

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE
ACTIVIDADES ACADÉMICAS, INVESTIGACIÓN E INTERACCIÓN
SOCIAL PREGRADO Y POSGRADO.
CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD
PÚBLICA DE EL ALTO”**

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

Postulante:	Lizeth Gutierrez Aruquipa
Tutor Metodológico:	M. Sc. Lic. Ing. Marisol Arguedas Balladares
Tutor Especialista:	M. Sc. Lic. Ing. Juan Fernando Huanca Alavi
Tutor Revisor:	Lic. Ing. Yolanda Escobar Mancilla

EL ALTO – BOLIVIA

2024

DECLARACION JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Lizeth Gutierrez Aruquipa estudiante, con C.I. 6944386 LP mediante la presente declaro de manera pública que la propuesta del proyecto “SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE ACTIVIDADES ACADEMICAS, INVESTIGACIÓN E INTERACCIÓN SOCIAL PREGRADO Y POSGRADO “Es original, siendo el resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados.

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

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

De identificarse falsificación, plagio, fraude o que el PROYECTO DE GRADO haya sido publicado anteriormente, asumo las consecuencias y sanciones que mi acción se deriven, responsabilizándome por toda las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la carrera de ingeniería de sistemas de la universidad pública de el alto.

El Alto, noviembre del 2024

Lizeth Gutierrez Aruquipa
CI: 6944386 LP
gutierrezlizeth681@gmail.com

DEDICATORIA

El presente proyecto de grado va dedicado a mis padres, por su apoyo incondicional y amor infinito. Sin su confianza en mis sueños y su esfuerzo incansable, este logro no habría sido posible.

A mis tutores, por su guía, paciencia y conocimientos compartidos, que me permitieron crecer tanto académica como personalmente. Gracias por estar siempre dispuestos a ayudarme y por motivarme a dar lo mejor de mí.

AGRADECIMIENTO

En primer lugar, agradezco profundamente a mis padres, quienes siempre han sido mi mayor fuente de apoyo, tanto emocional como económico. Su amor incondicional, sacrificio y aliento constante me han permitido llegar hasta aquí. Gracias por siempre creer en mí y por darme la oportunidad de alcanzar mis sueños.

A la Lic. Ing. Marisol Arguedas Balladares, por su orientación, paciencia y profesionalismo. Su conocimiento y guía han sido cruciales en el desarrollo de este trabajo.

Al Ing. Juan Fernando Huanca Alavi, por Sus consejos y sugerencias han enriquecido mi investigación y me han impulsado a dar lo mejor de mí.

A la Lic. Ing. Yolanda escobar Mancilla, A por su dedicación y enseñanzas a lo largo de mi formación. Gracias por siempre estar dispuestos a compartir su sabiduría y por motivarme a seguir adelante en los momentos de incertidumbre.

INDICE DEL CONTENIDO

CAPITULO I.....	1
1.1. INTRODUCCION.....	1
1.2. ANTECEDENTES.....	2
1.2.1. Antecedentes Institucionales	2
1.2.2. Antecedentes internacionales	4
1.3. PLANTEAMIENTO DEL PROBLEMA.....	7
1.3.1. Problema general.....	7
1.3.2. Problemas específicos.....	7
1.3.3. Formulación del problema.....	8
1.4. OBJETIVOS	8
1.4.1. Objetivo general.....	8
1.4.2. Objetivos específicos	9
1.5. JUSTIFICACIÓN.....	9
1.5.1. Justificación Técnica.....	9
1.5.2. Justificación Económica.....	10
1.5.3. Justificación Social.....	10
1.6. METODOLOGIA.....	10
1.6.1. Metodología de ingeniería UWE (UML-Based Web Engineering)	11
1.6.2. Métricas de calidad	11
1.6.3. Control de seguridad ISO/IEC 27001.....	12
1.6.4. Estimación de costos – COCOMO II.....	12
1.6.5. Pruebas de software	13
1.7. HERRAMIENTAS	14
1.7.1. Lenguaje de programación	14

1.7.2. Framework - Angular	14
1.7.3. Gestor de Base de Datos.....	15
1.7.4. Diseño de Interfaces	15
1.8. LIMITES Y ALCANCES	15
1.8.1. Limites	15
1.8.2. Alcances	16
1.9. APORTES	16
CAPITULO II.....	17
2.1. INTRODUCCION.....	17
2.1.1. Sistema.....	17
2.1.2. Sistema Integrado.....	18
2.1.3. Actividad Académica.....	18
2.1.4. Investigación.....	19
2.1.5. Interacción Social.....	20
2.2. INGENIERIA DE REQUERIMIENTOS.....	20
2.1.6. Requerimientos funcionales.....	21
2.1.7. Requerimientos no funcionales.....	22
2.3. INGENIERIA DE SOFTWARE.....	22
2.4. UWE – UML – Based Web Engineering	23
2.1.8. Captura, análisis y especificación de requisitos	24
2.1.9. Diseño del Sistema	24
2.1.10. Codificación del sistema	18
2.1.11. Pruebas	18
2.1.12. Instalación o fase de implementación	19
2.1.13. Mantenimiento	19
2.5. LENGUAJE DE MODELO UNIFICADO UML	19
2.6. ARQUITECTURA CLIENTE/SERVIDOR.....	20

2.7. MÉTRICAS DE CALIDAD	21
2.1.14. Norma ISO/IEC 25000	21
2.8. SEGURIDAD DE SOFTWARE ISO-27001	38
2.9. PRUEBAS DE FUNCIONAMIENTO	39
2.9.4. Prueba de Caja blanca	39
2.9.5. Pruebas de ruta básica	40
2.9.6. Prueba de Caja negra.....	42
2.9.7. Pruebas de estrés.....	44
2.10. ESTIMACIÓN DE COSTO	45
2.11. HERRAMIENTAS	49
2.11.1. Lenguaje de programación	49
2.11.2. Base de datos	50
2.11.3. Framework.....	51
2.11.4. Editores de Código	54
2.11.5. Enterprise Architect.....	56
CAPITULO III.....	57
3.1. MARCO APLICATIVO.....	57
3.2. APLICACIONES DE LA METODOLOGIA UWE	57
3.2.1. Captura, análisis y especificaciones de requerimientos	57
CAPITULO IV	97
4. PRUEBAS DE EVALUACION Y RESULTADOS.....	97
4.1. METRICAS DE CALIDAD ISO 25000	97
4.1.1. Funcionalidad	97
4.1.2. Confiabilidad	103
4.1.3. Usabilidad	105
4.1.4. Eficiencia	107
4.1.5. Mantenibilidad.....	109

4.1.6. Portabilidad.....	110
4.1.7. Seguridad	112
4.2. PRUEBAS	114
4.2.1. Prueba de caja blanca	114
4.2.2. Prueba de caja negra.....	117
4.2.3. Prueba de estrés.....	117
4.2.4. Pruebas de accesibilidad:	118
CAPITULO V	119
5. ESTIMACION DE COSTO	119
5.1. COCOMO II	119
CAPITULO VI	124
6. CONCLUSION Y RECOMENDACIONES	124
6.1. CONCLUSIONES	124
6.2. RECOMENDACIONES.....	125
7. REFERENCIAS BIBLIOGRAFICAS	127

INDICE DE TABLAS

Tabla 1. Definición de métricas de calidad y fórmulas de la adecuación funcional	23
Tabla 2. Definición de métricas y fórmulas de la eficiencia de desempeño	24
Tabla 3. Definición de métricas y fórmulas de la compatibilidad	26
Tabla 4. Definición de métricas y fórmulas de usabilidad	27
Tabla 5. Definición de metricas y formualsa de la fiabilidad.....	29
Tabla 6. Definición de metricas y formulas de la seguridad	30
Tabla 7. Definición de métricas y fórmulas de la mantenibilidad	32
Tabla 8. Definicipon de etricas y formulas de la portabilidad	33
Tabla 9. ISO/IEC 25040- proceso parar evaluación de software	35
Tabla 10. Herramientas usadas para la ingeniería de requerimientos	57
Tabla 11. Requerimientos funcionales.....	58
Tabla 12. Requerimientos no funcionales.....	59
Tabla 13. Descripción de actores.....	60
Tabla 14. Descripción de caso de uso - Gestión de usuarios	63
Tabla 15. Gestión de permisos	65
Tabla 16. Descripcion de caso de uso- Gestion de informacion de pagina web	66
Tabla 17. Descripción de caso de uso-gestión de noticias	68
Tabla 18. Descripción de caso de uso-Gestión de asignaturas	70
Tabla 19. Descripcion de caso de uso-Gestion de malla curricular.....	71
Tabla 20. Descripcion de caso de uso-Gestion de autoridades	73
Tabla 21. Descripcion de caso de uso- Gestion de trabajos de investigacion.....	74
Tabla 22. Descripcion de caso de uso-Gestion de libros digitales	75
Tabla 23. Descripcion de caso de uso-Generacion de Reportes	77
Tabla 24. Numero de entradas de usuario.....	97
Tabla 25. Numero de salidas de usuario.....	98
Tabla 26. Número de peticiones de usuario.....	98
Tabla 27. Parametros de medición de punto función	99
Tabla 28. Punto funcion sin ajustar.....	100
Tabla 29. Valoración de porcentajes.....	101
Tabla 30. Factores de complejidad	101
Tabla 31. Parametros de medicion de usabilidad	105
Tabla 32. Usabilidad del sistema	106

Tabla 33. Escala de valores de eficiencia	107
Tabla 34. Valoracion de la eficiencia del sistema.....	108
Tabla 35. Valores para la mantenibilidad	110
Tabla 36. Resultados.....	112
Tabla 37. Caso de prueba de ingreso al sistema.....	117
Tabla 38. Coeficientes de a.b.c.d de COCOMO II	119
Tabla 39. Ecuacione de COCOMO II	120
Tabla 40. Calculo de atributos FAE	121
Tabla 41. Resumen de costo	123

INDICE DE FIGURAS

Figura 1. Organigrama.....	4
Figura 2. Metodología UWE.....	23
Figura 3. Caso de uso.....	25
Figura 4. Diagrama de Clases	25
Figura 5. Modelo de Navegación	17
Figura 6. Modelo de Presentación UWE	18
Figura 7. Sistema Cliente/Servidor	21
Figura 8. Familia de normas ISO/IEC 25000	22
Figura 9. Características de calidad ISO/IEC 25010	22
Figura 10. Prueba de caja blanca	40
Figura 11. Diagrama de flujo.....	41
Figura 12. Matriz de grado	42
Figura 13. Prueba de caja negra.....	43
Figura 14. Diagrama de caso de uso general del sistema	62
Figura 15. Diagrama de caso de uso-Gestion de usuario	63
Figura 16. Diagrama de caso de uso- Gestion de informacion para pagina web	66
Figura 17. Diagrama de caso de uso-Gesyion de Noticias	68
Figura 18. Diagrama de caso de uso- Gestion de asignatura	69
Figura 19. Diagrama de caso de uso-Gestion de Malla Curricular	71
Figura 20. Diagrama de caso de uso-gestion de trabajos de investigacion	72
Figura 21. Diagrama de caso de uso-Gestion de libros digitales	75
Figura 22. Diagrama de caso de uso-Generacion de reportes.....	76
Figura 23. Diseño conceptual	78
Figura 24. Modelo de navegacion - Pagina web	79
Figura 25. Modelo de navegacion -Administrador	80
Figura 26. Modelo de navegacion - Secretaria	81
Figura 27. Modelo de navegacion- Bibliotecaria	82
Figura 28. Modelo de navegacion - Docente	82
Figura 29. Prototipo de presentacion de la pagina principal.....	83
Figura 30. Prototipo de presentacion informacion instutucional	83
Figura 31. Prototipo de presentacion -asignaturas y contenido	84
Figura 32. Prototipo de presentacion biblioteca digital.....	84

Figura 33. Prototipo de presentacion de Inicio sesion.....	85
Figura 34. Prototipo de presentacion de inicio de mmenu	85
Figura 35. Prototipo de presentacion - gestion de usrios	86
Figura 36. Pagina principiapl	86
Figura 37. Pagina web-carrera.....	87
Figura 38. Pagina web-plan de estudio-materias	87
Figura 39. Pagina web-actividades academicas	88
Figura 40. Pagina web-Biblioteca digital	88
Figura 41. Pagnia web-mostrar libros digitales	89
Figura 42. Pagina web - inicio Sesion.....	89
Figura 43. Modulo principal - administrador	90
Figura 44. Gestion de asignaturas	90
Figura 45. Gestion de trabajos de investigacion	91
Figura 46. Busqueda de trabajos de investigacion.....	91
Figura 47. Gestion de libros digitales	92
Figura 48. Resgistro de nuevo libro	92
Figura 49. Gestion de noticias	93
Figura 50. Registro de nueva noticia	93
Figura 51. Gestion de datos institucionales	94
Figura 52. Gestion de usuarios	94
Figura 53. Otorgar permisos	95
Figura 54. Resgitro de nuevo usarop	95
Figura 55. Formato de resportes en pdf.....	96
Figura 56. Prueba caja blanca	115

RESUMEN

Los sistemas de información y las tecnologías de la información han transformado la forma en que operan las instituciones en la actualidad. Su uso permite mejorar el manejo de la información, automatizando procesos administrativos de forma más eficiente y ágil. La carrera de Medicina de la Universidad Pública de El Alto se encuentra en un proceso de acreditación internacional. En este contexto, las nuevas tecnologías son un pilar fundamental para el tratamiento de la información. Por esta razón, se requiere una herramienta que facilite el control de las actividades académicas, investigación e interacción social. Esta herramienta será de gran ayuda para tener un registro de todas las actividades de la Carrera de Medicina. El proyecto actual consiste en diseñar y desarrollar un sistema que permita la centralización de toda la información relevante de las actividades, así mismo la creación de repositorio de trabajos de investigación y libros digitales. Estos datos para la toma de decisiones de la carrera. El sistema se desarrollará bajo la metodología UWE, que garantiza una buena organización del flujo de trabajo y el cumplimiento de los plazos de entrega. Para el desarrollo se utilizarán las siguientes tecnologías: Angular, NestJs, MariaDB, TypeScript, NodeJs y Git.

Palabras clave: UWE, ISO 25000, información.

ABSTRACT

Information systems and information technologies have transformed the way institutions operate today. Its use allows you to improve information management, automating administrative processes in a more efficient and agile way. The Medicine degree at the Public University of El Alto is in an international accreditation process. In this context, new technologies are a fundamental pillar for information processing. Therefore, a tool is required that facilitates the control of academic activities, research and social interaction. This tool will be of great help to have a record of all the activities of the medical career. The current project consists of designing and developing a system that allows the centralization of all the relevant information of the activities, as well as the creation of a repository of research works and digital books. These data for career decision making. The system will be developed under the UWE methodology, which guarantees good organization of the work flow and compliance with delivery deadlines. The following technologies will be used for development: Angular, NestJs, MariaDB, TypeScript, NodeJs and Git.

Keywords: UWE, ISO 25000, information.

LISTADO DE ABREVIATURAS

MCV	Modelo Vista Controlador
DB	Base de Datos (Data Base)
SQL	Lenguaje de consulta estructurado (Structured Query Language)
Css	Hojas de estilo en cascada (Cascading Style Sheets)
API de JR	Interfaz de programación de aplicaciones de Jasper Reports
ISO	Organización Internacional de Normalización
COCOMO	Constructive Cost Model (Modelo constructivo de costos)
HTML	Hyper text markup language
API	Interfaz de programación de aplicaciones
UML	Lenguaje Unificado de Modulado
SQuaRE	System and software Quality Requirements and Evaluation

CAPITULO I

MARCO

PRELIMINAR

**INGENIERÍA
DE
SISTEMAS**

UNIVERSIDAD PUBLICA DE EL ALTO



1.1. INTRODUCCION

Actualmente el tratamiento de la información se ha convertido en un factor muy importante, permite mejorar la productividad tanto en instituciones públicas como privadas, organizando la información de una manera más eficiente, siendo un apoyo en los procesos administrativos y en la toma de decisiones oportunas, confiables y efectivas.

En carrera de Medicina de la Universidad Pública de El Alto se realizan diferentes tipos de actividades, tanto académicas como sociales. En las actividades académicas podemos mencionar trabajos de investigación e interacción social, manejo de libros digitales, los trabajos de investigación y los libros digitales, son administradas de manera manual. Y en actividades sociales podemos mencionar a los convenios, bautizos, aniversario de carrera, confraternizaciones, congresos y otros.

Considerando lo mencionado, se propone la implementación del Sistema Integrado para la centralización de actividades académicas, investigación e interacción social Pregrado y Posgrado para un manejo de la información optimo, claro y preciso. El sistema permitirá tener un registro eficiente de las actividades académicas, en cuanto a la malla curricular y su contenido, administrar los trabajos de investigación, extensión e interacción social estudiantil, trabajos de producción intelectual de los docentes, brindar información oportuna y actualizada que ayude en la toma de decisiones.

Para el desarrollo del presente proyecto se utilizó la metodología UWE (UML Based Web Engineering), esta metodología tiene una variedad de diagramas que nos permitirá modelar el sistema; para el desarrollo del sistema se utilizó el framework Angular, angular sigue el patrón arquitectónico Modelo-Vista-Controlado (MVC) el cual permite tener un software organizado, y como gestor de Base de Datos esta MySQL.

1.2. ANTECEDENTES

1.2.1. Antecedentes Institucionales

La carrera de Medicina de la Universidad Pública de El Alto, convocó a concurso de méritos en el mes de septiembre del 2002 para la elección de docentes, de manera dar continuidad a las actividades ante el abandono de los anteriores profesores. Se realizó el análisis del plan curricular de la Carrera de medicina, y se decide realizar las “primeras jornadas académicas” que se llevaron a cabo el mes de octubre de 2002, el cual permitió ver la gran deficiencia organizacional, administrativa y principalmente académica. Como resultado de estas jornadas sobrevienen modificaciones tanto en el plan curricular como en el desempeño técnico administrativo, entre las más importantes a citar:

- Se anualiza la gestión de la carrera de medicina
- Adecuación a los programas analíticos de estudio de la UMSA
- Reordenamiento del nivel cursado por los alumnos considerando su avance teórico o práctico.

En el mes de abril 2004, se llevan a cabo las II Jornadas Académicas de la Carrera, habilitando el Tercer año con las materias correspondientes, basadas en el Pensum del Sistema Universitario Boliviano. En la perspectiva de la necesidad de contar con el Perfil del futuro Médico de la U.P.E.A se realiza las III Jornadas Académicas de la Carrera, en el afán de lograr la construcción del Plan Curricular que contempla el internado rotatorio. En el mes de junio del 2005 se inicia la Gestión Académica y finaliza en agosto del 2006. En fecha 25 y 26 de agosto se lleva adelante la Primera Pre Sectorial de la Carrera de Medicina de la U.P.E.A. donde se realiza la aprobación del nuevo Plan Curricular. En el mes de junio del 2006 se efectuó la Segunda Pre Sectorial de la Carrera de Medicina de la U.P.E.A. En fecha 15, al

29 de abril y del 1 al 13 de junio del 2010 se efectuó la Tercera Pre Sectorial de la Carrera de Medicina de la U.P.E.A. (Machicao, 2007, p.9)

- **Misión**

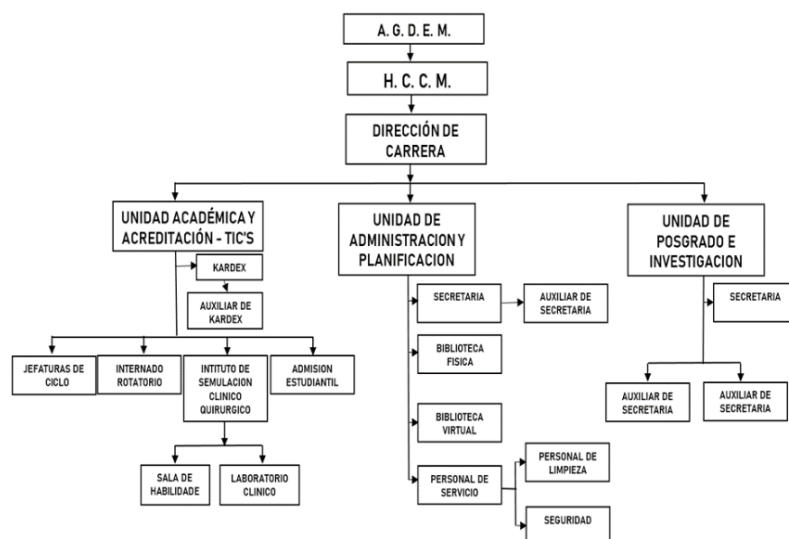
Formar profesionales de la salud, médicos generales, críticos y humanistas, con valores éticos, compromiso social y voluntad de superación, que responden a las necesidades de la salud con innovación y excelencia educativa, capaces de difundir conocimientos, tecnología y avances científicos, para contribuir a la salud y desarrollarse de la investigación científica en beneficio del ser humano y del país.

- **Visión**

Referente en la formación médica con valores espirituales éticos con solvencia científica, tecnológica de calidad y calidez, reconocida socialmente y acreditada, preservando la excelencia y consolidando su reconocimiento nacional e internacional.

- **Objetivo**

Fomentar la realización de investigación y proponer soluciones sobre las problemáticas en salud, que son prioritarias para el país, incluyendo su dimensión Internacional, que requieran la participación de investigadores en diferentes disciplinas de las Ciencias de la salud. Incidir en la formación de nuevos médicos mediante la incorporación y formación de un amplio número de estudiantes; impulsar la colaboración entre entidades universitarias y con otras instituciones y organismos, Nacionales e Internacionales especializados.

Figura 1.*Organigrama Institucional*

Nota. Organigrama Institucional de la carrera de Medicina – U.P.E.A. gestión 2024

1.2.2. Antecedentes internacionales

- (Mora Romero & Sanchez Jarquin, 2019, p. 9) “Desarrollo de un Sistema de Información para la gestión de proyectos de responsabilidad social del programa de ingeniería de sistemas de la Universidad Católica de Colombia” por Juan Manuel Pascagaza Gutierrez, con el sistema de información se ha logrado automatizar los procesos operativos, suministro una plataforma de información necesaria para ayudar a la toma de decisiones, se tiene la información necesaria la alcance y permite la gestión de los proyectos.
- (Arias Jaramillo & Cholca Catucuamba, 2021, p.3) “Diseño e Implementación de un repositorio digital de proyectos Tecnológicos de la carrera de las ciencias experimentales Informática de la UCE en el periodo 2020” por Edison David Arias Jaramillo, Jhonatan Stalyn Cholca Catucuamba. La elaboración del repositorio digital ayudara a la organización, prevención y difusión de la producción digital de la carrera de Pedagogía

de las ciencias experimentales de la informática de la universidad central del Ecuador, se pobra visibilizar las propuestas tecnológicas desarrollas por el estamento estudiantil, exhibiendo de forma sistemática, amigable y de fácil acceso a la comunidad, para esto se empleó metadatos para la organización interna de su base de datos, una página web que es de libre acceso a la población en general.

1.2.3. Antecedentes nacionales

- (Yolanda, 2011, p.2) “Sistema de Gestión de Información Académica, caso carrera de Medicina-UMSA” por Yolanda Chambi Quispe. El proyecto de Sistema de Gestión de Información Académica se desarrolló a partir de la necesidad de recopilar, almacenar, y difundir información académica a la carrera de Medicina. El objetivo de del sistema es trabajar de forma eficaz, y eficiente. El sistema se desarrolló con Visual WADE para el diseño de las interfaces y PHP como lenguaje de programación bajo la plataforma de LINUX.
- (Copacondo Rodriguez, 2011, p.3) “Portal Web para el Arzobispado de Nuestra Señora de La Paz” por José Miguel Copacondo Rodriguez. El desarrollo e implementación del portal web para el Arzobispado, Zend PHP como herramienta Framework, fue con tecnología web 2.0 que es la facilidad donde interactúa el usuario y administrador cuando ingresan al portal.
- (Choque Condori, 2022, p.3) “Histoteca virtual de Citohistologia y Patologia oral caso: facultad de Otontologia” por Yesica Rosa Choque Condori. El centro de referencia en Histología y Patología Oral de la facultad de odontalgia pretende encaminarse al buen manejo de su información, ya que tienen que manipular grandes cantidades de información acumulada durante varios años. Evidenciando los problemas del centro, con el sistema se mejora en el proceso académico de los estudiantes de la carrera para proveer una información segura, confiable y oportuna a aquellos que lo requieran.

1.2.4. Antecedentes locales

- (Orozco Sullca, 2020, p.4) “Plataforma Web Cultus para la gestión de recursos Visuales caso: Unidad Educativa Adventista Shalom” por Univ. Abraham Alexander Orozco Sullcata. Con la pandemia del COVID-19 que se expandió a nivel mundial, y viendo la seguridad de la salud de los estudiantes de la unidad educativa se optó por la creación de esta plataforma web, con este se buscó aportar a la unidad educativa con la gestión de material educativo, gestión de notas y otros. Para el desarrollo e implementación de la plataforma se utilizó el método UWE, lenguaje de programación java web y JavaScript, bootstrap como framework y MariaDB como Base de datos.
- (Sangalli Paco, 2020, p.3) “Sistema de Control y Seguimiento de Proyectos Educativos caso: Gobierno Autónomo municipal de Puerto Perez” por Anchelho Sangalli Paco. En vista de que el municipio de Puerto Pérez no contaba con un sistema de seguimiento para sus proyectos, se optó por el desarrollo de sistema de control y seguimiento, el cual brinda la información optima y eficaz hacia los pobladores. El sistema se desarrolló con la metodología ágil UWE, con el lenguaje de programación PHP y MYSQL como gestor de base de datos.
- (Gutierrez Zapata, 2020, p.3) “Sistema de Gestión y Digitalización Bibliotecaria caso: Carrera de Odontología U.P.E.A”, por Erick Roger Gutierrez Zapata. La carrera de Odontología no contaba con recursos virtuales necesarios para los estudiantes, de tal manera que con este sistema de gestión y digitalización bibliotecaria, optimizara la obtención de libros para los estudiantes. El desarrollo del sistema fue elaborado con el método UWE, se utilizó el lenguaje de programación PHP y MariaDB como gestor de Base de Datos.

1.3. PLANTEAMIENTO DEL PROBLEMA

La dirección de la carrera de Medicina de la Universidad Pública de El Alto, maneja información de los trabajos de investigación, extensión pregrado, posgrado además de libros digitales. Esta información no está debidamente registrada ni organizada, en consecuencia, se pierde la veracidad en la información, e incrementa los tiempos en respuesta a alguna petición, en algunos casos se tiene el extravío de documentación importante de los trabajos de investigación. Actualmente esta información está siendo administrada de manera manual, por tal razón existe el riesgo del extravío de información, además que el nivel de seguridad está siendo muy baja. Por otro lado, en el sitio web actual que tienen la carrera, no refleja todas las actividades de los acontecimientos académicos importantes (malla curricular, contenido de materias, convenios, proyectos, titulaciones, visitas a hospitales, interacción social, ferias científicas entre otros). Ver Anexo A.

1.3.1. Problema general

La carrera de Medicina de la U.P.E.A. enfrenta un manejo ineficiente de la información, relacionada con sus actividades académicas, de investigación e interacción social. Esta deficiencia resulta en demoras significativas en el tiempo de respuesta a las solicitudes recibidas y afecta negativamente la toma de decisiones institucional.

1.3.2. Problemas específicos

- Deficiente administración manual en cuanto a los trabajos de investigación, extensión e interacción social, ocasionando el extravío de trabajos, así mismo existe demora en una petición de búsqueda de algún trabajo, de entre 20 a 30 minutos lo que conlleva una pérdida de tiempo.
- Los trabajos de producción intelectual son administrados de manera manual, y existe la posibilidad de extravío de trabajos de investigación, pérdida de tiempo en la búsqueda.

- Actualmente los libros digitales están siendo almacenados en Google drive, además de diferentes dispositivos físicos, tales como CD's, Flash Memory, entre otros), lo cual genera una demora en la búsqueda de los mismos,
- Para la obtención de documentos importantes como ser la malla curricular, contenida mínimo de cada materia, se debe solicitar a instancias administrativas de la carrera, esto significa realizar una solicitud y esperar de entre 24 a 48 horas para la respuesta.
- Las actividades académicas, sociales, interacciones, no están siendo registradas en ningún lugar, solo están siendo mostradas en algunas redes sociales.

1.3.3. Formulación del problema

¿De qué manera se apoyará a la carrera de Medicina de la U.P.E.A. en cuanto a las Actividades Académicas, de Investigación e Interacción Social, tal que su gestión sea eficiente?

1.4. OBJETIVOS

1.4.1. Objetivo general

Desarrollar un Sistema Integrado para la Centralización de Actividades Académicas, Investigación e Interacción Social Pregrado y Posgrado, que permitirá la gestión eficiente de los trabajos de investigación, proporcionará datos oportunos y confiables en tiempo real.

1.4.2. Objetivos específicos

- Mejorar la organización de los trabajos de investigación con un repositorio, tal que disminuirá en gran medida los extravíos, minimizando los tiempos de búsqueda de los trabajos.
- Crear una base de datos para la gestión de los trabajos de producción intelectual, para evitar los extravíos de trabajos de investigación.
- Sistematizar en un repositorio los libros digitales, con esto se reducirá el uso de dispositivos de almacenamiento y beneficiará a los estudiantes siendo así que ya no tendrán que comprar libros que tienen un costo elevado.
- Brindar información oportuna acerca de la malla curricular, contenido de las materias para la disminución de tiempo en la solicitud de dicha información.
- Diseñar interfaces para la visualización de todos los acontecimientos académicos, sociales e instituciones, para tener un mejor control de los acontecimientos y así aportar en la toma de decisiones.

1.5. JUSTIFICACIÓN

1.5.1. Justificación Técnica

Para el desarrollo del proyecto se utilizará un servidor para la ejecución del proyecto, la carrera de medicina cuenta con un servidor disponible y espacio libre suficiente. se realizará el relevamiento de información para el desarrollo del proyecto.

El sistema gestionará toda la información de los trabajos de investigación de docentes y estudiantes, utilizando para ello la metodología orientada a objetos, y para el modelado y análisis del sistema se utilizará el método UWE.

1.5.2. Justificación Económica

Con la implementación del sistema, la carrera de medicina disminuirá en gastos de compras de material de escritorio (papelería), pérdidas económicas que se dan en impresiones de documentos, así como compra de libros físicos.

En cuanto a las herramientas para el desarrollo de la plataforma, para el alojamiento del sistema, la carrera de medicina cuenta con un hosting y un dominio que no está siendo utilizado en su totalidad. De modo que se aprovechará el espacio disponible, y se alojará la plataforma, entonces se tendrá un gasto adicional menos con relación a lo mencionado.

1.5.3. Justificación Social

A medida que avanza el tiempo, existe mayor cantidad de estudiantes y así mismo crece el desarrollo de proyectos. Entonces la carrera de medicina deberá ser capaz de responder a las demandas que se vayan generando.

Con el sistema, el plantel administrativo y/o usuarios habilitados, podrán contar con la información rápida y oportuna y segura, facilitará el control de las investigaciones y/o libros, ver en qué etapa se encuentra el proyecto de investigación. Todo esto para en la toma de decisiones para el bien de la carrera de medicina.

1.6. METODOLOGIA

El desarrollar un determinado software no es una labor fácil, tal que existen numerosas metodologías que inciden en las distintas dimensiones del proceso de desarrollo. Las metodologías permiten mejorar la interacción entre los desarrolladores y el cliente, tienen por objetivo asegurar que el producto final cumpla con los requerimientos del cliente. Para el desarrollo del proyecto se aplica el siguiente recurso metodológico.

1.6.1. Metodología de ingeniería UWE (UML-Based Web Engineering)

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

Para el modelado y análisis del sistema utilizaremos UWE que sirve para modelar aplicaciones web, presta una especial atención a la sistematización y personalización (sistemas adaptivos). Provee de perfiles UML, meta modelos, un proceso de desarrollo dirigido por modelos, y herramientas de soporte para el diseño sistemático de aplicaciones web. Utiliza notación basada en UML, para aplicaciones web en general y para aplicaciones adaptativas en particular.

Para el desarrollo de la metodología UWE, se tiene las siguientes etapas:

- Captura, análisis y especificación de requisitos
- Diseño de sistema
- Codificación del sistema
- Pruebas al sistema
- Instalación o fase de implementación
- Mantenimiento

1.6.2. Métricas de calidad

La ingeniería de software, no está basada en leyes cuantitativas básicas, sino más bien se realiza un conjunto de medidas conocidas como métricas, las cuales proporcionan una diferencia de la calidad de algún producto de software (Pressman, 20210).

- **Control de Calidad ISO 25000.-** ISO 25000: (SQuaRE-System and Software Quality Requirements and Evaluation). Requisitos y evaluación de calidad de productos de software.

Calidad de producto, calidad de procesos; son aspectos más importantes en el desarrollo de software. ISO/IEC 25000 constituye una serie de normas basadas en ISO/IEC 9126 y en ISO/IEC 14598 cuyo objetivo principal es guiar el desarrollo de los productos de software mediante la especificación de requisitos y evaluación de características de calidad. (AENOR2023)

1.6.3. Control de seguridad ISO/IEC 27001

Con el fin de preservar la información, se ha demostrado que no es suficiente con la implementación de controles y procedimientos de seguridad realizados frecuentemente sin un criterio común establecido.

La organización internacional de estandarización (ISO), a través de las normas recogidas en ISO/IEC 27000, establece una implementación efectiva de la seguridad de la información empresarial desarrolladas en las normas ISO 27001/ISO 27002 (Alonso, 2015).

1.6.4. Estimación de costos – COCOMO II

Es un modelo que permite estimar el costo, el esfuerzo, y el horario en la planificación de un nuevo proyecto de desarrollo de software. Uno de los principales factores que afectan la productividad incluye la aptitud personal, la experiencia, el proceso de desarrollo, el tamaño del proyecto. COCOMO II ha logrado la madurez necesaria del modelo para conseguir estimaciones de gran precisión. (Gomez, Migani, & Otazu, 2017)

1.6.5. Pruebas de software

Es un proceso de evaluación donde se ve la funcionalidad de un software, los errores que pueda tener. Con las pruebas que se realiza al software, aseguramos que lo desarrollado cumpla con todos los estándares requeridos.

- **Caja blanca**

El objetivo de las pruebas de caja blanca es comprobar la calidad e integridad del código del software. También se lo conoce como pruebas estructurales o de Cristal. Este método se centra en como diseñar los casos de pruebas atendiendo el comportamiento interior y la estructura del programa. (Florin, 2022)

- **Caja negra**

El objetivo de las pruebas de caja negra es verificar que el sistema funciona como esperar el usuario final. También conocido como Pruebas de Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de pruebas. (Florin, 2022)

- **Pruebas de Estrés**

Las pruebas de estrés, es medir el rendimiento del software a partir de la sobrecarga de sus indicadores normales de funcionalidad, quiere decir que se somete la aplicación a un alto nivel de carga de trabajo con las pruebas de estrés de software. El tráfico de información de transacciones y de usuarios se incrementa hasta sobrepasar la tasa normal. Uno de los beneficios de la prueba de estrés es; se conoce a precisión los máximos del producto, esto implica conocer los comportamientos en momentos críticos y sobre el manejo a los fallos, además se ve el rendimiento del software.

1.7. HERRAMIENTAS

1.7.1. Lenguaje de programación

- **JavaScript (JS):** JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, utilizado principalmente en el desarrollo web, pensado para agregar potencial de interacción y dinamismo a las páginas web. JavaScript también se utiliza en aplicaciones de servidor (con tecnologías como Node.js) y en el desarrollo de aplicaciones móviles y de escritorio. JavaScript se ha expandido más allá del desarrollo de interfaz, que es donde comenzó. Recientemente, JavaScript ha llegado al back-end, o dorsal de desarrollo web. (Sanchez Maza, 2012)
- **TypeScript:** TypeScript (TS) es un lenguaje de programación construido a un nivel superior de JavaScript (JS). Esto quiere decir que TypeScript dota al lenguaje de varias características adicionales que hacen que podamos escribir código con menos errores, más sencillo, coherente y fácil de probar, en definitiva, más limpio y sólido. (Serrano Sanchez , 2020)
- **NestJS.** - Nest proporciona una arquitectura de aplicaciones, que permite a los desarrolladores crear aplicaciones altamente comprobables, escalables, poco acopladas y de fácil mantenimiento.

1.7.2. Framework - Angular

Angular es un framework de JavaScript de código abierto escrito en TypeScript, su objetivo es desarrollar en una sola página, ya que Google se encarga del mantenimiento y contantes actualizaciones para la mejora del proyecto. (Puciarelli, 2020)

1.7.3. Gestor de Base de Datos

- **MariaDB.** - Es un sistema gestor de Base de Datos relacionales (DBMS), potente y versátil, capaz de satisfacer a sitios y plataformas web. Además, es multiplataforma, por lo que se puede instalar en Windows, Linux y Mac. En servidores web encuentra sobre la plataforma Linux, habitualmente y forma parte de la arquitectura LAMP (Linux, Apache, MySql), con una arquitectura Cliente/Servidor. (Hueso Ibañez, 2016)

1.7.4. Diseño de Interfaces

- **Css (Cascading Style Sheets).** Es uno de los lenguajes esenciales para la creación de páginas web. Es un lenguaje de hojas de estilo, es decir, te permite aplicar estilos de manera selectiva a elementos en documentos HTML.

