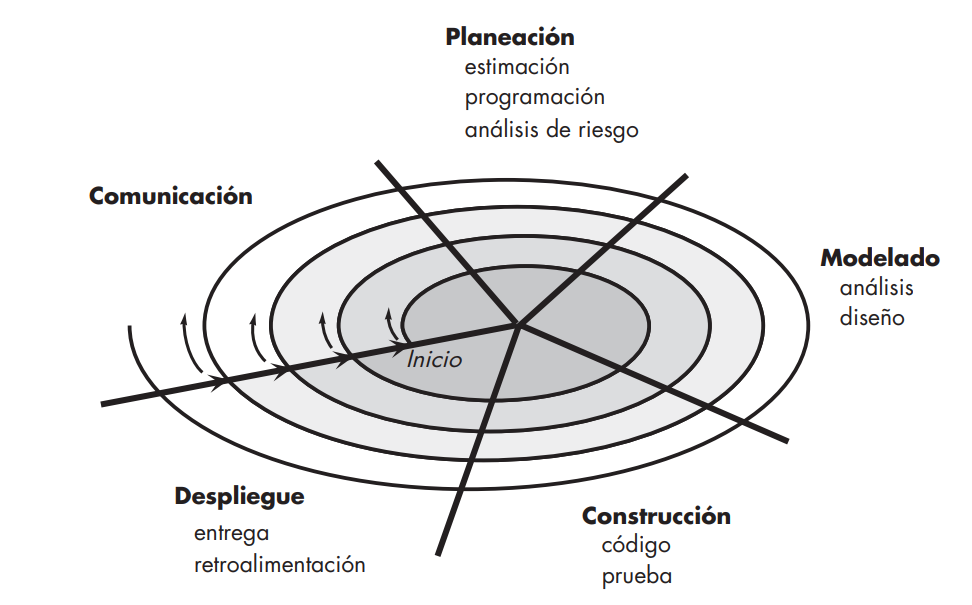
Pressman (2010) nos indica que, el modelo en cascada, a veces llamado ciclo de vida clásico, sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción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

**Figura 2.2** Modelo en Cascada  
**Fuente:** Pressman, 2010

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

#### Modelo en Espiral

(Pressman, 2010) nos indica que, el modelo espiral es un modelo evolutivo del proceso de desarrollo de software que combina la naturaleza iterativa de los prototipos con los aspectos controlados y sistemáticos del modelo en cascada. Este modelo permite un desarrollo rápido de versiones cada vez más completas.

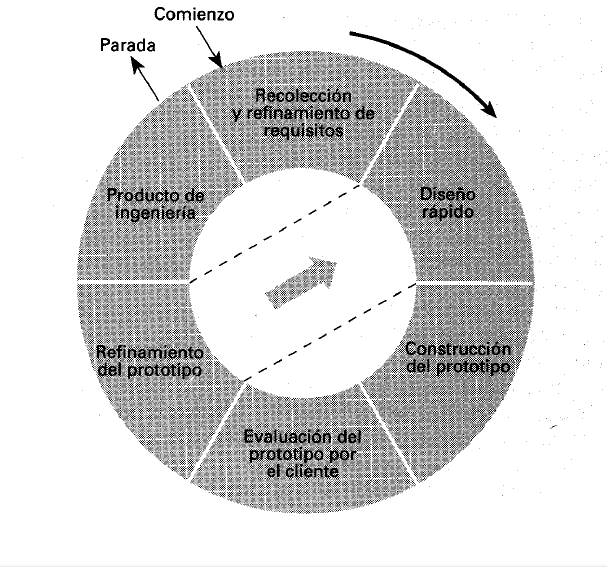
De acuerdo con Boehm (como se cita en Pressman, 2010), el modelo de desarrollo en espiral es un generador de procesos que se enfoca en la gestión de riesgos y guía la ingeniería concurrente en sistemas intensivos en software. Tiene dos características distintivas principales. La primera es el enfoque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

**Figura 2.3** Modelo en Espiral  
**Fuente:** Pressman, 2010

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

#### Modelo de Prototipos

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

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

**Figura 2.4** Modelo de Prototipo  
**Fuente:** EcuRed, s. f.

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

### Metodología Ágil en el Desarrollo de Software

La Metodología Ágil en el Desarrollo de Software es un enfoque altamente flexible y colaborativo para la creación de software que ha ganado amplia aceptación en la industria. Se diferencia de los modelos de desarrollo tradicionales, como el Modelo en Cascada, al enfocarse en la adaptabilidad, la colaboración constante y la respuesta ágil al cambio. La metodología ágil promueve la entrega continua de software funcional y valora la comunicación constante con los clientes y los miembros del equipo. Uno de los manifiestos más conocidos de la Metodología Ágil es el "Manifiesto Ágil" (Beck et al. 2001), que destaca valores clave, como la satisfacción del cliente a través de la entrega temprana y continua de software útil, y la bienvenida al cambio incluso en etapas tardías del desarrollo. Este enfoque se traduce en ciclos de desarrollo cortos, prototipado constante y la capacidad de responder rápidamente a las necesidades cambiantes del cliente.

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

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

#### Conceptos clave de metodologías ágiles de desarrollo.

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

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

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

#### 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.

## Proceso Unificado Ágil (AUP).

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

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

### Principios del Proceso Unificado Ágil (AUP).

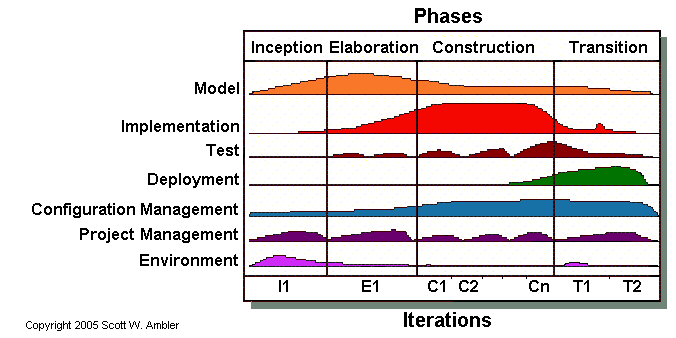
AUP se fundamenta en los siguientes principios, de acuerdo con EcuRed (s. f.):

* **Simplicidad**: Todo se describe concisamente utilizando poca documentación, no miles de ellas.
* **Agilidad**: El ajuste a los valores y principios de La Alianza Ágil.
* **Centrarse en actividades de alto valor**: La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.
* **Herramienta de la independencia**: Usted puede usar cualquier conjunto de herramientas que desea con el AUP. Se sugiere utilizar las herramientas más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.
* **Usted querrá adaptar este producto para satisfacer sus propias necesidades**: La metodología AUP es un producto de fácil uso utilizando cualquier herramienta. No es necesario comprar una herramienta especial, o tomar un curso, para adaptar esta metodología.

Estos conceptos se resumen en:

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

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

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

**Figura** **2.5** Ciclo de vida de AUP  
**Fuente:** Ambler, 2006

### Fases e Hitos del Ciclo de Vida.

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

#### Fase de Inicio.

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

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

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

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

#### Fase de Elaboración.

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

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

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

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

#### Fase de Construcción

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

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

Para salir de la fase de construcción, se debe superar el hito de **“Capacidad operativa inicial”** (IOC por sus siglas en inglés). El problema principal aquí es si la versión actual del sistema está lista para pasar a su entorno de prueba de preproducción para las pruebas de aceptación y del sistema. Si el equipo supera este hito, el proyecto pasa a la fase de “Transición”, de lo contrario, puede ser redireccionado o cancelado.

