

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL”

CASO: ESCUELA SUPERIOR DE FORMACIÓN DE MAESTROS “MCAL. ANDRÉS DE SANTA CRUZ Y CALAHUMANA”

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

Postulante : Univ. David Carani Quispe
Tutor Metodológico : M. Sc. Lic. Ing. Maricel Yarari Mamani
Tutor Revisor : Lic. Ing. Adelaida Ximena Pastrana Arcani
Tutor Especialista : M. Sc. Lic. Juan Carlos Sarzuri Patzi

EL ALTO – BOLIVIA
2023

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo **David Carani Quispe**, estudiante con **CI. 6001668 LP** mediante la presente **declaro** de manera pública que la propuesta del **PROYECTO DE GRADO** titulada **“REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL”** es original, siendo resultado de 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.

Asimismo, me hago responsable ante la universidad o terceros de cualquier 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 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 2023

David Carani Quispe
CI:6001668 LP
Correo: david.carani2407@gmail.com

DEDICATORIA

Este proyecto va dedicado a:

Mis padres Gregorio y Catalina (+) por todo su amor y ser el pilar fundamental de mi educación.

Mis hermanas Elvira, Miguelina, Gladis, Mónica y Hilda por darme siempre ese apoyo incondicional durante todo este tiempo.

AGRADECIMIENTO

A Dios por darme muchas bendiciones y ser la fortaleza que me impulsa a seguir adelante.

A mis padres y hermanas por estar presentes en todo momento y apoyarme de manera continua.

A mis tutores Lic. Juan Carlos Sarzuri Patzi y Ing. Adelaida Ximena Pastrana Arcani por sus sugerencias, observaciones y correcciones, brindándome constante ánimo y consejos para culminar el proyecto

A la Carrera de Ingeniería de Sistemas por los años de enseñanza y realizar mi vida profesional.

Al Lic. Edy Lupa Juárez Director General de la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana”, por darme su confianza, tiempo y colaboración en la realización de este proyecto.

¡GRACIAS!

RESUMEN

El Sistema Educativo Plurinacional se va involucrando cada vez más al mundo digital al uso y aplicación de las nuevas tecnologías de la Información y la Comunicación TICs, por esta razón, las TICs adquiere mucha importancia por contribuir en el desarrollo de nuevas formas de enseñanza y aprendizaje de la comunidad educativa.

En muchas instituciones educativas, se observa que las nuevas tecnologías aún están ausentes, por tal razón se da la iniciativa de desarrollar un sistema que atienda los requerimientos de investigación de la comunidad educativa; con la finalidad de difundir la información académica y el producto académico que realizan los estudiantes de esta institución.

Para el desarrollo del sistema se aplica la metodología OOHDM que propone el diseño de aplicaciones hipermedia y Web mediante un proceso de cinco fases que son: obtención de requerimientos, diseño conceptual, diseño navegacional, diseño abstracto de la interfaz y la implementación.

Para la implementación se hace uso de JavaScript y HTML como lenguaje de programación, PostgreSQL que es gestor de base de datos y DSpace software de código abierto para la creación de Repositorios y Bibliotecas Digitales que provee herramientas para la administración de colecciones digitales, también se aplica framework front-end de Bootstrap para adaptar la interfaz del sistema al tamaño dispositivo.

La estimación del costo se la realiza aplicando COCOMO II, luego se lleva a cabo las pruebas de caja blanca y caja negra, el cual muestran que los elementos del sistema funcionen de manera adecuada, la NORMA ISO 9126 muestra en nivel de aceptabilidad del sistema, la: usabilidad, funcionalidad, confiabilidad y eficiencia.

ÍNDICE GENERAL

CAPÍTULO I

1. MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES.....	2
1.2.1. Antecedentes Institucionales	2
1.2.1.1. Visión.....	2
1.2.1.2. Misión.....	3
1.2.1.3. Organigrama.....	3
1.2.2. Antecedentes afines al Proyecto de Grado.....	4
1.2.2.1. Antecedentes Internacionales.....	4
1.2.2.2. Antecedentes Nacionales	4
1.2.2.3. Antecedentes Locales.....	5
1.3. PLANTEAMIENTO DEL PROBLEMA	6
1.3.1. Problema Principal.....	6
1.3.2. Problemas Secundarios.....	6
1.3.3. Formulación del Problema	7
1.4. OBJETIVOS.....	7
1.4.1. Objetivo General.....	7
1.4.2. Objetivos Específicos.....	7
1.5. JUSTIFICACIÓN.....	8
1.5.1. Justificación Técnica.....	8
1.5.2. Justificación Económica.....	8
1.5.3. Justificación Social.....	9
1.6. METODOLOGÍA	9
1.6.1. Metodología OOHDM (Método de Diseño Hipermedia Objeto Orientado)	9
1.6.2. Métricas de calidad	10
1.6.3. COSTOS.....	11
1.6.4. Seguridad	11
1.6.5. Pruebas de Software	12
1.7. MÉTODOS DE RECOLECCIÓN DE DATOS.....	12
1.7.1. Técnicas de investigación	13
1.8. HERRAMIENTAS	13
1.9. LIMITES Y ALCANCES	14

1.9.1. Limites	14
1.9.2. Alcances	15
1.10. APORTES.....	15
1.10.1. Aporte Teórico	15
1.10.2. Aporte Práctico	15
CAPITULO II	
2. MARCO TEÓRICO	16
2.1. REPOSITORIO.....	16
2.1.1. Tipos de Repositorio	16
2.2. METODOLOGIA OOHDM (Método de Diseño Hipermedia Objeto Orientado) ..	17
2.2.1. Fase I. Obtención de Requerimientos	18
2.2.2. Fase II. Diseño Conceptual.....	20
2.2.3. Fase III. Diseño Navegacional	20
2.2.4. Fase IV. Diseño de Interfaz Abstracta.....	24
2.2.5. Fase V. Implementación	24
2.3. LENGUAJE UNIFICADO DE MODELADO (UML)	26
2.3.1. Modelos	27
2.3.2. Elementos Comunes a Todos los Diagramas	27
2.3.2.1. Notas	27
2.3.2.2. Dependencias	28
2.3.3. Diagramas de Estructura Estática.....	28
2.3.3.1. Clases.....	29
2.3.3.2. Objetos	29
2.3.3.3. Asociaciones.....	30
2.3.3.3.1. Nombre de la Asociación y Dirección.....	30
2.3.3.3.2. Multiplicidad	31
2.3.3.3.3. Roles.....	32
2.3.3.3.4. Agregación.....	32
2.3.3.3.5. Clases Asociación.....	33
2.3.3.3.6. Asociaciones N-Alias	34
2.3.3.3.7. Navegabilidad	34
2.3.3.4. Herencia	35
2.3.3.5. Elementos Derivados.....	35
2.3.4. Diagrama de Casos de Uso	36

2.3.4.1. Elementos.....	36
2.3.4.1.1. Actores.....	36
2.3.4.1.2. Casos de Uso	36
2.3.4.1.3. Relaciones entre Casos de Uso.....	37
2.3.5. Diagramas de Interacción	38
2.3.5.1. Diagrama de Secuencia.....	38
2.3.5.2. Diagrama de Colaboración	39
2.3.6. Diagrama de Estados	40
2.4. CALIDAD DE SOFTWARE	41
2.4.1. ISO/IEC 9126.....	42
2.4.1.1. Características de ISO 9126	43
2.5. PRUEBAS DE SOFTWARE.....	45
2.5.1. Método de la Caja Blanca	45
2.5.1.1. Prueba de ruta básica.....	45
2.5.2. Método de la Caja Negra	46
2.6. MODELO PARA EL ANÁLISIS DE COSTOS	48
2.6.1. Cocomo II	48
2.6.1.1. Modelo de Diseño Temprano.....	48
2.6.1.2. Puntos de Función no Ajustados.....	49
2.6.1.3. Puntos de Función Ajustados	49
2.6.1.4. Estimación del Esfuerzo.....	51
2.6.1.6. Costo Total del Sistema	55
2.7. HERRAMIENTAS DE APLICACIÓN WEB.....	55
2.7.1. Servidor HTTP Apache	55
2.7.1.1. Apache Maven	56
2.7.1.2. Instalación de Maven	56
2.7.1.3. Apache Tomcat.....	57
2.7.2. DSpace	57
2.7.3. Lenguaje de Programación	57
2.7.3.1. HTML.....	58
2.7.3.2. JavaScript.....	58
2.7.3.3. CSS	59
2.7.4. Gestor de base de Datos	59
2.7.4.1. PostgreSQL	60