1.8. LIMITES Y ALCANCES

1.8.1. Limites

El sistema tiene los siguientes límites:

- Solo usuarios habilitados por el administrador del sistema tendrán acceso al sistema dependiendo el rol asignado.
- El sistema no controlará el seguimiento académico a los estudiantes.
- La parte administrativa del sistema, no estará disponible para los estudiantes, solo tendrá el acceso el personal administrativo académico.
- El sistema realizará la gestión de la malla curricular y su contenido.
- El sistema realizará la gestión de los libros digitales.
- El sistema realizará la gestión de los trabajos de investigación de los distintos niveles de grado (Pregrado-internado-Postgrado)

1.8.2. Alcances

Basado en las principales necesidades de la carrera de medicina, la plataforma cuenta con los siguientes módulos y funcionalidades:

- Módulo de administración para la gestión de usuarios, permisos.
- Módulo de gestión del contenido de la plataforma, fotografías, autoridades de carrera, misión, visión, objetivo de carrera.
- Módulo de logueo.
- Módulo de gestión de la malla curricular y su contenido.
- Módulo de vista de la malla curricular, materias con su contenido de avance.
- Modulo para la gestión de libros digitales.
- Módulo de vista de libros y su descarga.
- Módulo de gestión de asignaturas.
- Módulo de gestión para las investigaciones, extensiones e interacción social Estudiantil. Así mismo de trabajos de investigación de producción intelectual
- Generación de reportes en formato DPF.

1.9. APORTES

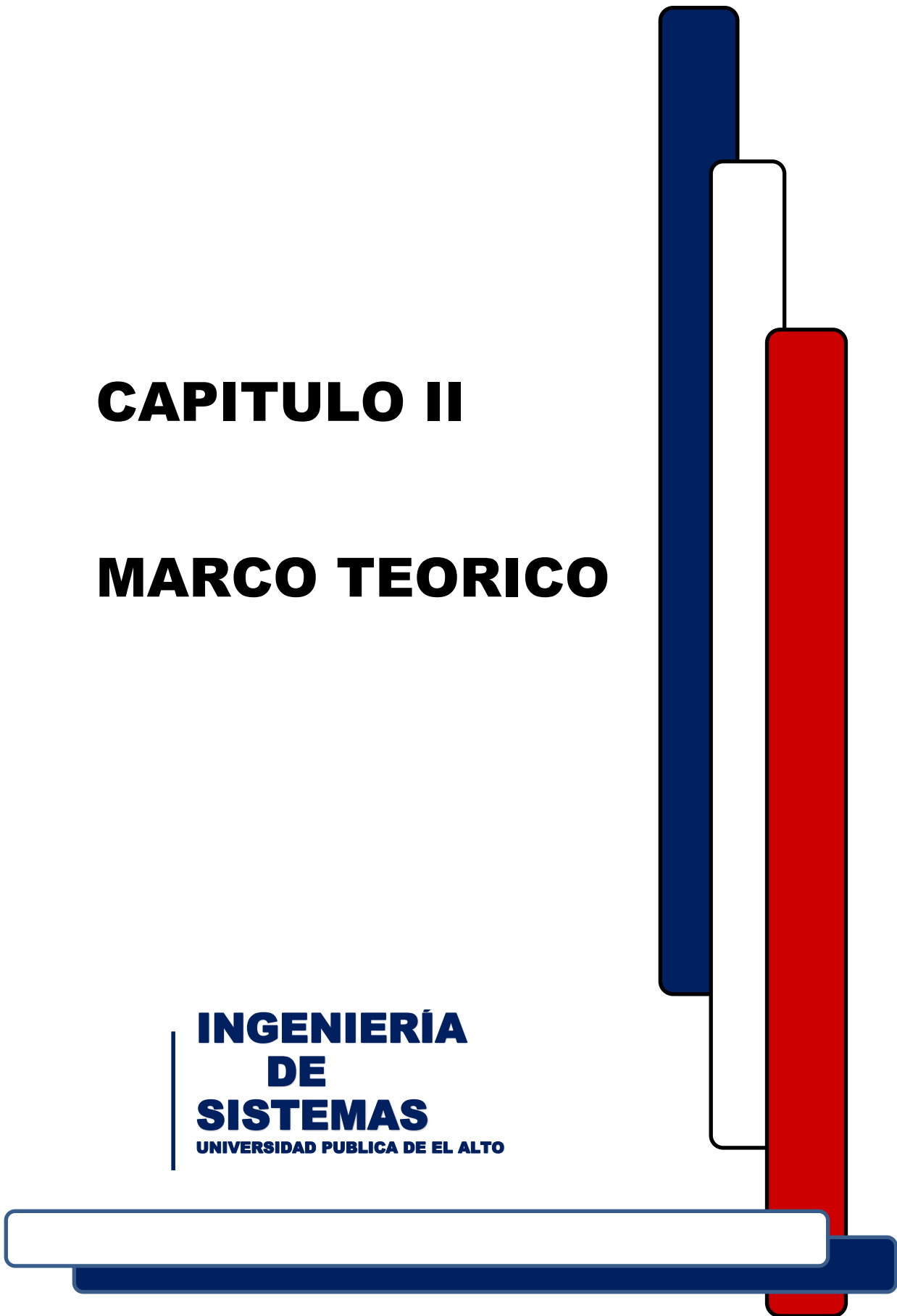
El sistema propuesto aportará en gran medida a la carrera de Medicina de la U.P.E.A., ya que permitirá tener información centralizada en la Base de Datos. Brindará información oportuna y segura en cuando a los trabajos de investigación. Las personas tendrán acceso libre a los libros digitales para las distintas materias, reduciendo así en costos en la adquisición de libros físicos. También se tendrá disponible la visualización de los últimos acontecimientos académicos y sociales.

CAPITULO II

MARCO TEORICO

**INGENIERÍA
DE
SISTEMAS**

UNIVERSIDAD PÚBLICA DE EL ALTO



2.1. INTRODUCCION

Para el desarrollo del proyecto es necesario e importante seguir una especificación que permita a los desarrolladores de software, tener una disciplina para todas las etapas del desarrollo del software, desde un inicio de un requerimiento hasta las pruebas que se realizan al software.

En este capítulo se verá la descripción de conceptos, definiciones y teorías fundamentales que se consideran para realizar el presente proyecto, ya sea técnicas, métodos, técnicas de análisis, tipos de pruebas de software e implementación del software.

2.1.1. Sistema

Sistema es un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, este está siendo dotado de una estructura, una composición, y un entorno particular.

En informática, se entiende por un sistema a un conjunto de datos ordenados conforme a una serie de instrucciones o algoritmos, que permiten su ubicación y su recuperación rápida y simple. (Hernandez Trasobares, 2003)

Según García Peñalvo (2018) en el texto “Ingeniería de software I” refiere lo siguiente:

Un sistema es un conjunto de partes o elementos organizados u relacionados que interactúen entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia. Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software). Cada sistema existe dentro de otro más grande, por lo tanto, un sistema puede estar en formato por subsistemas y

partes, y a la vez puede ser parte de un supersistema. Los sistemas tienen límites o fronteras que los diferencian del ambiente.

Ese límite puede ser físico (gabinete de una computadora) o conceptual. Si hay algún intercambio entre el sistema y el ambiente a través de ese límite, los sistemas abiertos, de lo contrario, el sistema es cerrado. El ambiente es el medio en extremo que envuelve física o conceptualmente a un sistema. Un sistema tiene interacción con el ambiente, del cual recibe entradas y al cual se le devuelven salidas. El ambiente también puede ser una amenaza para el sistema (p.6).

Un sistema es un conjunto de elementos que están relacionados entre sí para lograr un objetivo en común, se puede tratar de elementos materiales o conceptuales.

2.1.2. Sistema Integrado

Un sistema integrado es un sistema único diseñado para gestión de múltiples aspectos de las operaciones de una organización, este también se rige en normas, como la gestión de calidad.

Un sistema integrado implica la fusión de los sistemas formales existentes y la aplicación de mejores prácticas específicas en toda la organización.

Algunos de los beneficios de un sistema integrado son: evitar la duplicación de esfuerzo, uso efectivo de tiempo, hacer uso eficiente de los recursos para implementar y administrar sistemas. (Gisbert Soler & Ruslan, 2014)

2.1.3. Actividad Académica

Una actividad académica es cualquier acción dirigida a reforzar y complementar contenidos estudiados en el ámbito académico. Las actividades académicas están enfoca

aquellas operaciones que se realizan dentro del marco del proceso de enseñanza aprendizaje, encaminadas a reforzar los conocimientos que van adquiriendo, a desarrollar nuevas habilidades y vincular al campo de trabajo y con su entorno social. Las actividades académicas pueden ser exposiciones, concursos, paneles, conferencias, congresos, cursos de regularización, visitas a empresas, organizaciones.

2.1.4. Investigación

La investigación es un proceso intelectual y experimental, que comprende un conjunto de métodos aplicados de modo sistemático, este tiene la finalidad de indagar un tema, ya sea científico, humanístico, social o tecnológico.

La investigación puede llegar a tener varios objetivos, por tanto, este lleva a cabo mediante un proceso metódico, de forma organizada y objetiva a fin de que los resultados reflejen la realidad tanto como sea posible (Sabino, 2014)

Características de la investigación:

- Recoger información de diversas fuentes útiles para la investigación.
- Trabajo empírico, por tanto, se basa en la observación y experiencia del que investiga.
- Se desarrolla de manera organizada y coherente, por ello se basa en una metodología de investigación.
- Debe ser objetiva, mostrar los resultados obtenidos tal cual sin omisiones.
- Es innovadora, es decir, los resultados deben exponer nuevos conocimientos.
- Amplio uso del discurso descriptivo y analítico.

- Debe ser replicable, en caso de que otra persona desee volver a realizar la investigación. (Sabino, 2014)

Una actividad de investigación se refiere a trabajos que los estudiantes o docentes van realizando a lo largo de la gestión académica, estas investigaciones son elaboradas de acuerdo a la materia que este cursando.

2.1.5. Interacción Social

El ser humano por naturaleza es social e interactúa constantemente con su entorno. Podemos decir que el ser humano comienza a interactuar desde nace, con la familia, en instituciones educativas, trabajo. La interacción también se lleva a cabo a través de dispositivos electrónicos son el simple hecho de enviar o recibir mensajes. Entonces Interacción social es el acto de emitir un mensaje que genere un estímulo en otra persona.

Las interacciones sociales son vínculos creados por los seres humanos en su manera de convivencia grupal. (Esneca, 2022)

Una actividad de interacción social se refiere a, actividades sociales como, aniversarios de carrera, visitas a hospitales entre otros.

2.2. INGENIERIA DE REQUERIMIENTOS

La ingeniería de requerimientos es un proceso que comprende todas las actividades de requerimientos para crear y mantener un documento de requerimientos del sistema a desarrollar.

Es un proceso donde se aplica un método estructurado, el cual analiza el sistema y desarrolla un conjunto de modelos gráficos del mismo que actúan como una especificación del sistema.

Para el desarrollo de un proyecto de software la ingeniería de requerimientos cumple un papel muy importante, ya que su tarea principal consiste en la generación de especificaciones correctas que describan con claridad, de esta manera se optimizará en la recolección de información de requerimientos; esto nos servirá como base fundamental para el desarrollo del software, al realizar la ingeniería de requerimientos, se utilizará una serie de técnicas, métodos sistemáticos para asegurar que la información relevada es correcta y completa. (Arias Chavez, 2005)

2.1.6. Requerimientos funcionales

“Los requerimientos funcionales son las descripciones explícitas del comportamiento que debe tener una solución de software y que información debe manejar” (Arias Chavez, 2005).

Por tanto, los requerimientos funcionales expresan las capacidades o cualidades que debe tener la solución para satisfacer los requerimientos de los interesados de proyecto. Estos deben proporcionar una descripción lo suficientemente detallada para permitir el desarrollo e implementación de la solución.

Los requerimientos funcionales deben ser:

- **Precisos.** - Evitar la ambigüedad a la hora de definir los requerimientos.
- **Completo.** - Descripción de todos los servicios y características.
- **Consistentes.** - No debe ocurrir contradicciones en la descripción. (Arias Chavez, 2005)

Los requerimientos funcionales son los que más influyen en si la solución será aceptada o no por los usuarios.

2.1.7. Requerimientos no funcionales

Son restricciones impuestas por un sistema que definen sus atributos de calidad. Por lo general en un proyecto se denota con adjetivos como seguridad, rendimientos y escalabilidad. Los requerimientos no funcionales ayudan a garantizar que el sistema satisfaga las necesidades del usuario.

Requerimientos no funcionales. Entonces podemos dividir en dos a los requerimientos no funcionales:

- Atributos de calidad: estas determinan su calidad general. Deben de ser atributos de calidad que incluyan seguridad, rendimiento, fácil de uso.
- Restricciones: son limitaciones impuestas al sistema, por ejemplo, el tiempo, recursos y el entorno.
- Los requerimientos no funcionales ayudan a; garantizar que sistema sea el adecuado para su propósito, a que sea escalable, seguro y confiable, además que este sea de fácil uso, y mantenimiento. En resumen; es lo que el sistema realizar pero que el usuario no puede ver.
- “Los requerimientos no funcionales a su vez se asocian con características particulares del sistema. Más bien, estos requerimientos especifican o restringen las propiedades emergentes del sistema” (Arias Chavez, 2005)

2.3. INGENIERIA DE SOFTWARE

Existen varias definiciones para ingeniería de software, uno de los más relevantes es de Sommerville, este es considerado uno de los padres de ingeniería de software; “Es una disciplina que comprende todos los aspectos de la producción del software” (Sommerville, 2005)

Para poder entender mejor, ingeniería de software es una disciplina que implica el uso de estructuras, herramientas y técnicas para la construcción de sistemas informáticos. Este incluye el análisis previo que se tiene para ver la situación antes del proyecto, se debe redactar todo el proyecto, el desarrollo y las pruebas necesarias del mismo, para garantizar el correcto funcionamiento del software antes de ponerlo en funcionamiento.

2.4. UWE – UML – Based Web Engineering

UWE es un método de ingeniería de software para el desarrollo de aplicaciones web, basado en UML. UWE describe una metodología de diseño sistemática, basada en la técnica UML, así como la notación y mecanismos de extensión de UML.

Esta es una herramienta que nos permite la modelación de aplicaciones web. En requisitos separa las fases de captura, definición y validación, además hace tratamientos especiales dependiendo del carácter de cada requisito. (Carelo, Moraga, & Piattini Velthuis, 2010)

Figura 2.

Metodología UWE



Nota. Adaptado de UML Diseño orientado a objetos, de (Alarcon, 2011)

A continuación, veremos las etapas de la metodología UWE:

2.1.8. Captura, análisis y especificación de requisitos

En esta etapa se reúne y especifican las características que debe cumplir el sistema, todo esto a través de la identificación de actores, los requerimientos funcionales y no funcionales, también se hace referencia al diseño y la generación de glosario y al prototipo del sistema.

- **Identificación de actores.** - Contiene actor y descripción del mismo.
- **Requerimientos Funcionales.** - Para definir los requerimientos funcionales debemos tomar en cuenta los siguientes aspectos: Lo que debe realizar el sistema, Numero de requerimiento funcional, Lo que el sistema debe hacer y funciones que debe tener.
- **Requerimientos no funcionales.** - Capturar las propiedades del sistema; los más importantes y notorios son el rendimiento, la disponibilidad, seguridad entre otros. Estos se definen a través de tablas, requerimientos no funcionales y su respectiva descripción.

2.1.9. Diseño del Sistema

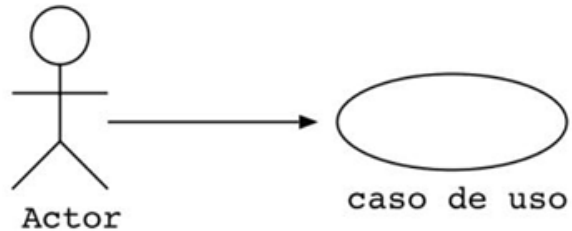
Este se basa en la especificación de los requisitos producidos por el análisis de los requerimientos (fase de análisis), en el diseño se define como se va cumpliendo estos requisitos, y la estructura que debe darse en la aplicación web.

- **Diagrama de Casos de Uso.** -

Secuencia de acciones que pueden ser representadas por garfios de un diagrama. Un caso de uso es una técnica de modelado que es utilizado para describir lo que es sistema debe de hacer, y estos son descripciones funcionales del sistema.

Figura 3.

Caso de uso



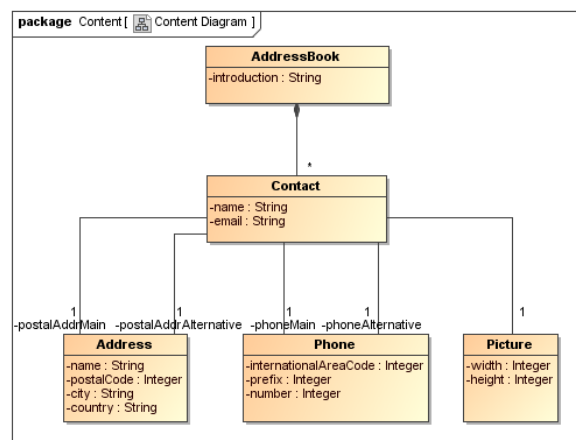
Nota. Adaptado de *El Modelo de casos de Uso*, de (Sparks & Systems, 2011)

- **Modelo conceptual. -**

A través de diagramas de clases se especifican el cómo están relacionados los contenidos del sistema. Esto incluye objetos aplicados en las actividades típicas que los usuarios realizan en la aplicación web.

Figura 4.

Diagrama de Clases



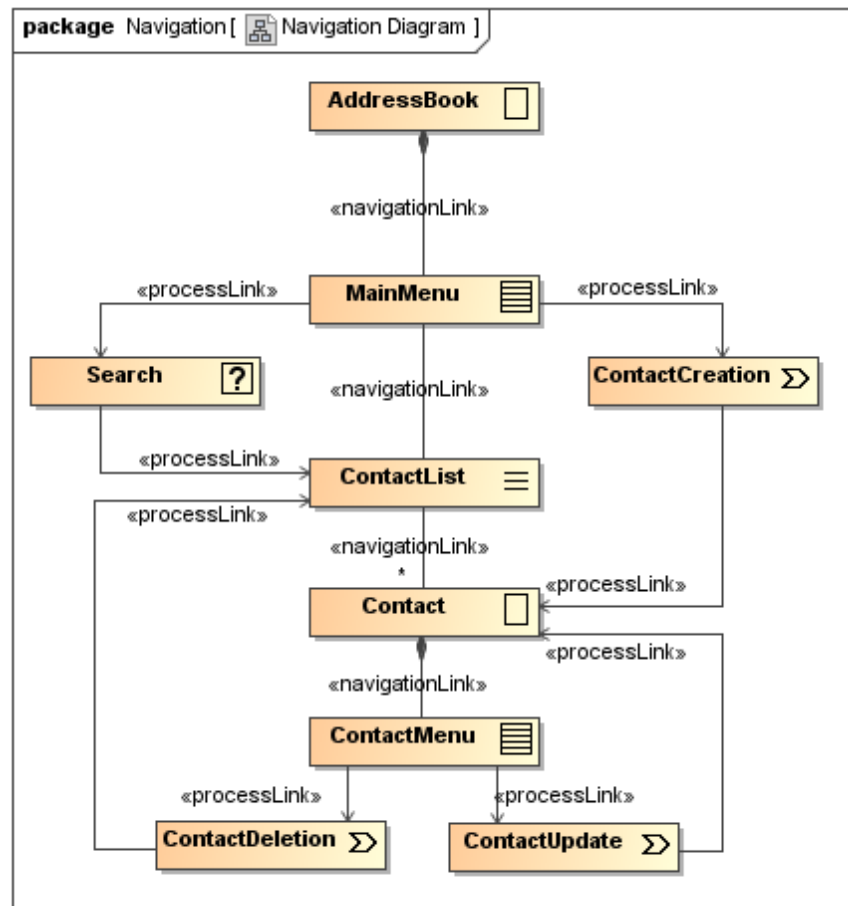
Nota. Adaptado de *Content Model*, de (Koch, 2016)

- **Modelo de navegación**

Para el modelo de navegación necesitamos un diagrama de contenidos nodos (nodes) y enlaces (link). Los enlaces de elementos de navegación se encuentran relacionados internamente dando así un panorama general del funcionamiento del aplicativo.

Figura 5.

Modelo de Navegación



Nota. Adaptado de tutorial – Navigation Model, de (Koch, 2016)

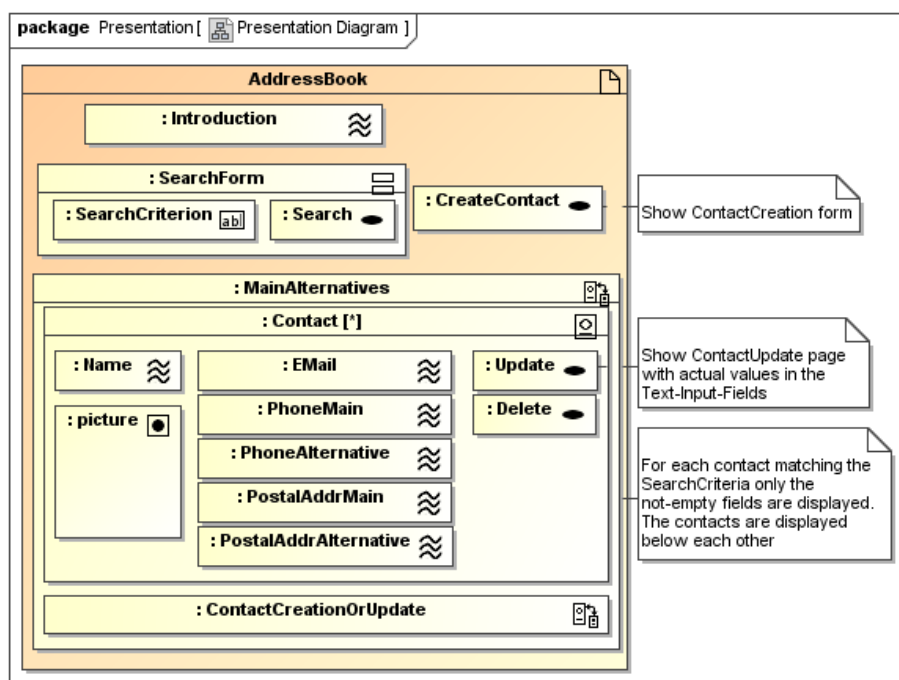
- **Modelos de presentación**

Se muestra una visión más abstracta de la aplicación, a su vez este se basa en el modelo de navegación y otros aspectos relevantes, las cuales serán ordenadas de acuerdo al

propósito de la aplicación. Los elementos de presentación son: ventana, entradas de texto, imágenes, audio y botones.

Figura 6.

Modelo de Presentación UWE



Nota. Adaptado de tutorial – Presentación Model, de (Koch, 2016)

2.1.10. Codificación del sistema

Esencialmente en esta etapa se realiza el desarrollo del aplicativo, es decir, se hace lo que se conoce como código fuente, en si la programación, aquí se utiliza el lenguaje de programación elegida, se plasma todo lo anterior (diseñar fase de la presentación).

2.1.11. Pruebas

Para realizar las pruebas al aplicativos, se utiliza las técnicas de Caja Negra o pruebas de comportamiento y caja Blanca o pruebas estructurales, que son las más utilizadas. Las pruebas de caja blanca son destinadas al código, por ellos se definen como pruebas

estructuradas. Y las pruebas de caja negra están dedicadas a evaluar las entradas y salidas del aplicativo.

2.1.12. Instalación o fase de implementación

Es el proceso donde los programas desarrollados son transferidos a un computador destinatario, inicializando, realizando la configuración, y poniendo en marcha la aplicación. Aquí se pone en marcha y ejecución de las anteriores etapas desarrolladas. Esto incluye la arquitectura y código fuente.

2.1.13. Mantenimiento

Es el proceso de cambiar o modificar y/o también actualizar el software para satisfacer las necesidades de los usuarios finales. El mantenimiento se realiza después de la implementación del software, por lo general es para la corrección de problemas o errores, mejorar el rendimiento (Quiroga, 20215).

2.5. LENGUAJE DE MODELO UNIFICADO UML

Es utilizado para el modelado de un sistema, UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real (LAR, 2000).

Existen nueve diagramas dentro de UML, los cuales pueden ser utilizados para modelar sistemas. La finalidad de los diagramas es presentar diferentes perspectivas del sistema del sistema, se los conoce como modelado, esto no es más que una representación gráfica del sistema. Diagramas que se utilizaran:

- Diagrama de casos de uso
- Diagrama de clases

2.6. ARQUITECTURA CLIENTE/SERVIDOR

Cliente/servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor. Normalmente, los servidores son ordenadores potentes dedicados a gestionar unidades de disco servidor de ficheros, impresoras servidor de impresoras, tráfico de red servidor de red, datos servidor de bases de datos o incluso aplicaciones servidor de aplicaciones, mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores. Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios clientes y procesos que responden a estos servicios servidores. Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos (Lijan Mora, 2022).

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. Es una relación entre el cliente-servidor que corresponde al patrón de solicitud y respuesta.

Figura 7.

Sistema Cliente/Servidor



Nota. Adaptado de tutorial – Presentación Model, de (Koch, 2016)

2.7. MÉTRICAS DE CALIDAD

Las métricas de calidad son esencialmente las que definen de una manera u otra la calidad del software.

- **Métrica:** hace referencia a números que van midiendo los parámetros predeterminados, por ejemplo, el número de líneas de código o el volumen de documento asociada a este mismo.
- **Calidad:** refiere a la satisfacción de necesidad implícita o explícita según un parámetro, en cumplimiento de requisitos. En el caso del software cumplir con todas las funcionalidades que se definieron en los requerimientos. (Carelo, Moraga, & Piattini Velthuis, 2010)

2.1.14. Norma ISO/IEC 25000

SQuaRE (*System and Software Quality Requirements and Evaluation*). Es una familia de normas que tienen el objetivo común de evaluar la calidad del producto. La ISO/IEC 25000 es el resultado de la evolución de otras normas, especialmente la norma ISO/IEC 9126, que

describe las particularidades de la calidad del producto, e ISO/IEC 1459 que esta abordada el proceso de evaluación del producto de software.

Figura 8.

Familia de normas ISO/IEC 25000



Nota: Adaptado de Normas ISO/IEC 25000, por (Normas ISO 25000, 2022)

2.7.2 ISO/IEC 25010 - System and software quality models:

Describe el modelo de calidad para el producto software y para la calidad en uso. Esta Norma presenta las características y subcaracterísticas de calidad frente a las cuales evaluar el producto software.

Figura 9.

Características de calidad ISO/IEC 25010



Nota: Adaptado de ISO/IEC 25000, por (ISO/IEC 25010, 2022)

- **Adecuación Funcional.** - Hace referencia a la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas. Cuando el software se ejecuta bajo determinadas condiciones.

Tabla 1.

Definición de métricas de calidad y fórmulas de la adecuación funcional

	METRICA	FORMULA	DESCRIPCION
COMPLETITUD FUNCIONAL	Compleitud de la implementación funciona	$CP = \left[\frac{(x - y) * 100}{x} \right]$ x: Número total de funcionalidades y: Número de funcionalidades incorrectas	Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
CORRELACION FUNCIONAL	Correlación entre las funcionalidades	$CF = \left[\frac{(x - y) * 100}{x} \right]$ x: Número total de funcionalidades no correlacionadas y: Número total de funcionalidades no correlacionadas	Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.

	METRICA	FORMULA	DESCRIPCION
PERTENECIA FUNCIONAL	Compleitud de la idoneidad funcional	$CF = \left[\frac{(x - y) * 100}{x} \right]$	Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Nota: La plataforma tiene la capacidad de ser funcionalmente correcto, cuenta con 3 subcaracterísticas para evaluación correctamente.

- **Eficiencia de desempeño.** - Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Tabla 2.

Definición de métricas y fórmulas de la eficiencia de desempeño

	METRICA	FORMULA	DESCRIPCION
COMPORTAMIENTO TEMPORAL	Tiempo de respuesta	$TPR = \frac{(x1 + x2 + x3)}{3}$ x: Tiempo 1 de respuesta en 1 minuto	Los tiempos de respuesta y procesamiento de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas.

		METRICA	FORMULA	DESCRIPCION
UTILIZACION RECURSOS	DE	Utilización de CPU	$CP = (10 * TP) - [11 * (TP - 1)]$	Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
			$UR = [x - (11 * \frac{x}{10 - 1})]$ x: porcentaje de uso del procesador	
CAPACIDAD		Tamaño de memoria usada por el proceso	$C = (\frac{x}{10}) - [11 - ((\frac{x}{100}) - 1)]$ x: Tamaño de la memoria usada por el proceso	Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

Nota. Funciona de acuerdo a una cantidad específica de recursos, bajo distintos escenarios ya sea en hardware o software.

- **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. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Tabla 3.

Definición de métricas y fórmulas de la compatibilidad

	METRICA	FORMULA	DESCRIPCION
COEXISTENCIA	Número de programas causantes de conflictos	$C = (10 * x) - [(10 - x) - (10 * x)]$ x: Numero de programas causantes de conflictos	Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
INTEROPERABILIDAD	Capacidad de intercambiar información con otras plataformas	I: capacidad de intercambiar información con otras plataformas	Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

Nota. Llevar a cabo sus funciones cuando comparten el mismo entorno sea 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. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Tabla 4.

Definición de métricas y fórmulas de usabilidad

	METRICA	FORMULA	DESCRIPCION
RECONOCIBILIDAD DE LA ADECUACION	Capacidad del software de detectar varios conceptuales	l: capacidad de software de detectar vacíos conceptuales	Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
APRENDIZABILIDAD	Numero de características game-based-learning que cumple la plataforma	$A = \frac{10 * x}{7}$ x: Numero de características game-based-learning que cumple	Capacidad del producto que permite al usuario aprender su aplicación.
OPERABILIDAD	Claridad de los mensajes	$O = \frac{(x - y) * 100}{x}$ x: Número total de entradas y: Número de entradas de validación	Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
PROTECCION CONTRA ERRORES DE USUARIOS	Verificación de entradas validas	$PEU = \frac{(x - y) * 100}{x}$	- Capacidad del sistema para proteger a los usuarios de hacer errores.

	METRICA	FORMULA	DESCRIPCION
ESTETICA DE LA INTERFAZ DEL USUARIO	Numero de principios de la evaluación heurística de la interfaz de usuario	$E = \frac{10 * x}{12}$ x: Numero de principios de estética que cumple	Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario
ACCESIBILIDAD	Número de sistemas operativos compatibles	$NA = \frac{10 * x}{4}$ x: Número total de sistemas operativos compatibles	Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Nota. Debe ser comprendida, entendida y usada y finalmente resuelta por parte del usuario final.

- **Fiabilidad-** Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Tabla 5.

Definición de métricas y fórmulas de la fiabilidad

	METRICA	FORMULA	DESCRIPCION
MADUREZ	Numero de fracasos de prueba	$M = \frac{(x - y) * 100}{x}$ x: Numero de pruebas realizadas y: Número total de fallas detectadas	Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales,
DISPONIBILIDAD	Tiempo de operación de servicios	$D = \frac{(x - y) * 100}{x}$ x: tiempo de uso en la plataforma en segundos y: tiempo de no disponibilidad de la plataforma	Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
TOLERANCIA FALLOS	A Tiempo perdido por fallas	$TF = (x - y) * NI$ x: tiempo total de uso de la plataforma en minutos y: Sumatoria de tiempo de duración de fallos	Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.

	METRICA	FORMULA	DESCRIPCION
CAPACIDAD DE RECUPERACION	Tiempo promedio de recuperación	$TPR = \frac{(x1 + x2 + x3)}{3}$ x: tiempo de respuesta en 1 minuto $CR = (10 * TP) - (11 * (TP - 1))$	Capacidad del producto software para recuperar los datos directamente afectados y restablecer el estado deseado del sistema en caso de interrupción o fallo.

Nota. Capacidad de desempeñar una función específica, se divide en 4 características para evaluación aspectos de la fiabilidad.

- **Seguridad.** - Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Tabla 6.

Definición de métricas y fórmulas de la seguridad

	METRICA	FORMULA	DESCRIPCION
CONFIDENCIALIDAD	Numero de métodos de recuperación y validación de datos	$M = (x - y) * NI$ x: número de formas de recuperación o validación de datos y: número de formas de recuperación o validación de datos no validad en la plataforma	Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente. Si se cumple que $x=y$ es afirmativo Si x es mayor que y es negativo

	METRICA	FORMULA	DESCRIPCION
	INTEGRIDAD	$I = (x - y) * NI$ x: número de solicitudes de cambio y: número de mensajes de alerta recibidos	Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
	AUTENTICIDAD	Numero de formas de autenticación $A = (x - y) * NI$ X: número de formas posibles de autenticación Y: número de formas que no permite autenticación	Capacidad de demostrar la identidad de un sujeto o un recurso Si se cumple que x=y es afirmativo Si x es mayor que y es negativo
	RESPONSABILIDAD	Numero de la plataforma de rastrear ingresos maliciosos x: número de ingresos maliciosos y: número de rastreos exitosos	Capacidad de rastrear de forma inequívoca las acciones de una entidad.

Nota. Para proteger la información y los datos. Personas no autorizadas no pueden leer o modificar datos.

- **Mantenibilidad.** - Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Tabla 7.

Definición de métricas y fórmulas de la mantenibilidad

	METRICA	FORMULA	DESCRIPCION
MODULARIDAD	Número de módulos concatenado	$M = \frac{(x - y) + 100}{10}$ x: Número total de módulos y: número de módulos no concatenados	Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás
REUSABILIDAD	Números de atributos y funciones heredadas	$R = \frac{(x - y) * NI}{10}$ x: suma de atributos y funciones de todos los objetos del software y: suma de atributos y funciones heredadas de todos los objetos	Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
ANALIZABILIDAD	Fallas diagnosticadas	$A = \frac{(x - y) * 100}{10}$ x; número total de fallas encontradas y: número total de fallas no diagnosticadas	- Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
CAPACIDAD DE SER MODIFICADO	Complejidad de modificación	$CM = \frac{x}{y}$ x: número de modificaciones y: tiempo que le toma al desarrollador hacer las modificaciones	Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.

	METRICA	FORMULA	DESCRIPCION
CAPACIDAD DE SER PROBADO	Numero de reinicios para pruebas	$A = \frac{(x - y) * 100}{x}$ x: número de total de reinicios y: número de reinicios fallidos	- Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

Nota. La evaluación de la mantenibilidad se divide en 5 características.

- **Capacidad para ser probado. 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. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Tabla 8.

Definición de métricas y fórmulas de la portabilidad

	METRICA	FORMULA	DESCRIPCION
ADAPTABILIDAD	Capacidad de adaptarse a otros entornos	A: capacidad de adaptarse a otros entornos, tabla de cumplimiento	Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
CAPACIDAD PARA SER INSTALADO	Numero de pasos previo a la de instalación	$D = (10 * x) - (10 * (x - 1))$ x: número de pasos para la instalación	Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.

	METRICA			FORMULA	DESCRIPCION
CAPACIDAD PARA SER REEMPLAZADO	Capacidad de ser reemplazado			CR=capacidad de ser reemplazado tabla de cumplimiento	Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

Nota. Es la capacidad de la plataforma para ser transferir exitosamente a cualquier entorno, y este de divide en 3 características.

2.7.4. ISO/IEC 2504n – división de evaluación de calidad

Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Esta división se encuentra formada por:

- **ISO/IEC 25040 - Evaluation reference model and guide** (Modelo de referencia, evaluación y guía). - Propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.

Tabla 9.

ISO/IEC 25040- proceso para evaluación de software

ACTIVIDAD	DESCRIPCION
Actividad 1	Establecer los requisitos de la evaluación Establecer el propósito de la evaluación: Tarea destinada a indicar el propósito por el cual la organización quiere evaluar la calidad de un producto de software. Obtener los requisitos de calidad del producto: Requiere identificar las partes interesadas en el producto (desarrolladores, posibles adquirentes, usuarios, proveedores, entre otros). Además, se describen los requisitos de calidad del producto utilizando un modelo de calidad. Identificar las partes del producto que se deben evaluar: Se deben identificar y documentar las partes del producto de software 47 incluidas en la evaluación. Tener en cuenta que el tipo de producto a evaluar depende de la fase en el ciclo de vida en que se realiza la evaluación. Definir el rigor de la evaluación: Se busca definir el rigor de la evaluación basándose en el propósito y en el uso previsto del producto de software. Al hablar de riesgos se hace referencia a diferentes tipos, tales como riesgos para la seguridad, riesgos económicos o riesgos ambientales.