#### Fase de Transición

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

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

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

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

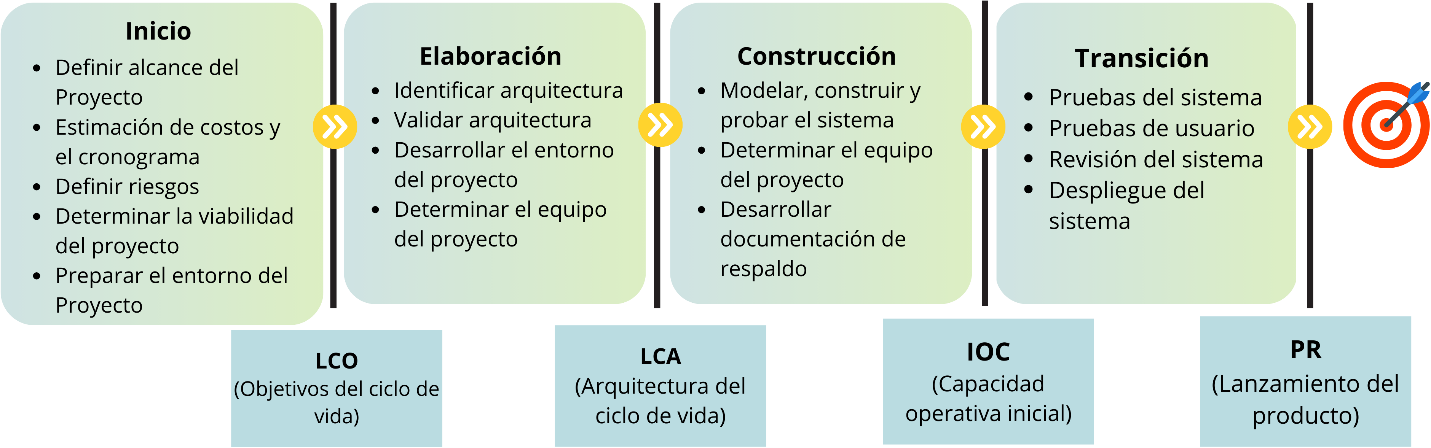
### Disciplinas

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

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

### Hitos

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

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

**Figura 2.6** Fases e Hitos de AUP  
**Fuente:** Ambler, 2006

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

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

### Ventajas y desventajas de la implementación.

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

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

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

## Ingeniería Web

(Escribano Santamarina, 2010) La ingeniería web es una nueva área de la ingeniería del software que abarca procesos, técnicas y modelos orientados a los entornos Web. Consiste en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad.

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

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

### La Ingeniería Web Dirigida por Modelos

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

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

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

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

### Metodologías en Ingeniería Web

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

(Escribano Santamarina, 2010) destaca las siguientes metodologías:

* **Web Services Distributed Management (WSDM):** Es una especificación basada en servicios web para gestionar y monitorizar el estado de otros servicios. Es un estándar OASIS (Organization for the Advancement of Structured Information Standards), y WSDM consiste en dos especificaciones:
* Management Using Web Services (MUWS): define como representar y como acceder a las interfaces de gestión de recursos expuestos como servicios web. Define un conjunto básico de operaciones de gestión sobre los servicios, tales como identificación, métricas, configuración y relaciones, además de un formato de eventos estándar.
* Management Of Web Services (MOWS): define como manejar servicios web como recursos y como describir y acceder a las capacidades de gestión utilizando MUWS. Esta especificación permite a las aplicaciones de gestión de servicios web interoperar entre sí.
* **WebML (Web Modeling Language):** Es una metodología de modelado visual de aplicaciones web, centrada especialmente en las aplicaciones de uso intensivo de datos, separando el contenido de la información en páginas, navegación y presentación, que se pueden definir y desarrollar de forma independiente. Permite la especificación de operaciones de manipulación de datos para actualizar la aplicación. Cuenta con cuatro perspectivas: el Modelo Estructural (de los datos de la aplicación), el Modelo de Hipertexto (para cada hipertexto describe qué páginas lo componen, y cómo navegan), el Modelo de Presentación (disposición y apariencia gráfica), y el Modelo de Personalización (para definir operaciones específicas para usuarios ó grupos de usuarios, ya que se almacenan como entidades en el Modelo Estructural). Dispone de una herramienta CASE (WebRatio).
* **OOWS:** Es una extensión del método OO-Method (ya basado en modelos), al cual se les añaden a las técnicas de modelado conceptual, capacidades de expresar la hipermedia inherente a las aplicaciones web. Consta de cinco modelos, que se especifican a continuación:
  + El Modelo de Objetos define la estructura y las relaciones estáticas entre clases identificadas en el dominio del problema.
  + En el Modelo Dinámico se describen las posibles secuencias de servicios y los aspectos relacionados con la interacción entre objetos.
  + El Modelo Funcional captura la semántica asociada a los cambios de estado entre los objetos motivados por la ocurrencia de eventos o servicios.
  + El Modelo de Navegación define la semántica navegacional asociada a las clases de los objetos del modelo.
  + El Modelo de Presentación captura los requisitos básicos de presentación de información. Está fuertemente basado en el modelo de navegación y permite definir, de una manera abstracta, la estructura lógica de presentación de los objetos navegacionales en la interfaz de usuario.
* **NDT (Navigational Development Techniques):** Es una metodología orientada a la Ingeniería Dirigida por Modelos (MDE). Aunque en un principio se centraba en las fases de ingeniería de requisitos y análisis, se ha ido ampliando a otras fases del ciclo de vida. Define un conjunto de metamodelos para las fases de requisitos y análisis, y una serie de transformaciones y reglas que permiten obtener los modelos de análisis a partir de ellos. Los metamodelos se representan a partir de MOF (MetaObject Facility), mientras que las transformaciones se definen mediante QVT (Query/View/Transformation). Consta de una herramienta denominada NDT Suite

## UWE (UML-Based Web Engineering).

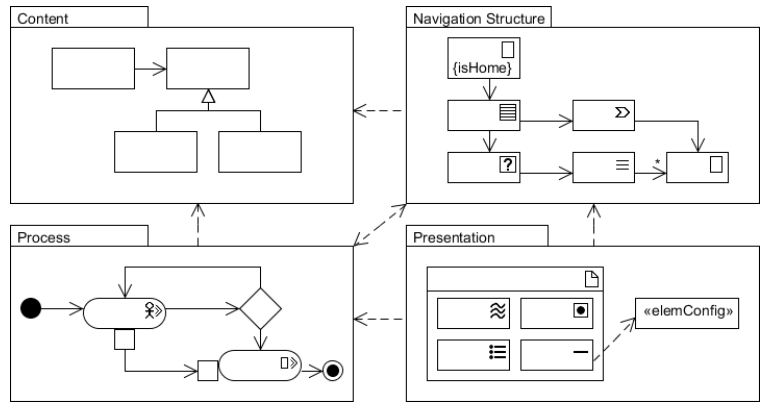
(Nieves-Guerrero et al., 2014) UWE es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Language) para sus modelos y sus métodos, lo que facilita la transición. La metodología define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

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

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

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

(Rotta et al., 2016) UWE utiliza notación y diagramas UML para el análisis y diseño de aplicaciones Web. UWE cubre aspectos de navegación, presentación, procesos de negocios, así como también, aspectos de adaptación. En la figura 2.7 se muestran las dependencias que existe entre los distintos tipos de modelos en UWE.