2.7.4.2. Funciones de PostgreSQL	60
2.7.4.3. Ventajas de PostgreSQL.....	61
2.7.5. Patrón de Arquitectura Modelo Vista Controlador (MVC).....	62
CAPITULO III	
3. MARCO APLICATIVO	64
3.1. DESARROLLO DE LA METODOLOGÍA OOADM	64
3.1.1. Obtención de Requerimientos.....	64
3.1.1.1. Identificación de Roles y Tareas	64
3.1.2. Especificación de Escenarios	65
3.1.3. Especificación de Casos de Uso.....	66
3.1.4. Especificación de UIDS (User Integration Diagram).....	72
3.2. DISEÑO CONCEPTUAL.....	75
3.3. DISEÑO DE NAVEGACIÓN	77
3.4. DISEÑO DE INTERFAZ ABSTRACTA	78
3.5. IMPLEMENTACIÓN.....	82
CAPITULO IV	
4. CALIDAD, COSTOS, SEGURIDAD Y PRUEBAS	93
4.1. MÉTRICAS DE CALIDAD	93
4.1.1. NORMA ISO 9126	93
4.1.1.1. Funcionalidad.....	93
4.1.1.2. Confiabilidad	96
4.1.1.3. Mantenibilidad.....	97
4.1.1.4. Usabilidad	98
4.1.1.5. Portabilidad.....	99
4.1.1.6. Nivel de Aceptabilidad.....	99
4.1.2. Pruebas de software	99
4.1.2.1. Pruebas de Caja Negra.....	100
4.1.2.2. Prueba de Caja Blanca	104
4.2. ESTIMACIÓN DE COSTOS.....	106
4.2.1. Modelo de Diseño Temprano.....	106
4.2.1.1. Calculo de los Puntos de Función no Ajustados	106
4.2.1.2. Calculo de los Puntos de Función Ajustados	107
4.2.1.3. Estimación del Esfuerzo.....	109
4.2.1.4. Estimación del Cronograma.....	112

4.2.1.5. Costo Total del Sistema	112
CAPITULO V	
5. CONCLUSIONES Y RECOMENDACIONES	113
5.1. CONCLUSIONES	113
5.2. RECOMENDACIONES	114
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE TABLAS

Tabla 1. Definición de contextos	23
Tabla 2. Resumen del OOHDM	25
Tabla 3. Puntos función sin ajustar	49
Tabla 4. Factores de complejidad	50
Tabla 5. Lineas de código por cada punto función.....	51
Tabla 6. Factores de escala.....	52
Tabla 7. Valores de los factores de escala	53
Tabla 8. Factores de ajustes.....	54
Tabla 9. Valores de factores de ajustes	54
Tabla 10. Definición y especificación de escenarios	65
Tabla 11. Caso de uso registro de administrador.....	67
Tabla 12. Caso de uso evaluación de trabajo de grado	68
Tabla 13. Caso de uso registro de trabajo de grado	69
Tabla 14. Caso de uso publicación de trabajo de grado	69
Tabla 15. Caso de uso administración de administrador	70
Tabla 16. Caso de uso administración de administrador general.....	71
Tabla 17. Factores de ponderación	94
Tabla 18. Ajustes de complejidad	94
Tabla 19. Ajustes de usabilidad	98
Tabla 20. Resultados obtenidos.....	99
Tabla 21. Casos de prueba.....	106
Tabla 22. Puntos función sin ajustar	106
Tabla 23. Factores de complejidad	107
Tabla 24. Lineas de código por cada punto de función.....	108
Tabla 25. Factores de escala.....	110
Tabla 26. Valores de los factores de escala	110
Tabla 27. Factores de ajuste	111
Tabla 28. Valores de los factores de ajuste	111

ÍNDICE DE FIGURAS

Figura 1. Organigrama institucional	3
Figura 2. Metodología OOHDM	18
Figura 3. Diagrama de interacción	19
Figura 4. Esquema conceptual	20
Figura 5. Diagrama de contexto.....	23
Figura 6. Mapeo de los objetivos	24
Figura 7. Logo de ULM	26
Figura 8. Ejemplo de nota.....	27
Figura 9. Dependencias.....	28
Figura 10. Notación para clases	29
Figura 11. Objetos	30
Figura 12. Asociación	31
Figura 13. Multiplicidad.....	31
Figura 14. Roles de una asociación.....	32
Figura 15. Agregación.....	33
Figura 16. Clase de asociación.....	33
Figura 17. Asociación ternaria	34
Figura 18. Herencia	35
Figura 19. Atributo derivado.....	36
Figura 20. Diagrama de casos de uso	37
Figura 21. Diagrama de secuencias	38
Figura 22. Diagrama de colaboración	39
Figura 23. Diagrama de estados.....	41
Figura 24. Características ISO/IEC 9126	43
Figura 25. Características de la vista de uso	44
Figura 26. Notación de tráfico de flujo	46
Figura 27. Caso de uso de administración.	66
Figura 28. Caso de uso registro de administrador	67
Figura 29. Caso de uso de evaluación de trabajo de grado.....	68
Figura 30. Caso de uso registro de trabajo de grado	68
Figura 31. Caso de uso publicación de trabajo de grado	69
Figura 32. Caso de uso administración de administrador	70
Figura 33. Caso de uso administración de administrador general	71

Figura 34. UID autenticación de usuario	72
Figura 35. UID registro de administrador	72
Figura 36. UID evaluación de trabajo de grado.....	73
Figura 37. UID registro de trabajo de grado.....	73
Figura 38. UID publicación de trabajo de grado	74
Figura 39. UID administración de administrador	74
Figura 40. UID administración de administrador general	75
Figura 41. Diagrama de clases del repositorio institucional	76
Figura 42. Diseño de navegación de la pagina principal del sistema	77
Figura 43. Diseño de navegacion del sistema académico	78
Figura 44. Pagina principal del sistema	79
Figura 45. Página de modalidad de graduación.....	79
Figura 46. Búsqueda de información	80
Figura 47. Visualizar y descargar archivos	80
Figura 48. Autenticación y acceso de administrador	81
Figura 49. Envío y publicación de documentos	81
Figura 50. Interfaz de usuario	82
Figura 51. Interfaz de página principal.....	83
Figura 52. Interfaz de comunidades y carreras	83
Figura 53. Interfaz de subcomunidades y modalidades de graduación.....	84
Figura 54. Interfaz publicación de documentos.....	85
Figura 55. Interfaz descarga de documentos.....	85
Figura 56. Interfaz de acceso y autenticación de administrador.....	86
Figura 57. Interfaz de administrador	86
Figura 58. Interfaz de nuevo registro	87
Figura 59. Interfaz de registro de usuario administrador.....	87
Figura 60. Interfaz herramientas de administrador.....	88
Figura 61. Interfaz crear nueva comunidad.....	88
Figura 62. Interfaz crear una colección	89
Figura 63. Interfaz listado de colecciones	89
Figura 64. Interfaz enviar documentos.....	90
Figura 65. Interfaz publicar documentos	90
Figura 66. Interfaz editar documentos	91
Figura 67. Interfaz retirar, borrar y actualizar documentos.....	91

Figura 68. Interfaz publicar noticias	92
Figura 69. Interfaz editar noticias.....	92
Figura 70. Interfaz inicio de sesión de sistema	100
Figura 71. Interfaz modificar contraseña.....	101
Figura 72. Interfaz modulos enlaces del sistema	102
Figura 73. Interfaz búsqueda de información	102
Figura 74. Interfaz información académica	103
Figura 75. Interfaz descargar archivo	103
Figura 76. Prueba de caja blanca	104
Figura 77. Alta de administrador	105
Figura 78. Reporte de alta de administrador	105

CAPÍTULO I

1. MARCO PRELIMINAR

En este capítulo se describen los antecedentes institucionales, se identifica la problemática que la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana” presenta, y se plantea las soluciones a estos. También, se dan a conocer los objetivos, las justificaciones, límites y alcances del proyecto a desarrollar.