ACTIVIDAD	DESCRIPCION
Actividad 2 Especificar la evaluación	Seleccionar los módulos de evaluación: El evaluador es encargado de seleccionar las métricas de calidad, técnicas y herramientas que cubran todos los requisitos de la evaluación. Dichas métricas deben poder compararse con los criterios definidos para poder tomar decisiones. Para esta tarea, se puede tener en cuenta la norma ISO/IEC 25020. Definir los criterios de decisión para las métricas: Tarea encargada de definir los criterios de decisión, los cuales son umbrales numéricos que se pueden relacionar con los requisitos de calidad y con los criterios de evaluación para decidir la calidad del producto. Definir los criterios de decisión de la evaluación: Se deben definir los criterios para las diferentes características evaluadas. Estos resultados, en un mayor nivel de abstracción, permiten realizar la valoración de la calidad del producto en forma general.
Actividad 3 Diseñar la evaluación	Planificar las actividades de la evaluación: Requiere planificar las actividades de la evaluación, teniendo en cuenta la disponibilidad de los recursos humanos y materiales necesarios, el presupuesto, los métodos de evaluación y estándares adaptados, las herramientas de evaluación, entre otros.

ACTIVIDAD	DESCRIPCION	
Actividad 4	Ejecutar la evaluación	Realizar las mediciones: Realizar las mediciones sobre el producto de software para obtener los valores de las métricas seleccionadas e indicadas en el plan de evaluación. Todos los resultados deben ser registrados. Aplicar los criterios de decisión para las métricas: Aplicar los criterios para las métricas sobre valores obtenidos en la medición de un producto. Aplicar los criterios de decisión de la evaluación: Aplicar los criterios de decisión de la evaluación, generando como resultado el grado en el que el producto cumple con los requisitos de calidad establecidos.
Actividad 5	Concluir la evaluación	Revisar los resultados de la evaluación: El evaluador y el cliente (si existe) son los encargados de revisar los resultados obtenidos en la evaluación, con el objetivo de realizar una mejor interpretación y una mejor detección de errores Crear el informe de evaluación: Una vez analizados los resultados, se busca elaborar un informe de evaluación, indicando 49 los requisitos, los resultados, las limitaciones y restricciones, el personal evaluador, entre otros. Revisar la calidad de la evaluación: El evaluador se encarga de revisar los resultados de la evaluación y la validez del proceso, de los indicadores y de las métricas aplicadas. En base a esto se obtiene un feedback, el cual debe servir para mejorar el proceso de evaluación. Tratar los datos de la evaluación: Al concluir la evaluación, el evaluador debe realizar el tratamiento de los datos según lo acordado con el cliente, ya sea devolviéndolos, modificándolos, guardándolos.

2.8. SEGURIDAD DE SOFTWARE ISO-27001

La norma ISO 27001 es un estándar internacional que establece los requisitos para la implementación, mantenimiento y, mejora continua de un sistema de gestión de la seguridad de la información (SGSI). Este sistema se realiza para proteger la confidencialidad, integridad y disponibilidad de la información. La norma proporciona un marco para la seguridad de la información de ya que ayuda a identificar y gestionar los riesgos de seguridad de la información de una manera más efectiva.

La norma ISO 27001 tiene las siguientes características:

- **Confidencialidad de datos:** Tiene como propósito garantizar la seguridad al momento de ingresar a la información, no permite divulgar la información a personas ajenas a la institución o empresa, con ello se busca conseguir una seguridad óptima que solo los usuarios registrados puedan ingresar.
- **Disponibilidad de datos:** tiene como propósito que los datos o información del sistema estén disponibles para los usuarios en lo posible las 24 horas del día
- **Integridad de datos:** Hace referencia a los datos que no pueden ser alterados por ningún tipo de personal. Teniendo como referencia al rol que pertenezca el usuario y la jerarquía en la institución.

El proceso de implantación de la norma ISO 27001 se divide en cuatro fases: planificación, implementación, evaluación y mejora continua.

- **Fase de planificación.** - En esta fase se identifica los requisitos de seguridad de la información y establece un plan para implementar el SGSI.

- **Fase de implementación.** - La fase de implementación incluye la creación de políticas, procedimiento y controles para proteger la información.
- **Fase de evaluación.** - Durante esta fase se evalúa la eficacia de su SGSI e identifica área de mejora-
- **Fase de mejora continua.** - Esta fase implica la identificación y aplicación de mejoras a los procesos y controles del SGSI. (GlobalSuite, 2024)

2.9. PRUEBAS DE FUNCIONAMIENTO

Las pruebas de funcionamiento tienen como objetivo encontrar errores en el sistema, entonces todo software debe diseñarse e implementarse teniendo en cuenta la “con probabilidad”. Así mismo las pruebas deben mostrar un conjunto de características que logren la meta de encontrar la mayor cantidad de errores con el mínimo esfuerzo (Pressman, pág. 412)

2.9.4. Prueba de Caja blanca

La prueba de caja blanca o pruebas estructurales, es una técnica de diseño de prueba que usa la estructura de control para obtener casos de prueba,

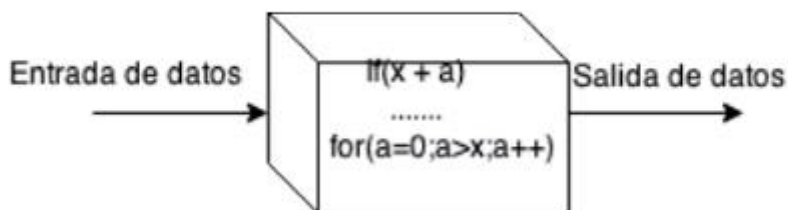
Dentro de esta estructura de control podemos encontrar la estructura componente, puede ser sentencias de decisiones, caminos distintos de código. En pruebas puede salir los siguientes casos:

- Garantizar que las rutas del código se revisen al menos una vez
- Revisa las condiciones lógicas
- Ejecuta los bucles de sus fronteras y de dentro de sus fronteras

- Revisan la estructura de datos

Figura 10.

Prueba de caja blanca



Nota. Adaptado de *pruebas de software, fundamentos técnicos* (p.31), por (Sanchez Maza, 2012)

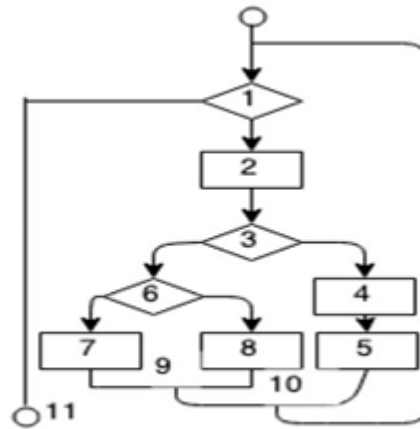
2.9.5. Pruebas de ruta básica

Es un método donde se definen casos de pruebas en función al flujo o rutas lógicas, el objetivo de las rutas básicas es definir la cantidad de rutas independientes, de modo que la cantidad de casos de pruebas necesarias se pueda definir para maximizar la cobertura de la prueba.

La prueba de ruta básica implica la ejecución de todos los bloques posibles en un programa y logra la máxima cobertura de ruta con el menor número de casos de prueba. Es un método híbrido de pruebas de rama y métodos de prueba de ruta,

la presentación que se conoce como grafo de flujo y las rutas de programa independiente:

- **Notación de grafo de flujo:** se realiza a partir de PDL (Program Design Language) o pseudocódigo, con el objetivo de mapear el grafo de flujo correspondiente.

Figura 11.*Diagrama de flujo*

Nota. Adaptado de ingeniería de software (p. 415), por (Pressman, 2010)

- **Complejidad ciclomatica:** Proporciona una medición de software, es un conjunto básico que garantiza que toda la ilustración será ejecutada al menos una vez.

Permite definir el número de caminos independientes del conjunto básico de un programa.

Número de regiones de gráfico de flujo que corresponde a la complejidad Ciclomática.

La complejidad Ciclomática $V(G)$ para un grafo de flujo G se define como

$$V(G) = E - N + 2$$

Donde E es el número de aristas del grafo de flujo

N el número de nodos del grafo de flujo

También se define como

$$V(G) = P + 1$$

Donde P es el número de nodos predicho contenidos en el grafo de flujo

- **Matrices de Grafos:** Una matriz de grafo es una matriz cuadrada cuyo tamaño (es decir, número de filas y columnas) es igual al número de nodos del gráfico de flujo. Cada fila y columna corresponde a un nodo identificado y las entradas de la matriz corresponden a conexiones (una arista) entre nodos (Pressman, pág. 420)

Figura 12.

Matriz de grado

Conectado a nodo		1	2	3	4	5
Nodo						
1			a			
2						
3		d		b		
4		c			f	
5		g	e			

Matriz de grafo

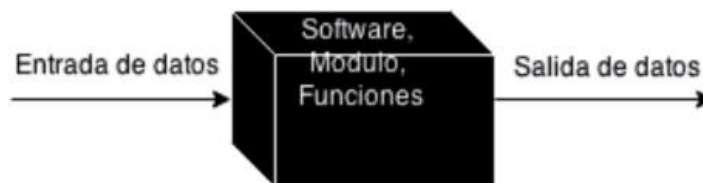
Nota. Adaptado de ingeniería de software (p. 420), por (Pressman, 2010)

2.9.6. Prueba de Caja negra

Llamadas también pruebas de comportamiento, son las que utilizan el análisis de especificación, tanto funcional como no funcional, sin tener en cuenta la estructura interna del sistema para diseñar casos de pruebas. (Maldonado Ramirez & Alva Zapata, 2021)

Figura 13.

Prueba de caja negra



Nota. Adaptado de pruebas de software (p.15), por (Maldonado Ramírez A. Z., 2021)

Las técnicas más comunes son:

- **Partición de equivalencia:** Es un método de caja negra que divide en dominio en entrada de un programa en clase de datos que pueden derivarse casos de prueba. Se espera que el programa tenga un comportamiento común. Según su técnica consta de dos pasos:

Identificar las clases de equivalencia

Identificar los casos de prueba

- **Análisis de valores límite:** completan a la partición equivalente, se debe verificar y poner mucha atención en los límites deben estar correctamente definidos y programados. Esta técnica nos lleva a elegir los casos de prueba que ejercen los valores límites.
- **Transición de estado:** Nos indica que todos los sistemas se mueven por transiciones de un paso a otro, se puede guiar de transiciones válidas e inválidas.
- **Tablas de decisión:** Consideran que para encontrar el resultado esperado deben estar en conjunto varias condiciones.

2.9.7. Pruebas de estrés

Las pruebas de estrés de software también conocidas como pruebas de resistencia, son utilizados durante la etapa de testing, con el objetivo de probar los límites de un sistema y prever escenarios de riesgo ante cargas extremas (Maldonado Ramirez & Alva Zapata, 2021)

Si el sistema no se encuentra en condiciones óptimas responderá con errores o comportamientos anormales, como fallos de código o bloqueos de información es por eso que es indispensable realizar las pruebas de estrés para evitar cuando ya se ponga en producción.

Tipos de prueba de estrés

- **De aplicación:** Se efectuarán ante posibles errores asociados al bloqueo de información y fallas en la conexión.
- **Transaccionales:** Se enfocan en las aplicaciones que tienen un intercambio de datos entre sí.
- **Simétricas:** Son aplicables cuando es necesario probar el funcionamiento de dos o más sistemas que operan bajo el mismo servidor.
- **Exploratorias:** Se ejecutan cuando el sistema es susceptible a fallos en condiciones inesperadas, como la saturación de la plataforma. (Maldonado Ramirez & Alva Zapata, 2021)

2.9.8. Pruebas de accesibilidad

Las pruebas de accesibilidad de un software son un conjunto de evaluaciones diseñadas para asegurar que una aplicación o página web sea accesible para el usuario final.

Estas pruebas verifican que el software cumpla con los estándares de accesibilidad establecidos y sea usable.

- **Elementos que se evalúan en las pruebas de accesibilidad:**

Textos alternativos: Para imágenes y otros elementos no textuales, como gráficos o videos.

Navegación mediante teclado: Verificar que todas las funciones sean accesibles sin usar un ratón.

Contraste y colores: Asegurar que el texto y los elementos sean legibles para personas con daltonismo o problemas de visión.

Tamaño y legibilidad de la fuente: Asegurarse de que el texto sea legible para personas con baja visión o dificultades de lectura.

Estructura lógica y navegación: Verificar que la estructura del contenido sea clara y lógica para que los usuarios con discapacidades cognitivas puedan navegar fácilmente.

Accesibilidad móvil: Asegurarse de que la aplicación o sitio web sea accesible también en dispositivos móviles, donde la interfaz puede ser más limitada.

En resumen, las pruebas de accesibilidad buscan garantizar que todos los usuarios, independientemente de sus capacidades, puedan interactuar con el software de manera efectiva y sin barreras. (Revilla Muñoz & Carreras Montoto, 2020)

2.10. ESTIMACIÓN DE COSTO

La estimación de costos para proyectos de software, es un proceso para predecir los costos totales asociados con el desarrollo y su entrega. Los costos pueden incluir desde el personal y los equipos hasta los materiales, herramientas y servicios externos.

Existen tres principales parámetros que se deben calcular en los costos de software.

Costo de esfuerzo

Costos de hardware y software

Costos de viajes y capacitación

2.10.1. COCOMO II

Una de las tareas más importantes en la planificación de proyectos es la estimación de costos, el cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo para el desarrollo.

El modelo COCOMO se convirtió en uno de los modelos de estimación de costos más utilizados y estudiado en la industria, esta evolución a COCOMO II que es una jerarquía de modelos de estimación.

Se divide en tres modelos distintos de desarrollo:

- **Modelo1. Composición de aplicaciones:** Este modelo es utilizado para estimar el esfuerzo y tiempo de desarrollo, se usa durante las primeras etapas, cuando es primordial la elaboración de prototipos de las interfaces de usuario, la valoración del rendimiento y la evaluación de la madurez en la tecnología.

La fórmula propuesta de este modelo es:

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

Dónde:

PM: Meses personas

NOP: Tamaño de nuevo software a desarrollar expresado en puntos objeto y se calcula:

$$NOP = \frac{OP * (100 - \%reuso)}{100}$$

OP = Tamaño del software a desarrollar expresado en puntos Objeto

%reuso = Porcentaje de reusó que se espera lograr en el proyecto

PROD: es la productividad promedio determinada a partir del análisis de datos de proyectos.

- **Modelo 2. Diseño preliminar:** este modelo toma en cuenta la exploración a diferentes arquitecturas del sistema, este es usado una vez estabilizadas los requisitos y establecida la arquitectura del software.

la fórmula para el cálculo de esfuerzo:

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

$PM_{estimado}$: Es el esfuerzo Nominal ajustado por 7 factores que reflejan aspectos propuso del proyecto

EM_i : Corresponde a los factores de costo tienen un efecto multiplicativo.

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

A: Es una constante que captura los efectos sobre el esfuerzo (A=2.94)

KSLOC: es el tamaño del software expresado en líneas de código.

B: es un factor exponencial de escala, toma en cuenta las características relacionadas.

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

B < 1.0 es el proyecto exhibe económica de escala

Si el $B=1.0$ las economías y deseconomías de escala de equilibrio

Si el $B > 1.0$ el proyecto muestra deseconomía de escala.

- **Modelo 3. Post-arquitectura:** es el más detallado y se aplica cuando a la arquitectura del proyecto está completamente definida. Esta fase se aplica durante el desarrollo y mantenimiento del software.

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

Como todos los modelos de estimación para software, los modelos COCOMO II requieren información sobre el dimensionamiento. Como parte de la jerarquía del modelo, están disponibles tres diferentes opciones de dimensionamiento: puntos objeto, puntos de función y líneas de código fuente (Pressman, 2010)

Fórmula para hallar el factor de complejidad TCF:

$$TCF = (0.65 + 0.04 * PF)$$

El procesamiento de datos el punto función se basa de la siguiente formula:

$$PF = cuenta\ total * TCF$$

Factor LDC/PF se calcula con la fórmula:

$$LDC = PF * factor \frac{LDC}{PF}$$

Número estimado de líneas de código distribuido en miles se calcula con:

$$KLCD \frac{LDC}{1000}$$

Las ecuaciones de COCOMO básico tiene la siguiente forma:

$$E = a_b(KLCD)^{bb}$$

$$D = c_b D^{db}$$

Dónde:

E: esfuerzo aplicado en personas por mes

D: tiempo de desarrollo de mese cronológicos

KLDC: número estimado de líneas de código (en miles)

2.11. HERRAMIENTAS

2.11.1. Lenguaje de programación

- **JavaScript (JS)**

JavaScript es un lenguaje de programación de alto nivel y multiplataforma, lenguaje que es utilizado principalmente para la creación de aplicaciones web interactivas. Es un lenguaje de scripting que se ejecuta en el lado del cliente, lo que esto significa que se ejecute el navegador web del usuario final. JavaScript es utilizado para agregar interactividad, funcionalidad y dinamismo a las páginas web.

JavaScript proporciona las herramientas necesarias para desarrollar aplicaciones web modernas, con versatilidad, con alto rendimiento. (Haverbeke, págs. 6-8)

- **TypeScript**

Es un lenguaje de programación libre y de código abierto, es un superconjunto de JavaScript que añade tipos estáticos y objetos basados en clases. Algunas de las ventajas

que ofrece son: Agilidad de detectar errores y bugs, Sintaxis intuitiva, Usa NPM (Node Package Manager), es un gestor de paquetes con el que tiene infinidad de acceso a librerías, Permite trabajar con frameworks como Angular, jQuery, mongoDB, Ract y Vue.

- **NestJS. -**

Es un framework NodeJS construido sobre TypeScript y NodeJS, esta herramienta nos permite tener una estructura modular de código, lo que facilita el desarrollo de aplicación empresariales brindando solidez y escalabilidad. NestJS se centra en la creación de aplicaciones Backend.

NestJS tiene un enfoque basado en componentes, lo que significa que la aplicación se divide en módulos que contienen componentes, controladores y servicios relacionados. Está basado en TypeScript, por tanto, puede utilizar todas las características de TypeScript en su código como tipos estáticos, clases, herencias, legibilidad, interfaces, decoradores, entre otros (Ritter, 2020)

2.11.2. Base de datos

- **MariaDB/Mysql**

MariaDB es una de las bases de datos relacionales de código abierto más popular. Está creado por los desarrolladores originales de MySQL, es de código abierto, por tanto, MariaDB es optado por las distribuciones de Linux.

MariaDB se basa con los valores de rendimiento, estabilidad y apertura. A pesar de que MariaDB es una base de datos relacional, ofrece funciones similares a las de NoSQL en la versión 10. Entonces MariaDB ofrece las mismas características que MySQL, y se puede usar como sustituto directo del servidor de base de datos MySQL; diseñado para brindar

velocidad, fiabilidad y facilidad de uso, por tal razón puede ser utilizado para tareas de procedimientos tanto pequeño como a nivel empresarial. (MariaDB Foundation, 2009)

2.11.3. Framework

Es un marco de trabajo que tiene como objetivo facilitar la solución de problemas que surgen cuando estas programando, los frameworks aceleran el proceso de desarrollo, facilitando tareas como la organización del código. Un Framework también proporciona una estructura base para el código. Uno de los beneficios de un framework es la reutilización de código cuantas veces se requiera, esto reduce tiempo de trabajo y mayor productividad. (Bolloso Chacin, 2016)

- **Angular**

Es un framework de JavaScript de código abierto, escrito en TypeScript. Su objetivo principal es el desarrollo de aplicaciones de una sola página. Google se encarga del mantenimiento y constantes actualizaciones de mejoras de este framework. Algunas de las características de angular son: enrutamiento, administración de formularios, comunicación cliente-servidor.

La arquitectura de angular está en un marco MVC (modelo-vista-controlador), el cual proporciona herramientas necesarias para construir aplicaciones robustas y que puedan ser utilizados para todo tipo de proyectos.

Angular incluye: Un marco basado en componentes para crear aplicaciones web escalables.

Una colección de bibliotecas bien integradas que cubren una amplia variedad de funciones, incluido el enrutamiento, la gestión de formularios, la comunicación cliente-servidor y más.

Un conjunto de herramientas de desarrollo para ayudarlo a desarrollar, compilar, probar y actualizar su código.

Con Angular, está aprovechando una plataforma que puede escalar desde proyectos de un solo desarrollador hasta aplicaciones de nivel empresarial. Angular está diseñado para que la actualización sea lo más sencilla posible, así que aproveche los últimos desarrollos con el mínimo esfuerzo.

Lo mejor de todo es que el ecosistema de Angular consta de un grupo diverso de más de 1,7 millones de desarrolladores, autores de bibliotecas y creadores de contenido. (Angular.io, 2022) Angular es una plataforma que permite desarrollar aplicaciones web en la sección cliente utilizando HTML y JavaScript para que el cliente asuma la mayor parte de la lógica y descargue al servidor con la finalidad de que las aplicaciones ejecutadas a través de Internet sean más rápidas. El hecho de estar mantenido por Google, así como una serie de innumerables razones técnicas, ha favorecido su rápida adopción por parte de la comunidad de desarrolladores.

Permite la creación de aplicaciones web de una sola página (SPA: single-page application) realizando la carga de datos de forma asíncrona. Además de mejorar el rendimiento de las aplicaciones web, su utilización en dispositivos móviles está optimizada ya que, en ellos, los ciclos de CPU y memoria son críticos para su óptimo funcionamiento. Ello permite el desarrollo de aplicaciones móviles híbridas con Ionic 2. Gracias al uso de componentes, se puede encapsular mejor la funcionalidad facilitando el mantenimiento de las aplicaciones. Permite el uso de TypeScript (lenguaje desarrollado por Microsoft) con las ventajas que supone poder disponer de un tipado estático y objetos basados en clases.

Todo ello, gracias a la especificación ECMAScript 6, que es la base sobre la que se apoya TypeScript. Gracias a un compilador(transpilador) de TypeScript, el código escrito en

este lenguaje se traducirá a JavaScript original En la web Angular de podemos observar el diagrama de arquitectura que muestra cómo se relacionan los elementos principales de una app los cuales son los siguientes: Módulos, Componentes, Templates, Metadata, Data binding, Directives, Services, Dependency injection (Boada Oriols & Gómez Gutiérrez, 2019)

- **HTML**

HTML (Lenguaje de modelado de Hiper texto), son etiquetas para la construcción de páginas web, como imágenes, listas, videos entre otros, en donde HTML es el encargado de la estructura de la página. HTML es un lenguaje de marcación de elementos para la creación de documentos hipertexto. (Curso completo de HTML, pág. 2)

- **Css**

Css (Cascading Style Sheets –“Hojas de estilo en cascada”) es un lenguaje de hojas de estilos creado para controlar el aspecto de los documentos electrónicos que fueron definidos con HTML y XHTML. Básicamente en un lenguaje que maneja el diseño y presentación de las páginas web, es como el usuario final lo ve.

Se les denomina hojas de estilo en cascada, porque puede tener una o varias de las propiedades heredadas de otras. Al crear una página web, se utiliza principalmente HTML/XHTML para marcas los contenidos, es decir, para designar la función de cada elemento dentro de la página; párrafo, título, texto, tablas, lista de elementos, entre otros. Una vez creado los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cas elemento; color, tamaño, tipo de letra, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página,

Css permite separar el contenido HTML de su presentación final, es decir permite crear código HTML limpio y estructurado, mientras que el estilo se define como un archivo CSS separado.

Este orden mejora la legibilidad del código y facilita el mantenimiento, así como la actualización de los estilos.

En cuanto al rendimiento, Css permite cargar estilos externos en un archivo, con esto el navegador almacenado en cache los estilos y aplicara a todas las páginas con estilo, esto mejora el rendimiento al reducir la cantidad de datos que deben transferirse entre servidores y el cliente. (Gauchat, 2017)

- **NodeJS**

Es un entorno de tiempo de ejecución de JavaScript, este entorno de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un proyecto escrito en JavaScript. NodeJS fue creado por los desarrolladores de JavaScript, lo transformaron de algo que solo podía ejecutarse en el navegador en algo que se podría ejecutar en cualquier ordenador como si fuese una aplicación independiente. El objetivo principal de NodeJs es usar el modelo de entrada y salida sin bloqueo y controlado por eventos para seguir siendo liviano y eficiente frente a aplicaciones en tiempo real de uso de datos que se ejecutan en dispositivos. NodeJs realmente se caracteriza en la creación de aplicaciones de red rápidas, ya que es capaz de manejar una gran cantidad de conexiones simultaneas con alto nivel de rendimiento, lo que equivale a una alta escalabilidad. (Kiessling & Junge, 2015)

2.11.4. Editores de Código

Los editores de código permiten editar código fuente en diversos lenguajes de programación y ofrecen múltiples herramientas para facilitar el trabajo e incrementar la productividad.

- **Visual Studio Code**

Visual Studio Code es editor de código fuente desarrollado por Microsoft, que está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js, además cuenta con extensiones para diferentes tipos de lenguajes (C++, C#, Java, Python, PHP, Go) y tiempo de ejecución (.NET, Unity). VS Code tiene variedad de características para agilizar el trabajo, y es lo que hace que sea el preferido de los desarrolladores:

Multiplataforma: Es una característica importante en el desarrollo de cualquier aplicación. VS Code está disponible para Windows, macOS y Linux

IntelliSense: Está relacionada con la edición de código, autocompletado y resaltado de sintaxis, lo que permite agilizar a la hora de escribir código. Tal como su nombre lo indica, proporciona diferentes sugerencias de código y terminaciones, inteligentes en base a los tipos de variables, funciones y otros. Con la ayuda de extensiones se puede personalizar y conseguir un IntelliSense más completo para cualquier tipo de lenguaje.

Extensiones: Las extensiones nos permiten personalizar y agregar funcionalidad adicional de forma modular y aislada. Realmente las extensiones nos permiten tener una mejor experiencia y lo más importante no afecta en el rendimiento del editor, ya que se ejecuta en procesos independientes.

Depuración: VS Code incluye la función de depuración que ayuda a la detección de errores en el código. Esto permite agilizar el trabajo ya que se disminuirá el tiempo de búsqueda de errores de código, una de las ventajas es que es capaz de detectar pequeños errores de forma automática antes de ejecutar el código o la depuración como tal.

Control de versiones: VS code tiene compatibilidad con Git, por lo que se puede revisar diferencias, organizar archivos, realizar commits desde el editor, y hacer push y pull desde cualquier servicio de gestión de código fuente. (Gamarra, 2023)

2.11.5. Enterprise Architect

Es una plataforma de modelado, diseño y administración, colaborativa, basada en UML 2.5 y estándares relacionados. Enterprise Architect ayuda a trazar especificaciones de alto nivel para el análisis, diseño, implementación modelos de mantenimiento y prueba, usando UML, SysML, BPMN y otros estándares abiertos.

Enterprise Architect es ágil, intuitiva y extensible, con características importantes para dominios específicos totalmente integradas. Permite visualizar, analizar, modelar, probar, y mantener un amplio rango de sistemas, software, procesos y arquitecturas. (Enterprise Architect, 2023)

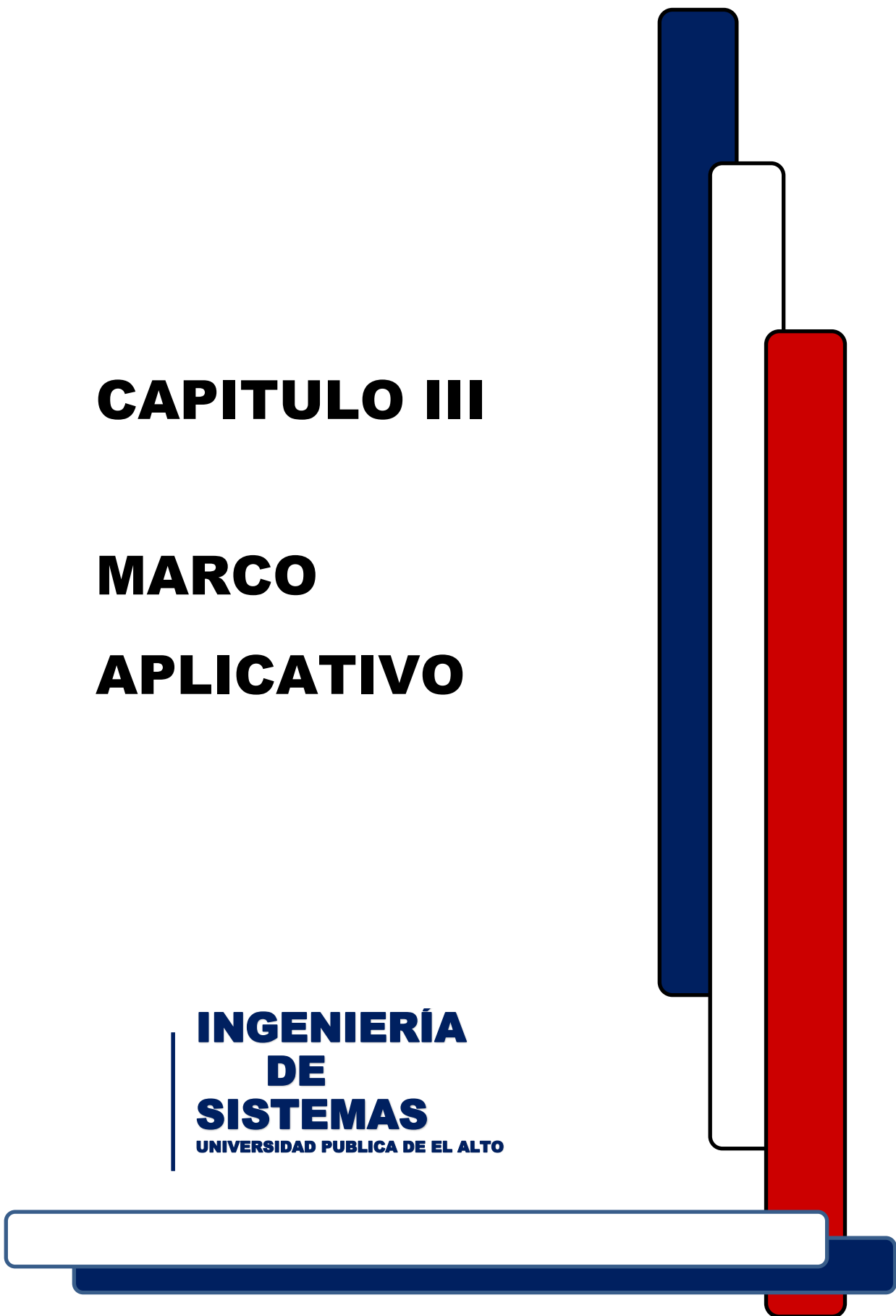
CAPITULO III

MARCO

APLICATIVO

**INGENIERÍA
DE
SISTEMAS**

UNIVERSIDAD PUBLICA DE EL ALTO



3.1. MARCO APLICATIVO

En esta parte del capítulo, se da paso al diseño y desarrollo del sistema, los diversos procedimientos y comportamientos más relevantes que se desarrollaran, con el fin de identificar correctamente los requerimientos, que son necesarios para el desarrollo del sistema, también se identificarán los principales actores con sus respectivas funciones dentro del sistema, de esta forma se logrará una adecuación desarrollada del sistema.

3.2. APLICACIONES DE LA METODOLOGIA UWE

3.2.1. Captura, análisis y especificaciones de requerimientos

Se utilizará como base de requerimiento, la ingeniería de requerimientos; requerimientos. Requerimientos funcionales y requerimientos no funcionales. Para el desarrollo correcto del sistema se empleará las herramientas de: captura, análisis, y especificaciones de requerimientos.

Tabla 10.

Herramientas usadas para la ingeniería de requerimientos

HERRAMIENTA	CARACTERISTICAS
Observación	Se prosiguió con la observación al proceso de almacenamiento de información, a los administrativos, docentes. Así también se observó la función de cada uno de los involucrados en el proceso.
Documentación	Se verificó, el tipo de documentos que se maneja y que de qué manera este es almacenado.
Entrevistas	Se realizó entrevistas con el personal involucrado en el proceso, los cuales fueron: Director de carrera, secretaria, docente

Luego de realizar el análisis de requerimientos, se procedió a la ejecución de las herramientas de la metodología UWE.

2.10.4.1. Requerimientos funcionales

Tabla 11.

Requerimientos funcionales

ROL	DESCRIPCION	FUNCION
RF-1	Acceder al sistema	Accede al sistema 4 tipos de usuarios (Administrador, Bibliotecaria, secretaria, Docentes)
RF-2	Gestionar usuarios	Permite crear, listar, actualizar, reset de contraseñas y eliminar registros.
RF-3	Gestionar permisos	Permite la asignación de permisos a los usuarios dependiendo del rol.
RF-4	Administración del software	Permite al usuario con el perfil del administrador, tener acceso total a los módulos y operaciones de la gestión de la información; creación, listado, actualización.
RF-5	Perfil de usuario	Muestra la información del usuario y permite la actualización de la contraseña.
RF-6	Gestionar el contenido de la plataforma, fotografías, autoridades de carrera, misión, visión, objetivo de carrera.	Permite crear, editar, activar, inactivar
RF-7	Gestionar la malla curricular y su contenido.	Permite crear. Editar, eliminar
RF-8	Gestionar materias de cada gestión.	Permite crear, editar, eliminar, leer.
RF-9	Gestionar la carga de contenidos de cada materia.	Permite crear, editar, eliminar, leer, carga PDF.

ROL	DESCRIPCION	FUNCION
RF-10	Gestionar libros digitales	Permite crear, editar los libros PDF's.
RF-11	Gestionar trabajos de investigación en sus diferentes niveles de grado.	Permite crear, editar, activar, inactivar.
RF-12	Generar reportes para informes de investigaciones, libros, trabajos de investigación	Permite creación de reportes en formato PDF.

2.10.4.2. Requerimientos no funcionales

Tabla 12.

Requerimientos no funcionales

ROL	FUNCION
RNF-1	El sistema debe funcionar en los diferentes navegadores. Tales como: Google Chrome, Mozilla Firefox, IE/Edge, entre otros.
RNF-2	El Sistema debe ser amigable para el usuario, con una interfaz gráfica acorde a los requerimientos.
RNF-3	El sistema deberá tener dos tipos de menús: unos que ser para el usuario general, y otro para los que manipularas la parte administrativa.
RNF-4	El sistema deberá encriptar las contraseñas de los usuarios.
RNF-5	El inicio de sesión tendrá un determinado periodo de tiempo activo
RNF-6	El sistema deberá responder a todas las solicitudes del usuario.
RNF-6	Registro de actividades con hora y fecha
RNF-7	La información que se mostrará en la página web, serán los últimos que se registraron
RNF-8	La información que se mostrará en la página web, serán los últimos que se registraron

2.10.4.3. Diseño del sistema

Para el diseño del sistema se utilizará: descripción de los actores, diagramas de casos de uso, diagrama de clases, modelo conceptual y modelos navegacional.

- **Descripción de actores**

La descripción de los actores nos permitirá identificar a los actores que interactúen en el sistema, teniendo en cuenta, el rol que cumple cada uno de los diferentes usuarios y entidades dentro de la carrera de Medicina entorno al manejo de trabajos de investigación de pregrado y postgrado, además de las diferentes actividades que se van realizando.

Tabla 13.