**Figura 2.7** Dependencia entre modelos.  
**Fuente:** Nora Koch, 2016.

### Modelos - Fases de UWE.

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

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

Las fases, modelos o etapas a utilizar son:

#### Fase de Requerimientos.

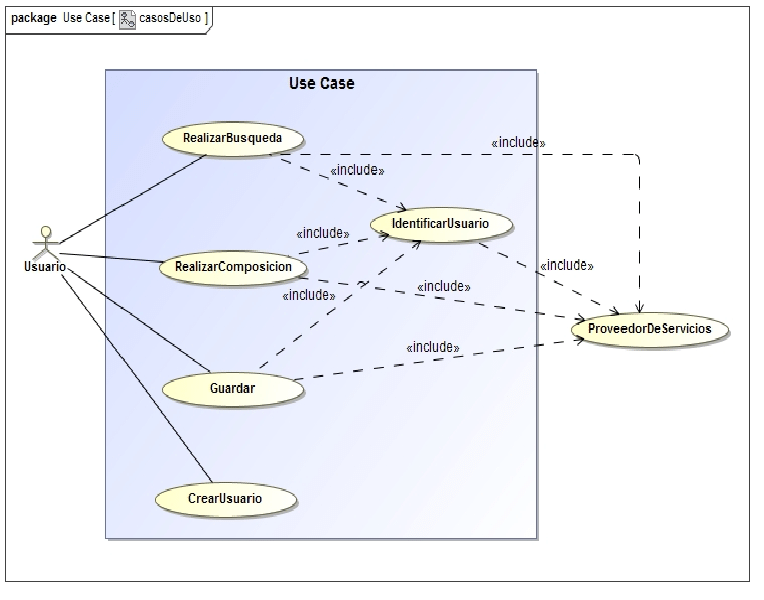
(Busch, 2016) El modelo de requisitos (a veces también llamado modelo de casos de uso) define la funcionalidad de una aplicación y los actores que la utilizan a un alto nivel de abstracción. Para ello se pueden utilizar diagramas de casos de uso.

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

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

(Galiano, 2012) En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

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



#### 

**Figura 2.8** Modelo de Requerimientos.  
**Fuente:** Nieves-Guerrero et al., 2014

#### Modelo de Contenido.

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

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

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

#### Modelo de Contenido.  Modelo de Navegación.

**Figura 2.9** Modelo de Contenido.

**Fuente:** Nieves-Guerrero et al., 2014

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

(Galiano, 2012) Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. El primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionarán.

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

#### Modelo de Presentación.

**Figura 2.10** Modelo de Navegación

**Fuente:** Nora Koch, 2016.

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

(Galiano, 2012) Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

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

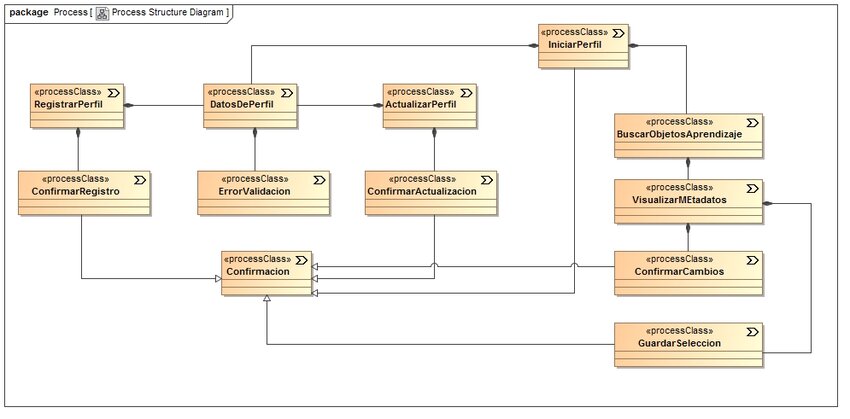
#### Modelo de Proceso

**Figura 2.11** Modelo de Presentación

**Fuente:** Nieves-Guerrero et al., 2014

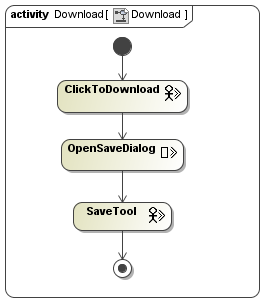
(Rotta et al., 2016) Representa los aspectos dinámicos de la aplicación Web y especifica funcionalidad cómo transacciones y workflows de actividades. Se modela mediante un diagrama de actividades de UML, y es resultado de refinar el diagrama de actividades modelado durante la especificación de requerimientos. Muestra el flujo de la ejecución representado por nodos de actividad conectados, los nodos de control que proveen constructores de flujo de control como decisiones y sincronización y nodos de objetos que representan el flujo de datos.

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

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

**Figura 2.12** Estructura del Proceso

**Fuente:** Nieves-Guerrero et al., 2014

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

**Figura 2.13** Flujo del Proceso

**Fuente:** Nora Koch, 2016

#### Relación entre Modelos.

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

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

## Modelos de UWE.  Historia Clínica

**Figura 2.14** Relación entre modelos

**Fuente:** Nieves-Guerrero et al., 2014

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

Según Laín Entralgo (1978, citado en Sarsanedas Castellano, s. f.) la historia clínica es el documento fundamental donde se recoge la información confiada por el enfermo al médico, para obtener el diagnóstico el tratamiento y la posible curación de la enfermedad.

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

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

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

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

### Características de una Historia Clínica.

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

* **Su práctica es obligatoria.** Ningún acto médico hospitalario o de consultorio debe efectuarse sin su correspondiente registro en la historia clínica. En las instituciones de salud se exige la historia clínica como elemento indispensable para ejercer una medicina de calidad. Por otro lado, en caso de complicaciones (salvo en algunos casos de extrema urgencia y corto tiempo disponible), su ausencia no tiene excusa.
* **Es irreemplazable.** La escritura de la historia no puede ser reemplazada por la memoria del médico. Es lógico que no se puedan conocer detalles de cada paciente, ni por el número ni por la complejidad individual de cada ser humano.
* **Es privada y pertenece al paciente.** Aquí se integran los conceptos de confidencialidad, secreto profesional e información.