1.1. INTRODUCCIÓN

El presente proyecto, está enfocado a un proceso ágil, dinámico y práctico de la administración y almacenamiento de documentos de grado. Que permita a la comunidad investigadora un acceso fácil y abierto a todo tipo de contenido digital almacenado, incluido textos, imágenes y documentos con información de mucha importancia; se quiere optimizar los recursos de tiempo y distancia en la búsqueda de información.

Almacenar documentos de trabajo de grado con sus respectivos formatos digitales en un espacio físico y que estos sean manipulados para realizar consultas, provocan un deterioro a largo plazo, ocasiona la pérdida de información de estos documentos. Implementando una tecnología de consulta de archivos y documento, podemos evitar este tipo de inconvenientes.

Se propone desarrollar un Repositorio Institucional (RI) de información que reúna, preserve, divulgue y de acceso a la producción intelectual y académica. Un sistema, que brindare a los estudiantes, tutores, coordinadores y otros investigadores, la facilidad de consultar Trabajos de Grado que se hayan desarrollado en la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana”.

Para el desarrollo del proyecto, se empleará la tecnología OpenSource, que desarrolla el software de manera descentralizada y colaborativa, flexible y duradero; emplearemos lenguajes de programación y PostgreSQL como gestor de base de

datos. También emplearemos el software DSpace para crear repositorios digitales, es un software gratis y fácil de instalar "listo para usar" y completamente personalizable para adaptarse a las necesidades de cualquier organización.

1.2. ANTECEDENTES

1.2.1. Antecedentes Institucionales

La Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana” fue fundada el año 1958, desde entonces va formando maestros del área técnica con una gran variedad de denominaciones en atención a las políticas educativas vigentes.

Actualmente se encuentra ubicado en el barrio de Munaypata de la ciudad de La Paz. Tiene el objetivo de formar profesionales siguiendo el lineamiento Socio Comunitario y Productivo en marco a la LEY N° 070 Avelino Siñani - Elizardo Pérez, a nivel Licenciatura en las carreras: Electrónica y Electricidad, Electromecánica, Mecánica Automotriz, Industria de alimentos y Gastronomía.

Sus modalidades de graduación son:

- Excelencia Académica
- Recuperación de Saberes y Conocimientos
- Proyecto Productivo Técnico Tecnológico
- Investigación Educativa Bajo el Enfoque de la Investigación Acción Participativa
- Producción de Textos Para el Desarrollo Curricular

1.2.1.1. Visión

Escuela Superior, líder en la formación de maestros técnicos, tecnológicos emprendedores y productivos, con alto compromiso social para el Estado Plurinacional de Bolivia.

1.2.1.2. Misión

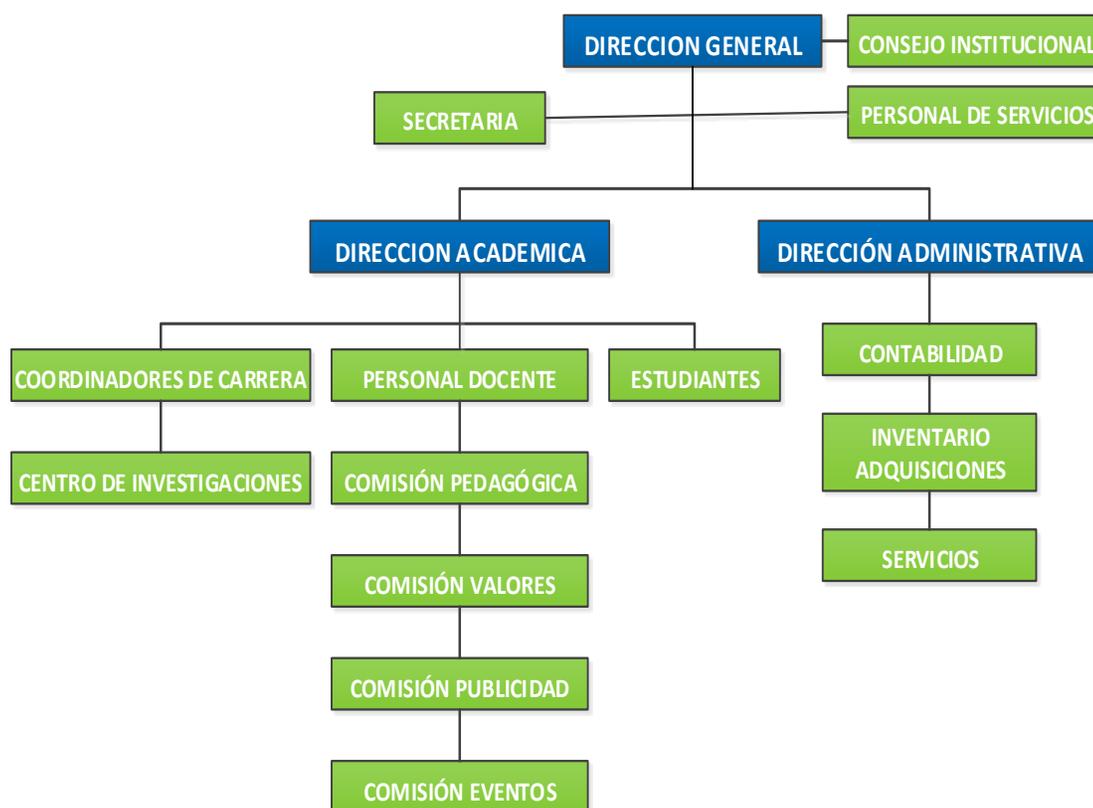
Formar maestros técnico-tecnológicos con excelencia profesional, para los subsistemas regular y alternativa en sus diferentes áreas, niveles, modalidades y especialidades que requiere el Sistema Educativo Plurinacional, aplicando metodologías que respondan a la educación descolonizadora, productiva, sostenible.

1.2.1.3. Organigrama

La Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana”, en su estructura funcional se rige en el marco legal de la Ley N°070, Código de Educación, Resolución Administrativa, y Reglamento de Faltas y Sanciones.

Figura 1

Organigrama Institucional



Nota. El organigrama muestra la estructura funcional de la E.S.F.M. "Mcal. Andrés de Santa Cruz y Calahumana". Fuente: E.S.F.M. "Mcal. Andrés de Santa Cruz y Calahumana"

1.2.2. Antecedentes afines al Proyecto de Grado

1.2.2.1. Antecedentes Internacionales

- Pinzón, M. (2001). *Diseño e implementación de una biblioteca virtual basada en el buscador para universidades*. Esta biblioteca virtual está basada en un motor de búsqueda que les permita a sus usuarios buscar, encontrar y consultar la información que estará organizada por categorías tales como tipos de documentos, materia, título, autor, etc. Utiliza herramientas de desarrollo HTML, JavaScript, php, motor de base de datos y software de acceso libre.
- Sánchez, J. (2015). *Diseño de biblioteca virtual con acceso personalizado y capacitación de electrónica básica en el centro de proyección social monte de galilea usme*. Este proyecto se realiza con miras al mejoramiento de la biblioteca Centro de Proyección Social Monte de Galilea USME, este proyecto de investigación está marcado en la función de proyección social.
- Paredes, B. (2009). *Diseño y aplicación de una biblioteca virtual de tesis digitales de la escuela de ingeniería industrial*. El diseño y aplicación de una biblioteca virtual mejora la entrega de información de tesis digitales, beneficiando a los estudiantes y dependientes de la biblioteca. La biblioteca es fundamental en el desarrollo académico de los estudiantes, en la actualidad este proceso carece de aspectos importantes que no permiten el uso de nuevas herramientas y recursos bibliográficos que eleven el nivel tecnológico de la biblioteca.