Descripción de actores

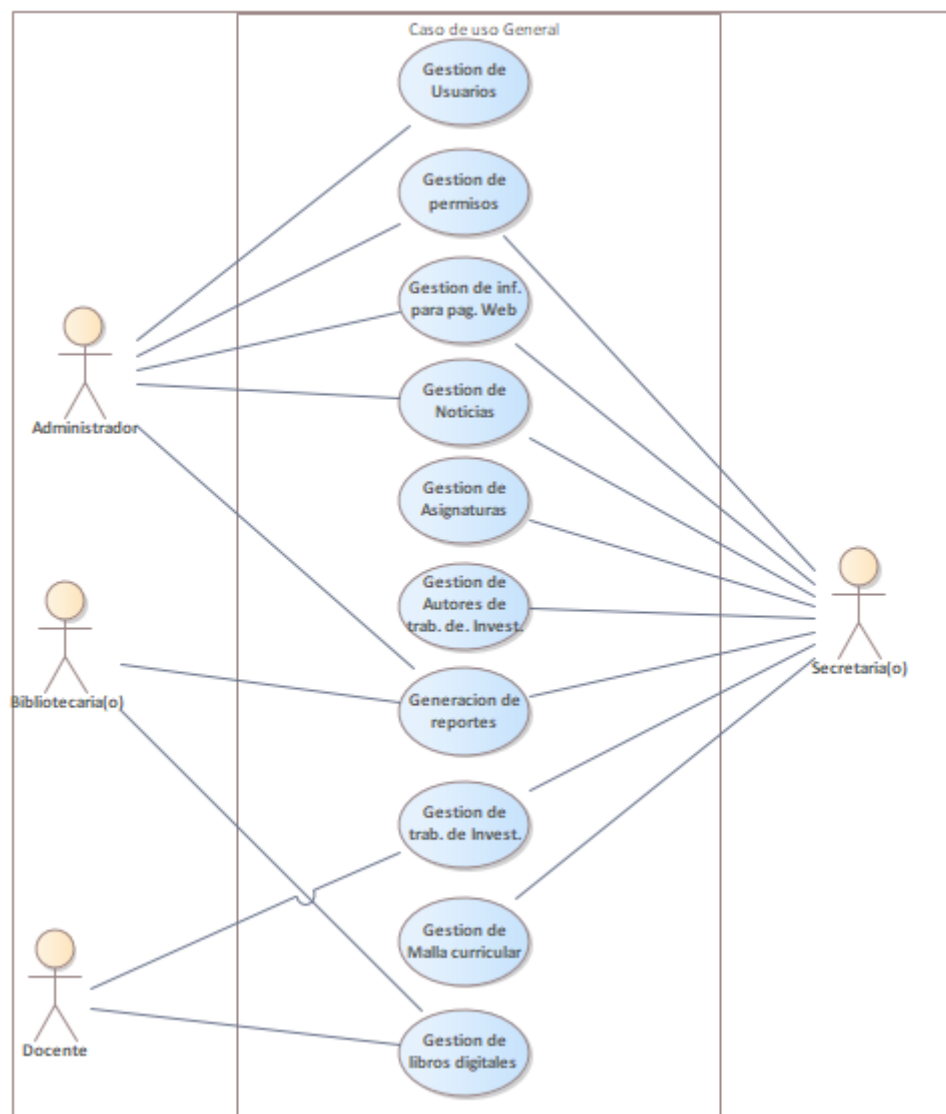
DESCRIPCION	RESPONSABILIDADES
Administrador	- Se encarga de gestionar los usuarios dentro del sistema web, la asignación de roles y permisos. - Se encarga de la gestión de la información que se muestra en la web (noticias, autoridades, actividades recientes, misión, visión, fotografías) - Se encarga de administrar los libros digitales. - Se encarga de administrar las materias y su contenido. - Se encarga de la gestión de las investigaciones hechas por los estudiantes (investigación, extensión e interacción social). Pregrado-Internado Rotatorio. - Se encarga de la gestión de las investigaciones de posgrado. - Se encarga de la gestión de materias y su contenido - Se encarga de la gestión de la malla curricular

DESCRIPCION	RESPONSABILIDADES
Secretaria	- Se encarga de la gestión de la información que se muestra en la web (noticias, autoridades, actividades recientes, misión, visión, fotografías) - Se encarga de administrar los libros digitales. - Se encarga de la gestión de las investigaciones hechas por los estudiantes (investigación, extensión e interacción social). Pregrado-Internado Rotatorio. - Se encarga de la gestión de las investigaciones de posgrado. - Se encarga de la gestión de materias y su contenido - Se encarga de la gestión de la malla curricular
Bibliotecaria	- Se encarga de administrar los libros digitales.
Docente	- Se encarga de la gestión de las investigaciones hechas por los estudiantes (investigación, extensión e interacción social). Pregrado-Internado Rotatorio.
Público General	- Visualización de información principal de la carrera de medicina, actividades de la carrera, visualización y/o descarga de libros digitales, de la malla curricular de la carrera.

2.11. Diagrama de caso de uso general del sistema

Figura 14.

Diagrama de caso de uso general del sistema



2.12. Diagrama de casos de uso específicos del sistema

Figura 15.

Diagrama de caso de uso-Gestión de usuario

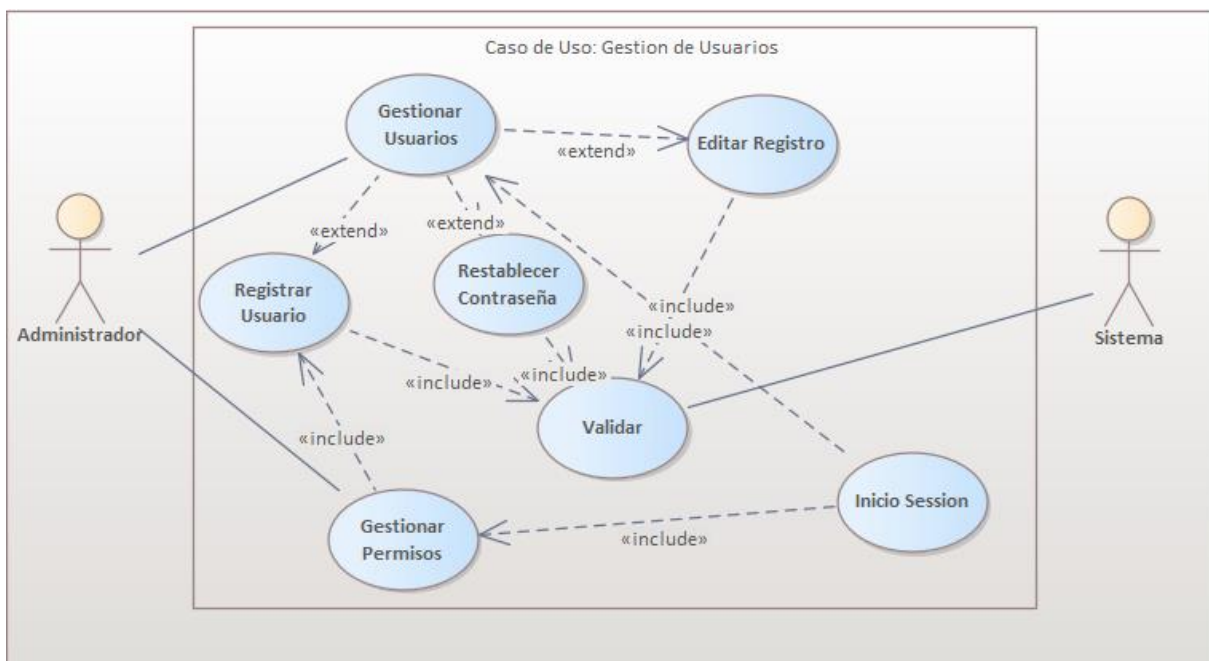


Tabla 14.

Descripción de caso de uso - Gestión de usuarios

Caso de uso	Gestión de Usuarios
Identificador	CU01
Descripción	El sistema permite gestionar a los usuarios registrados en el mismo
Actores	Administrador
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema

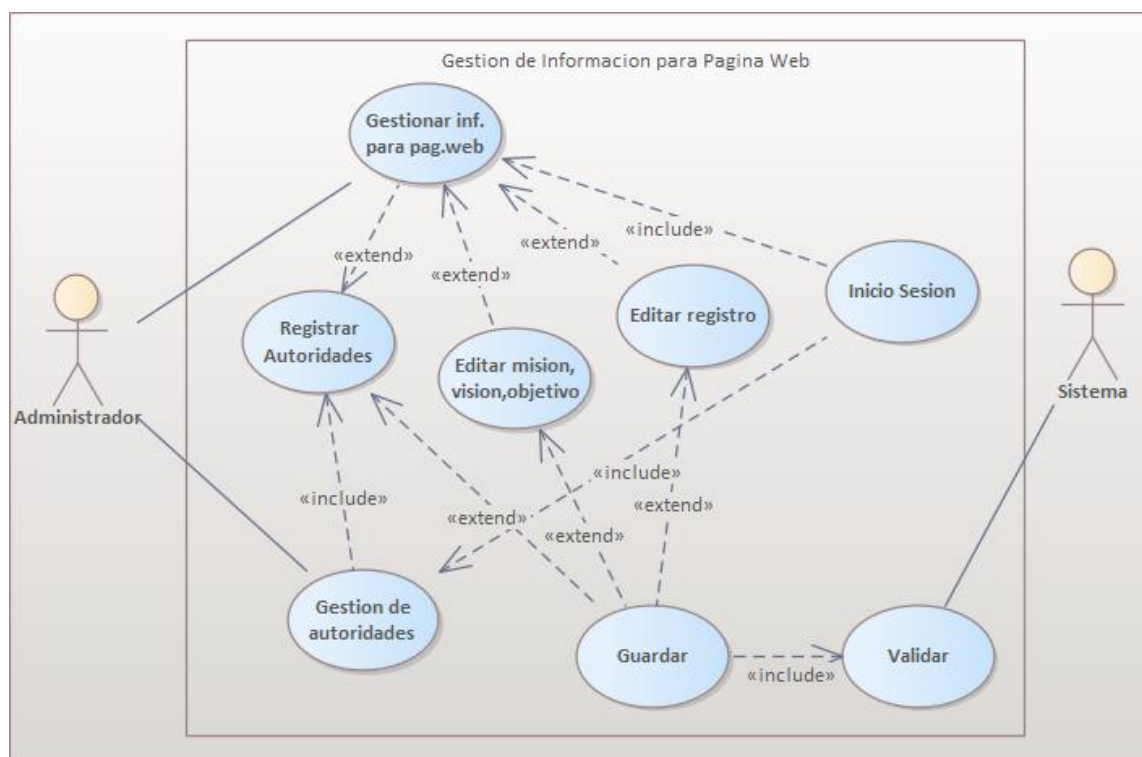
Caso de Uso	Gestión de usuarios
	El caso de uso responde cuando el administrador selecciona la opción de usuarios. El sistema despliega una lista de usuarios registrados 1. El usuario solicita crear, modificar, reset password, activar o desactivar un usuario.
Flujo de eventos	3. El sistema solicita la información para crear, modificar, reset de password, activar o desactivar un Usuario 4. El sistema valida los datos y realiza la acción solicitada
Post condiciones	El usuario logra crear modificar, reset password, activar o desactivar un usuario
Flujo alternativo	Si el usuario intenta crear a un usuario con datos ya existentes muestra un mensaje de error “El usuario ya existe”

Tabla 15.*Gestión de permisos*

Caso de uso	Gestión de Permisos
Identificador	CU01-1
Descripción	El sistema permite gestionar a los permisos que se otorgara a los usuarios.
Actores	Administrador, secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema
	El caso de uso responde cuando el administrador selecciona la opción de Permisos.
	El sistema despliega una lista de
	Usuarios registrados.
Flujo de eventos	- El usuario selecciona un registro determinado. - El usuario otorga los permisos pertinentes. - El usuario modifica el registro El sistema valida los datos y realiza la acción solicitada
Post condiciones	El usuario logra crear modificar el registro, otorgando permisos.
Flujo alternativo	Si el usuario intenta crear quitar permisos el sistema mostrará el mensaje de "No corresponde".

Figura 16.

Diagrama de caso de uso- Gestión de información para página web



Nota. Elaboración Propia

Tabla 16.

Descripción de caso de uso- Gestión de información de página web

Caso de uso	Gestionar información para página web
Identificador	CU02
Descripción	El sistema permite gestionar información que se verá reflejado en la página web
Actores	Administrador-secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema

Caso de uso	Gestionar información para página web
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y seleccionar Página: -El sistema muestra lista de autoridades registradas, muestra información importante para edición (misión, visión, objetivo): - El usuario modifica la descripción de Misión, Visión y objetivo. - El sistema valida la información y realiza la acción. - El usuario solicita crear un nuevo registro para autoridades. - El sistema solicita los datos necesarios para la creación del nuevo registro. - El usuario registra los datos requeridos - El sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada
Flujos alternativos	Si el usuario ingresa información ya registrado, el sistema muestra mensaje de "datos duplicados"

Figura 17.

Diagrama de caso de uso-Gestión de Noticias

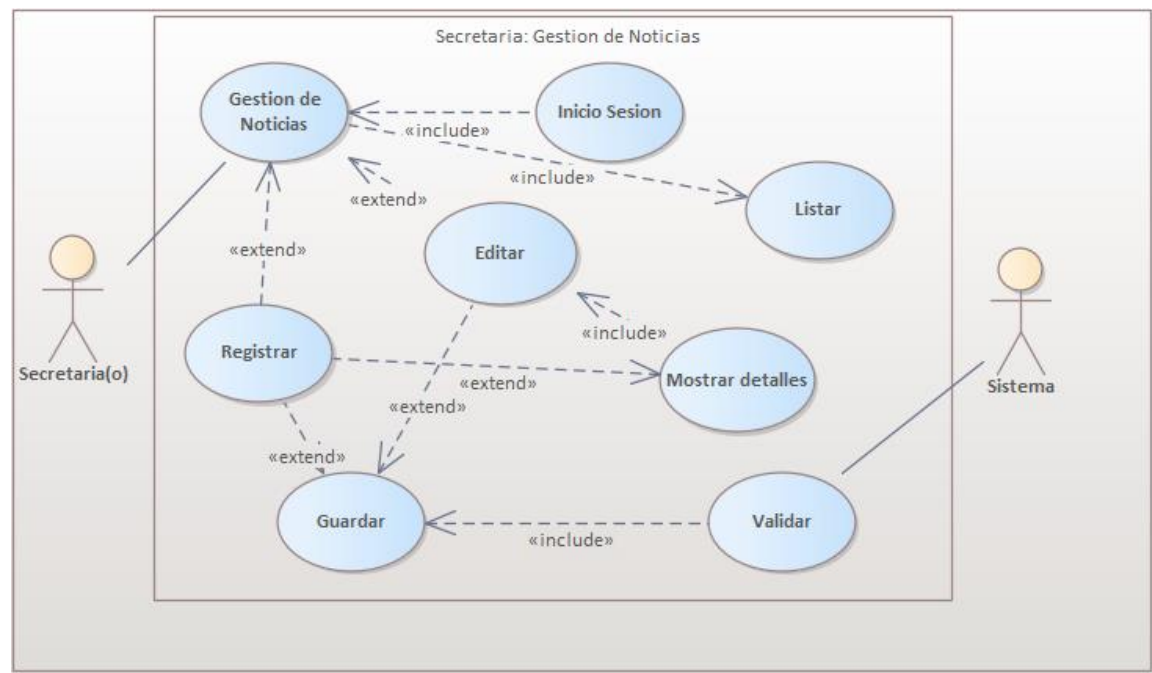


Tabla 17.

Descripción de caso de uso-gestión de noticias

Caso de uso	Gestión de Noticias
Identificador	CU03
Descripción	El sistema permite gestionar información que se verá reflejado en la página
Actores	Administrador-secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema.

Caso de uso	Gestión de Noticias
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y seleccionar Noticias: -El sistema despliega lista de las noticias, activas e inactivas - El usuario solicita crear una nueva noticia. - El sistema solicita los datos necesarios para la creación de nueva noticia - el usuario carga y/o registra los datos requeridos - El sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada
Flujos alternativos	Si el usuario ingresa información ya registrado, el sistema muestra mensaje de "datos duplicados"

Figura 18.

Diagrama de caso de uso- Gestión de asignatura

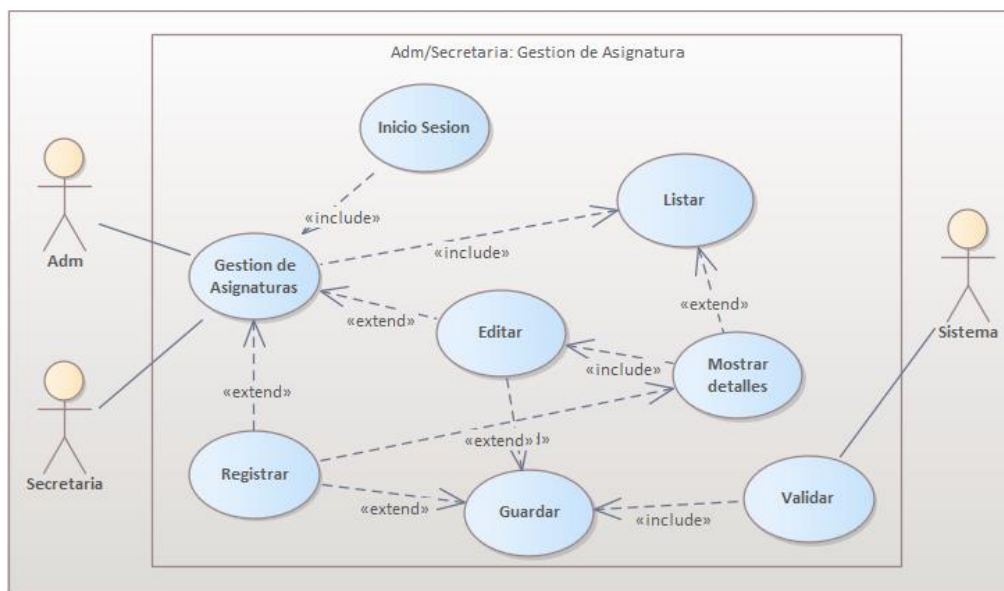


Tabla 18.*Descripción de caso de uso-Gestión de asignaturas*

Caso de uso	Gestión de asignaturas
Identificador	CU04
descripción	El sistema permite la gestión de materias con su respectivo contenido, además de visualizarlo en la página web
Actores	Administrador-secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y selecciona adm asignaturas: -El sistema despliega lista de las asignaturas registradas -El usuario solicita crear o insertar una nueva asignatura -El sistema solicita los datos necesarios para la creación del registro de la asignatura. -El usuario carga y/o registra los datos requeridos -El sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada
Flujo alternativo	Si el usuario ingresa una misma asignatura ya registrado, el sistema muestra un mensaje de "ya existe el registro"

Figura 19.
Diagrama de caso de uso-Gestión de Malla Curricular

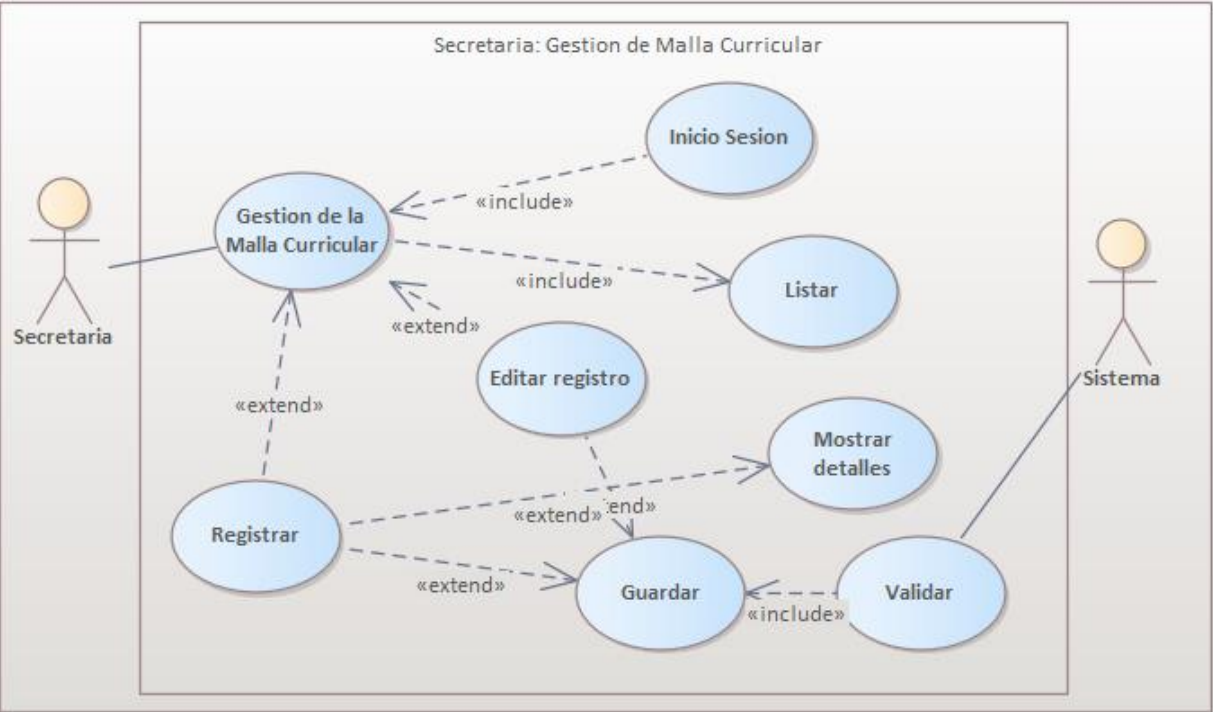


Tabla 19.
Descripción de caso de uso-Gestión de malla curricular

Caso de uso	Gestión de malla curricular
Identificador	CU05
Descripción	El sistema permite la gestión de la malla curricular
Actores	Administrador-secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema

Caso de uso	Gestión de malla curricular
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y selecciona malla curricular: -El sistema despliega lista de todas las mallas curricular registradas -El usuario solicita crear o insertar una nueva malla curricular -El sistema solicita los datos necesarios para la creación del nuevo registro de la malla curricular. -El usuario carga y/o registra los datos requeridos -El sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada
Flujo alternativo	Si el usuario ingresa una misma asignatura ya registrado, el sistema muestra un mensaje de "ya existe el registro"

Figura 20.

Diagrama de caso de uso-gestión de trabajos de investigación

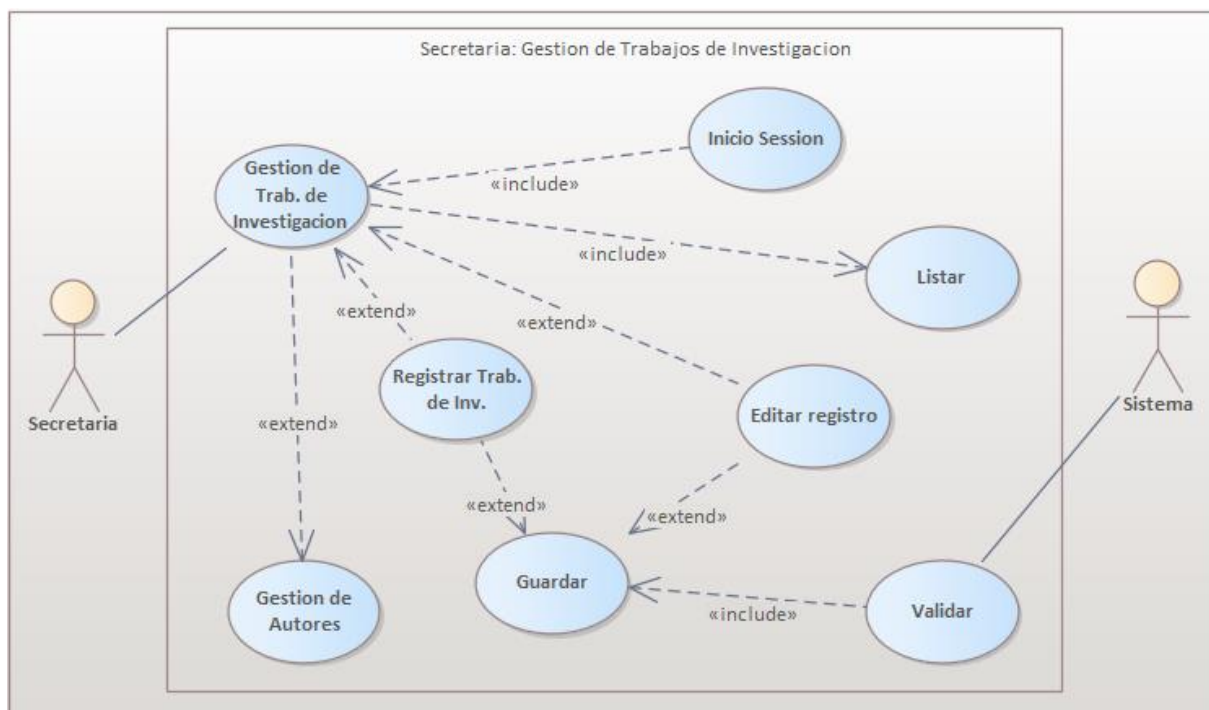
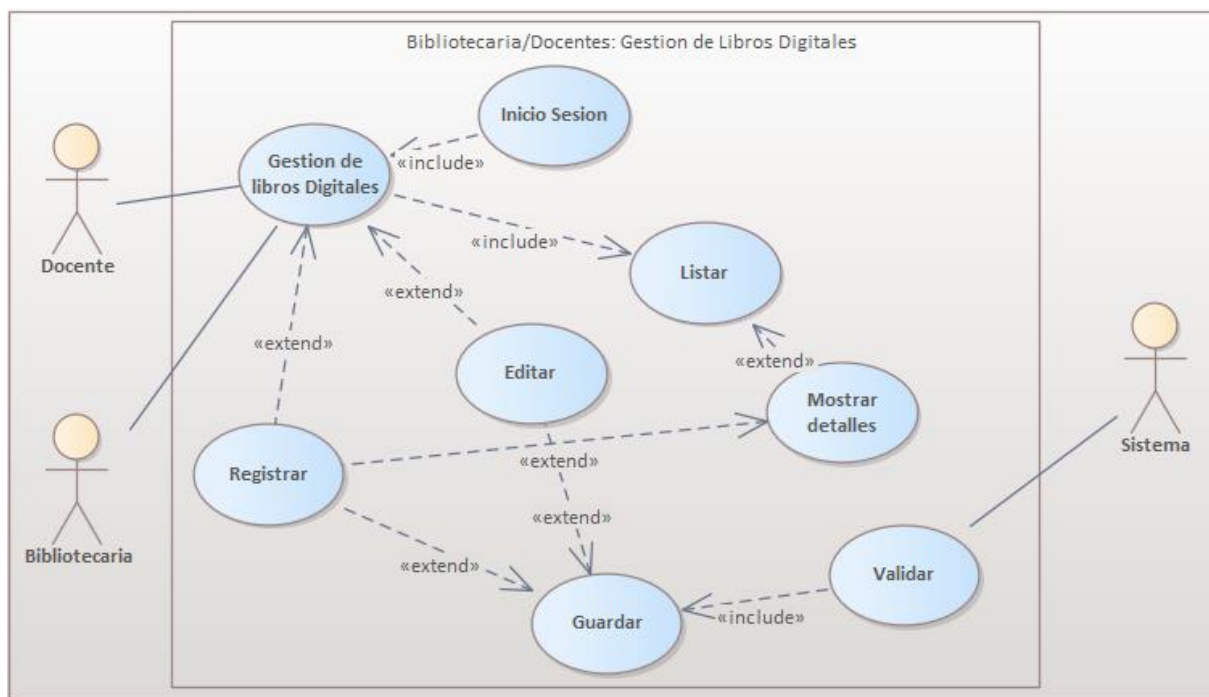


Tabla 20.*Descripción de caso de uso-Gestión de autoridades*

Caso de uso Gestión de Autores de Trabajo de Investigación	
Identificador	CU06
descripción	El sistema permite la gestión de los autores de los trabajos de investigación en sus diferentes niveles de educación (Pregrado/Posgrado)
Actores	Administrador-secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y seleccionar autores: -El sistema despliega lista todos los autores registrados. -El usuario solicita crear un nuevo registro de autor. -El sistema solicita los datos necesarios para la creación del nuevo registro para autor. -El usuario carga y/o registra los datos requeridos -El sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada
Flujo alternativo	Si el usuario ingresa una misma asignatura ya registrado, el sistema muestra un mensaje de "ya existe el registro"

Tabla 21.*Descripción de caso de uso- Gestión de trabajos de investigación*

Caso de uso	Gestión de Trabajos de Investigación
Identificador	CU06-1
Descripción	El sistema permite la gestión de los trabajos de investigación, nivel académico de posgrado
Actores	Administrador-secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y selecciona Trabajos de investigación: -el sistema despliega el registro de todos los trabajos registrados. -el usuario solicita crear o insertar una nueva malla curricular -el sistema solicita los datos necesarios para la creación del nuevo registro de la malla curricular. -el usuario carga y/o registra los datos requeridos -el sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada
Flujo alternativo	Si el usuario ingresa una misma asignatura ya registrado, el sistema muestra un mensaje de "ya existe el registro"

Figura 21.*Diagrama de caso de uso-Gestión de libros digitales***Tabla 22.***Descripción de caso de uso-Gestión de libros digitales*

Caso de uso	Gestión de libros digitales
Identificador	CU07
Descripción	El sistema permite la gestión de los libros digitales (PDF)
Actores	Administrador-secretaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema

Caso de uso	Gestión de libros digitales
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y selecciona Libros: -el sistema despliega lista de todas las mallas curricular registradas -El usuario solicita crear o insertar una nueva malla curricular -El sistema solicita los datos necesarios para la creación del nuevo registro de la malla curricular. -El usuario carga y/o registra los datos requeridos -El sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada
Flujo alternativo	Si el usuario ingresa una misma asignatura ya registrado, el sistema muestra un mensaje de "ya existe el registro"

Figura 22.

Diagrama de caso de uso-Generación de reportes

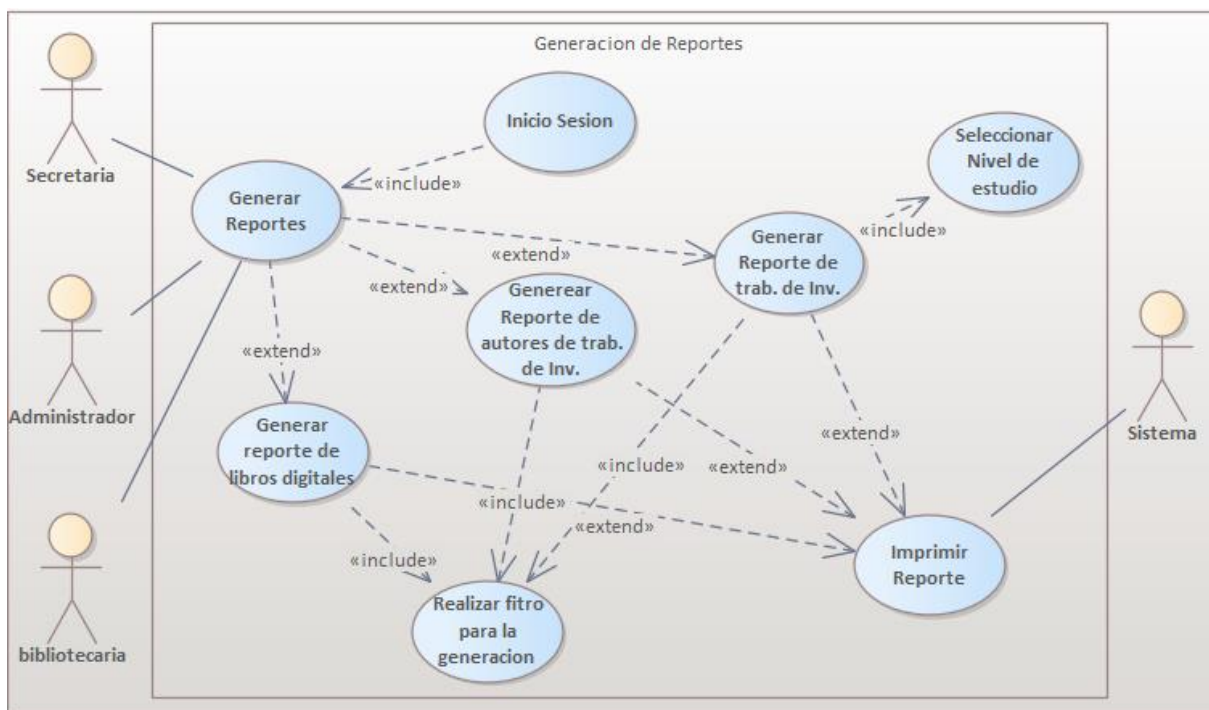


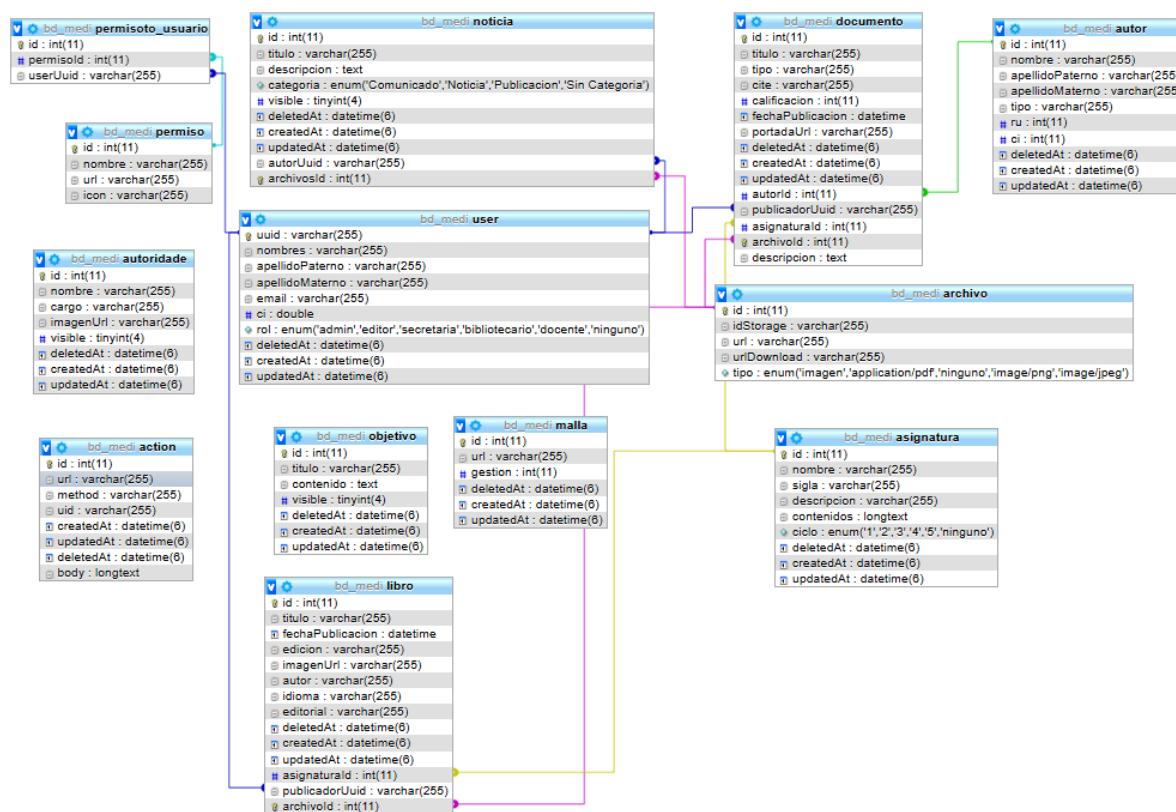
Tabla 23.*Descripción de caso de uso-Generación de Reportes*

Caso de uso	Generación de Reportes
Identificador	CU08
Descripción	El sistema permite la generación de reportes de los diferentes registros que existe en el sistema (trabajos de investigación, libros digitales, lista de autores)
Actores	Administrador, secretaria, Bibliotecaria
Precondiciones	El usuario debe estar registrado en el sistema, haber iniciado sesión y posteriormente haber ingresado con éxito al sistema
Flujo de eventos	El caso de uso responde cuando el usuario ingresa y se despliega a la barra de menú y selecciona libros digitales, Documentos, Autores: -El sistema despliega lista de todos los registros. -El sistema permite el filtro para realizar el reporte. -El sistema permite seleccionar el tipo de archivo para el reporte (Excel ó PDF) -El Usuario tiene la opción de generar reportes de los diferentes registros que existe. -El sistema valida toda la información y realiza la acción solicitada.
Post condiciones	El usuario logra la acción solicitada.
Flujo alternativo	Si el usuario no selecciona los campos, el sistema muestra el mensaje de # los campos son obligatorios”.

2.13. Diseño conceptual

Figura 23.

Diseño conceptual



2.14. Modelo de navegación

A continuación, se muestra la interacción de los usuarios con el software a través de la navegación por el sistema.

Figura 24.