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

* **Confidencialidad.** El secreto médico es uno de los deberes principales del ejercicio médico cuyo origen se remonta a los tiempos más remotos de la historia y que hoy mantiene toda su vigencia. La obligación de secretos es uno de los temas del derecho sanitario que más preocupa dada la creciente dificultad de su mantenimiento, el secreto no es absoluto, en la práctica médica pueden surgir situaciones de conflicto entre el deber de secreto y el principio de beneficencia del médico, fuerte protección legal del derecho a la intimidad. El secreto médico, la confidencialidad e intimidad y la historia clínica, son tres cuestiones que se implican recíprocamente y se relacionan. La historia clínica es el soporte documental biográfico de la asistencia sanitaria administrada a un paciente, por lo que es el documento más privado que existe en una persona.
* **Seguridad.** Debe constar la identificación del paciente, así como de los facultativos y personal sanitario que intervienen a lo largo del proceso asistencial.
* **Disponibilidad.** Aunque debe preservarse la confidencialidad y la intimidad de los datos en ella reflejada, debe ser así mismo un documento disponible, facilitándose en los casos legalmente contemplados, su acceso y disponibilidad.
* **Es Única.** La historia clínica debe ser única para cada paciente por la importancia de cara a los beneficios que ocasiona al paciente la labor asistencial y la gestión y economía sanitaria.
* **Es Legible.** Una historia clínica mal ordenada y difícilmente inteligible perjudica a todos, a los médicos, porque dificulta su labor asistencial y a los pacientes por los errores que pueden derivarse de una inadecuada interpretación de los datos contenidos en la historia clínica.

### Gestión de Historias Clínicas.

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

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

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

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

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

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

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

* Todos los datos son accesibles desde un solo lugar
* Se cumple con la ley de protección de datos
* Personalizar las fichas de historia clínica
* Seguimiento personalizado
* Conexión con otros módulos
* Alto nivel de escalabilidad

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

### Historia Clínica Electrónica (HCE).

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

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

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

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

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

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

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

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

# CAPÍTULO III MARCO APLICATIVO



## Introducción

En el presente capítulo se realizará el análisis y diseño para el proyecto mencionado, utilizando haciendo uso de los instrumentos métodos, técnicas, y herramientas mencionadas en el capítulo anterior. Para el desarrollo del mismo haremos uso de la metodología AUP en sus fases (iniciación, elaboración, construcción y transición). Siguiendo las fases del modelado UML, y así mismo de la ingeniería de software.

|  |  |
| --- | --- |
| **FASE** | **APLICACIÓN** |
| Iniciación | Análisis de los requerimientos, definir la arquitectura en base a los requerimientos del cliente |
| Elaboración | Definir requerimientos funcionales.  Definir casos de uso.  Definir los modelos UML. |
| Construcción | Codificación del software  Implementar la solución.  Pruebas del desarrollador. |
| Transición | Liberar el sistema. |

**Tabla 3.1** *Fases De Implementación AUP   
Elaboración propia*

Las fases de AUP serán desglosados iterativamente. Mientras que los modelos de UML se integrarán en la fase de elaboración de AUP, de esta manera AUP y UML realizar un proceso en el que se podrán integrar.

## Fase de Inicio

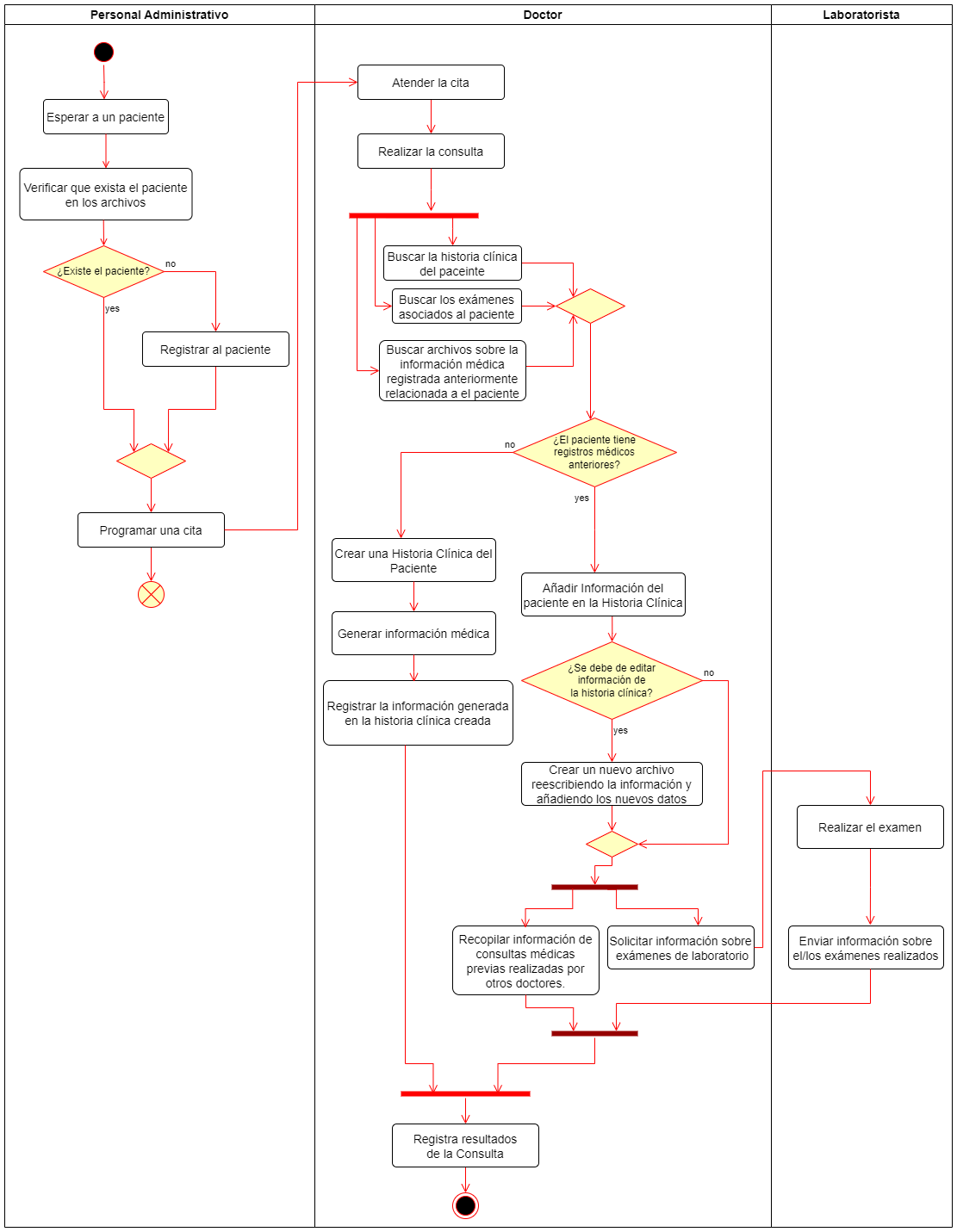
La fase inicial tiene como propósito identificar los requerimientos, creando historias de usuario que se convertirán luego en requisitos para el desarrollo del software. Esta etapa es breve y se enfoca en las actividades de modelado empresarial y sus requerimientos a través del modelo de casos de uso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

* **Entrevista**: Entrevistas directas con la directora del Instituto de Genética y el médico a cargo de la unidad de genética médica.
* **Observación**: Observación de los procesos que el personal realiza para la obtención, registro y acceso a la información de los datos clínicos de un paciente.
* **Documentación**: El instituto proporcionó documentación de anteriores gestiones, así como un archivo de Access que servía como una alternativa para almacenar información, aunque no era utilizado. Se obtuvieron modelos de historias clínicas de distintas administraciones, además de archivos de Excel con datos de pacientes.