1.2.2.2. Antecedentes Nacionales

- Costa, W. (2021). *Diseño de una biblioteca virtual en la carrera de mecánica industrial de la facultad tecnología de la Universidad Mayor de San Andrés*. El proyecto se desarrolla con el modelo Cliente Servidor, utiliza herramientas: Macromedia Dreamweaver como editor de código HTML profesional para el diseño visual y la administración de sitios y páginas web, PHP (acrónimo de

“PHP: Hipertext Preprocessor”) un lenguaje de programación con variables, sentencias condicionales, bucles y funciones. Por medio de las redes los usuarios de la información se benefician de las bibliotecas virtuales por que establecen las intranets y los portales que permite el acceso a los diferentes soportes de información que cuentan las bibliotecas, como son los bases de datos, publicaciones electrónicas y otros.

- Quispe, H. (2019). *Digitalización del archivo académico de la unidad de kardex de la carrera de derecho, Universidad Mayor de San Andrés*. El proyecto presenta una propuesta para la digitalización de documentos con servicios automatizados, para la gestión de información y documentos aplica software libre, editor de texto, programa PostgreSQL, digitalización de imágenes, ficheros y aplicaciones. Se desarrolla bajo normas de gestión de archivos y normativas legales y estándares de normas ISO.

1.2.2.3. Antecedentes Locales

- Gutiérrez, E. (2020). *Sistema de gestión y digitalización bibliotecaria*. Actualmente con el avance de la ciencia y la tecnología contar, con un sistema automatizado es una necesidad en cualquier entidad, por tal razón se ha propuesto el Sistema de Gestión y Digitalización Bibliotecaria, para que brinde un mejor servicio al personal de biblioteca como ser: préstamo de libros, descargas de libros digitalizados, reportes y estadísticas.
- Orihuela, G. (2023). *Diseño e implementación de una biblioteca virtual y repositorio de proyectos de grado y tesis*. Hoy en día las tecnologías de la información y la comunicación juegan un papel fundamental, no solo como mediadores instrumentales, sino como nuevos procesos de enseñanza y aprendizaje, generando cambios importantes en la enseñanza académica y la investigación, El desarrollo de la tecnología de la información afecta sin duda a las funciones tradicionales de los centros bibliotecarios, que deben afrontar nuevos retos en la sociedad de la información.

1.3. PLANTEAMIENTO DEL PROBLEMA

La Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana”, cuenta con un depósito de trabajos de grado con sus respectivos formatos digitales, mismos que se encuentran en la Dirección Académica, y con el transcurso de cada gestión se van acumulando más documentos sin tener un lugar apropiado donde almacenarlo.

El espacio físico con la que cuenta la biblioteca de la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana” también es insuficiente; la cantidad de trabajos de grado en resguardo, implican una manipulación y deterioro de estos documentos a largo plazo, sabiendo que los docentes y estudiantes de la misma institución, son quienes realizan la consulta de estos archivos para fines académicos.

1.3.1. Problema Principal

Los estudiantes y docentes de la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana” no tienen un acceso eficiente a la producción intelectual ni a los trabajos de grado realizados, no ofrecen un servicio de búsqueda de información porque no existe un sistema de consultas bibliográficas dentro de la institución,

1.3.2. Problemas Secundarios

De acuerdo al diagnóstico efectuado a la biblioteca de la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana”, se ha logrado evidenciar algunas falencias:

- El espacio físico de funcionamiento de la biblioteca es muy reducido.
- El almacenamiento de la cantidad de trabajos es de forma exponencial.
- Información bibliográfica desactualizada que refleja el descontento de la población estudiantil.

- Documentos de trabajo de grado en deterioro por la manipulación constante de los mismos.
- Trabajos de grado repetido, similar o copia de proyectos anteriormente presentados.
- Falta de acceso a redes bibliográficas con otras bibliotecas de instituciones superiores y universidades.
- Las horas de atención se limita, porque las actividades académicas de los estudiantes son en los mismos horarios.

1.3.3. Formulación del Problema

En función a los problemas mencionados anteriormente, surge la siguiente interrogante:

¿De qué manera se podrá mejorar la administración y la gestión de los productos académicos finales desarrollados en la Escuela Superior de Formación de Maestros Mcal. Andrés de Santa Cruz y Calahumana?

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar un Repositorio Institucional que cumpla los estándares específicos de almacenamiento bibliográfico y que permita realizar la búsqueda de información de acuerdo a las modalidades de graduación en la Escuelas Superiores de Formación de Maestros.

1.4.2. Objetivos Específicos

- Diseñar un repositorio que permita almacenar documentos en formato digital con sus respectivas medidas de seguridad.
- Desarrollar una página web de acuerdo a normas de calidad y seguridad que requiere un sistema.

- Realizar el módulo de búsqueda de información que permita visualizar y las descargas los productos académicos finales autorizados.
- Configurar el módulo de almacenamiento de archivos digital para que los lectores puedan acceder a estos las 24 horas.
- Disponer del módulo de reportes y acceso al sistema para generar estadísticas de forma automática.
- Configurar el acceso a la producción intelectual y académica de las diferentes carreras según las modalidades de graduación de la institución.

1.5. JUSTIFICACIÓN

1.5.1. Justificación Técnica

La administración de un servidor en la nube, requiere de Internet con un ancho de banda adecuado, para la configuración de la base de datos, la página web y otros servicios que requiera el sistema; la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana” cuenta con el servicio de internet adecuado para el desarrollo del sistema.

Para el acceso al presente sistema de información no será necesario aplicar tecnología de punta, sino simplemente la terminal o computadora que cada uno tiene, ya que se tendrá acceso desde cualquier equipo mediante una conexión de red a internet.

1.5.2. Justificación Económica

Con la implementación de este sistema, se optimizará las principales tareas de búsqueda de información, se facilitará el documento de Modalidad de Titulación o el documento de Trabajo Final de Titulación en formato digital, minimizando gastos en papelería e impresiones, permitiendo a toda persona realizar consultas desde todo lugar que tenga acceso a internet.

Las características económicas del proyecto son:

- Desarrollo del proyecto, no generará gastos económicos a la institución:
- Servidor, el costo del alquiler del servidor en la nube no tendrá un costo elevado, será cubierto por el desarrollador del proyecto.
- Herramientas de desarrollo, Se empleará para el desarrollo del presente proyecto tecnología OpenSource (frameworks, plantillas, lenguajes de programación, gestor de base de datos).
- El proyecto y su implementación no generarán ingresos económicos a la institución.

1.5.3. Justificación Social

El Repositorio Institucional facilitara el acceso, almacenamiento y envío rápido de la información y a la vez beneficiará a toda la comunidad investigadora y sediento de conocimiento de la mismas y otras Instituciones, quienes a su vez mejoraran el proceso de obtención y aprovechamiento de toda información proveniente de la Escuela Superior de Formación de Maestros.

Además de mejorar la búsqueda de información, que tendrá un nivel de atención elevado y dejando la posibilidad de que otras instituciones superiores se vean en la necesidad de implementar sistemas similares, que permita al estudiante gestionar su propio proceso de aprendizaje.