Modelo de navegación - Pagina web

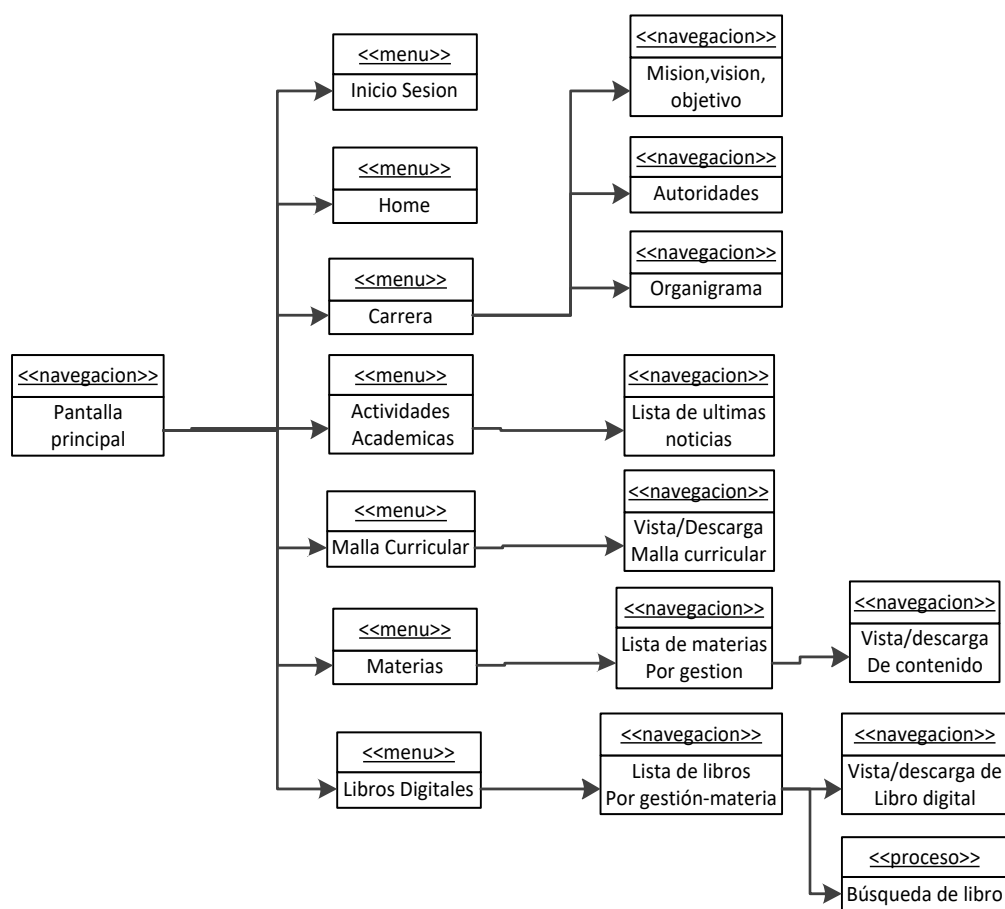


Figura 25.

Modelo de navegación -Administrador

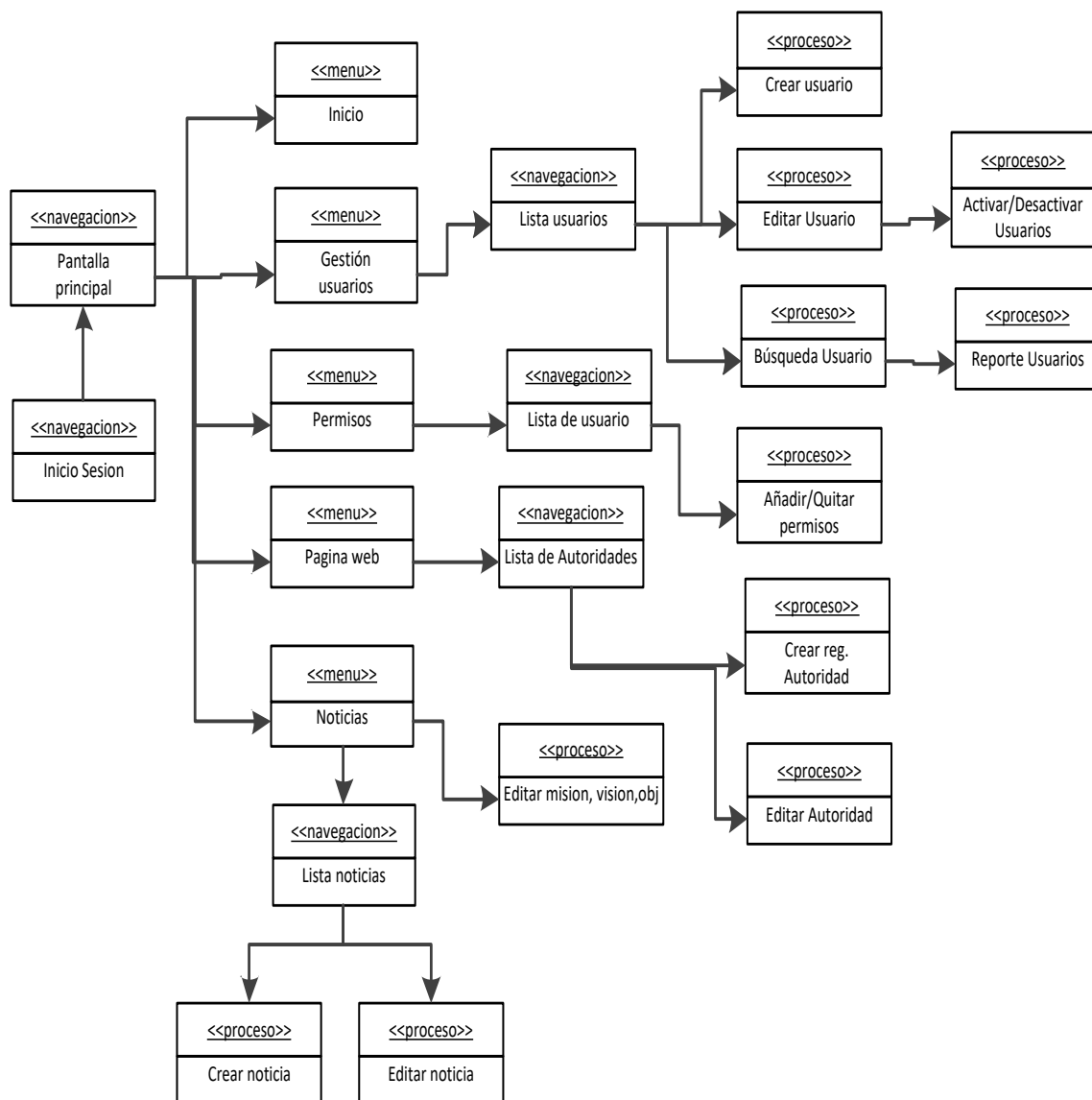


Figura 26. Modelo de navegación - secretaria

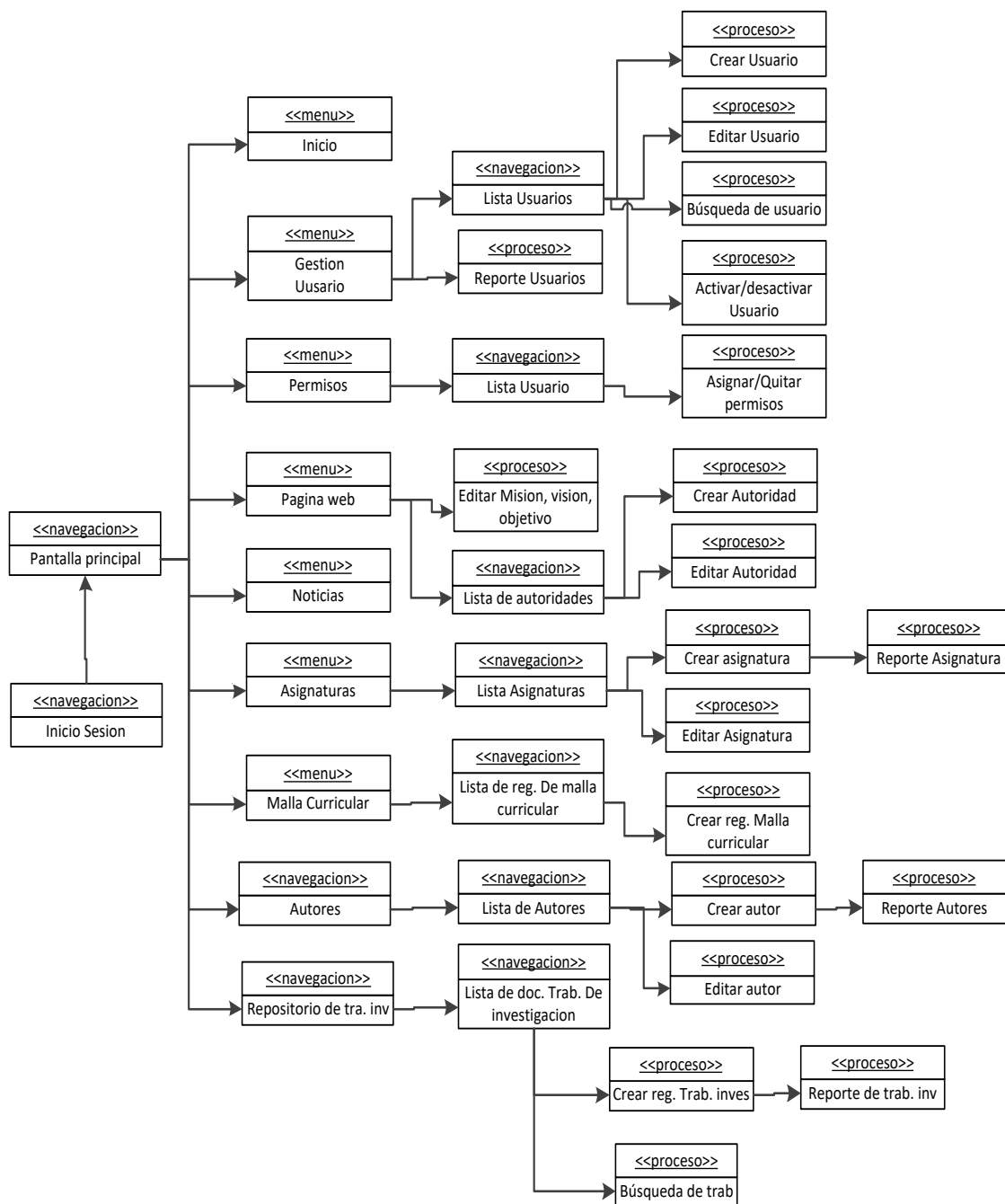
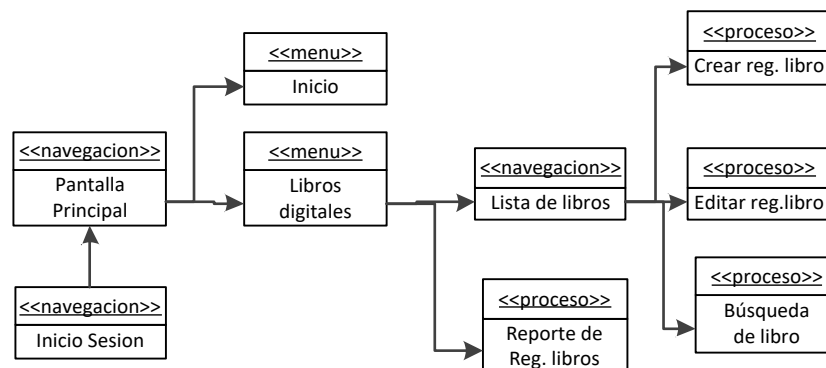
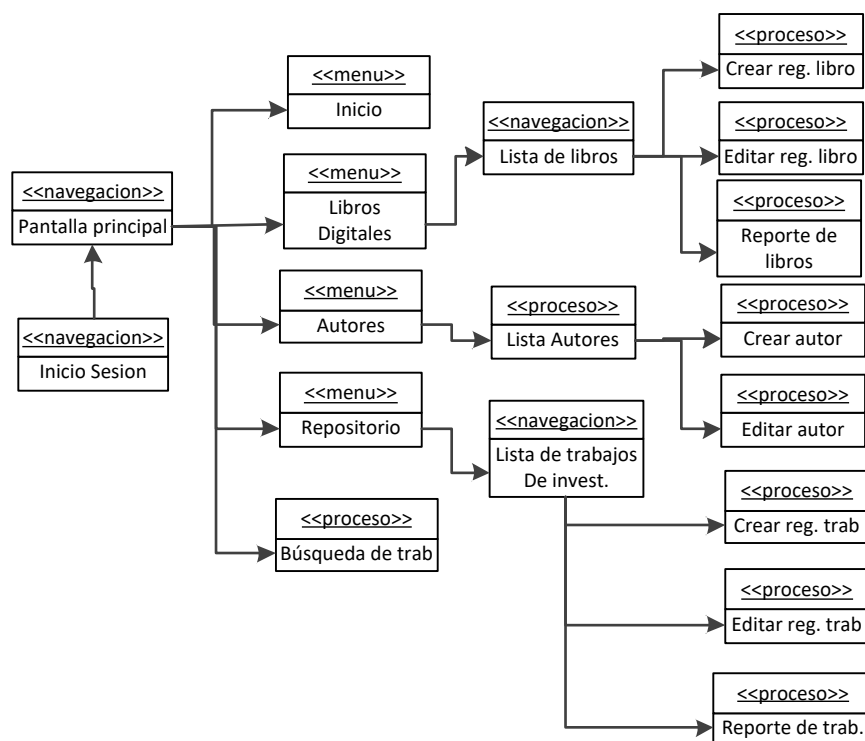


Figura 27.*Modelo de navegación- Bibliotecaria***Figura 28.***Modelo de navegación - Docente*

2.15. Modelo de presentación

En el desarrollo del diseño navegacional se busca que el usuario pueda localizar la información de su interés, de manera fácil y rápida de manera que tenga una experiencia adecuada y de esta manera llegar a la satisfacción del usuario.

Figura 29.

Prototipo de presentación de la página principal

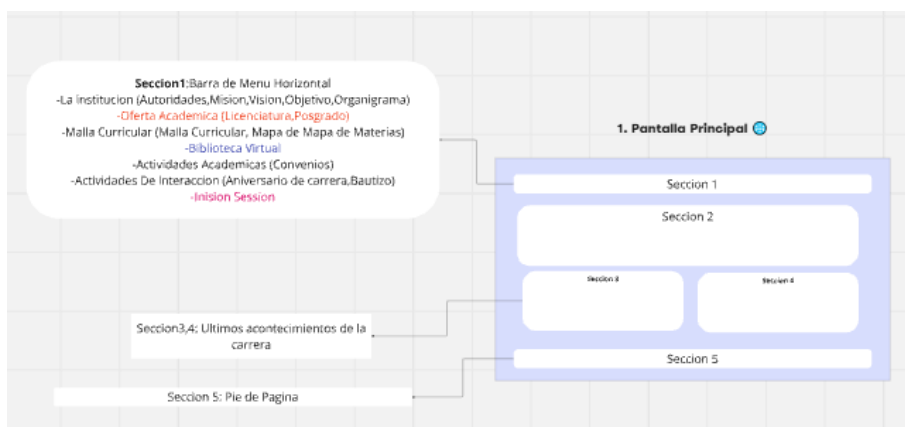


Figura 30.

prototipo de presentación información institucional

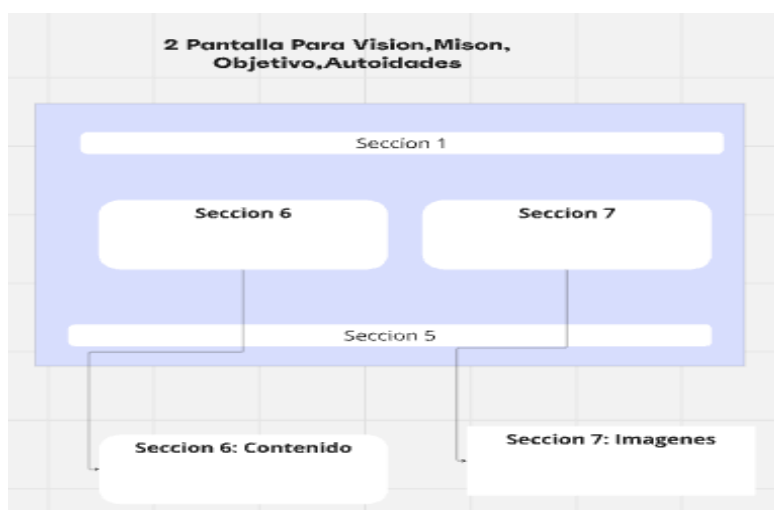


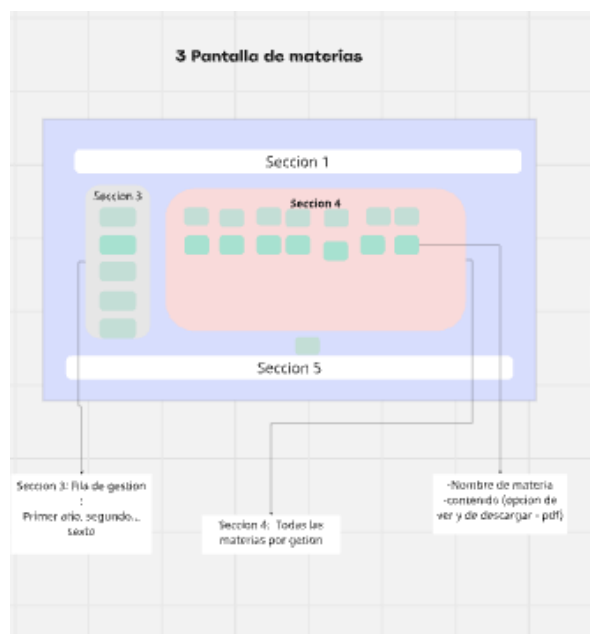
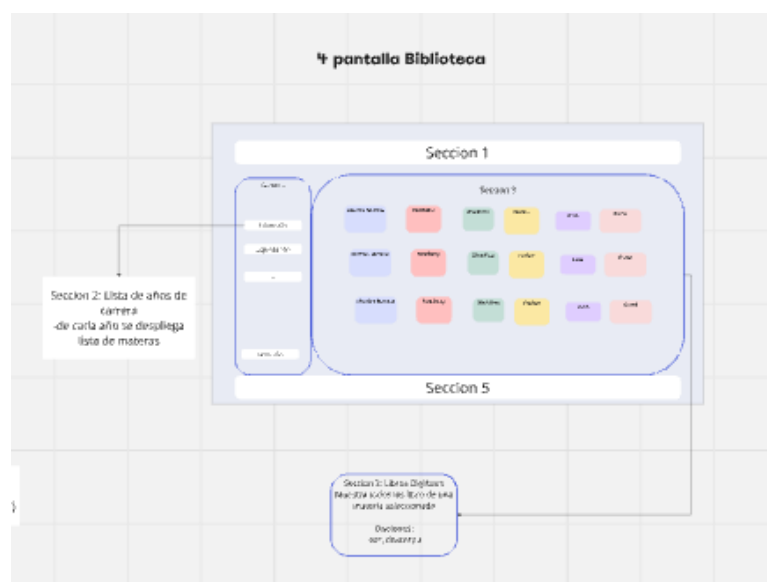
Figura 31.*Prototipo de presentación -asignaturas y contenido***Figura 32.***Prototipo de presentación biblioteca digital*

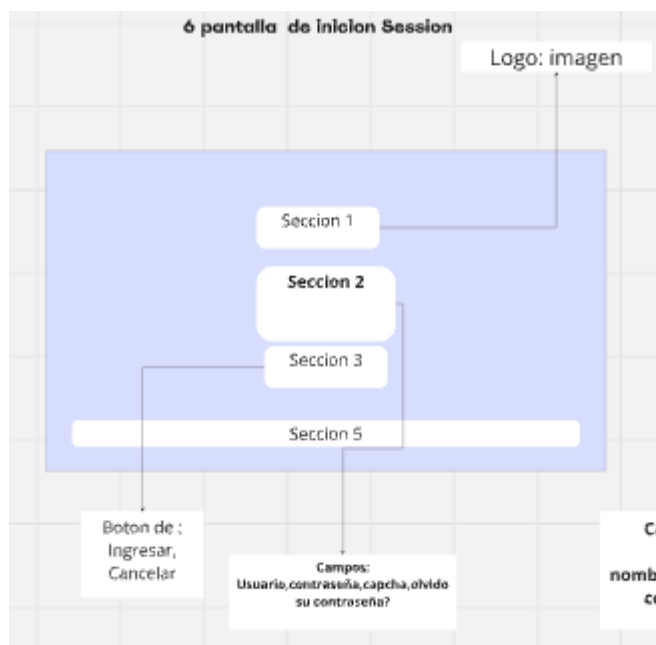
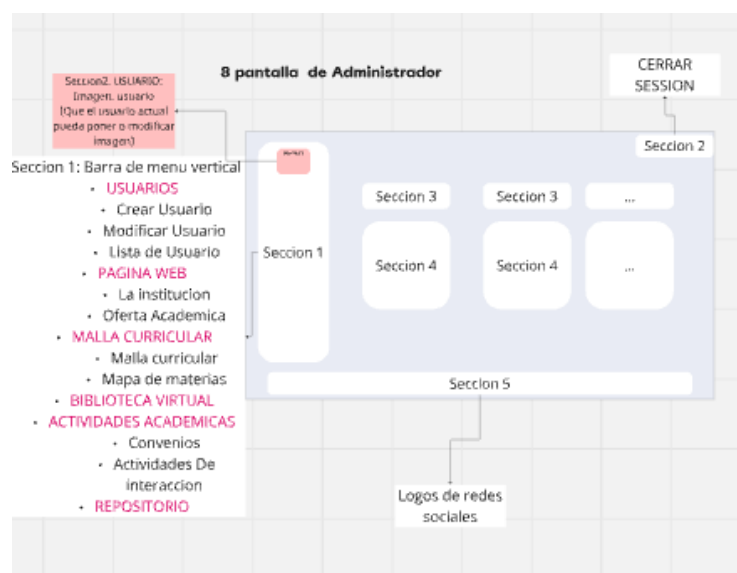
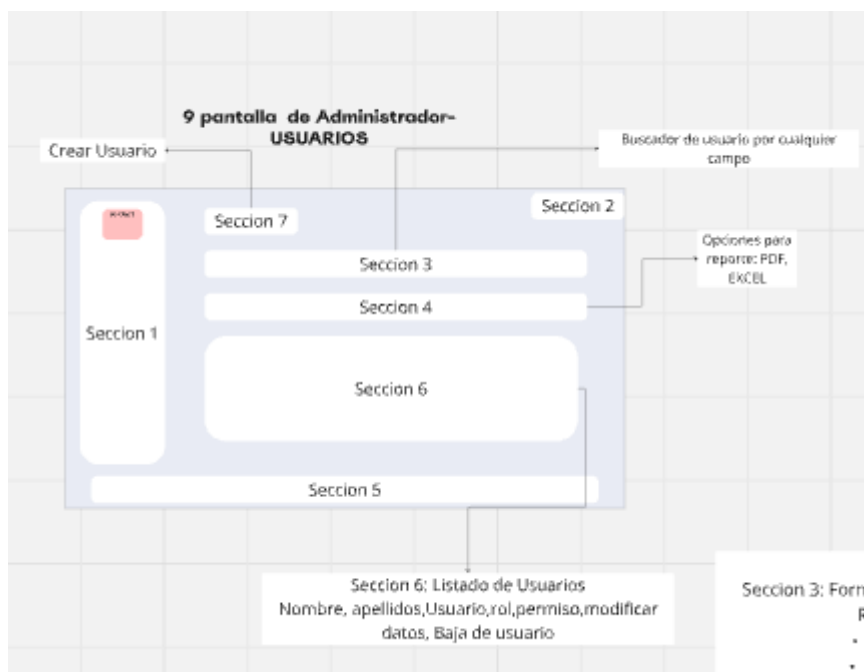
Figura 33.*Prototipo de presentación de Inicio sesión***Figura 34.***Prototipo de presentación de inicio de menú*

Figura 35.*Prototipo de presentación - gestión de usuarios*

2.16. Modelo de implementación

Figura 36.*Página principal*

Figura 37.

Página web-carrera



Figura 38.

Página web-plan de estudio-materias

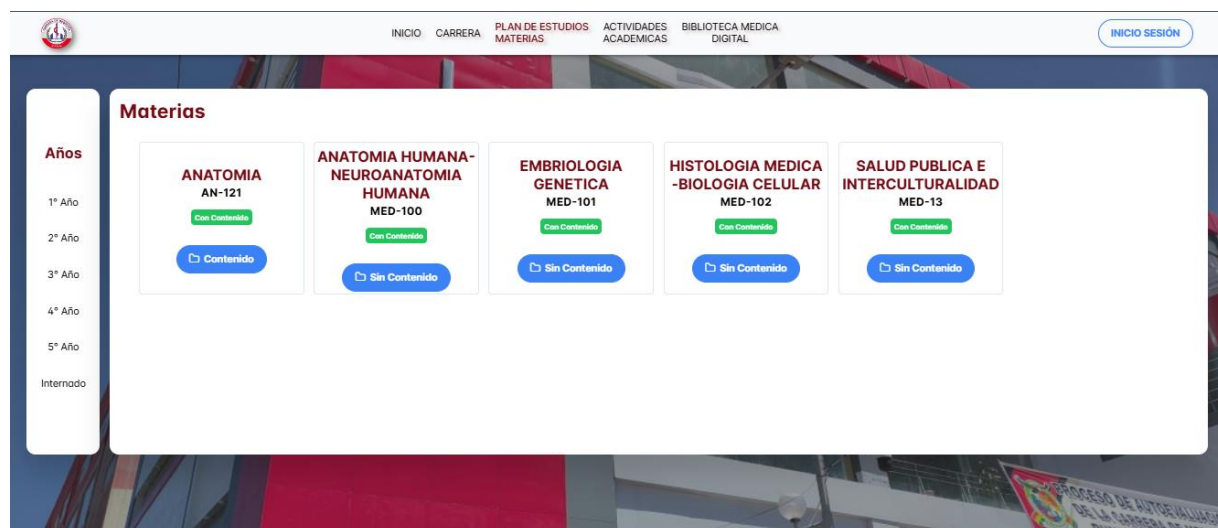


Figura 39.

Página web-actividades académicas



Figura 40.

Página web-Biblioteca digital

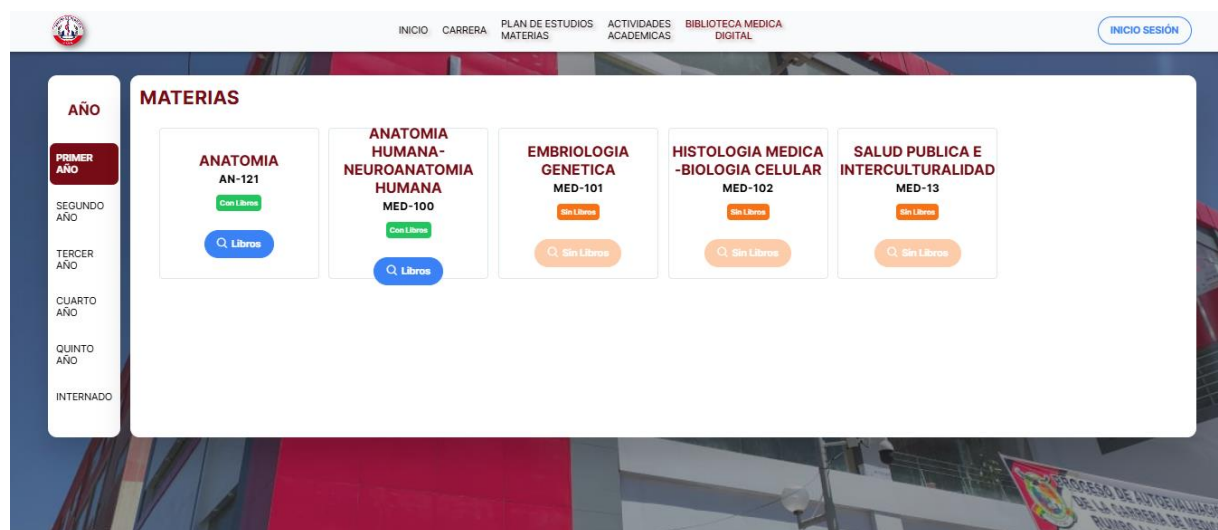
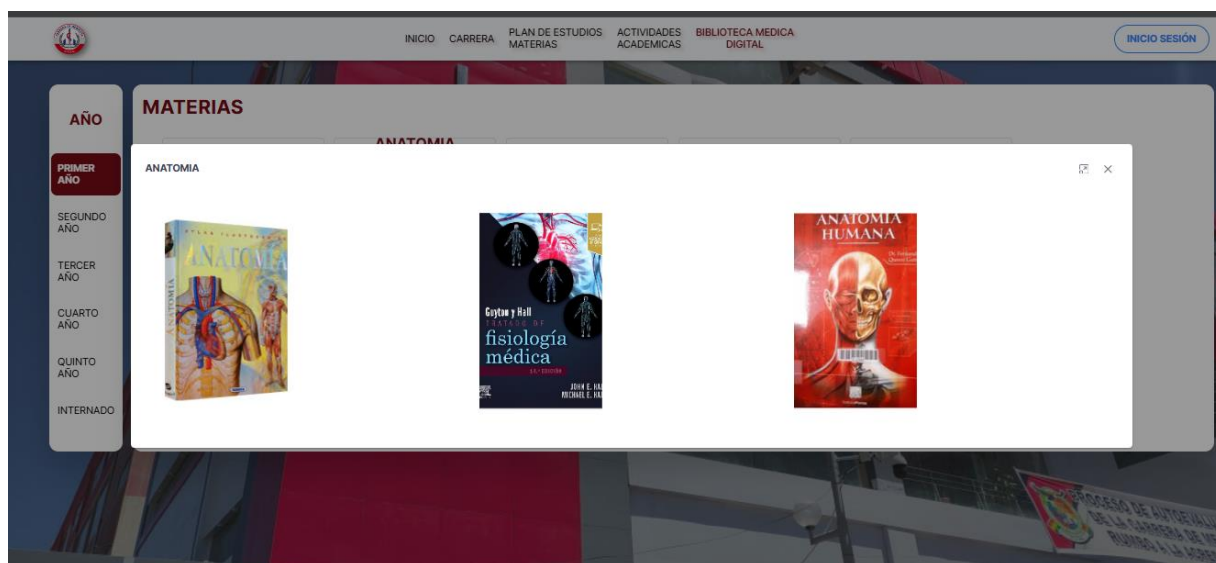


Figura 41.

Página web-mostrar libros digitales

**Figura 42.**

Página web - inicio Sesión

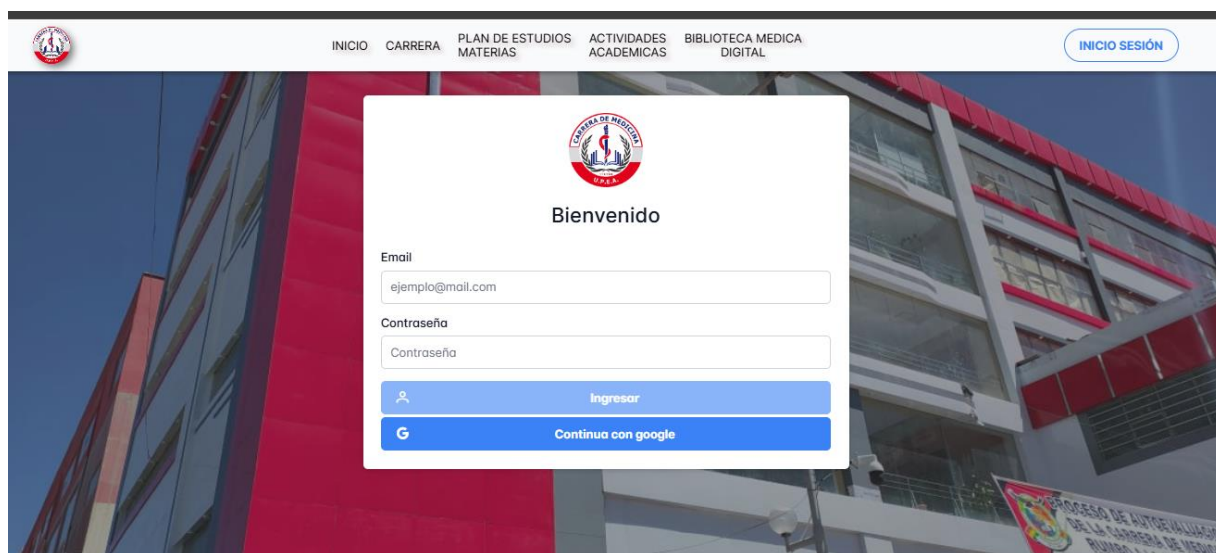


Figura 43.

Modulo principal - administrador

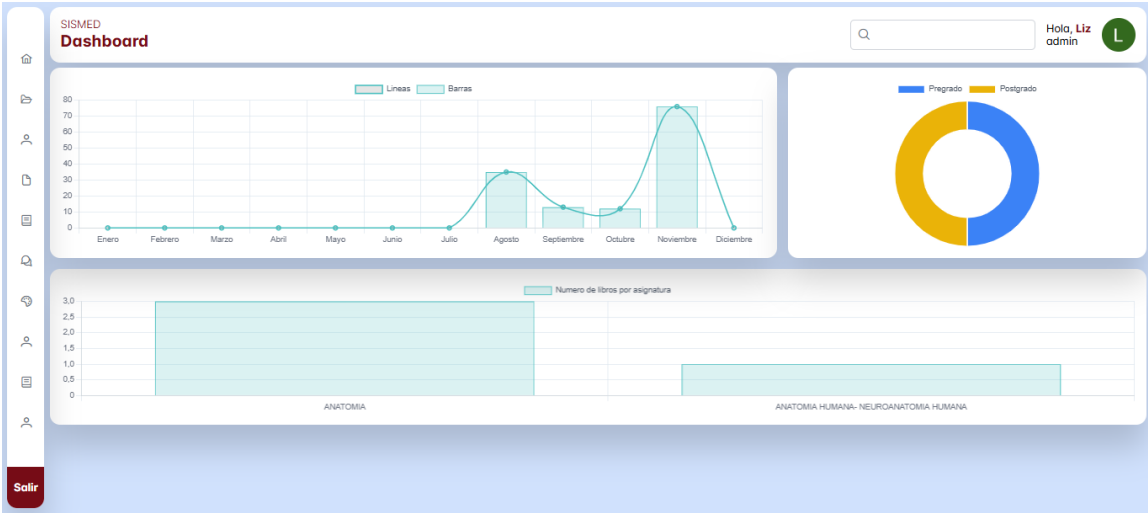


Figura 44.

Gestión de asignaturas

SIS-MED DASHBOARD						
GESTIÓN DE ASIGNATURAS						
ID	ASIGNATURA	AÑO	SIGLA	DESCRIPCION/PRE REQUISITO	CONTENIDO	FECHA DE CREACIÓN
6	SALUD PUBLICA E INTERCULTURALIDAD	1	MED-13	Ninguno	✓	Nov 12, 2024
5	HISTOLOGIA MEDICA -BIOLOGIA CELULAR	1	MED-102	Ninguno	✓	Nov 12, 2024
4	EMBRIOLOGIA GENETICA	1	MED-101	Ninguno	✓	Nov 12, 2024
3	ANATOMIA HUMANA- NEUROANATOMIA HUMANA	1	MED-100	Ninguno	✓	Nov 12, 2024
2	FISIOLOGIA	2	FIO-655	MATERIA DE SEGUNDO AÑO	✓	Nov 10, 2024
1	ANATOMIA	1	AN-121	MATERIA DE PRIMER AÑO	✓	Nov 7, 2024

Figura 45.

Gestión de trabajos de investigación

SIS-MED DASHBOARD

REPOSITORIO-TRAB. DE INVESTIGACION

+ NUEVO PDF Excel

id	TITULO	FECHA PUBLICACIÓN	RESUMEN	PREGRADO/POSGRADO	FECHA CREACION	AUTOR	NOMBRE DE DOC.
2	Estudio De Celular Madre	Nov 4, 2024	Es Una Investigacion De Alto Interes Para El Estudio De Celulas.	Postgrado	Nov 10, 2024	Ana Lima Lopez	Documentostenologas De La Internet.pdf
1	Nuevo Tra	Nov 6, 2024	Sass	Pregrado	Nov 7, 2024	Ana Lima Lopez	Documentosflujo De Sistema.pdf

SALIR

Figura 46.

Búsqueda de trabajos de investigación

SIS-MED DASHBOARD

Inicio Asignatura Autores Documentos Libros Noticias Pagina Permisos Repositorio Usuarios

Título Cite

Palabra Clave + Seleccione una Categoría

Seleccione una Autor Seleccione una Materia Seleccione una Año

Buscar

SALIR

Figura 47.

Gestión de libros digitales

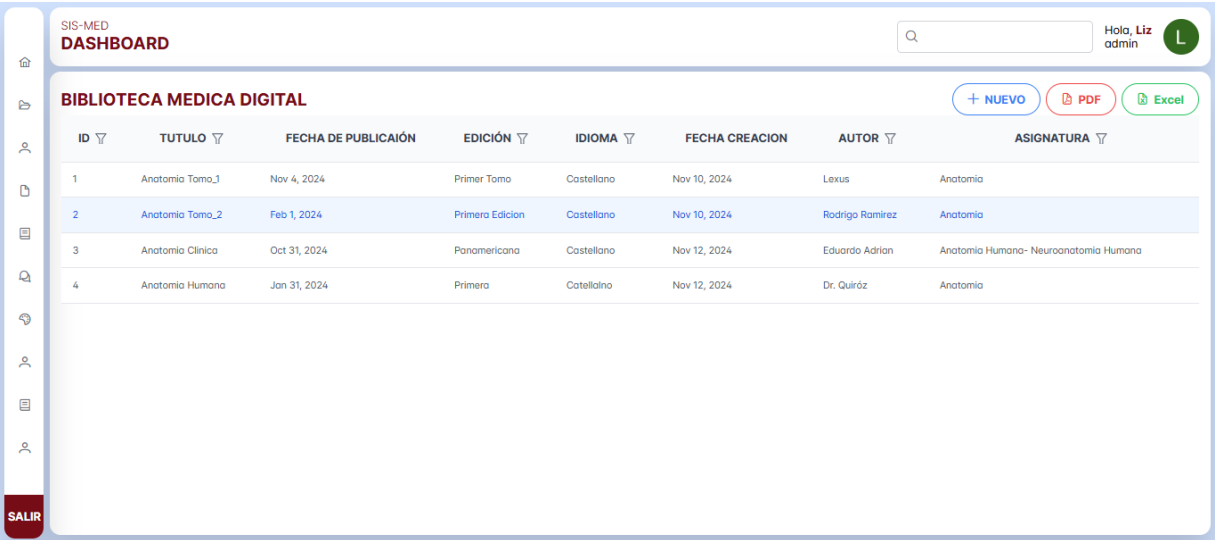


Figura 48.