### Situación actual

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

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



**Diagrama 3.1** Diagrama de Actividades del flujo actual del sistema

### Especificación de Requerimientos

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

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

#### Requerimientos Funcionales

**Tabla 3.1** Tabla de Requerimientos Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requerimiento** | **Descripción** | **Prioridad** |
| RF1 | El administrador podrá crear y gestionar cuentas de usuario dentro del sistema. | El sistema deberá permitirá el registro de nuevos usuarios según el rol | 5 |
| RF2 | Cada usuario podrá iniciar sesión con un correo válido. | El sistema permitirá el inicio de sesión de usuarios únicamente a través de una dirección de correo electrónico y contraseña que haya sido validada y registrada previamente | 5 |
| RF3 | El personal administrativo podrá registrar un paciente | El sistema permitirá el registro de paciente | 5 |
| RF4 | El personal administrativo podrá modificar los datos esenciales de un paciente, además de poder buscar pacientes utilizando su número de carnet de identidad. | El sistema permitirá editar solo la información de los datos personales del paciente, además de facilitar la búsqueda mediante el carnet de identidad. | 4 |
| RF5 | El personal administrativo podrá agendar una cita. | El sistema posibilitará el registro de una cita, lo cual incluirá la selección de un doctor y la elección de una fecha para dicha cita. | 5 |
| RF6 | El doctor podrá ver a los pacientes registrados, además de buscarlos por su carnet de identidad. | El sistema podrá listar a los pacientes, además de facilitar la búsqueda mediante el carnet de identidad. | 4 |
| RF7 | El doctor podrá visualizar la historia clínica de un paciente. | El sistema permitirá la visualización de la información de la historia clínica de un paciente. | 4 |
| RF8 | El doctor podrá visualizar las citas que debe de atender. | El sistema permitirá la visualización de las citas. | 4 |
| RF9 | El doctor podrá atender una cita, es decir, llevar a cabo una consulta médica. Además de editar o crear la historia clínica en el proceso de dicha consulta | El sistema permitirá el registro de una consulta en la historia clínica, incluyendo la capacidad de actualizar la información de la misma o crearla si así fuera el caso. | 5 |
| RF10 | El laboratorista podrá registrar la descripción de los exámenes realizados a un paciente en su historia clínica. | El sistema permitirá registrar descripciones de exámenes en la historia clínica, brindando la posibilidad de editar dichas descripciones. | 5 |
| RF11 | El laboratorista y el personal administrativo podrán visualizar la historia clínica de un paciente. | El sistema permitirá visualizar la información de una historia clínica | 4 |
| RF12 | Generación de reportes. | Los datos registrados en la historia clínica podrán ser impresos. | 4 |

#### Requerimientos no Funcionales

**Tabla 3.2** Tabla de Requerimientos no funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requerimientos** | **Descripción** | **Prioridad** |
| RFN1 | USABILIDAD | Debe ser fácil de usar, con la ayuda de interfaces intuitivas. | 5 |
| RFN2 | SEGURIDAD | El ingreso y visualización del sistema estará restringido bajo autenticación y autorización por roles  Seguridad  Todos los datos transferidos entre el cliente y el servidor deberán estar cifrados utilizando TLS/SSL.  Las contraseñas de los usuarios deberán almacenarse de forma segura usando algoritmos de hash modernos y salting.  El acceso a historias clínicas estará restringido solo a personal autorizado. | 5 |
| RFN3 | EFICIENCIA | El almacenamiento de datos debe ser eficiente para asegurar que las historias clínicas se guarden y recuperen de manera rápida y segura. | 5 |
| RFN4 | PORTABILIDAD | El sistema debe brindar comodidad al usuario y a las demás áreas que necesitan acceder al sistema. Además de funcionar en toda la infraestructura que cuenta el Instituto de Genética. | 5 |