1.6. METODOLOGÍA

1.6.1. Metodología OOHDM (Método de Diseño Hipermedia Objeto Orientado)

OOHDM es una metodología de desarrollo propuesta por Rossi y Schwabe (ROSSI 1996) para la elaboración de aplicaciones multimedia y tiene como objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia.

OOHDM está basada en HDM, en el sentido de que toma muchas de las definiciones, sobre todo en los aspectos de navegación, planteadas en el modelo de

HDM. Sin embargo, OOHDM supera con creces a su antecesor, ya que no es simplemente un lenguaje de modelado, sino que define unas pautas de trabajo, centrado principalmente en el diseño, para desarrollar aplicaciones multimedia de forma metodológica.

OOHDM como técnica de diseño de aplicaciones hipermedia, propone un conjunto de tareas que según Schwabe, Rossi y Simone pueden resultar costosas a corto plazo, pero a mediano y largo plazo reducen notablemente los tiempos de desarrollo al tener como objetivo principal la reusabilidad de diseño, y así simplificar el coste de evoluciones y mantenimiento.

Esta metodología plantea el diseño de una aplicación de este tipo a través de cinco fases que se desarrollan de un modo iterativo. Estas fases son:

- Fase I. Obtención de Requerimientos
- Fase II. Diseño Conceptual
- Fase III. Diseño Navegacional
- Fase IV. Diseño de Interfaz Abstracta
- Fase V. Implementación

1.6.2. Métricas de calidad

Las métricas de calidad de software es un conjunto de medidas utilizadas para estimar la calidad de un proyecto a desarrollar, entre otros conceptos, y que permiten comparar o planificar estas aplicaciones. Si no se mide, no hay una forma real de determinar si se está mejorando y si no se está mejorando, se está perdido. (Barrientos, 2018).

- **ISO/IEC 9126**

La ISO 9126 es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-característica consta de un conjunto de atributos que son medidos por una serie de métricas. Prieto M. (2017).

Las métricas de la ISO 9126 son:

- Funcionalidad
- Confiabilidad
- Facilidad de uso
- Eficiencia
- Facilidad de mantenimiento
- Portabilidad

1.6.3. COSTOS

En modelo COCOMO es uno de los sistemas de estimación de costes más utilizados en proyectos de desarrollo de software. La estandarización de su uso y la facilidad de la aplicación del mismo junto con la aproximación al coste real, han convertido a este modelo en uno de los referentes en este tipo de proyectos. (Calero, 2010).

- **COCOMO II**

El modelo COCOMO original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evolucionó hacia un modelo de estimación más exhaustivo, llamado COCOMO II que requiere 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.

1.6.4. Seguridad

La seguridad de la información es una responsabilidad de gobernanza corporativa. No puede ser vista como una iniciativa aislada del equipo de Tecnología de la Información, y sí como un tópico de estrategia de negocio.

Dentro de esta perspectiva, las organizaciones han luchado para proteger informaciones controladas, críticas y/o confidenciales de accesos indebidos que pueden causarle daños irreversibles.

- Las normas ISO/IEC 27001 y la ISO/IEC 27002 especifican los requerimientos para establecer, implementar, operar, monitorear, revisar, mantener y mejorar un SGSI, además especifica los requerimientos para la implementación de controles de seguridad frente a las necesidades de toda la organización, frente a un proceso específico o un servicio, según el objetivo y los alcances del SGSI que se haya definido. Francisco S. Edgar R. y Miriam V. (2015), Revista Tecnológica ESPOL – RTE, Vol. 28.

1.6.5. Pruebas de Software

La prueba de software es el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe hacer. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento. Hay diferentes tipos de pruebas de software, cada una con objetivos y estrategias específicos:

- Prueba de aceptación
- Pruebas de integración
- Pruebas de unidad
- Pruebas funcionales
- Pruebas de rendimiento
- Pruebas de regresión
- Pruebas de estrés
- Pruebas de usabilidad

1.7. MÉTODOS DE RECOLECCIÓN DE DATOS

Hay muchas maneras de recolectar información en una investigación. El método elegido depende de la pregunta de investigación que se formula. Algunos métodos de recolección de información incluyen encuestas, entrevistas, pruebas, evaluaciones, observaciones, revisión de registros existentes etc.

1.7.1 Técnicas de investigación

- **Observación**

Es un método de recolección de datos consiste en el registro sistemático, válido y confiable de comportamientos y situaciones observables, a través de un conjunto de categorías y subcategorías. Hernández S. (2014)

- **Entrevista**

Es un método de recolección en donde implica que una persona calificada (entrevistador) aplica el cuestionario a los participantes; el primero hace las preguntas a cada entrevistado y anota las respuestas. Hernández S. (2014).

- **Lectura de normas vigentes**

De esta manera se recopiló información acerca de las normativas vigentes de escuelas superiores de formación de maestros y la educación técnica tecnológica.

1.8. HERRAMIENTAS

Las herramientas de desarrollo de software que se utilizan en las etapas de análisis, diseño y desarrollo del sistema se detallan a continuación:

HTML, es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y videos en la página.

CSS, es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas.

JavaScript, es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás. (Está bien, no todo, pero es sorprendente lo que puedes lograr con unas pocas líneas de código JavaScript).

Framework, es una herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas. Los framework hacen que el desarrollador no esté continuamente “reinventado la rueda” y se centre en el problema que quiere resolver y no en la implementación de funcionalidades que normalmente son de uso común y que ya están resueltas por otros.

Apache HTTP Server, es un software de servidor web de código abierto para plataformas Unix. Permite a los propietarios de sitios web servir contenido en la web y es uno de los servidores más confiables.

PostgreSQL, es un gestor de bases de datos relacional que presenta varias características por las que destaca, siendo uno de los mejores y más utilizados motores de BD en la actualidad; por el cual este será el gestor de base de datos utilizados en la implementación de este proyecto.

1.9. LIMITES Y ALCANCES

1.9.1. Limites

El repositorio tiene las siguientes limitantes:

- El Sistema, no tendrá relación con recursos humanos de la institución.
- La Gestión de Archivos y documentos solo lo podrán realizar los administradores.
- El administrador es el responsable de verificar que la información compartida es verídica.
- El Repositorio, no permitirá que personas ajenas elimine, compartan ni modifiquen la información.
- El Proyecto, no tendrá relación directa con otros repositorios dentro y fuera de la institución.

1.9.2. Alcances

Con la implementación del Repositorio, se pretende mejorar la atención a los docentes y estudiantes de manera óptima y personalizada. Se desarrollará con:

- Módulo de Administración.
- Módulo de Gestión.
- Módulo de Consulta de proyectos
- Procesos de emisión de documentos
- Estadísticas

1.10. APORTES

1.10.1. Aporte Teórico

Brindar a los estudiantes, tutores y coordinadores la facilidad de consultar los Trabajos de Grado que se hayan desarrollado en la Escuela Superior de Formación de Maestros “Mcal. Andrés de San Cruz y Calahumana” sin restringir el acceso a estos documentos a estudiantes o docentes externos a la institución.

1.10.2. Aporte Práctico

Este proyecto optimizará los recursos de tiempo y distancia, logrando como resultado un proceso ágil, dinámico y práctico de la administración y almacenamiento de documentos de grado.

CAPÍTULO II

2. MARCO TEÓRICO

En esta etapa se establece la información necesaria sobre esta metodología y se incluye conceptos sobre el análisis de costos, pruebas y calidad de software. Al finalizar se describen las herramientas que serán utilizadas por el desarrollador. Todo, con el fin de tener una base teórica para el desarrollo del sistema y que permita alcanzar los objetivos planteados.

2.1. REPOSITORIO

Los repositorios son sistemas de información o espacios centralizados que almacenan, mantienen, organizan materiales científicos académicos y difunde información digital, habitualmente archivos digitales, como apoyo a la investigación y el aprendizaje, a la vez que garantizan el acceso a la información.