Registro de nuevo libro

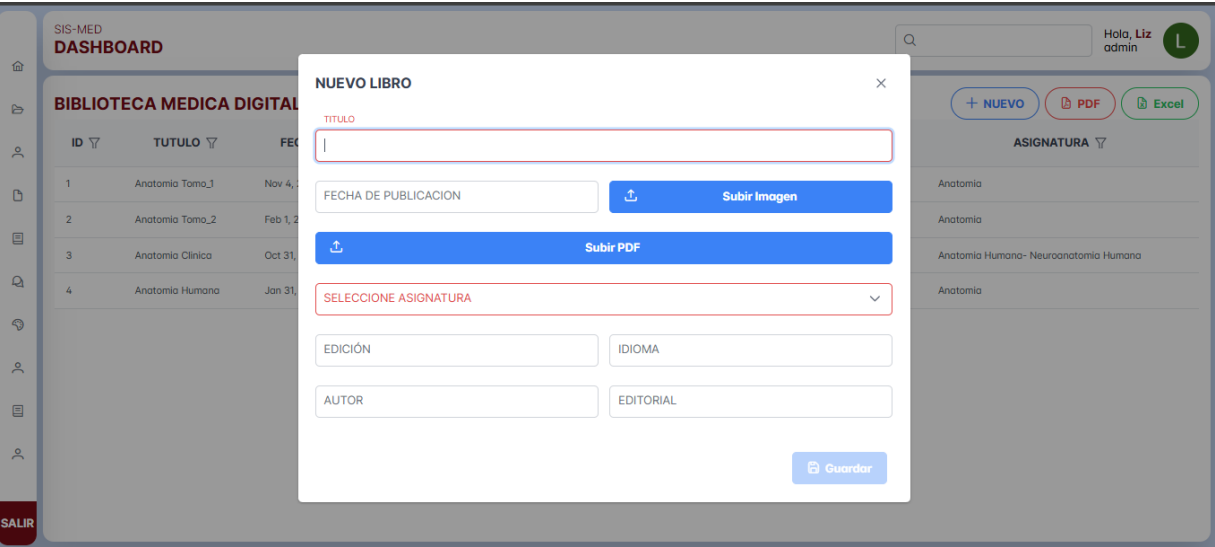


Figura 49.

Gestión de noticias

SIS-MED
DASHBOARD

Hola, Liz
admin

Inicio

Asignar

Auto

Docu

Libro

Noti

Pag

Perm

Repo

Usua

SALIR

GESTIÓN DE NOTICIAS

+ Nuevo

PDF

Excel

TÍTULO	CONTENIDO/DESCRIPCIÓN	Imagen	FECHA CREACIÓN	ACTIVO / NO ACTIVO
LA UPEA ABRE LA CARRERA DE MEDICINA EN CARANAVI PARA EL 2024	La carrera de medicina, expande sus horizontes. tal es el caso en el municipio de Caranavi. Los Jóvenes de la población de Caranavi podrán estudiar la carrera de medicina.	images/LA_UPEA_ABRE_LA_CARRERA_DE_MEDICINA_EN_CARANAVI_PARA_EL_2024	Nov 12, 2024	✓
VIII OLIMPIADA NACIONAL - ANATOMIA I, NEUROANATOMIA	Muchas felicitaciones a toda la carrera de Medicina*, hoy nuestra delegación de estudiantes y docentes, llevaron en alto el nombre de nuestra querida Universidad Pública de El Alto, demostrando la "capacidad académica" que tienen todos nuestros "estudiantes y docentes". Fue un gran trabajo para nuestros delegados por lo que	images/VIII_OLIMPIADA_NACIONAL_-_ANATOMIA_I_,_NEUROANATOMIA	Nov 12, 2024	✓
EL EMBARAZO PARTO PUERPERIO	la carrera de Medicina UPEA Bolivia La Paz-El Alto , donde tocamos un tema muy importante "El Embarazo Parto Puerperio".	images/EL_EMBARAZO_PARTO_PUERPERIO	Nov 12, 2024	✓
NOTICIA	EN LA SEMANA ANIVERSARIO	images/NOTICIA	Nov 7, 2024	✓
This is Fine	This is Fine	images/This_is_Fine	Nov 7, 2024	✓
MHGFD	DS DFDF DFDF	images/MHGFD	Nov 7, 2024	✓

Figura 50.

Registro de nueva noticia

SIS-MED
DASHBOARD

Hola, Liz
admin

Inicio

Asignar

Auto

Docu

Libro

Noti

Pag

Perm

Repo

Usua

SALIR

GESTIÓN DE NOTICIAS

+ Nuevo

PDF

Excel

TÍTULO	CONTENIDO/DESCRIPCIÓN	Imagen	FECHA CREACIÓN	ACTIVO / NO ACTIVO
LA UPEA ABRE LA CARRERA DE MEDICINA EN CARANAVI PARA EL 2024	La carrera de medicina, expande sus horizontes. tal es el caso en el municipio de Caranavi. Los Jóvenes de la población de Caranavi podrán estudiar la carrera de medicina.	images/LA_UPEA_ABRE_LA_CARRERA_DE_MEDICINA_EN_CARANAVI_PARA_EL_2024	Nov 12, 2024	✓
VIII OLIMPIADA NACIONAL - ANATOMIA I, NEUROANATOMIA	Muchas felicitaciones a toda la carrera de Medicina*, hoy nuestra delegación de estudiantes y docentes, llevaron en alto el nombre de nuestra querida Universidad Pública de El Alto, demostrando la "capacidad académica" que tienen todos nuestros "estudiantes y docentes". Fue un gran trabajo para nuestros delegados por lo que	images/VIII_OLIMPIADA_NACIONAL_-_ANATOMIA_I_,_NEUROANATOMIA	Nov 12, 2024	✓
EL EMBARAZO PARTO PUERPERIO	la carrera de Medicina UPEA Bolivia La Paz-El Alto , donde tocamos un tema muy importante "El Embarazo Parto Puerperio".	images/EL_EMBARAZO_PARTO_PUERPERIO	Nov 12, 2024	✓
NOTICIA	EN LA SEMANA ANIVERSARIO	images/NOTICIA	Nov 7, 2024	✓
This is Fine	This is Fine	images/This_is_Fine	Nov 7, 2024	✓
MHGFD	DS DFDF DFDF	images/MHGFD	Nov 7, 2024	✓

NUEVA NOTICIA

TÍTULO

B I U Sans Serif Normal

SELECCIONE CATEGORIA

VISIBLE? ☐

Ayuda

Guardar

Figura 51.

Gestión de datos institucionales

SIS-MED
DASHBOARD

OBJETIVO

Fomentar la realización de investigación y proponer soluciones sobre las problemáticas en salud, que son prioritarias para el país, incluyendo su dimensión internacional, que requieran la participación de investigadores en diferentes disciplinas de las Ciencias de la salud. Incidir en la formación de nuevos médicos mediante la incorporación y formación de un amplio número de estudiantes; impulsar la colaboración entre entidades universitarias y con otras instituciones y organismos, Nacionales e Internacionales especializados.

MISIÓN

Formar profesionales de la salud, médicos generales, críticos y humanistas, con valores éticos, compromiso social y voluntad de superación, que responden a las necesidades de salud con innovación y excelencia educativa, capaces de difundir conocimientos, tecnología y avances científicos, para contribuir a la salud y desarrollarse en la investigación científica en beneficio del ser humano y del país.

VISIÓN

Ser una Facultad de medicina generadora de profesionales, conocimientos, tecnologías e innovación; referente de superación en las ciencias de la salud y fuente permanente de actualización, para atender las necesidades del país, preservando la excelencia y consolidando su reconocimiento nacional e internacional.

AUTORIDADES INSTITUCIONALES

+ Nuevo

NOMBRE	CARGO	FOTOGRAFIA	VISIBLE?
Jhonny Santos Rojas Roque	Director De Carrera	Si Tiene	Visible
Juan	Director	Si Tiene	No Visible
Dr. Jonny Dd	Director De Carrera	Si Tiene	No Visible

MALLA CURRICULAR

Subir Nueva Malla

SALIR

Figura 52.

Gestión de usuarios

SIS-MED
DASHBOARD

LISTA DE USUARIOS

+ NUEVO USUARIO

PDF

Excel

Solicitudes 1

NOMBRE(S)	AP. PATERNO	AP. MATERNO	EMAIL	C.I.	Fecha de alta
Ana	Perez	Lima	abc@gmail.com	12345678	11/10/24, 12:16 PM
Otmar	Aguilar	Aguilar	ootmar1998@gmail.com	9991405	11/6/24, 10:34 PM
Otmar	Aguilar	Aguilar	ootmar1998@gmail.com	9991405	11/6/24, 10:33 PM
Liz	Gutierrez	Aruquipa	gutierrezlzt321@gmail.com	123456	8/13/24, 10:29 PM
Lizet	Gutierrez	Aruquipa	gutierrezlzt321@gmail.com	123456	8/5/24, 12:01 PM

SALIR

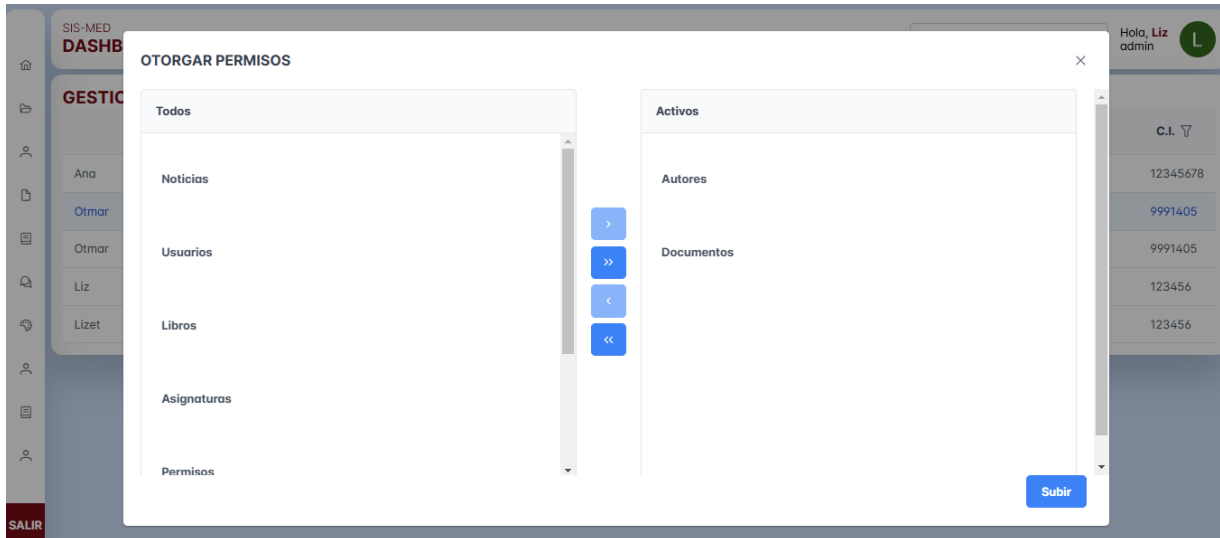
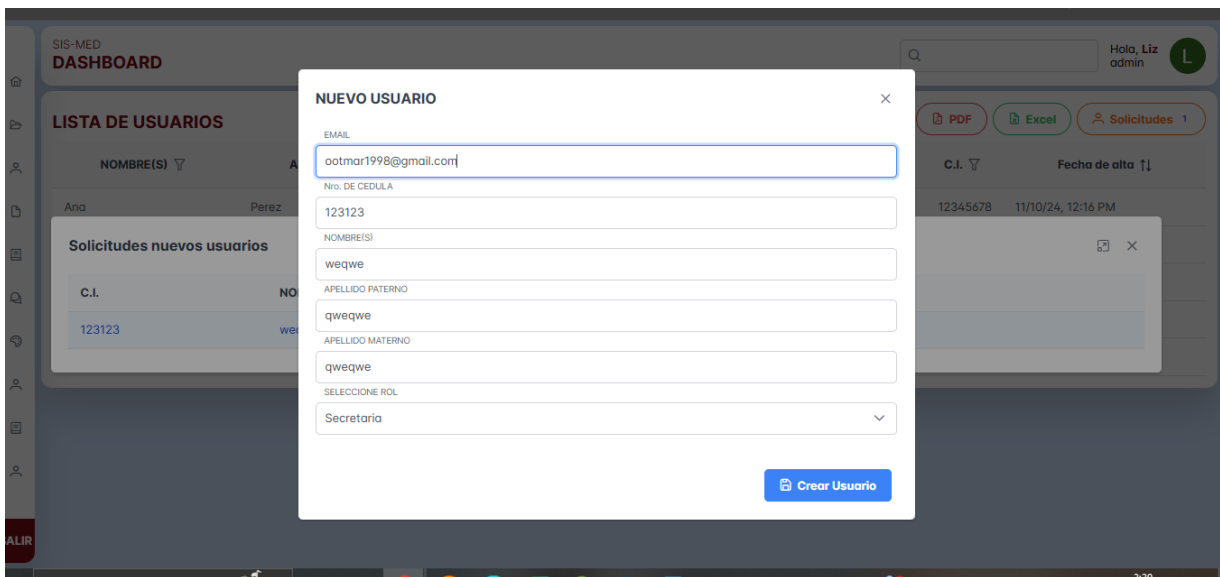
Figura 53.*Otorgar permisos***Figura 54.***Registro de nuevo usuario*

Figura 55.

Formato de reportes en PDF

5-93ef-c4e91990cfb91 / 1100%+



Universidad Pública de El Alto

Creada por Ley 2115 del 5 de Septiembre de 2000 y Autónoma por Ley 2556 del 12 de Noviembre de 2003

REPORTE DE USUARIOS

Nombres	Apellido Paterno	Apellido Materno	C.I.	Correo	Fecha de Alta
Ana	Perez	Perez	12345678	abc@gmail.com	10/11/2024, 12:16:13
Otmar	Aguilar	Aguilar	9991405	ootmar1998@gmail.com	6/11/2024, 22:34:24
Otmar	Aguilar	Aguilar	9991405	ootmar1998@gmail.com	6/11/2024, 22:33:40
Liz	Gutierrez	Gutierrez	123456	gutierrezzt321@gmail.com	13/8/2024, 22:29:25
Lizet	Gutierrez	Gutierrez	123456	gutierrezzt321@gmail.com	5/8/2024, 12:01:33

CAPITULO IV

PRUEBAS DE EVALUACIÓN Y RESULTADOS

**INGENIERÍA
DE
SISTEMAS**

UNIVERSIDAD PUBLICA DE EL ALTO



4. PRUEBAS DE EVALUACION Y RESULTADOS

4.1. METRICAS DE CALIDAD ISO 25000

En este capítulo se determina la calidad del sistema, ya que es uno de los aspectos más importantes dentro del desarrollo de software, del mismo modo de describir la seguridad del sistema.

Para el desarrollo del proyecto de aplicaron las métricas de calidad, donde se calcularon los parámetros que cumple el sistema, que permita especificar ordenadamente dichas características y atributos.

4.1.1. Funcionalidad

Para el desarrollo de esta técnica de estimación de software, toma en cuenta 5 características:

- **Número de entradas de usuario:** El número de entradas en la que el usuario proporciona datos al sistema.

Tabla 24.

Número de entradas de usuario

Entradas de usuario	N°
1 Administración de usuarios	1
2 Administración de secretaria	1
3 Administración de bibliotecaria	1
4 Administración de docente	30
Total	33

- **Número de salidas de usuario:** Es el número de salidas que el sistema proporciona, tales como: mensajes, notificaciones, reportes, alertas entre otros.

Tabla 25.*Número de salidas de usuario*

Entradas de usuario	N°
1 Administración de usuarios	4
2 Administración de secretaria	15
3 Administración de bibliotecaria	3
4 Administración de docente	3
Total	25

- **Número de peticiones de usuario:** se representa por la cantidad de entradas interactivas que genera una respuesta del software como salidas interactivas.

Tabla 26.*Número de peticiones de usuario*

Entradas de usuario	N°
1 Administración de usuarios	3
2 Administración de secretaria	6
3 Administración de bibliotecaria	3
4 Administración de docente	3
Total	15

- **Número de archivos:** Se cuenta cada archivo maestro lógico, quiere decir que se contara todos los archivos que se encuentren en el sistema (Base de Datos, cualquier archivo independiente). Es cada tabla, archivo entre otros.
- **Número de interfaces externas:** es la suma de todas las interfaces legibles por la máquina. Son las interfaces, discos, copias de seguridad, transmisiones de datos.

Tabla 27.

Parámetros de medición de punto función

Parámetros de medición	Cuenta
Número de Entradas de usuario	3
Número de salidas de usuario	6
Número de peticiones de usuario	3
Numero de archivos	3
Numero de interfaz externa	0

Una vez realizado el conteo de los parámetros de función, se pasa a realizar el cálculo de los puntos función sin ajustar. En la siguiente tabla refleja el cálculo de puntos función.

Tabla 28.*Punto función sin ajustar*

Parámetros de medición	Factor de ponderación				Total
	Cuenta	Simple	Medio	Complejo	
Número de entradas de usuario	33 x	3	4	6	198
Número de salidas de usuario	25 x	4	5	7	125
Número de peticiones de usuario	15 x	3	4	6	90
Numero de archivos	15 x	7	0	15	225
Numero de interfaz externa	0 x	5	7	10	0
Total					638

En la tabla anterior se puede apreciar la cuenta total de puntos de función sin ajustar, esto para determinar los valores ajustados, que se hace el empleo de los factores de complejidad listados a continuación:

Tabla valoración de porcentajes

Tabla 29.

Valoración de porcentajes

Importancia	0%	20%	40%	60%	80%	100%
Escala						
	No influye	Incidental	Moderado	Medio	Significativo	Esencial
Factor	0	1	2	3	4	5

Tabla 30.

Factores de complejidad

	Valoración	Factor
1	¿Requiere el sistema copias de seguridad y recuperación fiable?	4
2	¿Se requiere comunicación de datos?	4
3	¿existe funciones de procesos distribuidos?	3
4	¿es crítico el rendimiento?	4
5	¿Se ejecutará el sistema con un sistema operativo existente y fuertemente utilizado?	4
6	¿Requiere el sistema entrada de datos interactiva?	5
7	¿Facilidad operativa?	3
8	¿Se actualiza los archivos maestros de forma interactiva?	3
9	¿Son complejas las entradas, las salidas, los archivos o las peticiones?	3
10	¿procesamiento interno complejo?	3

	Valoración	Factor
11	¿Diseño de código reutilizable?	4
12	¿facilidad de instalación?	4
13	¿Soporta múltiples instalaciones en diferentes sitios?	4
14	¿facilidad de cambios?	5
Factor de ajuste de complejidad		53

Para el cálculo del punto función ajustado se utiliza la siguiente formula:

$$PF = \text{Conteo total} \left[0.65 + 0.01 * \sum fi \right]$$

Donde:

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

Entonces se tiene:

$$\sum(fi)= 53$$

$$PF = 638 * [0.65 + 0.01 * 53]$$

$$PF = 752.84$$

Considerando el máximo valor en la sumatoria de los factores de ajustes

$\sum(fi) = 70$, se calcula el siguiente valor:

$$PF_{max} = \text{Cuenta Total} \left[0.65 + (0.01 * \sum fi) \right]$$

$$PF_{max} = 638 [0.65 + (0.01 * 70)]$$

$$PF_{max} = 861.3$$

El cálculo de la funcionalidad se obtiene de la relación entre valor máximo con el valor obtenido del punto de función ajustado:

$$Funcionalidad = \frac{PF}{PF_{max}} * 100$$

$$Funcionalidad = \frac{752.84}{861.3} * 100$$

$$Funcionalidad = 0.874 * 100$$

$$Funcionalidad = 87.41\%$$

Entonces, la funcionalidad del sistema desarrollado llega a ser del 87.41%, para la satisfacción de las necesidades del usuario.

4.1.2. Confiabilidad

Confiabilidad es la capacidad de un sistema para mantener su nivel de ejecución bajo condiciones normales en un periodo de prueba establecido, cuyo criterio en la madurez, recuperabilidad, tolerancia a fallos, y cumplimiento de fiabilidad para determinar la confiabilidad del sistema, se toma en cuenta el tiempo en el que el sistema comienza a funcionar para realizar observaciones al sistema.

Donde se toma en cuenta que:

$$P(T \leq t) = F(t) \Rightarrow \text{Probabilidad de fallas}$$

$$P(T \leq t) = 1 - F(t) \Rightarrow \text{Probabilidad de trabajo sin fallas}$$

Para poder calcular la confiabilidad del sistema se toma en cuenta la siguiente función:

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

Donde:

f = funcionalidad del sistema

μ = Probabilidad de error del sistema

t = Tiempo de prueba del sistema

En un periodo de 20 días como tiempo de prueba, se define de cada 10 ejecuciones 1 falla.

$$F(t) = 0.87 * e^{-\frac{1}{10} * 20}$$

$$F(t) = 0.1177 * 100\%$$

$$F(t) = 11.77\%$$

Reemplazando:

$$P(T \leq t) = 0.1177 = 11.77\%$$

$$P(T \leq t) = 1 - 0.1177$$

$$P(T \leq t) = 0.8823 * 100\%$$

$$P(T \leq t) = 88.23\%$$

Por lo tanto, podemos mencionar que, la confiabilidad del sistema desarrollado es del 88.23% e un lapso de tiempo de 20 días de prueba.

4.1.3. Usabilidad

Para poder calcular la usabilidad de un sistema, se toma en cuenta el factor humano, para saber si el sistema cumple con los requisitos establecidos para el usuario final, la evaluación del sistema se realiza mediante encuestas a los usuarios del sistema. Esto para determinar la usabilidad del sistema, para esto se utiliza la siguiente formula:

$$PU = \left[\frac{\sum x_i}{n} * 100 \right]$$

Dónde:

x_i = Valores de evaluacion de cada pregunta

n = Numero de preguntas

Entonces se toma la siguiente tabla:

Tabla 31.

Parámetros de medición de usabilidad

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

Tabla 32.*Usabilidad del sistema*

N°	Pregunta	SI	NO	Resultado $\bar{X}_i$
1	¿El sistema es amigable para usted?	8	2	0.8
2	¿Puede realizar con facilidad las operaciones que el sistema brinda?	9	1	0.9
3	¿Las respuestas del sistema son entendibles?	9	1	0.9
4	¿El sistema cuenta con una interfaz agradable a la vista?	9	1	0.9
5	¿Le parece complicadas las funciones del sistema?	2	8	0.8
6	¿Para usted es difícil aprender a manejar el sistema?	1	9	0.9
7	¿Los resultados que proporciona el sistema le facilitan el trabajo?	10	0	1
8	¿Cuándo utilizo el sistema, se produjo algún tipo de error?	1	9	0.9
9	¿Necesito apoyo técnico para utilizar el sistema?	1	9	0.9
10	¿cree que el sistema tiene bastantes inconsistencias?	1	9	0.9
TOTAL				8.9

Con los resultados que se obtuvieron del cuestionaron, se calcula la usabilidad:

$$FU = \left[\frac{8.9}{10} * 100 \right]$$

$$FU = [0.89 * 100]$$

$$FU = 89\%$$

Entonces se llega a la conclusión de, que la usabilidad del sistema es del 89%.

4.1.4. Eficiencia

La eficiencia mide, si el sistema emplea óptimamente los recursos del sistema. Para calcular la eficiencia que tiene el sistema, se toma en cuenta la siguiente escala:

Tabla 33.

Escala de valores de eficiencia

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

Para valorar la eficiencia del sistema se tiene la siguiente tabla:

Tabla 34.*Valoración de la eficiencia del sistema*

N°	Pregunta	Valor
1	¿La distribución de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	4
2	¿Una secuencia de operaciones puede realizarse con facilidad de movimientos?	5
3	¿Los datos de salida están presentados de modo que se entienden de inmediato?	4
4	¿La operación jerárquica están organizadas, de manera que minimizan la navegación del usuario para que se ejecuten las mismas?	4
5	¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	5
TOTAL		22

Para calcular la eficiencia del sistema tenemos la siguiente fórmula:

$$E = \frac{\sum x_i}{n} * \frac{100}{n}$$

Donde:

$$\sum x_i = \text{Sumatoria de los valores de eficiencia}$$

n = Numero de preguntas

Entonces, reemplazamos los datos obtenidos:

$$E = \frac{22}{5} * \frac{100}{5}$$

$$E = 88\%$$

Concluimos que la eficiencia del sistema es del 88%

4.1.5. Mantenibilidad

La mantenibilidad permite el esfuerzo necesario para realizar modificaciones del sistema, este puede ser por errores, la inclusión de módulos no contemplado en un inicio o incrementar funciones dentro del sistema. El estándar IEEE94 sugiere la implantación del índice de madurez del sistema, ya que este proporciona un indicador de estabilidad del sistema desarrollado.

De modo que se tiene la siguiente ecuación, para el índice de madurez del sistema:

$$IMS = \left[\frac{Mt - (Fa + Fd)}{Mt} \right]$$

Donde:

Mt: Numero de modulos total de la version actual

Fa: Numero de modulos de la version actual que se añadieron

Fc: Numero de modulos de la version actual que se cambiaron

Fd: Numero de modulos de la version actual que se eliminaron en la version actual

Tabla 35.*Valores para la mantenibilidad*

	Valor
Mt	8
Fa	0
Fc	1
Fd	0

Reemplazando los datos en la ecuación, se obtiene:

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

$$IMS = 0.875 * 100$$

$$IMS = 87.5\%$$

Por lo tanto, se concluye que, el índice de mantenibilidad del sistema desarrollado es del 87.5%.

4.1.6. Portabilidad

Portabilidad es la capacidad de transferir del software de un ambiente a otro, en donde se considera lo siguiente:

- **Adaptabilidad:** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicar modificaciones.
- **Facilidad de instalación:** Es el esfuerzo necesario para instalar el software en un ambiente determinado.

- **Conformidad:** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- **Capacidad de reemplazo:** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

El sistema está desarrollado para ser implementado en cualquier servidor, que tenga instalado los servicios de apache, y las herramientas de JavaScript y MySQL.

En el caso del sistema desarrollado, se debe configurar en el marco de trabajo de Angular.

En cuanto a la ejecución, en este caso se está aplicando una tecnología web, eso quiere decir que el sistema puede ser ejecutado en cualquier dispositivo que cuenta con conexión a internet e instalado un navegador web.

Para obtener la portabilidad del sistema, se tiene la siguiente formula:

$$Portabilidad = 1 - \left(\frac{\text{Numero de dias para portar el sistema}}{\text{Numero de dias para implementar el sistema}} \right)$$

Reemplazando datos, tenemos lo siguiente:

$$Portabilidad = 1 - \left(\frac{1}{7} \right)$$

$$Portabilidad = 0.8571 * 100\%$$

$$Portabilidad = 85.71\%$$

Resultados:

Podemos mencionar que, de acuerdo a los resultados obtenidos se puede establecer que la calidad total del sistema, en base a los parámetros medidos anteriormente. La calidad

está directamente relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

Tabla 36.

Resultados

Características	Resultados
Funcionalidad	87.41%
Confiabilidad	88.23%
Usabilidad	89%
Eficiencia	88%
Mantenibilidad	87.5%
Portabilidad	85.71%
Evaluación Total	87.64%

La evaluación de calidad total del sistema desarrollado es de un 87.64%.

4.1.7. Seguridad

Para el desarrollo de un software, uno de los aspectos más importantes a tomar en cuenta es la implantación de normas de seguridad. La norma ISO 27001 evalúa y rectifica el cumplimiento de las normas, tal como la mejora continua en base a un conjunto de controles, que permite la reducción de riesgos a incidentes de seguridad en el funcionamiento del sistema dentro de la institución, para esto se toma en cuenta los siguientes tipos de seguridad:

- **Acceso controlado.** El acceso al sistema está determinado por un sistema de roles los cuales son: Administrador, secretaria, Bibliotecaria, Docente

Dependiendo a cada rol que tenga cada usuario, tiene permisos determinados y módulos que visitar. Al momento de iniciar sesión se genera un token JWT el cual expira después de cierto tiempo, desde que inicia el token contiene información del usuario como los datos generales y su rol, el mismo token se usa para la petición de http al servidor backend, el cual verifica si el token está firmado por el mismo backend para la verificación del usuario.

- **Autorización.** El usuario dependiendo al rol que tiene, tendrá acceso a limitada información, por ejemplo, en un caso de docente, solo podrá ver módulos para cargar trabajos de investigaciones.

El sistema determina la clasificación de la información dependiendo del rol del usuario como indica la ISO 27001:2013, nos menciona que se debe clasificar la información de la siguiente manera:

Confidencial: cuando el nivel de confidencialidad de la información se incrementa.

Restringido: para niveles de medios de confidencialidad

Uso interno: información con un nivel bajo de confidencialidad

Público: Cuando todas las personas pueden ver la información

- **Seguridad física**

Cuando el sistema se encuentra en la fase de producción, al estar alojado en los servidores de la unidad de SIE de la universidad Pública de El Alto, dicha unidad nos garantiza que nos proporciona los ambientes que cumplen con los requisitos de la ISO 27001:2013 los cuales son:

Asegurar el control de acceso a las áreas seguras, solo personas autorizado.

Establece un perímetro de seguridad, instalar controles de acceso a una instalación e implantar medidas contra inundaciones e incendios.

Controla y registra las visitas.

Por seguridad en oficinas y despachos, la información crítica estará alejada de zonas de acceso público. Se instalarán alarmas para la detección de intrusos.

Cualquier material que ingrese a la organización debe ser inspeccionada y registradas.

- **Criptografía**

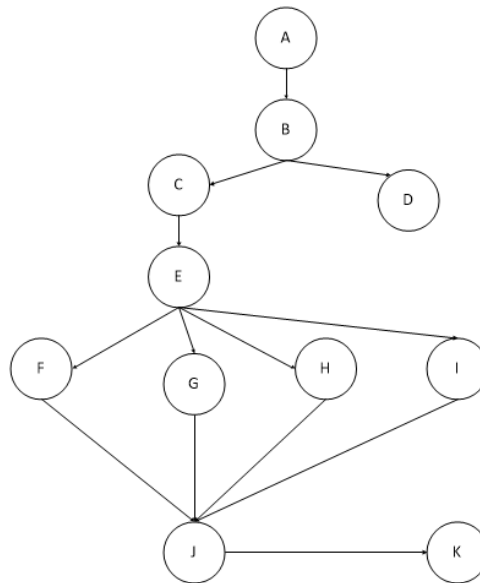
La información del usuario esta encriptada en el token JWT por una firma digital con SHA-256 de esta manera se garantiza la seguridad de la información del usuario, de la misma manera al momento de crear un usuario las contraseñas son encriptadas por la librería Bcrypt el cual lleva incorporado un valor llamado salt, que es un fragmento aleatorio que se usará para generar el hash asociado a la contraseña, y se guardará junto con ella en la base de datos. Así se evita que dos contraseñas iguales generen el mismo hash y los problemas que ello conlleva, por ejemplo, ataque por fuerza bruta a todas las contraseñas del sistema a la vez.

4.2. PRUEBAS

Para realizar las pruebas del funcionamiento del sistema, se realizó, las pruebas unitarias en cada iteración de los módulos, las cuales fueron descritas anteriormente. Se realizará la prueba de caja blanca y caja negra.

4.2.1. Prueba de caja blanca

Se diseñará la prueba de caja blanca respecto a la creación de usuario y autenticación del usuario. En la siguiente figura se muestra de manera general, y que pasos debe seguir:

Figura 56.*Prueba caja blanca*

Dónde:

(A): Ingreso de datos para la creación de usuario

(B): Verificar si el usuario existe

(c): Nuevo usuario

(D): Usuario ya existe

(E): Asignar rol

(F): Administrador

(H): Docente

(G): Bibliotecaria

(I); Secretaría

(J); Asignar permisos

(K): Guardar usuario

Determinar los caminos linealmente independientemente los caminos que deben ser probados.

Camino 1: A-B-C-E-F-J-K

Camino 2: A-B-C-E-G-J-K

Camino 3: A-B-C-E-H-J-K

Camino 4: A-B-C-E-I-J-K

Camino 5: A-B-D

Presentar los casos de prueba para forzar la ejecución de cada camino. Esta última condición establecida que para la ejecución de cientos caminos.

Camino 1: El administrador podrá ingresar el módulo de usuarios, ingresar los datos requeridos, asignar un rol de Administrador, se asignará automáticamente todos los permisos para los módulos, crea y guardar datos.

Camino 2: El administrador podrá ingresar el módulo de usuarios, ingresar los datos requeridos, asignar un rol como Bibliotecaria, se asignará automáticamente los permisos para los módulos de libros, crea y guardar datos.

Camino 3: El administrador podrá ingresar el módulo de usuarios, ingresar los datos requeridos, asignar un rol como Docente, se asignará automáticamente los permisos para los módulos de autores, y documentos, crea y guardar datos.

Camino 4: El administrador podrá ingresar el módulo de usuarios, ingresar los datos requeridos, asignar un rol como secretaria, se asignará automáticamente los permisos para todos los módulos excepto el módulo de permisos, crea y guardar datos.

Camino 5: El administrador podrá ingresar el módulo de usuarios, ingresar los datos requeridos, el sistema verifica los datos y muestra notificación de “el usuario ya existe”.

4.2.2. Prueba de caja negra

Para realizar las pruebas de caja negra, se hace uso de la técnica de partición de equivalencia que es una de las más efectivas, para el análisis de los valores valido e invalido de las siguientes interfaces:

Tabla 37.

Caso de prueba de ingreso al sistema

Casos de prueba de ingreso al sistema	
Código	Caso de prueba 01
Descripción de la prueba	Ingresar al sistema mediante email y contraseña, después de la validación correspondiente.
Condiciones de la ejecución	El usuario debe encontrarse habilitado previamente.
Pasos de ejecución	El sistema muestra la pantalla de autenticación, donde pide email y contraseña para ingresar.
Resultado esperado	El sistema valida los datos ingresados y permite al usuario el ingreso de acuerdo a su rol asignado.
Evaluación de prueba	Se realiza el ingreso al sistema con la verificación previa, en caso de que la verificación falle, el usuario es devuelto a la página de inicio.

4.2.3. Prueba de estrés

Las pruebas de estrés tienen como objetivo someter al sistema a situaciones que superen su capacidad normal de procesamiento, cuyo objetivo es identificar posibles debilidades, errores o fallos en el sistema. Estas pruebas ayudan en gran medida a determinar

si el sistema o la plataforma pueden manejar grandes volúmenes de usuarios concurrentes, cargas de trabajo intensas.

Para el proyecto desarrollado, se utilizó un servidor de los de la universidad, y se obtuvieron los siguientes resultados que van con el óptimo rendimiento que exige el sistema.