## Fase de Elaboración

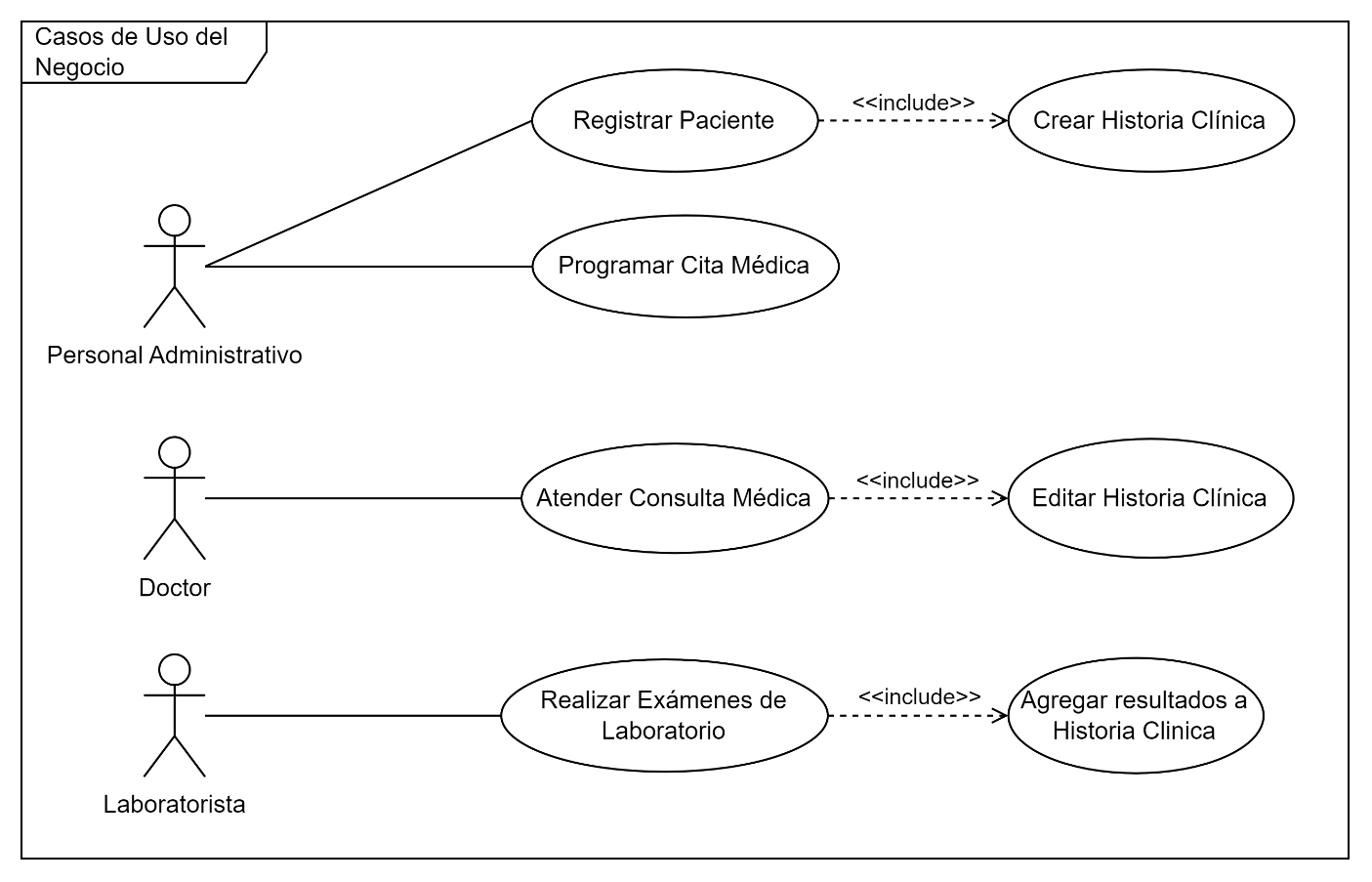
### Descripción de los Actores

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

**Tabla 3.3** Descripción de los Actores

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

### Modelo de Casos de Uso del Negocio

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

**Diagrama 3.2** Casos de uso del negocio

#### Descripción de los casos de uso del negocio

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

**Tabla 3.4** Descripción de caso de uso del negocio - Registra Paciente

|  |  |
| --- | --- |
| **CASO DE USO:** | REGISTRAR PACIENTE |
| **ACTORES:** | Personal Administrativo |
| **TIPO:** | Primario |
| **DESCRIPCIÓN:** | El personal administrativo verifica si el paciente está registrado.  Si no está registrado, se procede a registrar al paciente, creando su historia clínica con los datos personales del nuevo paciente.  Si el paciente ya está registrado, se omite el registro. |

**Tabla 3.5** Descripción de caso de uso del negocio - Programar Cita Médica

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

**Tabla 3.6** Descripción de caso de uso del negocio - Atender Consulta Médica

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

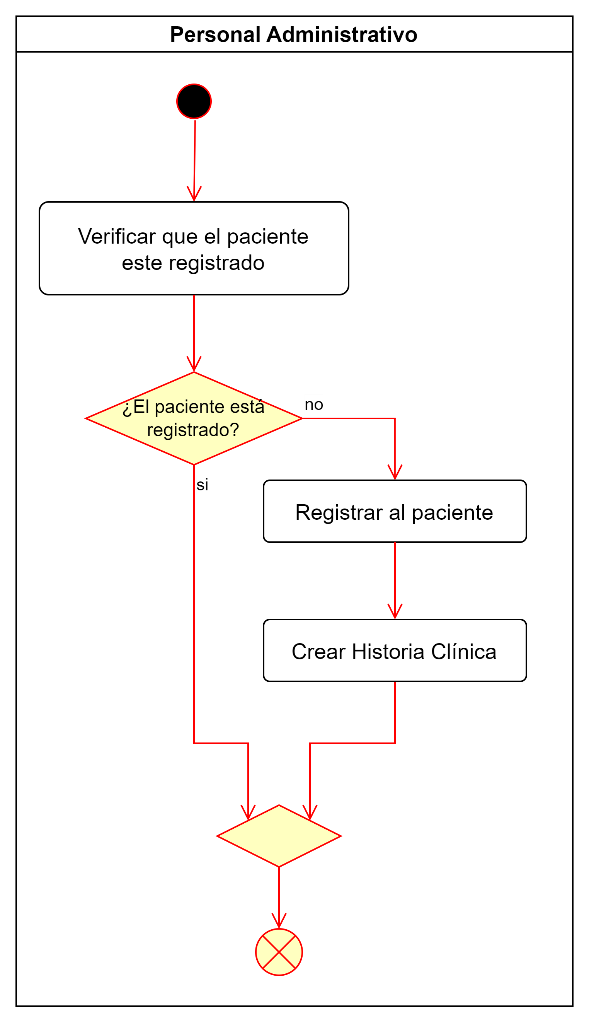
**Tabla 3.7** Descripción de caso de uso del negocio - Realizar Exámenes de Laboratorio

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

### Modelo de objetos del negocio

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

#### Registrar Paciente

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

**Diagrama 3.3** Registrar Paciente

#### Programar Cita Médica

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

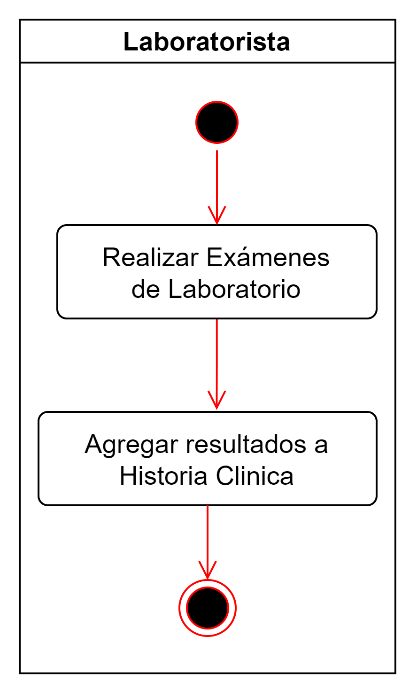
#### Atender Consulta Médica

**Diagrama 3.4** Programar Cita Médica

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

#### Realizar Exámenes de Laboratorio

**Diagrama 3.5** Atender Consulta Médica

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

**Diagrama 3.6** Realizar Exámenes de Laboratorio

## Fase de Construcción

### Casos de uso del Sistema

El modelo de casos de uso del sistema describe las funcionalidades y los actores que interactúan con el sistema. Se suele representar a través de los Diagramas de Casos de Uso.

# CAPÍTULO IV CALIDAD Y SEGURIDAD

## Introducción

## Norma ISO/IEC 25010

(ISO 25010, s. f.) El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.

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

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

Las características y subcaracterísticas proporcionan una terminología coherente para especificar, medir y evaluar la calidad de los sistemas y productos informáticos. También proporcionan un conjunto de características de calidad con las que se pueden comparar los requisitos de calidad establecidos para comprobar su integridad.

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

**Figura 4.1** Norma ISO/IEC 25010  
**Fuente:** ISO 25010, s. f.

### Características de la norma ISO/IEC 25010

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

* **Adecuación Funcional:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.
* **Eficiencia de desempeño:** Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.
* **Compatibilidad:** Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.
* **Usabilidad:** Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.
* **Fiabilidad:** Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.
* **Seguridad:** Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.
* **Mantenibilidad:** Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
* **Portabilidad:** Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