Los repositorios están compuestos por múltiples archivos digitales representativos de la producción intelectual que resulta de la actividad investigadora de la comunidad científica y tiene la finalidad de organizarla, preservarla y difundirla en modo de acceso abierto.

Los repositorios institucionales consisten en estructuras web interoperables de servicios informáticos, dedicadas a difundir la perpetuidad de los recursos científicos y académicos (físicos o digitales) de las universidades, a partir de la enumeración de un conjunto de datos específicos (metadatos), para que esos recursos se puedan recopilar, catalogar, acceder, gestionar, difundir y preservar de forma libre y gratuita, de manera que están estrechamente ligados a los ideales y objetivos del acceso abierto.

2.1.1. Tipos de Repositorio

Los repositorios se dividen en dos grandes grupos:

- **Los repositorios institucionales**, proveen contenidos producidos por una institución en particular, universidades o comunidades autónomas.
- **Los repositorios temáticos**, proveen contenido de un campo disciplinar o asunto en particular.

También pueden clasificarse de acuerdo con su tipo de contenidos. Aunque algunos repositorios albergan contenidos de varios tipos, en términos generales pueden clasificarse como:

- Repositorios científicos
- Repositorios de objetos de aprendizaje.
- Repositorios de datos.
- Repositorios de patrimonio cultural.

A nivel particular, podemos aprovechar alojamientos tipo Google Drive o OneDrive. Desde la Junta, al ofrecernos la herramienta Teams, cada grupo de trabajo también puede almacenar hasta un 1 Tb de contenidos.

2.2. METODOLOGIA OOHDM (Método de Diseño Hipermedia Objeto Orientado)

OOHDM es una metodología de desarrollo propuesta por “Rossi y Schwabe (ROSSI 1996)” para la elaboración de aplicaciones multimedia y tiene como objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia. OOHDM está basada en HDM, en el sentido de que toma muchas de las definiciones, sobre todo en los aspectos de navegación, planteadas en el modelo de HDM.

OOHDM como ya se ha comentado es una metodología de desarrollo para aplicaciones multimedia. Antes de comenzar a detallar cada una de las fases que propone, es necesario resaltar algunas de sus características:

- Está basada en el paradigma de la orientación a objetos. En esto se diferencia de su antecesor HDM.

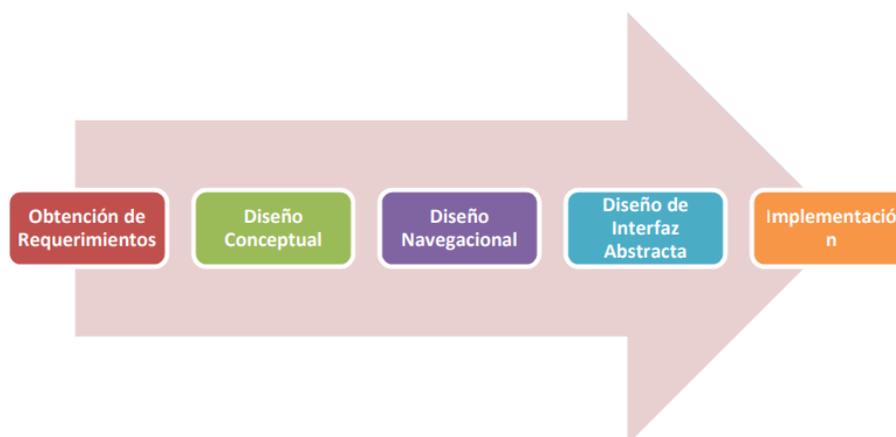
- Propone un modelo para representar a las aplicaciones multimedia, sino que propone un proceso predeterminado para el que indica las actividades a realizar y los productos que se deben obtener en cada fase del desarrollo.

OOHDM como técnica de diseño de aplicaciones hipermedia, propone un conjunto de tareas que según Schwabe, Rossi y Simone pueden resultar costosas a corto plazo, pero a mediano y largo plazo reducen notablemente los tiempos de desarrollo al tener como objetivo principal la reusabilidad de diseño, y así simplificar el coste de evoluciones y mantenimiento.

Esta metodología plantea el diseño de una aplicación de este tipo a través de cinco fases que se desarrollan de un modo iterativo. Estas fases son:

Figura 2

Metodología OOHDM



Nota. En la figura muestra las fases de la Metodología OOHDM. Fuente: Schwabe y Rossi (1996)

2.2.1. Fase I. Obtención de Requerimientos

La herramienta en la cual se fundamenta esta fase son los diagramas de casos de usos, los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema.

Según German (2003) primero que todo es necesario la recopilación de requerimientos. En este punto, se hace necesario identificar los actores y las tareas que ellos deben realizar. Luego, se determinan los escenarios para cada tarea y tipo

de actor. Los casos de uso que surgen a partir de aquí, serán luego representados mediante los Diagramas de Interacción de Usuario (UIDs), los cuales proveen de una representación gráfica concisa de la interacción entre el usuario y el sistema durante la ejecución de alguna tarea.

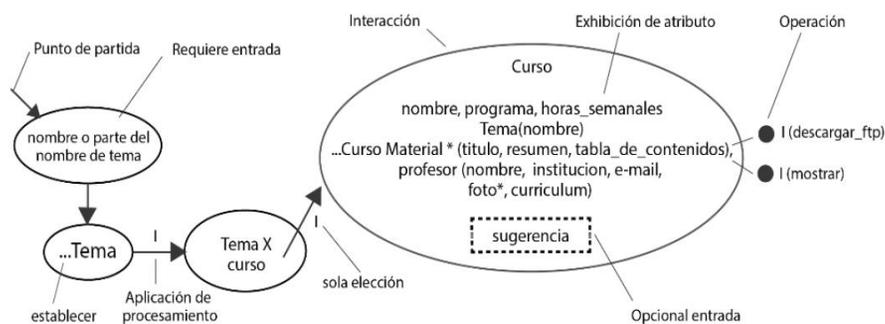
Con este tipo de diagramas se capturan los requisitos de la aplicación de manera independiente de la implementación. Ésta es una de las fases más importantes, debido a que es aquí donde se realiza la recogida de datos, para ello se deben de proporcionar las respuestas a las siguientes interrogantes:

- ¿Cuáles son los tópicos principales que serán atendidos?
- ¿Cómo los tópicos están relacionados entre sí?
- ¿Qué categoría de usuarios serán atendidos?
- ¿Cuáles son las tareas principales que serán abordadas?
- ¿Qué tareas corresponden a qué categoría de usuarios?
- ¿Los recursos disponibles son competitivos con la información levantada?

Con las preguntas mencionadas anteriormente, se puede recaudar de cierta manera las bases necesarias para la construcción de una aplicación hipertextual exitosa, sin embargo, mientras mayor sea el nivel de profundidad de la recolección de datos, mayor probabilidad de realizar una aplicación adecuada a las necesidades de los usuarios.