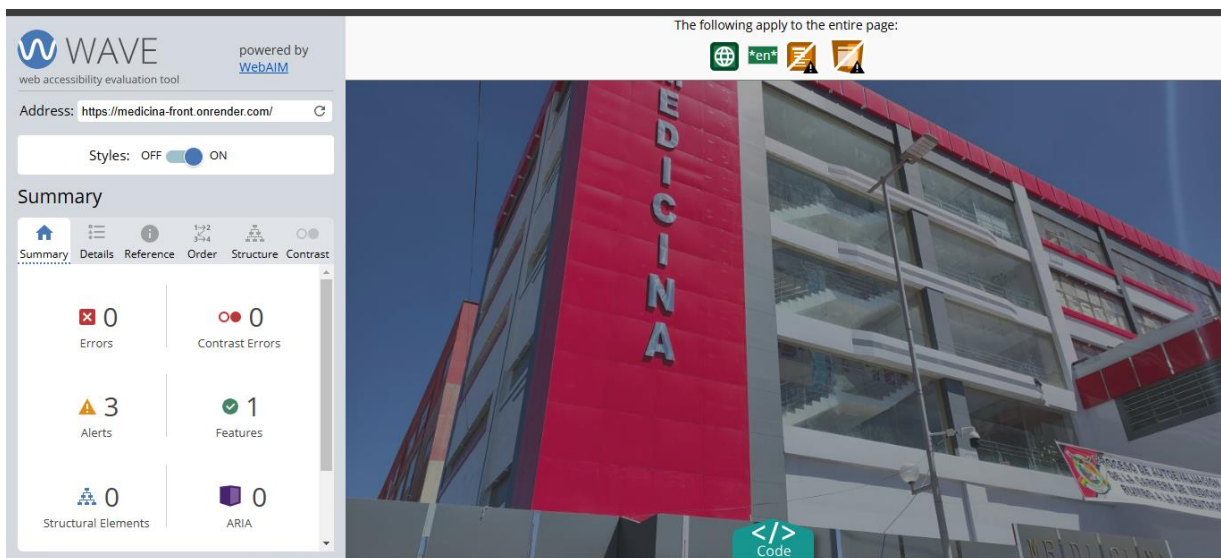
- usuarios establecidos:40
- Tiempo de prueba:10 min
- Solicitudes realizadas: 80
- Fallas HTTP:0
- Pico de solicitud: 7 solicitudes por segundo
- Tiempo de respuesta medida: 87 ms

4.2.4. Pruebas de accesibilidad:

Para esta prueba nos ayudaremos de la página, ingresando a la url: <https://wave.webaim.org/>

Figura 57.

Prueba de accesibilidad



CAPITULO V

ESTIMACIÓN DE COSTOS

**INGENIERÍA
DE
SISTEMAS**

UNIVERSIDAD PÚBLICA DE EL ALTO



5. ESTIMACION DE COSTO

5.1. COCOMO II

Para la realización del presente proyecto, es necesario planificar y estimar costos durante y hasta la finalización del mismo. Para estimar el costo total del sistema se tomará en cuenta los siguientes aspectos en cuanto a costos:

Para calcular el esfuerzo de trabajo, necesitamos hallar la variable KLDC (Kilo Líneas De Código). Para este proyecto se implementó 4050 líneas de código.

LDC = 3500

$$KLCD = \frac{LDC}{1000}$$

$$KLCD = \frac{3500}{1000}$$

$$KLCD = 3.5$$

Tabla 38.

Coeficientes de a.b.c.d de COCOMO II

Proyecto Software	A	B	C	D
Orgánico	3,2	1,05	2,5	0,38
Semi acoplado	3	1,12	2,5	0,35
Empotrado	2,8	1,2	2,5	0,32

Debido a que el proyecto desarrollado, no supera las 50000 líneas de código se utilizara los coeficientes para proyectos orgánicos. Se utilizará la siguiente ecuación para calcular el costo de software:

Tabla 39.*Ecuaciones de COCOMO II*

Variable	Ecuación	Tipo/unidad
Esfuerzo requerido por el proyecto	$E = a * (KLDC)^b * FAE$	Mes/persona
tiempo requerido por el proyecto	$T = c * (E)^d$	Mes
Número de personas requeridas para el proyecto	$NP = \frac{E}{T}$	Personas
costo total	$CT = SueldosMes * NP * T$	\$us

Para hallar FAE se utilizará la tabla de atributos multiplicadores.

Tabla 40.*Cálculo de atributos FAE*

Valor						
Atributos	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra alto
atributos de software						
Fiabilidad	0,75	0,88	1	1,15	1,4	
Tamaño de base de datos		0,94	1	1,08	1,16	
Complejidad	0,7	0,85	1	1,15	1,3	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1	1,11	1,3	1,66
Restricciones de memoria virtual			1	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1	1,15	1,3	
Tiempo de respuesta		0,87	1	1,07	1,15	
atributos de personal						
Capacidad de análisis	1,46	1,19	1	0,86	0,71	
experiencia en la aplicación	1,29	1,13	1	0,91	0,82	
calidad de los programadores	1,42	1,17	1	0,86	0,7	
Experiencia en la máquina virtual	1,21	1,1	1	0,9		
experiencia en el lenguaje	1,14	1,07	1	0,95		
atributos del proyecto						
técnicas actualizadas de programación	1,24	1,1	1	0,91	0,82	
utilización de herramientas de software	1,24	1,1	1	0,91	0,83	
restricciones de tiempo de desarrollo	1,22	1,08	1	1,04	1,1	

Por tanto, el factor de ajuste del proyecto será:

$$FAE = 1.00 * 1 * 1.15 * 1.00 * 1.06 * 1 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 0.95 * 1.00 \\ * 0.91 * 1.00$$

$$FAE = 0.91$$

Aplicando y reemplazando valores a la fórmula de esfuerzo, se tiene:

$$E = a * KLCD^b * FAE \text{ (mes/persona)}$$

$$E = 3.2 * 3.5^{1.05} * 0.91 \text{ (mes/persona)}$$

$$E = 10.85 \text{ (mes/persona)}$$

Cálculo del tiempo:

$$T = c * Esfuerzo^d \text{ (meses)}$$

$$T = 2.5 * 10.85^{0.38} \text{ (meses)}$$

$$T = 6.18 \text{ (meses)} \Rightarrow \text{equivale a 6 meses}$$

Cálculo del personal requerido:

$$NP = \frac{E}{T} \text{ (personas)}$$

$$NP = \frac{10.85}{6.18} \text{ (personas)}$$

$$NP = 1.75 \Rightarrow \text{equivale a 2 personas}$$

Entonces el cálculo total del costo del proyecto es:

Parar el cálculo se utilizará el sueldo mínimo nacional de Bolivia (Bs. 2500) lo que equivale a \$us 360 (cambio de moneda: \$us 1 = Bs. 6.96).

$$NP = sueldoMes * NP * T$$

$$NP = 360 * 2 * 7$$

$$NP = 5040$$

Entonces, para la realización del proyecto se requiere 2 personas, estimando un tiempo de trabajo de 7 meses, con un costo de \$us 5040, esto equivale a Bs. 35078.40.

Tabla 41.

Resumen de costo

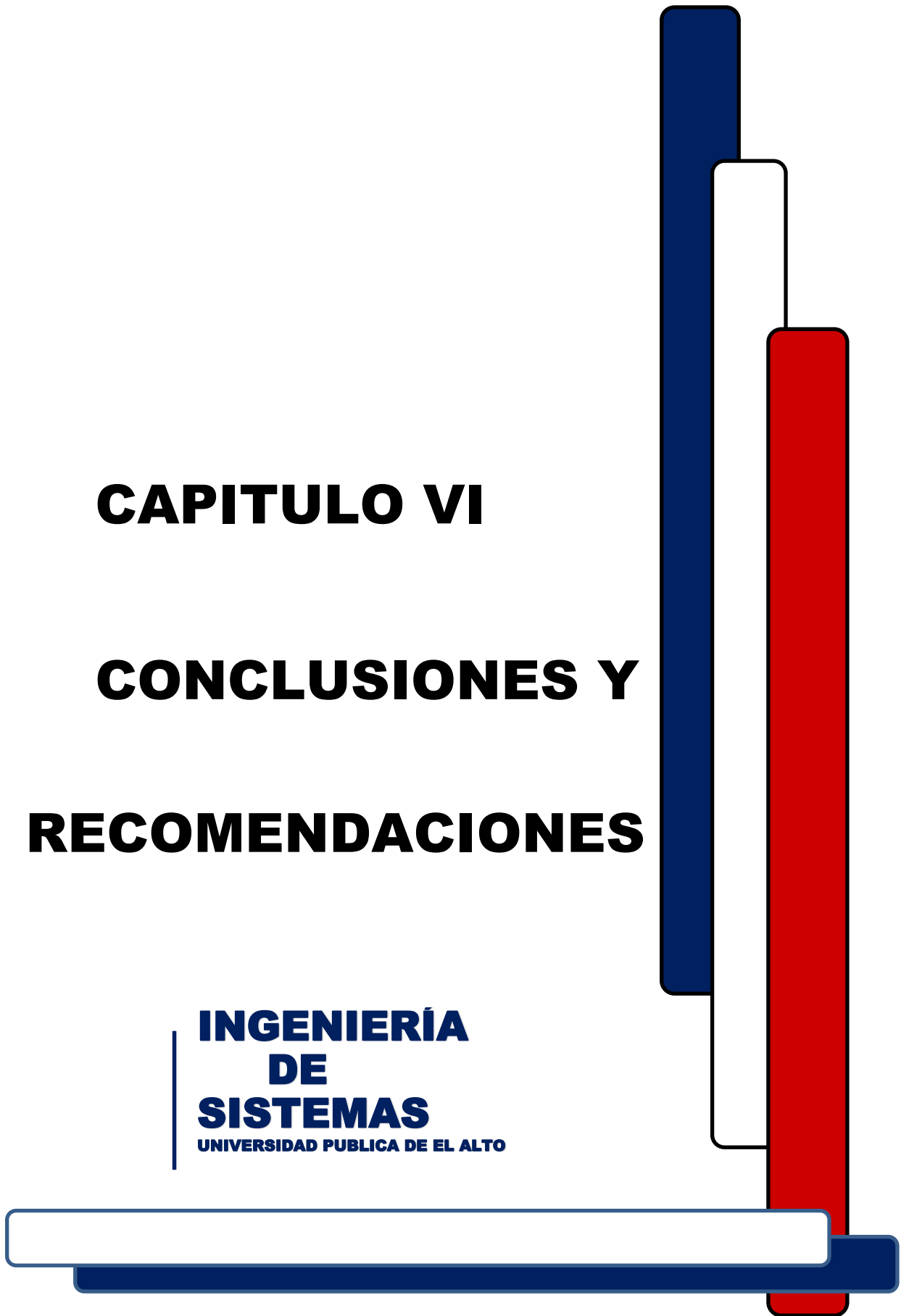
Esfuerzo	11 mes/persona
Tiempo	6 meses
Cantidad de personas requeridas	2 personas
Costo Total	35078.40 Bs.

CAPITULO VI

CONCLUSIONES Y

RECOMENDACIONES

**INGENIERÍA
DE
SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



6. CONCLUSION Y RECOMENDACIONES

Se desarrolló un sistema Integrado para la Centralización de Actividades Académicas, Investigación e Interacción Social Pregrado y Posgrado, que permite la gestión eficiente de los trabajos de investigación, y proporciona datos oportunos.

6.1. CONCLUSIONES

- Se desarrolló un repositorio para los trabajos de investigación, extensión e interacción social, para tener un mejor control de los trabajos disminuyendo la probabilidad de extravío de trabajos.
- Se diseñó una base de datos para la gestión de los trabajos de producción intelectual, reduciendo los extravíos de trabajos y minimizando tiempos en búsqueda.
- Se desarrolló un repositorio para la gestión de libros digitales, con esto se reduce la compra de libros físicos,
- Se creó una interfaz sencilla e intuitiva para la accesibilidad de los usuarios, interfaz que contiene información acerca de la malla curricular, contenido de las materias, minimiza en tiempo de respuesta a solicitudes.
- Se diseñó una interfaz para la visualización de todos los acontecimientos académicos, sociales e interacciones que la carrera va teniendo a lo largo de cada gestión.

- Tabla comparativa

Figura 58.*Tabla Comparativa*

ANTES DEL SISTEMA	DESPUES DEL SISTEMA
Extravíos de trabajos de investigación en un 10%	Mejor control de los trabajos de investigación, con un 0% de extravíos.
Los tiempos de búsqueda de los trabajos de investigación fueron de entre 5 a 10 o más minutos.	Se minimizo en el tiempo de búsqueda de los trabajos de investigación, en 10 segundos.
Existe libros digitales que los docentes dejan cada inicio de gestión, libros digitales almacenados en diferentes dispositivos físicos de almacenamiento.	Mejor control de los libros digitales, accesibilidad para los estudiantes y reducción de gasto económico en la adquisición de libros.
Demora en la búsqueda de los documentos importantes (Malla Curricular/contenido mínimo de asignaturas).	Accesibilidad libre a los documentos importantes (Malla Curricular/contenido mínimo de asignaturas).
Descentralización de publicaciones de actividades académicas, sociales. Con información desactualizada.	Visualización en tiempo real de las diferentes actividades que tiene la carrera.

6.2. RECOMENDACIONES

Para el siguiente proyecto se tienen las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.

- Se recomienda capacitar a las nuevas autoridades, nuevos administrativos de la carrera, para que puedan realizar operaciones en el sistema y así tener un mejor control de las actividades en generan que ofrece el sistema.
- El mantenimiento del software en determinados periodos de tiempo, tomando en cuenta las normas preestablecidas.

- Se recomienda cambiar la contraseña de manera periódica, por seguridad y proteger el acceso a personas ajenas.
- Cuando se requiera la ampliación o creación de nuevos módulos, es recomendable la revisión de la documentación, ya que el sistema presenta elementos que se pueden reutilizar para el desarrollo y así disminuir tiempo en el desarrollo de los mismos.

7. REFERENCIAS BIBLIOGRAFICAS

- AENOR2023. (s.f.). AENOR. Recuperado el 10 de 08 de 2023, de <https://www.aenorchile.com/certificacion/tecnologias-de-la-informacion/producto-software>
- Alarcon, R. (2011). *UML Diseño Orientado a Objetos con UML*. Grupo Eidos.
- Arias Chavez, M. (2005). *Ingenieria de Requerimientos y su importancia en el desarrollo de proyectos de Software*. Costa Rica: InterSedes.
- Arias Jaramillo, E. D., & Cholca Catucuamba, J. S. (2021). Diseño e implementacion de un repositorio digital de proyectos tecnologicos de la carrera de ciencias experimentales informaticos de la UCE .
- Bolloso Chacin, R. (2016). *ELEMENTARY: UN FRAMEWORK DE PROGRAMACION WEB*. Venezuela.
- Carelo, C., Moraga, A., & Piattini Velthuis, M. (2010). *Calidad del Producto y proceso software*. RA-Ma S.A. Editorial y Publicaciones.
- Choque Condori, Y. R. (2022). Histoteca Virtual de Citohistologia y patologia oral caso: facultada de odontologia.
- Copacondo Rodriguez, J. M. (2011). Portal web para el Arzobispado de nuestra Señora de La Paz.
- de las Heras del Dedo, R., Alvarez Garcia , A., & Lasa Gomez, C. (2018). *Metodos Agiles. Scrum, Kanban,Lean*. Anaya Multimedia 2018.
- Enterprise Architect. (2023). Obtenido de <https://sparxsystems.com/>
- Esneca. (2022). *esneca*. Obtenido de <https://www.esneca.com/blog/interaccion-social-formas-categorias/>
- Ferrer , J., & Garcia, V. (2018). *Curso completo de HTML*.
- Florin, B. (2022). *Tecnicas de prueba de software*. Obtenido de http://fcaenlinea.unam.mx/anexos/1728/Unidad_6/u6_act1.pdf
- Gamarra, F. D. (2023). *Visual Studio Code (Profesional)*.
- Gauchat, J. (2017). *El gran libro de HTML5, CSS# y JavaSript*. Barcelona.
- Gisbert Soler, V., & Ruslan, E. (2014). *Sistemas integardos de getsion y beneficios*. Area Innovacion y Desarrollo, S.L.
- GlobalSuite. (02 de 2024). *GlobalSuite*. Obtenido de <https://www.globalsuitesolutions.com/es/que-es-la-norma-iso-27001>
- Gomez, A., Migani, S., & Otazu, A. (2017). *Un Modelo de Estimacion de Proyectos de Software*.

- Gutierrez Zapata, E. R. (2020). Sistema de gestion y digitalizacion bibliotecaria caso: Carrera de Odontologia UPEA.
- Haverbeke, M. (2018). *Eloquent JavaScript*. 3ra edicion.
- Hernandez Trasobares, A. (2003). *Los Sistemas de Informacion*. Fundacion Dialnet.
- Hueso Ibañez, L. (2016). *Administracion de Sistemas de Gestores de Datos 2da Edicion*. Grupo Editorial RA-MA.
- ISO/IEC 25010. (2022). Obtenido de <http://www.iso25010.com>
- ISO27000. (s.f.). *ISO27000*. Recuperado el 02 de 08 de 2023, de <http://www.iso27000.es>
- Kiessling, M., & Junge, H. (2015). *El libro para principiante en Node.js*.
- Koch, N. (2016). *UWE - UML -based Web Wen Enginnering*. Obtenido de <https://uwe.pst.ifi.lmu.de/index.html>
- Maldonado Ramirez, A. Z. (2021). *Pruebas de Software*.
- Maldonado Ramirez, I., & Alva Zapata, J. (2021). *Prueba de software*.
- MariaDB Foundation. (2009). Obtenido de <https://mariadb.org/>
- McCarthy, R. (2020). *El Metodo Agile*.
- Mora Romero , R. A., & Sanchez Jarquin, L. J. (2018). Sistema web para el control de registro de los proyectos investigativos en la direccion de grado y posgrado de UNAN.
- Normas ISO 25000. (2022). Obtenido de <http://www.iso25000.com>
- Orozco Sullca, A. A. (2020). Plataforma web Cultus para la gestion de recursos virtuales caso: unidad educativa adventista Shalom.
- Pressman, R. S. (2010). Ingenieria del Software. En R. S. Pressman.
- Puciarelli, L. (2020). *Angular*. RedUsers.
- Ritter, K. (2020). *NestJS Buid a REST ful CRUD API*.
- Sabino, C. (2014). *El proceso de investigacion*. Episteme.
- Sanchez Maza, M. A. (2012). *JavaScript*. Innovacion y Cualificacion.
- Sangalli Paco, A. (2020). Sistema de control y seguimiento de proyectos educativos caso: Gobierno autonomo municipal de Puerto Perez.
- Serrano Sanchez , C. (2020). *TypeScript, Curso Practico*. RA-MA- S.A. Editorial y Publicaciones.
- Sommerville, I. (2005). *Ingenieria de Software*. Pearson Education S.A. Madrid 2005.
- Sparks, G., & Systems, S. (2011). *El Modelo de Casos de Uso*. Chile.
- Vazquez Carlos, E. (2020). Estimacion de Software con COSMiC. *FATTO consultoria y sistemas*, 7.

Yolanda, C. Q. (2011). *Sistema de gestion de informacioin academica caso carrera de medicina-UMSA*.

8. BIBLIOGRAFIA

Revilla Muñoz, O., & Carreras Montoto, O. (2020). Accesibilidad Web.

Ritter, K. (2020). NestJS Buid a REST ful CRUD API.

Sabino, C. (2014). El proceso de investigacion. Episteme.

Isotools. (s.f.). ISO 27001: Seguridad informática y seguridad de la información. Obtenido de <https://www.isotools.us/2015/01/05/iso-27001-seguridad-informatica-seguridadinformacion/>

Jwt.io. (2013). Introduction to JSON Web Tokens. Obtenido de <https://jwt.io/introduction>

Lucas, J. (4 de Septiembre de 2019). Qué es NodeJS y para qué sirve. Obtenido de <https://openwebinars.net/blog/que-es-nodejs/>

MariaDB Foundation. (2022). MariaDB Foundation. Obtenido de <https://mariadb.org/es/>

Martinez, A. (24 de Mayo de 2022). Concepto Definición. Obtenido de <https://conceptodefinicion.de/sistema/>

MDN. (2 de Diciembre de 2022). MDN.

https://developer.mozilla.org/es/docs/Glossary/World_Wide_Web Obtenido de

Meléndez Valladarez, S., Gaitan, M., & Pérez Reyes, N. (2016). Metodología Ágil De Desarrollo De Software Programacion Extrema.

Microsoft. (09 de 02 de 2023). ¿Qué es Git? Obtenido de <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>

Mongodb. (2022). What is MongoDB? Obtenido de <https://www.mongodb.com/docs/manual/>

NestJS . (s.f.). Introduction. Obtenido de <https://docs.nestjs.com/>

ANEXOS

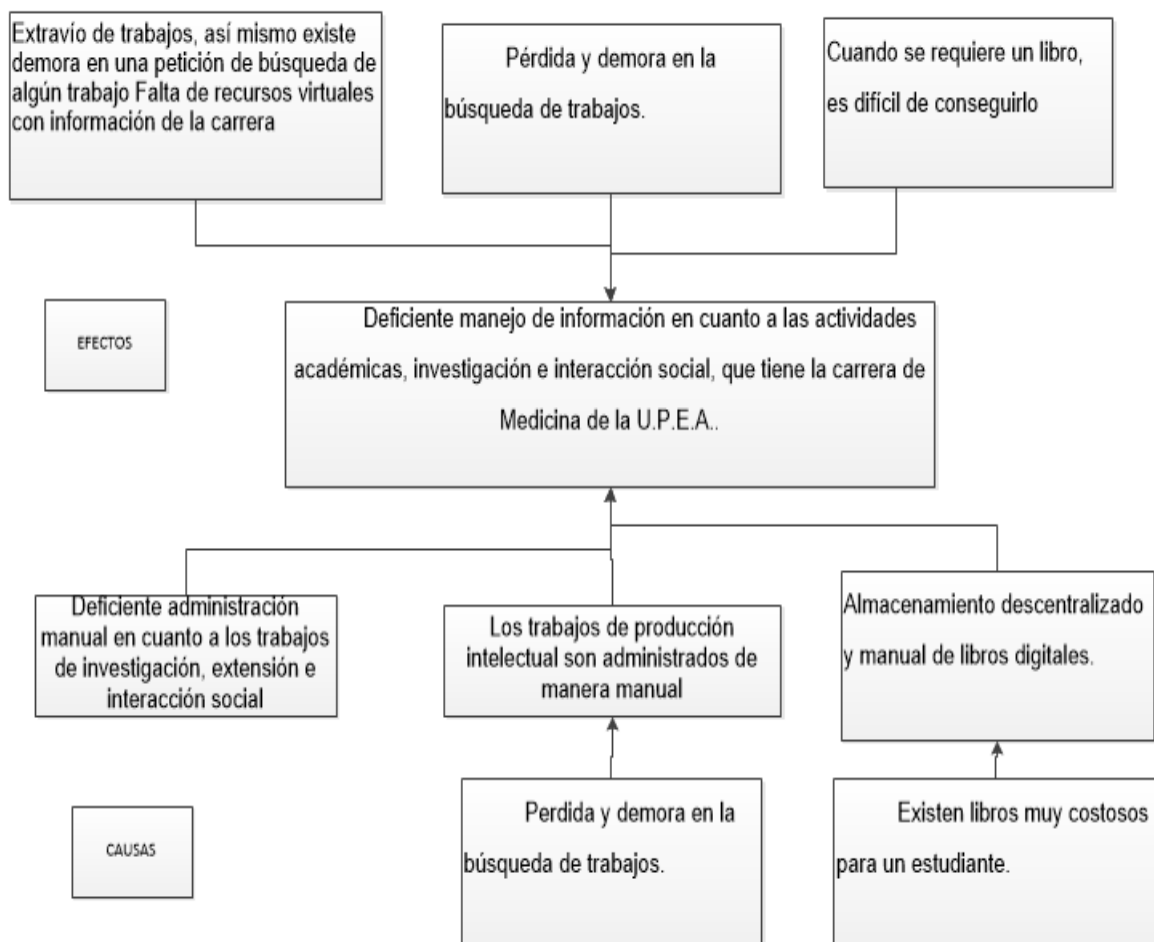
**INGENIERÍA
DE
SISTEMAS**

UNIVERSIDAD PÚBLICA DE EL ALTO

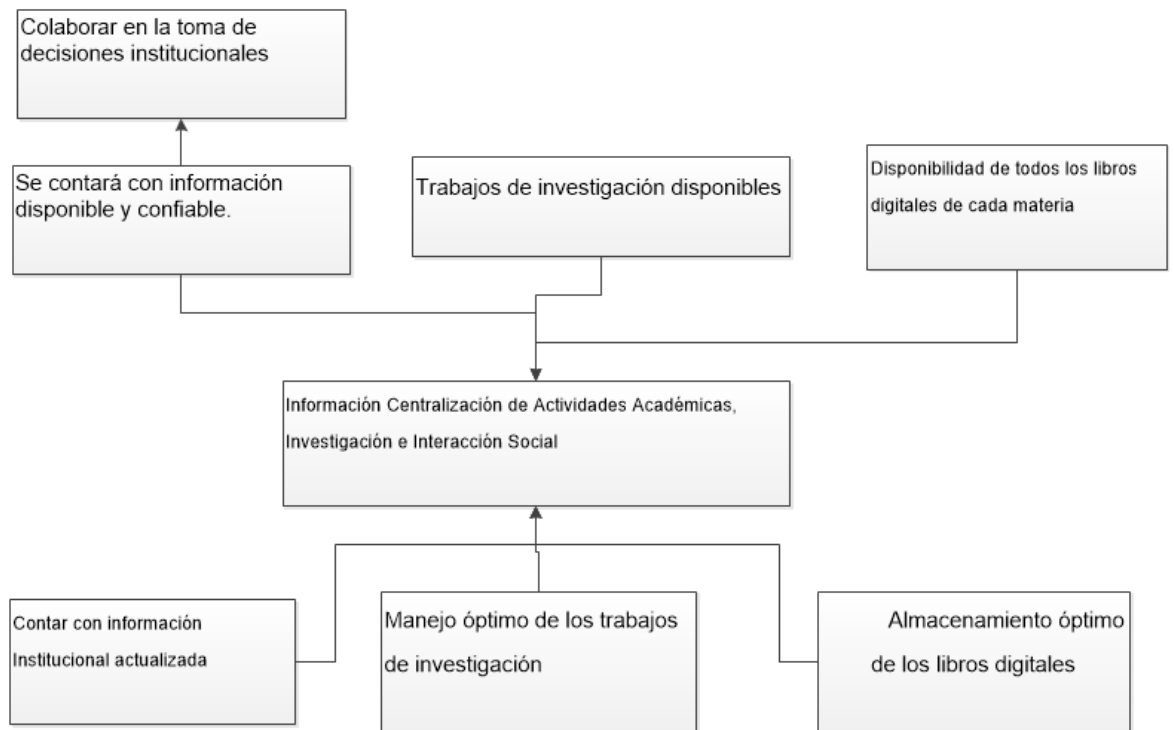


ANEXO A

ARBOL DE PROBLEMAS



ARBOL DE OBJETIVOS

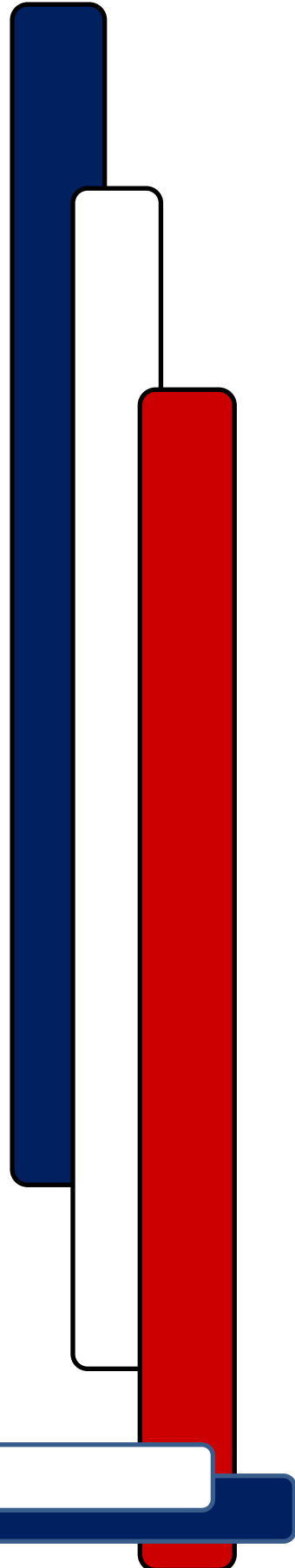


ANEXO B

AVALES

**INGENIERÍA
DE
SISTEMAS**

UNIVERSIDAD PÚBLICA DE EL ALTO



AVAL DE CONFORMIDAD

El Alto, noviembre de 2024

Señor:
Ing. WILLIAM ROQUE ROQUE
DIRECTOR DE CARRERA
INGENIERÍA DE SISTEMAS
Presente. —

REF. AVAL DE CONFORMIDAD

Distinguido director de carrera:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE ACTIVIDADES
ACADEMICAS, INVESTIGACIÓN E INTERACCIÓN SOCIAL PREGRADO Y POSGRADO

CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PUBLICA DE EL ALTO

MODALIDAD: PROYECTO DE GRADO

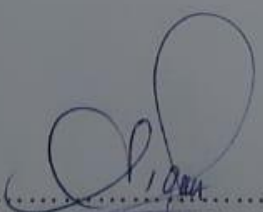
Univ. LIZETH GUTIERREZ ARUQUIPA

Registro Universitario: 200007408

Cedula de Identidad: 6944386 LP

Para su defenza pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


.....
M. Sc. Lic. Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II

AVAL DE CONFORMIDAD

El Alto, noviembre de 2024

Señor:

M. Sc. Lic. Ing. Marisol Arguedas Balladares

TUTOR METODOLÓGICO

TALLER DE GRADO II

Presente. —

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE ACTIVIDADES ACADEMICAS, INVESTIGACIÓN E INTERACCIÓN SOCIAL PREGRADO Y POSGRADO

CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PUBLICA DE EL ALTO

MODALIDAD: PROYECTO DE GRADO

Univ. LIZETH GUTIERREZ ARUQUIPA

Registro Universitario: 200007408

Cedula de Identidad: 6944386 LP

Para su defenza pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


.....
Lic. Ing. Yolanda Escobar Mancilla
TUTOR REVISOR

AVAL DE CONFORMIDAD

El Alto, noviembre de 2024

Señor:
M. Sc. Lic. Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. —

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE ACTIVIDADES
ACADEMICAS, INVESTIGACIÓN E INTERACCIÓN SOCIAL PREGRADO Y POSGRADO

CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PUBLICA DE EL ALTO

MODALIDAD: PROYECTO DE GRADO

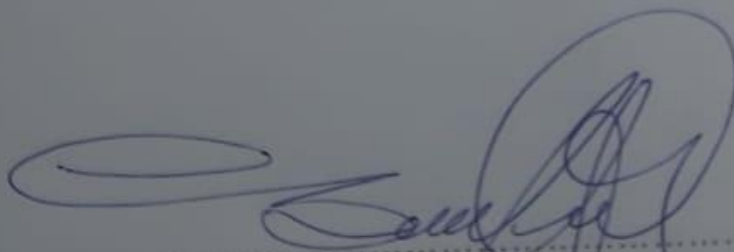
Univ. LIZETH GUTIERREZ ARUQUIPA

Registro Universitario: 200007408

Cedula de Identidad: 6944386 LP

Para su defenza pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,



M. Sc. Lic. Ing. Juan Fernando Huanca Alavi
TUTOR ESPECIALISTA



Universidad Pública De El Alto

*Creada por Ley 2115 de 5 de septiembre de 2000
Modificada por Ley 2556 de 12 de noviembre de 2003*

CARRERA DE MEDICINA



El Alto, septiembre de 2024

Señor:
M.SC. LIC. ING. MARISOL ARGUEDAS BALLADARES
TUTOR METODOLÓGICO - TALLER DE GRADO II
UNIVERSIDAD PUBLICA DE EL ALTO

REF.: **AVAL DE CONFORMIDAD**

Distinguido tutor metodológico,
Mediante la presente tengo el bien de comunicarle la conformidad del trabajo de grado.

TITULO: SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE ACTIVIDADES ACADÉMICAS, INVESTIGACIÓN E INTERACCIÓN SOCIAL PREGRADO Y POSGRADO.

CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PUBLICA DE EL ALTO

MODALIDAD: PROYECTO DE GRADO

UNIVERSITARIO: LIZETH GUTIERREZ ARUQUIPA

REGISTRO UNIVERSITARIO: 200007408

CEDULA DE IDENTIDAD: 6944386 LP.

De tal forma cabe recalcar que el SISTEMA satisface los requerimientos de la Carrera de Medicina, de esta forma se dio cumplimiento con los objetivos del presente.

En cuanto certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la carrera de ingeniería en Sistemas de la Universidad Pública de El Alto.

Atentamente.


Dr. Jhanny Santos Rojas Roque
DIRECTOR
CARRERA MEDICINA
UNIVERSIDAD PÚBLICA DE EL ALTO

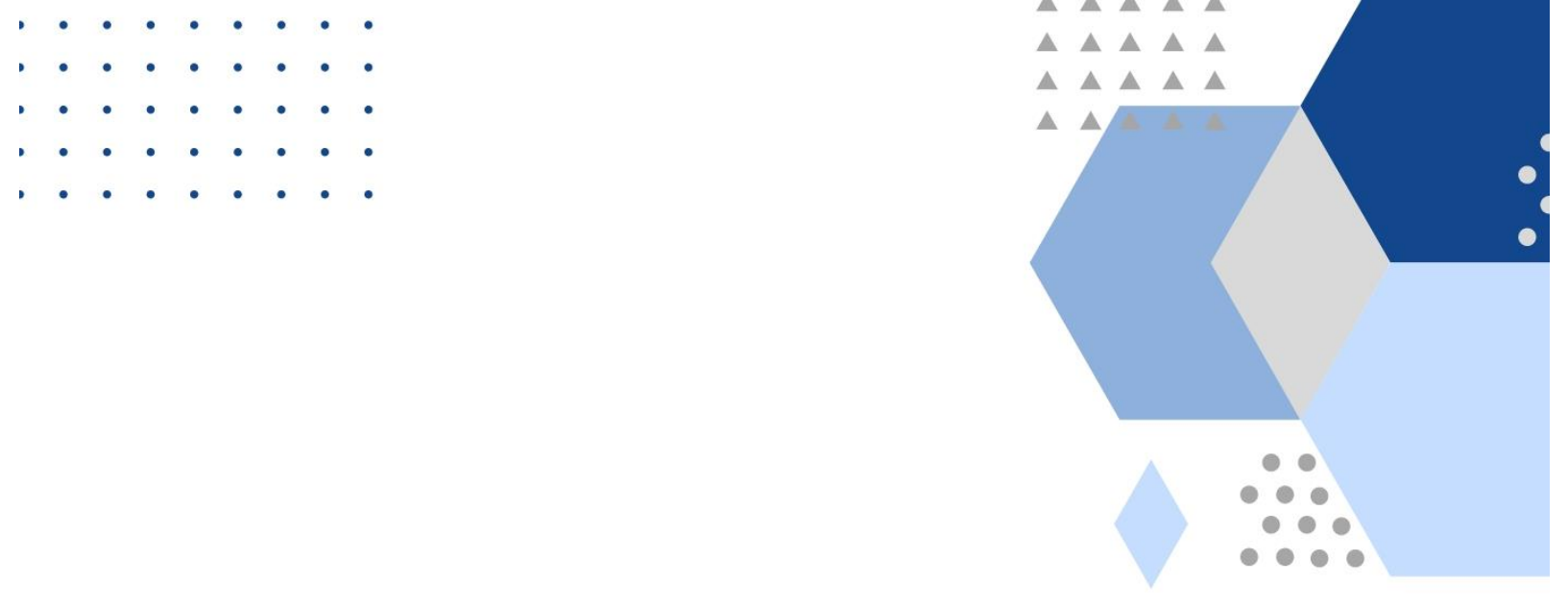


ANEXO C

MANUALES

**INGENIERÍA
DE
SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO





MANUAL DE ADMINISTRACION

VERSION 1.0

**SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE
ACTIVIDADES ACADÉMICAS, INVESTIGACIÓN E
INTERACCIÓN SOCIAL PREGRADO Y POSGRADO**

LIZETH GUTIERREZ ARUQUIPA



MANUAL DE ADMINISTRACIÓN

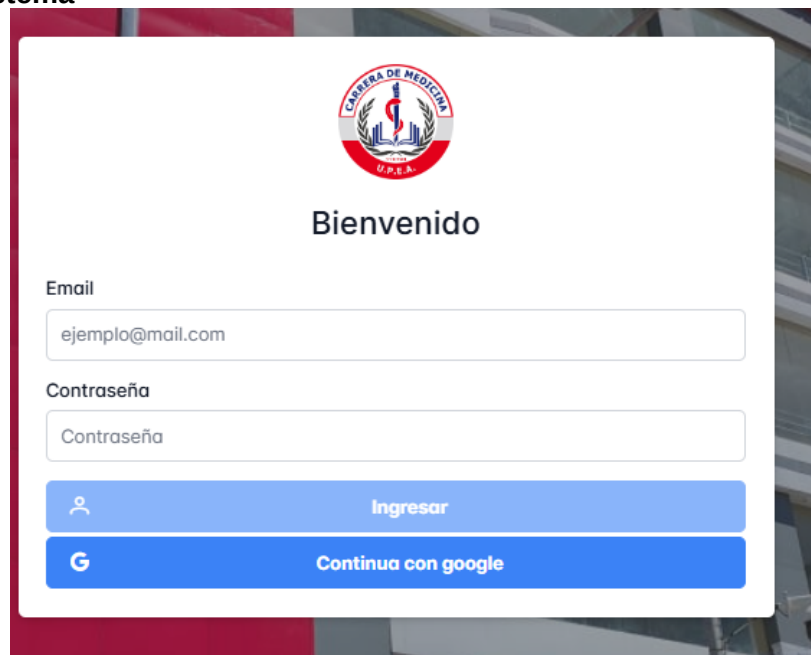
1. Introduccion.