# CAPÍTULO V COSTO BENEFICIO

## Introducción

## Modelo COCOMO II

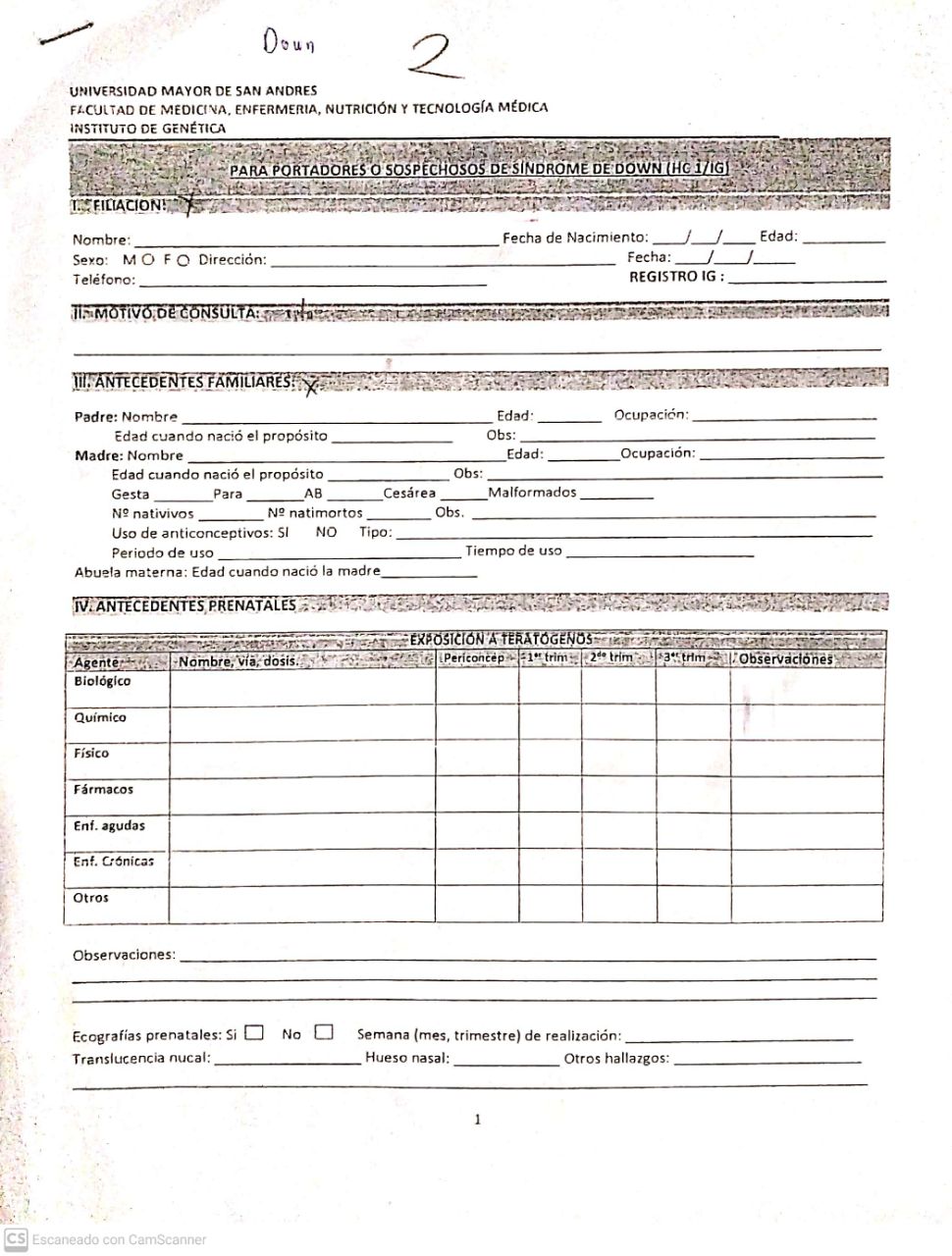
(Boehm, 2000) El modelo COCOMO II es una herramienta de estimación de costos de software desarrollada por Barry Boehm en la década de 1990. Este modelo se basa en los principios del modelo original COCOMO, pero con mejoras significativas. El Modelo COCOMO II se utiliza para estimar el costo, el esfuerzo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

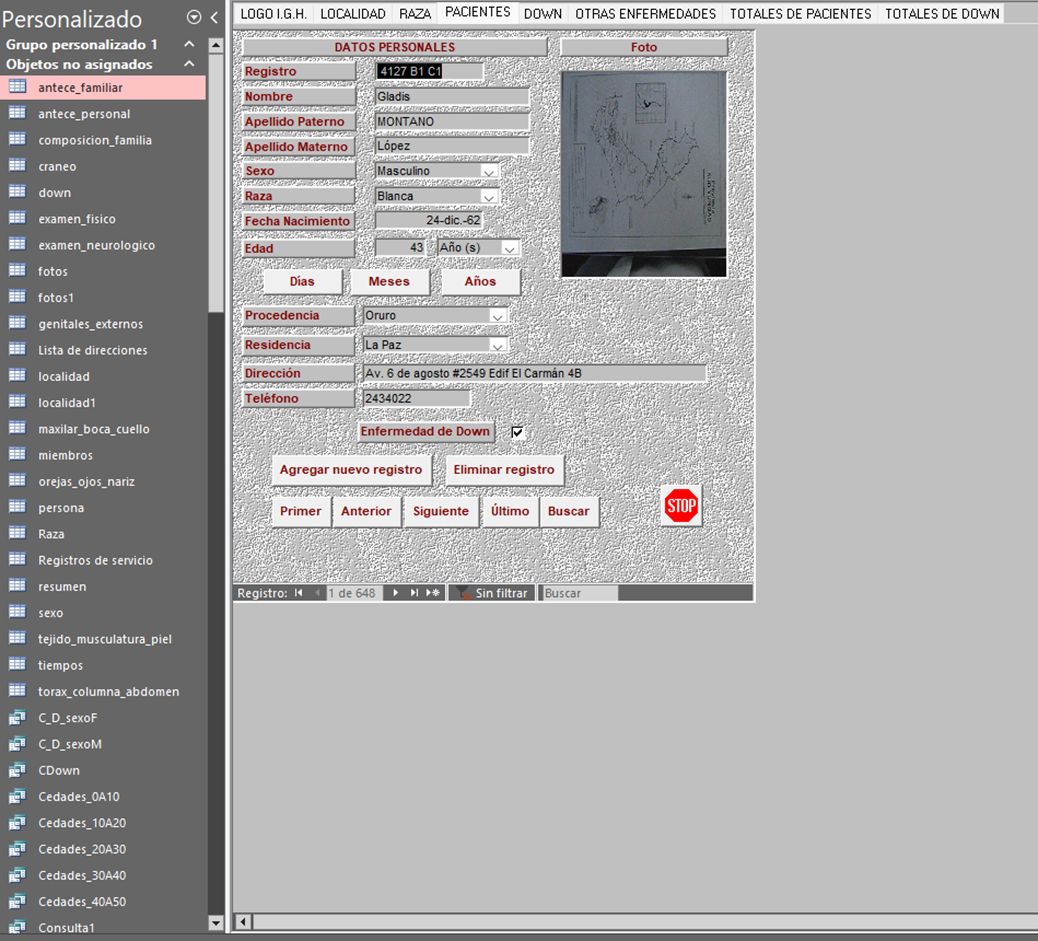
(Migani, 2014) Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

* Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
* Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
* Implementar una herramienta de software que soportara el modelo.
* Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

Cocomo II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura. Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores descriptos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información.