Figura 3

Diagrama de interacción



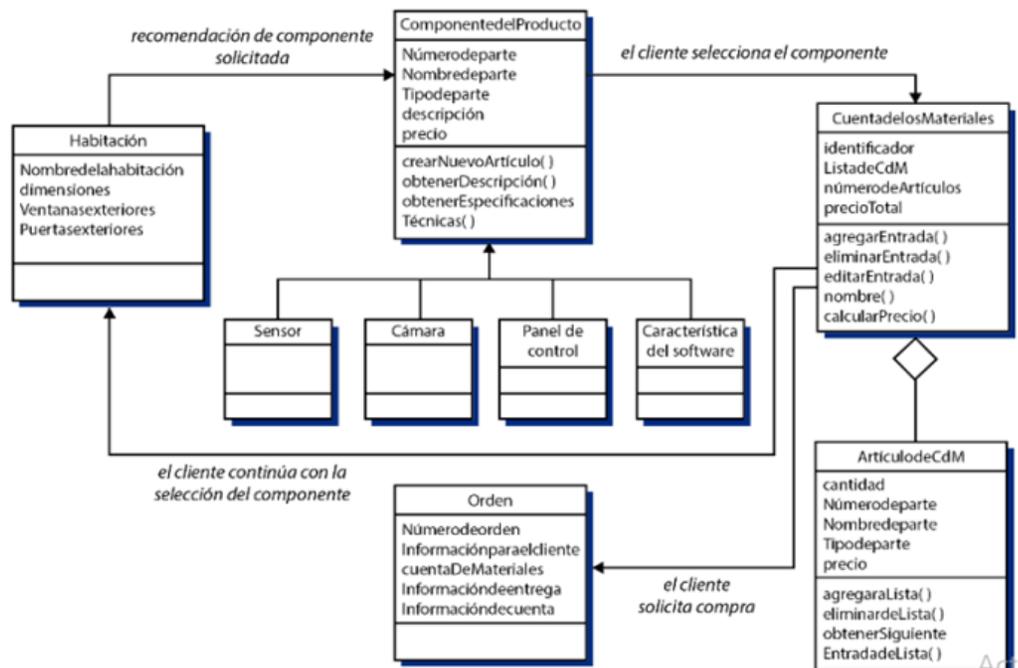
Nota. La figura muestra el diagrama de interacción de usuario, caso de uso “Buscar cursos dado un tema”. Fuente: Guell, Schawabe, & Patricia

2.2.2. Fase II. Diseño Conceptual

Se construye un modelo orientado a objetos, según Koch (2002) que represente el dominio de la aplicación usando las técnicas propias de la orientación a objetos. La finalidad principal durante esta fase es capturar el dominio semántico de la aplicación en la medida de lo posible, teniendo en cuenta el papel de los usuarios y las tareas que desarrollan. El resultado de esta fase es un modelo de clases relacionadas que se divide en subsistemas.

Figura 4

Esquema conceptual



Nota. En la figura muestra el esquema conceptual parcial para Casa Segura Asegurada. Fuente: Pressman (2010).

2.2.3. Fase III. Diseño Navegacional

En OOHDM una aplicación se ve a través de un sistema de navegación. En la fase de diseño navegacional se debe diseñar la aplicación teniendo en cuenta las tareas que el usuario va a realizar sobre el sistema.

Para ello, hay que partir del esquema conceptual desarrollado en la fase anterior. Hay que tener en cuenta que sobre un mismo esquema conceptual se pueden desarrollar diferentes modelos navegacionales (cada uno de los cuales dará origen a una aplicación diferente).

La estructura de navegación de una aplicación hipermedia está definida por un esquema de clases de navegación específica, que refleja una posible vista elegida.

En OOHDM hay una serie de clases especiales predefinidas, que se conocen como clases navegacionales: Nodos, Enlaces y Estructuras de acceso, que se organizan dentro de un Contexto Navegacional.

La semántica de los nodos y los enlaces son comunes a todas las aplicaciones hipermedia, las estructuras de acceso representan diferentes modos de acceso a esos nodos y enlaces de forma específica en cada aplicación.

- **Nodos:** Los nodos son contenedores básicos de información de las aplicaciones hipermedia. Se definen como vistas orientadas a objeto de las clases definidas durante el diseño conceptual usando un lenguaje predefinido y muy intuitivo, permitiendo así que un nodo sea definido mediante la combinación de atributos de clases diferentes relacionadas en el modelo de diseño conceptual. Los nodos contendrán atributos de tipos básicos (donde se pueden encontrar tipos como imágenes o sonidos) y enlaces.
- **Enlaces:** Los enlaces reflejan la relación de navegación que puede explorar el usuario. Ya sabemos que para un mismo esquema conceptual puede haber diferentes esquemas navegacionales y los enlaces van a ser imprescindibles para poder crear esas vistas diferentes.
- **Estructuras de Acceso:** Las estructuras de acceso actúan como índices o diccionarios que permiten al usuario encontrar de forma rápida y eficiente la información deseada. Los menús, los índices o las guías de ruta son ejemplos de estas estructuras. Las estructuras de acceso también se modelan como

clases, compuestas por un conjunto de referencias a objetos que son accesibles desde ella y una serie de criterios de clasificación de las mismas.

- **Contexto Navegacional:** Para diseñar bien una aplicación hipermedia, hay que prever los caminos que el usuario puede seguir, así es como únicamente podremos evitar información redundante o que el usuario se pierda en la navegación. En OOHDM un contexto navegacional está compuesto por un conjunto de nodos, de enlaces, de clases de contexto y de otros contextos navegacionales. Estos son introducidos desde clases de navegación (enlaces, nodos o estructuras de acceso), pudiendo ser definidas por extensión o de forma implícita.
- **Clase de Contexto:** Es otra clase especial que sirve para complementar la definición de una clase de navegación. Por ejemplo, sirve para indicar qué información está accesible desde un enlace y desde dónde se puede llegar a él.
- **La navegación no se encontraría definida sin el otro modelo que propone OOHDM:** el contexto navegacional. Esto es la estructura de la presentación dentro de un determinado contexto. Los contextos navegacionales son uno de los puntos más criticados a OOHDM debido a su complejidad de expresión.

Una vez definidas las clases de navegación, el OOHDM “estructura el espacio de navegación agrupando los objetos de navegación en conjuntos llamados contextos”.

cada definición de contexto incluye, además de los elementos que se incluyen en ella, la especificación de su estructura de navegación interna, un punto de entrada las restricciones de acceso en términos de clases de usuario y las operaciones, y una estructura de acceso asociada.

Existen 6 formas de definir contextos:

Tabla 1

Definición de contextos

Nombre	Acción	Grafica
Clase derivada simple	Son todos los objetos de una clase que satisface alguna propiedad.	
Grupo derivado de clase	Es un conjunto de contextos derivados de la clase sencilla, donde la definición de la propiedad de cada contexto es parametrizada.	
Enlace simple derivado	Son todos los objetos relacionados con un objeto determinado.	
Enlace grupo derivado	Es un conjunto de contextos de enlace derivado, cada uno de los cuales se obtiene variando el elemento de origen de la conexión.	
Arbitrario	Es un conjunto enumerado.	
Dinámico	Es un conjunto donde cambian los elementos durante la navegación. Asociado a los contextos, existen estructuras de acceso (índices). Índice Índice dinámico Índice con ordenamiento múltiples	   

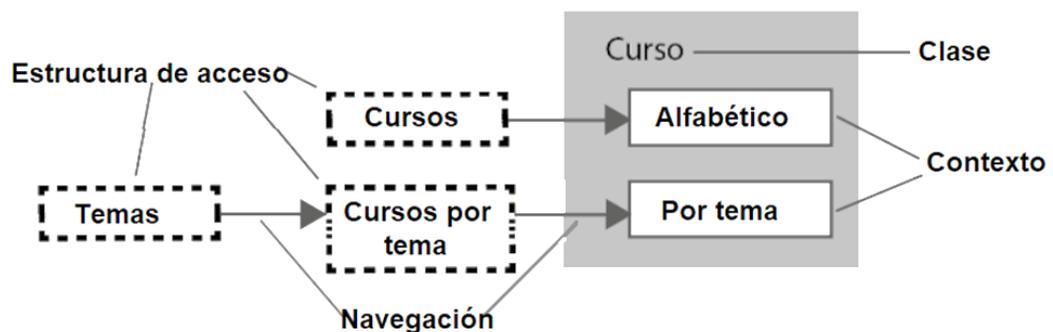
Nota. Datos tomados para la definición y contexto de navegación.

La Figura 5 ilustra un análisis parcial de la navegación, el diagrama de contexto representa tres estructuras de acceso (“Cursos”, “temas”, “curso por tema”) y dos contextos para la clase curso (“alfabético” y por materia).