El presente manual de administración pretende servir como guía para el uso adecuado del Sistema Integrado para la Centralización de Actividades Académicas, Investigación e Interacción Social pregrado y posgrado, para la parte administrativa

2. Objetivo

Brindar información necesaria para poder realzar la administración y configuración de los usuarios del sistema.

3. Guia del Sistema



The image shows a login interface for the 'CARRERA DE MEDICINA UPEL' system. At the top center is the university's logo, which features a caduceus and the text 'CARRERA DE MEDICINA' and 'UPEL'. Below the logo, the word 'Bienvenido' (Welcome) is displayed. There are two input fields: 'Email' with the placeholder 'ejemplo@mail.com' and 'Contraseña' (Password) with the placeholder 'Contraseña'. Below these fields are two buttons: a light blue button with a person icon and the text 'Ingresar' (Login), and a dark blue button with the Google 'G' logo and the text 'Continua con google' (Continue with google).

a. Ingresar email

b. Ingresar la contraseña

MODULO DE ADINISTRACIÓN DE USUARIO

The screenshot shows a dashboard titled 'MODULO DE ADINISTRACIÓN DE USUARIO'. At the top left is a 'DASHBOARD' link. Below it is the 'LISTA DE USUARIOS' section. Callout A points to the table header, B points to the '+ NUEVO USUARIO' button, C points to the 'Solicitudes 1' button, and D points to the table body.

NOMBRE(S) ▾	AP. PATERNO ▾	AP. MATERNO ▾	EMAIL ▾	C.I. ▾	Fecha de
Ana	Perez	Lima	abc@gmail.com	12345678	11/10/24, 12:16 PM
Otmar	Aguilar	Aguilar	ootmar1998@gmail.com	9991405	11/6/24, 10:34 PM
Otmar	Aguilar	Aguilar	ootmar1998@gmail.com	9991405	11/6/24, 10:33 PM
Liz	Gutierrez	Aruquipa	gutierrezzt321@gmail.com	123456	8/13/24, 10:29 PM
Lizet	Gutierrez	Aruquipa	gutierrezzt321@gmail.com	123456	8/5/24, 12:01 PM

A. Lista de usuarios

B. Registro de nuevo usuario

The 'NUEVO USUARIO' form contains the following fields:

- EMAIL (with a red error message above it)
- Nro. DE CEDULA
- NOMBRE(S)
- APELLIDO PATERNO
- APELLIDO MATERNO
- SELECCIONE ROL (dropdown menu)
- Guardar button

C. Registrar a las nuevas solicitudes

The table shows a list of new user requests. The first row is highlighted.

C.I.	NOMBRE	AP. PATERNO	AP. MATERNO
123123	weqwe	qweqwe	qweqwe

000015a

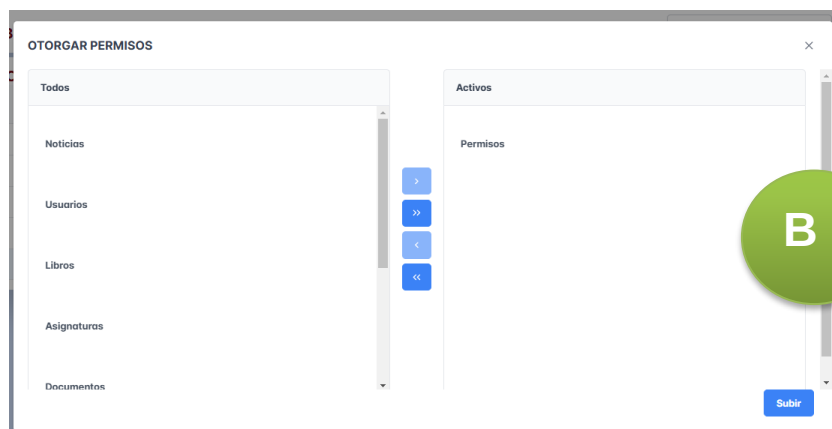


REPORTE DE USUARIOS

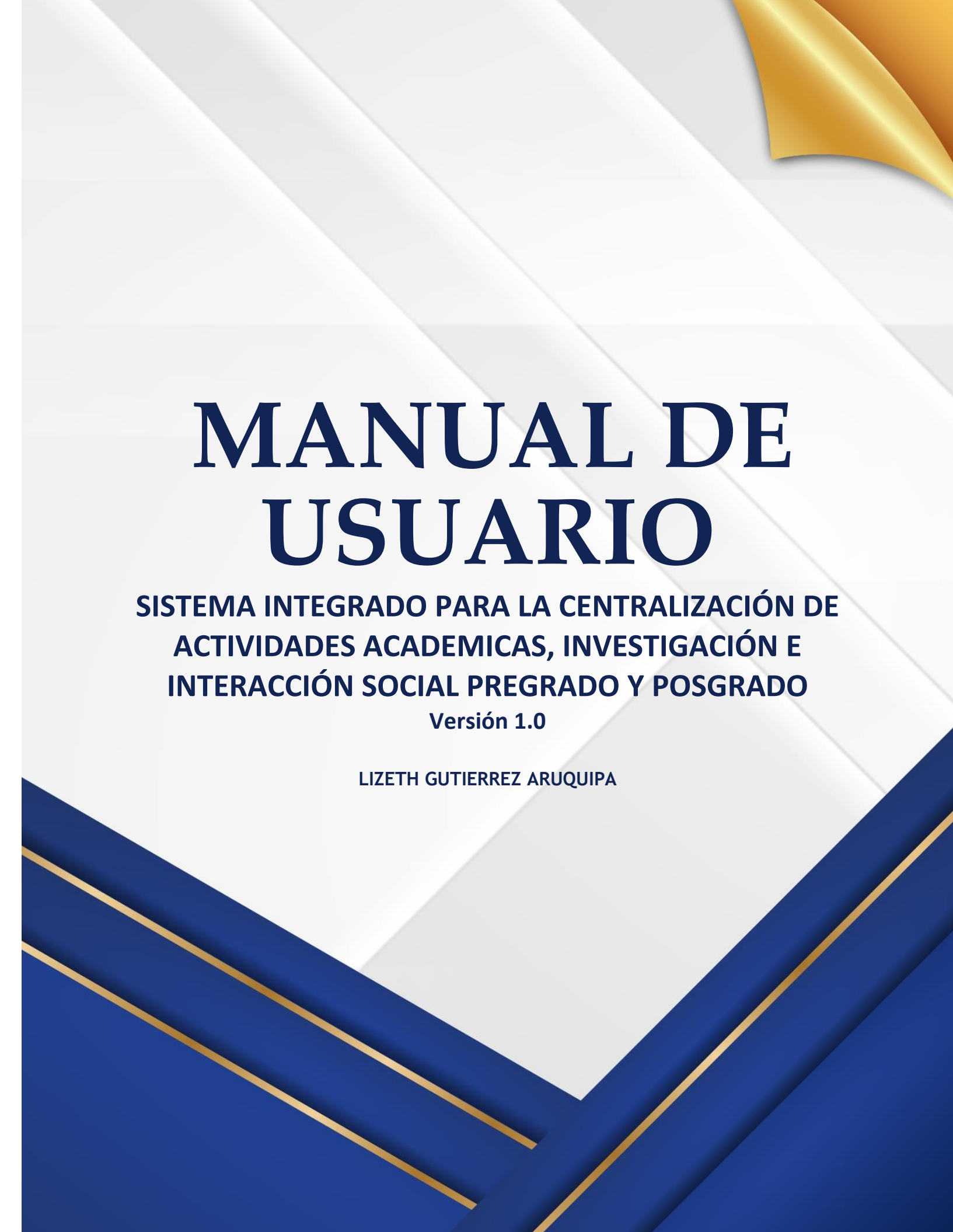
Nombres	Apellido Paterno	Apellido Materno	C.I.	Correo	Fecha de Alta
Ana	Perez	Perez	12345678	abc@gmail.com	10/11/2024, 12:16:13
Otmar	Aguilar	Aguilar	9991405	ootmar1998@gmail.com	6/11/2024, 22:34:24
Otmar	Aguilar	Aguilar	9991405	ootmar1998@gmail.com	6/11/2024, 22:33:40
Liz	Gutierrez	Gutierrez	123456	gutierrezlzt321@gmail.com	13/8/2024, 22:29:25
Lizet	Gutierrez	Gutierrez	123456	gutierrezlzt321@gmail.com	5/8/2024, 12:01:33

GESTION DE PERMISOS

PERSONAS DE INTERES				
NOMBRE(s) ▼	AP. PATERNO ▼	AP. MATERNO ▼	EMAIL ▼	C.I. ▼
Ana	Perez	Lima	abc@gmail.com	12345678
Otmar	Aguilar	Aguilar	ootmar1998@gmail.com	
Otmar	Aguilar	Aguilar	ootmar1998@gmail.com	
Liz	Gutierrez	Aruquipa	gutierrezlzt321@gmail.com	
Lizet	Gutierrez	Aruquipa	gutierrezlzt321@gmail.com	



- A.** Lista de usuarios
B. Ventana para otorgar permisos a un determinado usuario.



MANUAL DE USUARIO

**SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE
ACTIVIDADES ACADÉMICAS, INVESTIGACIÓN E
INTERACCIÓN SOCIAL PREGRADO Y POSGRADO**

Versión 1.0

LIZETH GUTIERREZ ARQUIPA

MANUAL DE USUARIO

1. INTRODUCCION.

El presente documento tiene como finalidad, servir de guía para el uso adecuado del SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE ACTIVIDADES ACADEMICAS, INVESTIGACIÓN E INTERACCIÓN SOCIAL PREGRADO Y POSGRADO.

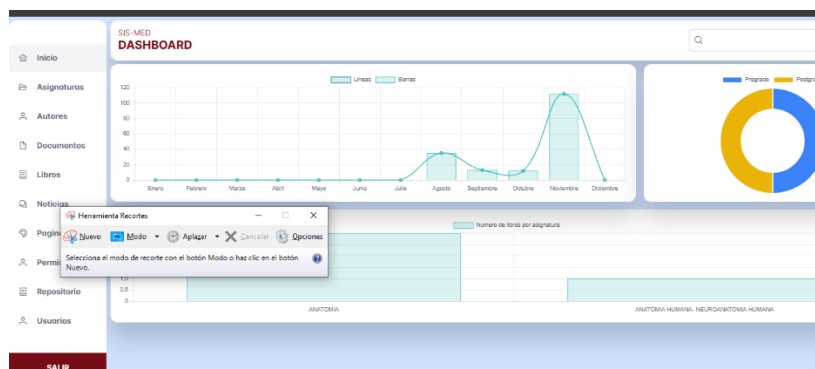
2. OBJETIVO DEL SISTEMA.

El objetivo del manual, es permitir que el usuario final pueda leer el conocimiento necesario para administrar el SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN DE ACTIVIDADES ACADEMICAS, INVESTIGACIÓN E INTERACCIÓN SOCIAL PREGRADO Y POSGRADO

3. GUIA DEL SISTEMA.

3.1. Inicio de sesión para el ingreso al sistema

3.2. Gestiones



Cuando el usuario logre ingresar al sistema podrá ver un Dashboard.

3.3. Gestión de asignaturas

SIS-MED DASHBOARD

Inicio Asignaturas Autores Documentos Libros Noticias Pagina Permisos Repositorio Usuarios

SALIR

Gestión de ASIGNATURAS

+ NUEVO REGISTRO PDF Excel

	ASIGNATURA ▾	AÑO ▾	SIGLA ▾	DESCRIPCION/PRE REQUISITO	CONTENIDO	FECHA DE CREACIÓN
7	LOGIA MEDICA	2	MES-200	MED-100 MED -101 MED - 102	✓	Nov 13, 2024
6	SALUD PUBLICA E INTERCULTURALIDAD	1	MED-13	Ninguno	✓	Nov 12, 2024
5	HISTOLOGIA MEDICA - BIOLOGIA CELULAR	1	MED-102	Ninguno	✓	Nov 12, 2024
4	EMBRIOLOGIA GENETICA	1	MED-101	Ninguno	✓	Nov 12, 2024
3	ANATOMIA HUMANA- NEUROANATOMIA HUMANA	1	MED-100	Ninguno	✓	Nov 12, 2024
2	FISIOLOGIA	2	FIO-655	MATERIA DE SEGUNDO AÑO	✓	Nov 10, 2024
1	ANATOMIA	1	AN-121	MATERIA DE PRIMER AÑO	✓	Nov 7, 2024

A. Lista de asignaturas registradas

B. Reportes en formato pdf y Excel

C. Registro de nueva asignatura

Nuevo Usuario

ASIGNATURA:

SIGLA:

DESCRIPCIÓN/PRE-REQUISITOS

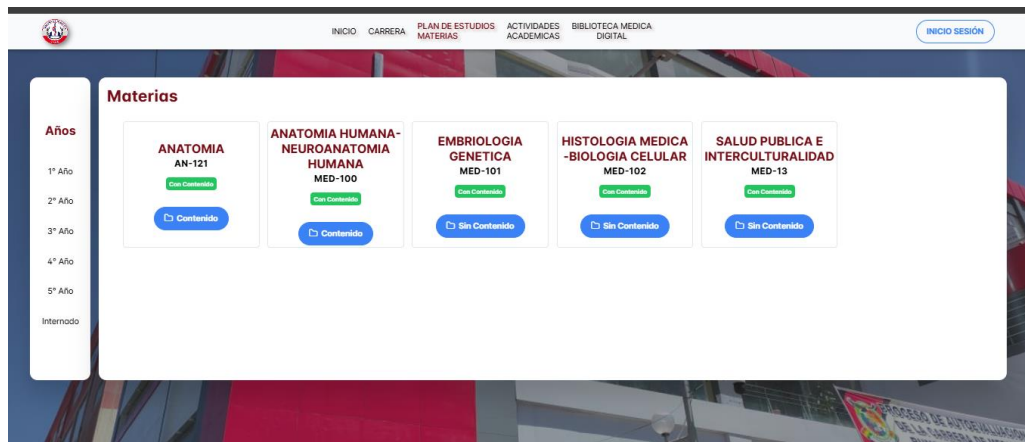
Subir Descripción PDF

SELECCIONE EL AÑO

Guardar

El usuario registra todos los campos requeridos, cargar un documento en formato pdf, selecciona el año al que corresponde la asignatura. Y finalmente presiona guardar.

Lo registrado se podrá visualizar al principio del sistema



3.4. Gestión de libros digitales

SIS-MED
DASHBOARD

Hola, Liz admin

BIBLIOTECA MEDICA DIGITAL

+ NUEVO PDF Excel

ID	TITULO	FECHA DE PUBLICACIÓN	EDICIÓN	IDIOMA	FECHA CREACION	AUTOR	ASIGNATURA
1	Anatomia Tomo_1	Nov 4, 2024	Primer Tomo	Castellano	Nov 10, 2024	Lexus	Anatomia
2	Anatomia Tomo_2	Feb 1, 2024	Primera Edicion	Castellano	Nov 10, 2024	Rodrigo Ramirez	Anatomia
3	Anatomia Clinica	Oct 31, 2024	Panamericana	Castellano	Nov 12, 2024	Eduardo Adrian	Anatomia Humana- Neuroanatomia Humana
4	Anatomia Humana	Jan 31, 2024	Primera	Cotellano	Nov 12, 2024	Dr. Quiróz	Anatomia

SALIR

A. Lista de libros digitales

B. Descarga de reportes de libros en formato pdf y/o Excel

C. Nuevo registro para libro

3.5. Gestión de noticias

Inicio

Asignaturas

Autores

Documentos

Libros

Noticias

Página

Permisos

Repositorio

Usuarios

SALIR

SIS-MED

DASHBOARD

Hola, Liz admin

GESTIÓN DE NOTICIAS

+ Nuevo

PDF

Excel

TÍTULO	CONTENIDO/DESCRIPCIÓN	Imagen	FECHA CREACIÓN	ACTIVO / NO ACTIVO
LA UPEA ABRE LA CARRERA DE MEDICINA EN CARANAVI PARA EL 2024	La carrera de medicina, expande sus horizontes, tal es el caso en el municipio de Caranavi. Los Jovenes de la población de Caranavi podran estudiar la carrera de medicina.	images/LA_UPEA ABRE LA CARRERA DE MEDICINA EN CARANAVI PARA EL 2024	Nov 12, 2024	✓
VIII OLIMPIADA NACIONAL - ANATOMIA I, NEUROANATOMIA	Muchas felicitades a toda la carrera de Medicina*, hoy nuestra delegación de estudiantes y docentes, llevaron en alto el nombre de nuestra querida Universidad Pública de El Alto, demostrando la "capacidad académica" de nuestros estudiantes y docentes. En un video sobre	images/VIII OLIMPIADA NACIONAL - ANATOMIA I, NEUROANATOMIA	Nov 12, 2024	✓
EL EMBARAZO PARTO PUERPERIO	la carrera de Medicina UPEA Bolivia La Paz-El Alto , donde tocamos un tema muy importante "El Embarazo Parto Puerperio".	images/EL_EMBARAZO PARTO PUERPERIO	Nov 12, 2024	✓
NOTICIA	EN LA SEMANA ANIVERSARIO	images/NOTICIA	Nov 7, 2024	✓
This is Fine	This is Fine	images/This_Is_Fine	Nov 7, 2024	✓
MHGFD	DS DFDF DFDF	images/MHGFD	Nov 7, 2024	✓

- A. Lista de noticias, comunicados, eventos que fueron suscitándose en la carrera.
- B. Descarga de reportes en formato pdf o Excel
- C. Creación de nueva noticia

NUEVA NOTICIA

TÍTULO

rt

B I U Sans Serif Normal

SELECCIONE CATEGORIA

VISIBLE?

+ Abir

Subir

Cancelar

 logo_upea.png 156.014 KB

Ayuda

Guardar

El usuario ingresa todos los campos requeridos, un título, una descripción en donde puede dar estilo al párrafo, seleccionar un tipo de noticia, seleccionar una imagen como portada de la noticia, subir y guardar el registro.

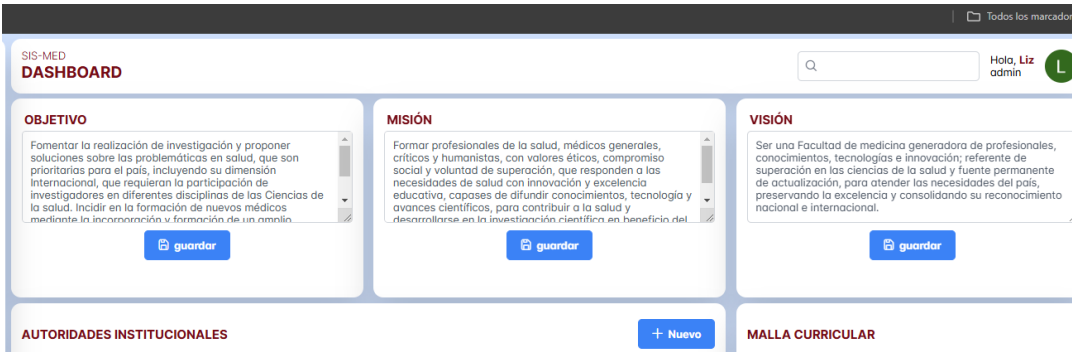
La noticia se verá reflejado en la pantalla principal.



3.6. Gestión de pagina

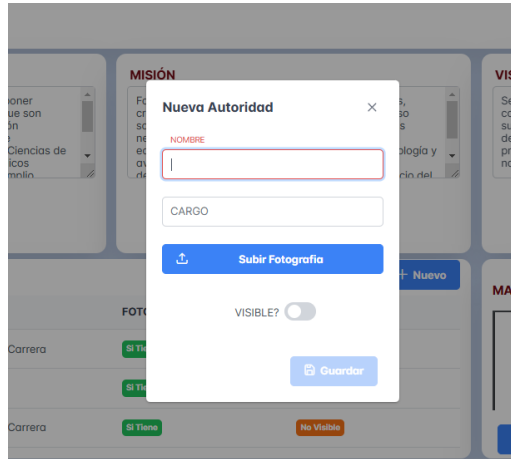


- A. Objetivo de la carrera
- B. Misión de la carrera
- C. Visión de la carrera



Hacer click en el texto y modificar si se requiere, y presiona guardar.

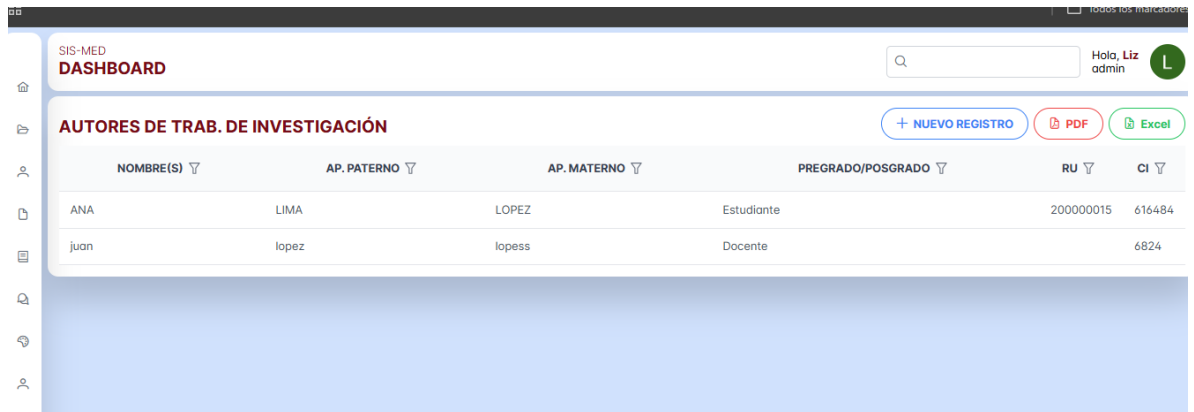
D. autoridades



The image shows a modal window titled "Nueva Autoridad" with a close button (X). Inside the modal, there is a text input field labeled "NOMBRE", another labeled "CARGO", a blue button with an upload icon labeled "Subir Fotografía", a toggle switch labeled "VISIBLE?" which is currently turned off, and a light blue button labeled "Guardar". The background shows a blurred view of a dashboard with various panels.

El usuario podrá ingresar datos de las autoridades vigentes, se tiene la opción de que sea visible o no.

3.7. Gestión de trabajos de investigación



The screenshot shows the "SIS-MED DASHBOARD" interface. At the top right, there is a search bar and a user profile for "Liz admin". Below the dashboard header, there is a section titled "AUTORES DE TRAB. DE INVESTIGACIÓN" with buttons for "+ NUEVO REGISTRO", "PDF", and "Excel". Below this is a table with the following data:

NOMBRE(S) ▾	AP. PATERNO ▾	AP. MATERNO ▾	PREGRADO/POSGRADO ▾	RU ▾	CI ▾
ANA	LIMA	LOPEZ	Estudiante	200000015	616484
juan	lopez	loposs	Docente		6824

- Para el registro de trabajos de investigación, primero se tiene que registrar el autor si no existiese en lista.

NUEVO REGISTRO

SELECCIONE: PREGRADO/POSGRADO

Nro. DE CEDULA

NOMBRE(s)

AP. PATERNO

AP. MATERNO

Guardar

Llenar los campos mínimos requeridos para la creación del nuevo autor de la investigación y finalmente presionar el botón de guardar.

SIS-MED DASHBOARD

REPOSITORIO-TRAB. DE INVESTIGACION

id	TITULO	FECHA PUBLICACIÓN	RESUMEN	PREGRADO/POSGRADO	FECHA CREACIÓN	AUTOR	NOMBRE DE DOC.
2	Estudio De Celular Madre	Nov 4, 2024	Es Una Investigacion De Alto Interes Para El Estudio De Celulas.	Postgrado	Nov 10, 2024	Ana Lima Lopez	Documentostenologos De La Internet.pdf
1	Nuevo Tra	Nov 6, 2024	Sass	Pregrado	Nov 7, 2024	Ana Lima Lopez	Documentosflujo De Sistema.pdf

- A. Lista de trabajos de investigación registrados.
- B. Generación de reportes en formato pdf o Excel.
- C. Nuevo registro de trabajo de investigación

Nuevo

TITULO

Subir PDF

DESCRIPCION/RESUMEN

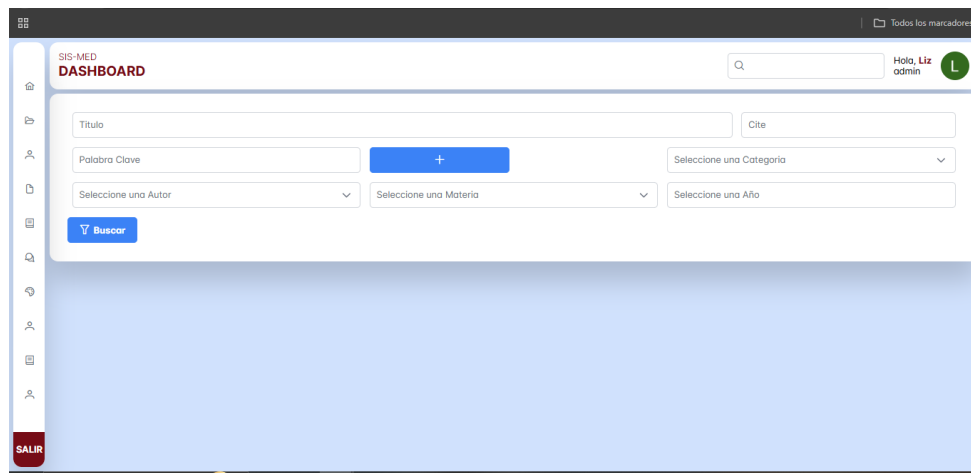
FECHA DE PUBLICACIÓN CITE

SELECCIONE AL AUTOR

PREGRADO/POSGRADO

Guardar

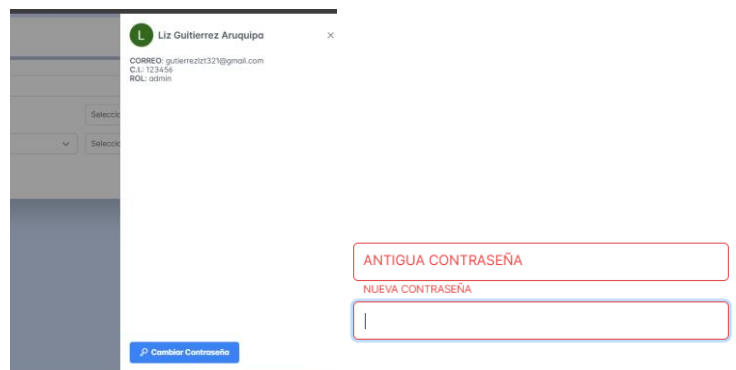
Introduce los datos requeridos, un título, carga el documento en pdf, la fecha de publicación, selecciona el autor de la investigación, y selecciona el tipo de grado, para finalmente guardar.



The screenshot shows the 'SIS-MED DASHBOARD' search interface. It features a sidebar on the left with icons for home, folders, users, documents, and a 'SALIR' (Logout) button at the bottom. The main content area has a search bar at the top right with the text 'Hola, Liz admin' and a profile icon. Below the search bar, there are several input fields: 'Titulo', 'Cite', 'Palabra Clave', 'Selecciona una Categoría' (dropdown), 'Selecciona una Autor' (dropdown), 'Selecciona una Materia' (dropdown), and 'Selecciona una Año'. A blue 'Buscar' (Search) button is located below the 'Palabra Clave' field. The bottom of the dashboard area is a large light blue rectangle.

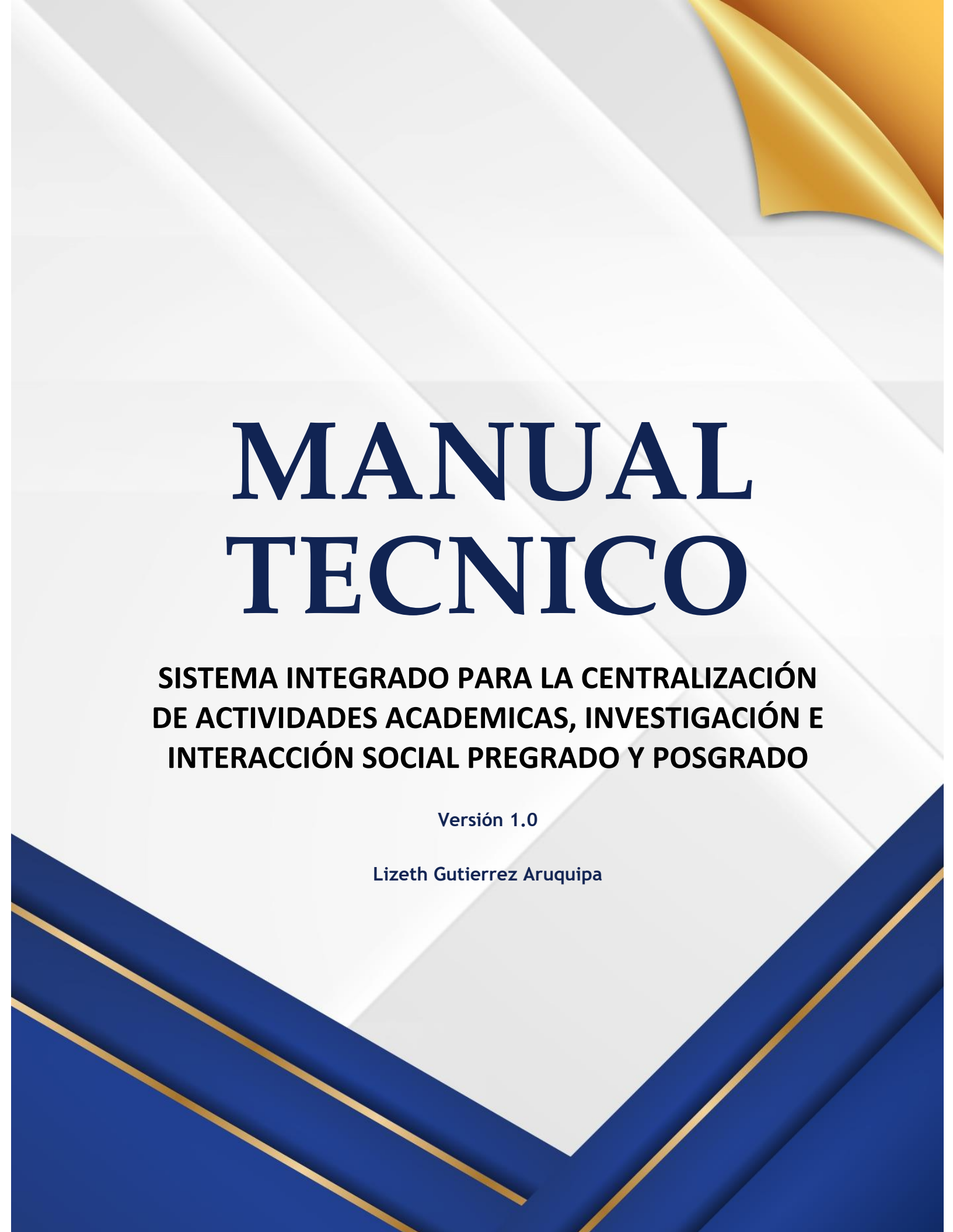
Para la busque de trabajos de investigación, se tiene un módulo dedicad, con todos los filtros.

3.8. Usuario



The screenshot shows a user profile page for 'Liz Guitierrez Ánuquipa'. The profile information includes 'CORREO: gultierrez1321@gmail.com', 'C.I: 123456', and 'RSL: admin'. Below the profile information, there are two dropdown menus labeled 'Selecciona' and 'Selecciona'. At the bottom left, there is a blue button labeled 'Cambiar Contraseña'. On the right side, there are two input fields: 'ANTIGUA CONTRASEÑA' and 'NUEVA CONTRASEÑA'. The 'NUEVA CONTRASEÑA' field has a cursor inside it.

El usuario podrá cambiar su contraseña estando en su perfil.



MANUAL TECNICO

**SISTEMA INTEGRADO PARA LA CENTRALIZACIÓN
DE ACTIVIDADES ACADEMICAS, INVESTIGACIÓN E
INTERACCIÓN SOCIAL PREGRADO Y POSGRADO**

Versión 1.0

Lizeth Gutierrez Aruquipa

MANUAL TECNICO

El Sistema de centralización, integración de actividades académicas fue desarrollado pensando en el servicio que necesita la carrera de Medicina para el mejor control de sus actividades. El propósito fundamental es facilitar las operaciones y minimizar tiempo de respuesta a peticiones.

A continuación, detallamos las herramientas que se usaron para el desarrollo del sistema.

1. Objetivo

Detallar la información necesaria para realizar la instalación, configuración y puesta en funcionamiento del aplicativo.

2. Objetivos específicos

- Definir claramente el proceso de instalación del aplicativo.
- Detallar los requerimientos mínimos de hardware y software que el funcionamiento del aplicativo.
- Describir las herramientas utilizadas en el desarrollo de la aplicación.

3. Requerimientos Técnicos

Una de las herramientas necesarias al momento de realizar desarrollo web es una computadora la cual debería cumplir con las siguientes características:

- Procesador: Intel i5 o AMD Ryzen 5 en adelante
- Memoria Ram_ 16GB
- Almacenamiento: Disco SSD (Estado sólido)

- Conexión a internet

Sin embargo, estos requerimientos son flexibles.

Para la implementación del proyecto deberá contar con el hardware recomendable de:

- 2 Cores 3.2 GHz. O superior
- 2 Gb de RAM
- 16 Gb de almacenamiento

Para el uso del software desarrollado los equipos o dispositivos deberán cumplir con las siguientes características:

- Procesos: Dual Core o superior
- Memoria: 2 Gb o superior
- Almacenamiento: 120 GB o mas
- Sistema Operativo: Windows, Linux, Mac
- Otros: Conexión a internet

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

Nestjs: Nest.js es un framework de desarrollo web basado en Node.js que utiliza TypeScript para proporcionar una estructura de programación sólida y altamente escalable.

Se inspira en otros framework de desarrollo populares, como Angular y Spring. Y se centra en la creación de aplicaciones de Backend altamente y modulares, aunque también puedes desarrollar aplicaciones web.

Es compatible e incluye integraciones por defecto con una amplia variedad de bases de datos y sistemas de autenticación. Además, se puede utilizar con una amplia variedad de módulos de terceros para añadir funcionalidades adicionales. También se integra con herramientas de testing y depuración para ayudar a probar y depurar.

Angular

Angular es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript.

Angular se basa en clases tipo “componentes”, cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades(variables) y métodos (funciones a llamar).

Angular es la evolución de AngularJS aunque incompatible con su predecesor.

MariaDB

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

MariaDB está escrito en C y C++ y es compatible con varios lenguajes de programación, incluidos C, C#, Java, Python, PHP. MariaDB también es compatible con todos los principales sistemas operativos, incluidos Windows, Linux y macOS.

Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, esta disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación.

NodeJs.

Node.js es un entorno en tiempo de ejecución, multiplataforma para la capa del servidor (en el lado del servidor) basado en JavaScript.

Node.js es un entorno controlado por eventos diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tiene que preocuparse con el bloqueo de procesos, pues no hay bloqueo.

NPM.

NPM es parte esencial de Node.js el entorno de ejecución de JavaScript en el lado del servidor basado en el motor V8 de Google. Es un seguramente la principal razón del gran éxito de Node permitiendo que cientos de desarrolladores puedan compartir paquetes de software entre distintos proyectos.

En este momento, se han publicado 1493231 paquetes por medio de su repositorio con mas de 27 mil millones de descargas.

NPM responde a las siglas de Node Package Manager, o manejo de paquetes de node, es la herramienta por defecto de JavaScript para la tarea de compartir e instalar paquetes.

Instalación del Backend

Para realizar la instalación se debe copiar el código fuente, el cual se encuentra en cd adjunto a este manual, una vez copiado la carpeta contenida en la carpeta de Backend

procedemos a introducir el siguiente comando en una consola.

- **npm install**

Para el desarrollo

- **npm run start:dev**

***para producción**

- **npm run start:prod**

Configuración de las Bases de datos

Para conectar la base de datos MariaDB o MySQL al servidor Backend se debe contar con las credenciales de acceso a la base de datos, con esos datos se procesa a crear el archivo .env dentro de la carpeta del proyecto con la siguiente estructura:

- DATABASE_HOST

- DATABASE_PORT

- DATABASE_USER

- DATABASE_PASSWORD

- DATABASE_NAME

- DATABASE_SSL

Instalación Frontend

Para realizar la instalación se debe copiar el código fuente el cual se encuentra en CD adjunto a este manual, una vez copiado la carpeta contenida en la carpeta Frontend procedemos a introducir el siguiente comando en una consola.

- `npm install`

Para el desarrollo

- `ng serve`

Para producción

- `ng build`