# ANEXOS





# REFERENCIAS BIBLIOGRÁFICAS

* World Health Organization. Regional Office for Europe. (2019, 10 de octubre). *Future of digital health systems: report on the WHO symposium on the future of digital health systems in the European region: Copenhagen, Denmark, 6–8 de Ferbrero de* 2019. <https://apps.who.int/iris/handle/10665/329032>
* 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>
* Ambler, S. (2002). The Agile Unified Process (AUP) Home Page. (s. f.). Ambysoft Inc. <http://www.ambysoft.com/unifiedprocess/agileUP.html> (Acceso: 30/06/2023)
* Nora Koch. (2016). ¿Qué es UWE? UWE - Tutorial. <https://uwe.pst.ifi.lmu.de/teachingTutorial.html> (Acceso: 30/06/2023)
* Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G. (2006). Object-Oriented Software Engineering. A Use Case Driven Approach. Pearson.
* Pressman, R. S. (2010). INGENIERIA DE SOFTWARE (7o edición revisada). McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V. A Subsidiary of The McGraw-Hill Companies, Inc.
* Admin. (2023, 28 marzo). Metodologías de desarrollo software con ejemplos. DiagramasUML.com. <https://diagramasuml.com/metodologias-de-desarrollo-software-con-ejemplos/> (Acceso: 29/09/2023)
* EcuRed. (s. f.). Modelo de prototipos - ECURED. Modelo de prototipos. <https://www.ecured.cu/Modelo_de_prototipos> (Acceso: 29/09/2023)
* Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. <http://agilemanifesto.org/principles.html>
* Schwaber, K., & Sutherland, J. (2017). The Scrum Guide.
* EcuRed. (n.d.). Agile Unified Process - ECURED. <https://www.ecured.cu/Agile_Unified_Process> (Acceso: 02/10/2023)
* Ambler, S. W. (2006). Agile UP Product. Ambysoft Inc. May 13, 2006. http://www.ambysoft.com/unifiedprocess/agileUP.html (Acceso: 13/08/2022)
* Escribano Santamarina, Á. (2010). Ingeniería web dirigida por modelos. Escuela Superior de Informática, Universidad de Castilla-La Mancha. <http://www.inf-cr.uclm.es/www/mpolo/asig/trabajos0910/Ingenieria_web_basada_en_modelos.pdf> (Acceso 30/10/2022).
* Pinzon, O. (2017). Ingeniería Web: Una Metodología para el Desarrollo de Aplicaciones Web Escalables y Sostenibles. LACCEI. <https://www.laccei.org/LACCEI2017-BocaRaton/student_Papers/SP277.pdf> (Acceso: 08/03/2023).
* Pressman, R. S. (2002). INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO (5o). Mcgraw-Hill/Interamericana de España, S. A. U. https://pdfcoffee.com/qdownload/ingenieria-del-software-5ta-edicion-roger-s-pressman-freelibrosorgpdf-6-pdf-free.html (Acceso: 08/03/2023)
* Vargas, M. S. (2017, 30 abril). INGENIERIA WEB GUIADA POR MODELOS. <http://marcelosalasvargas.blogspot.com/2017/04/ingenieria-web-guiada-por-modelos.html> (Acceso: 09/03/2023)
* Nieves-Guerrero, C., Ucán-Pech, J., & Domínguez, V. H. M. (2014). UWE en sistema de recomendación de objetos de aprendizaje. Aplicando Ingeniería Web: un método en caso de estudio. Revista latinoamericana de ingeniería de software, 2(3), 137. <https://doi.org/10.18294/relais.2014.137-143>
* Rotta, D. P., Pallotta, G. S., Klikailo, H. E., & Belloni, E. A. (2016). Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web. 19o Concurso de Trabajos Estudiantiles. Recuperado 9 de junio de 2023, de <http://sedici.unlp.edu.ar/bitstream/handle/10915/58143/Documento_completo.pdf-PDFA.pdf?sequence=1>
* Galiano, L. (2012, octubre). Metodología UWE aplicada a mi solución informática de mi proyecto. Recuperado 8 de julio de 2023, de <http://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>
* Busch, M. (2016). Evaluating & Engineering: an Approach for the Development of Secure Web Applications. Fakult¨at fur Mathematik, Informatik und Statistik ¨ der Ludwig-Maximilians-Universit¨at Munchen. <https://www.pst.ifi.lmu.de/~busch/thesisMarianneBusch.pdf>
* Luna Orosco, J., & Carrasco, M. (2008). Norma Técnica para el Manejo del Expediente Clínico. Dr. Javier Luna Orosco. <https://www.minsalud.gob.bo/images/Documentacion/dgss/Area_de_Calidad/64%20Norma%20expediente%20clinico.pdf>
* Sarsanedas Castellano, E. (s. f.). Sistema de información clínico: historia clínica. <https://openaccess.uoc.edu/bitstream/10609/139235/2/Documentación%20clínica_Módulo%202_Sistema%20de%20información%20clínico_%20historia%20clínica.pdf> (Acceso: 02/04/2023)
* 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>.
* Velito, A. S., & Tejada Soriano, S. R. (2010). La historia clínica como instrumento de calidad. <http://www.auditoriamedicahoy.com/biblioteca/La%20historia%20clínica%20como%20instrumento%20de%20calidad%20Tejada%20Velito.pdf> (Acceso: 05/04/2023)
* Chero-Farro, D., Cabanillas-Olivares, A., & Fernández-Mogollón, J. (2017). Historia clínica como herramienta para mejora del proceso de atención. Revista De Calidad Asistencial. <https://doi.org/10.1016/j.cali.2016.05.005>
* Torres Lloveras, P., & Sarsanedas Castellanos, E. (s. f.). La historia clínica (HC). <https://openaccess.uoc.edu/bitstream/10609/70507/5/Documentación%20clínica_Módulo%202_%20La%20historia%20clínica%20%28HC%29.pdf> (Acceso: 05/04/2023)
* Juliá, S. (2020, 23 septiembre). Cómo gestionar historias clínicas de pacientes. Nubimed. <https://www.nubimed.com/blog/como-gestionar-historias-clinicas-pacientes/> (Acceso: 05/04/2023)
* Davis, R. G. (2018, 12 septiembre). Gestión de historias clínicas en el área de infectología de un hospital estatal, Lima,2018. <https://repositorio.uwiener.edu.pe/handle/20.500.13053/2358> (Acceso: 05/04/2023)
* Chaudhry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., . . . Shekelle, P. G. (2006). Systematic review: Impact of health information technology on quality, efficiency, and costs of medical care. Annals of Internal Medicine, 144(10), 742-752. doi: 10.7326/0003-4819-144-10-200605160-00125 https://pubmed.ncbi.nlm.nih.gov/16702590/
* Zhou, L., Soran, C. S., Jenter, C. A., Volk, L. A., Orav, E. J., & Bates, D. W. (2015). The relationship. https://europepmc.org/article/pmc/pmc2705247
* Ghiglia, M. M. C. (2019). Historia clínica electrónica herramienta para la continuidad de asistencia. Revista médica del Uruguay. <https://doi.org/10.29193/rmu.35.3.6>
* 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
* Boehm, B. W. (2000). Software cost estimation with COCOMO II. Prentice-Hall PTR.
* Migani, S. (2014). UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE. unsj. <https://www.academia.edu/4853589/UN_MODELO_DE_ESTIMACION_DE_PROYECTOS_DE_SOFTWARE>
* ISO 25010. (s. f.). <https://iso25000.com/index.php/normas-iso-25000/iso-25010> (Acceso 22/10/2023)
* Britton, J. (2021, 6 mayo). What is ISO 25010? Perforce Software. <https://www.perforce.com/blog/qac/what-is-iso-25010#:~:text=ISO25010%20describes%20two%20quality%20models,a%20particular%20context%20of%20use>.
* 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 clinico. <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://esmirnasite.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 https://pubmed.ncbi.nlm.nih.gov/16702590/
* 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. https://europepmc.org/article/pmc/pmc2705247
* 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/portalIG/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). Tribalyte 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.