El subdiagrama compuesto de “temas”, “cursos por tema” y el contexto “curso por tema” representa una posible solución de navegación al UID.

Figura 5

Diagrama de contexto



Nota. Ejemplo de diagrama de contexto. Fuente: Guell, Schawabe, & Patricia

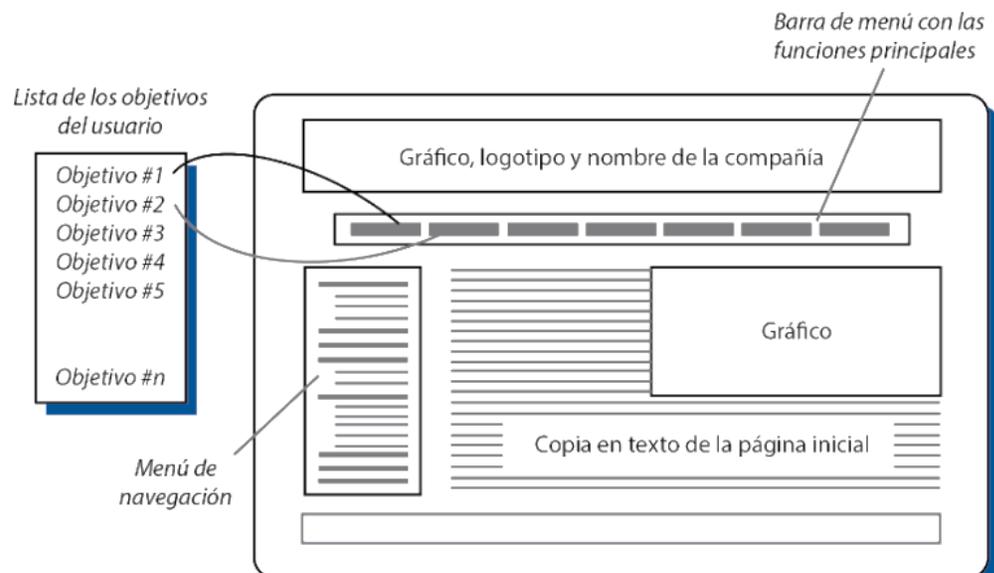
2.2.4. Fase IV. Diseño de Interfaz Abstracta

Una vez definida la estructura navegacional, hay que prepararla para que sea perceptible por el usuario y esto es lo que se intenta en esta fase. Esto consiste en definir qué objetos de interfaz va a percibir el usuario, y en particular el camino en el cuál aparecerán los diferentes objetos de navegación, qué objeto de interfaz actuará en la navegación, la forma de sincronización de los objetos multimedia y el interfaz de transformaciones.

Al haber una clara separación entre la fase anterior y esta fase, para un mismo modelo de navegación se pueden definir diferentes modelos de interfaces, permitiendo, así que el interfaz se ajuste mejor a las necesidades del usuario.

Figura 6

Mapeo de los objetivos



Nota. La figura muestra las acciones de la interfaz del usuario. Fuente: Pressman (2010).

2.2.5. Fase V. Implementación

Una vez obtenido el modelo conceptual, el modelo de navegación y el modelo de interfaz abstracta, sólo queda llevar los objetos a un lenguaje concreto de programación, para obtener así la implementación ejecutable de la aplicación.

Ventajas y Desventajas

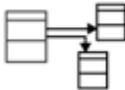
OOHDM es sin duda una de las metodologías que más aceptación ha tenido, y sigue teniendo, en el desarrollo de aplicaciones multimedia. Actualmente está sirviendo como base para el desarrollo de nuevas propuestas metodológicas para los sistemas de información web.

OOHDM es una propuesta basada en el diseño, que ofrece una serie de ideas que han sido asumidas por bastantes propuestas y que han dado muy buenos resultados. La primera de ellas es que hace una separación clara entre lo conceptual, lo navegacional y lo visual. Esta independencia hace que el mantenimiento de la aplicación sea mucho más sencillo.

Además, es la primera propuesta que hace un estudio profundo de los aspectos de interfaz, esencial no solo en las aplicaciones multimedia, sino que es un punto crítico en cualquiera de los sistemas que se desarrollan actualmente.

Tabla 2

Resumen del OOHDM

	 Diseño conceptual	 Diseño de la navegación	 Navegación abstracto de la interfaz	 Implementación
Productos del trabajo	Clases, subsistemas, relaciones, atributos	Vínculos de nodos, estructuras de acceso, controles de navegación, transformación de navegación	Objetos abstractos de la interfaz, respuestas a eventos externos, transformaciones	Webapp ejecutable
Mecanismo	Clasificación, composición, agregación, generalización, especificación	Mapeo entre objetos conceptuales y de navegación	Mapeo entre la navegación y los objetos perceptibles	Recursos proporcionado por el ambiente meta
Preocupaciones del diseño	Semántica de modelado del dominio de la aplicación	Toma en cuenta el perfil del usuario y la tarea. Hace énfasis en aspectos cognitivos	Modelado de los objetos perceptibles, implementación de las metáforas escogidas. Descripción de la interfaz para objetos de navegación	Corrección; desempeño de la aplicación completitud

Nota. Datos obtenidos de la metodología OOHDM. Fuente: Pressman (2010).

OOHDM hace uso también de la orientación a objetos y de un diagrama tan estandarizado como el de clases, para representar el aspecto de la navegación a

través de las clases navegacionales: índices, enlaces y nodos. Esta idea ha dado muy buenos resultados y parece muy adecuada a la hora de trabajar.

Sin embargo, y a pesar de esto, OOHDM presenta algunas deficiencias. OOHDM ha dejado fuera de su ámbito un aspecto esencial que es el tratamiento de la funcionalidad del sistema.

Además, OOHDM no ofrece ningún mecanismo para trabajar con múltiples actores. Por ejemplo, imaginemos que la interfaz y la navegación de la aplicación varía sustancialmente dependiendo de quién se conecte a la aplicación. El diagrama navegacional, los contextos navegacionales y los ADVs resultarían muy complejos para representar esta variabilidad.

2.3. LENGUAJE UNIFICADO DE MODELADO (UML)

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

Figura 7

Logo de UML



Nota. La figura muestra el logo del lenguaje unificado de modelado. Fuente: Ferré, Sánchez.

Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE.

En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

2.3.1. Modelos

Un modelo representa a un sistema software desde una perspectiva específica. Al igual que la planta y el alzado de una figura en dibujo técnico nos muestran la misma figura vista desde distintos ángulos, cada modelo nos permite fijarnos en un aspecto distinto del sistema. Los modelos de UML que se tratan en esta parte son los siguientes:

- Diagrama de Estructura Estática.
- Diagrama de Casos de Uso.
- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Estados.

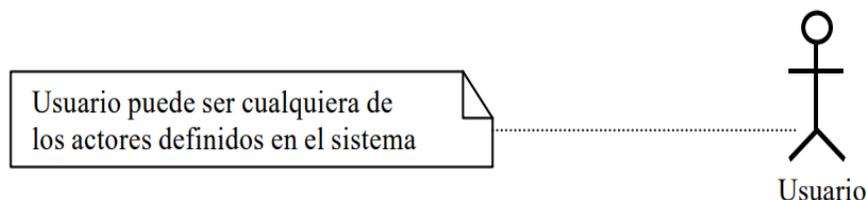
2.3.2. Elementos Comunes a Todos los Diagramas

2.3.2.1. Notas

Una nota sirve para añadir cualquier tipo de comentario a un diagrama o a un elemento de un diagrama. Es un modo de indicar información en un formato libre, cuando la notación del diagrama en cuestión no nos permite expresar dicha información de manera adecuada.

Figura 8

Ejemplo de nota



Nota. En la figura se muestra un ejemplo de cómo añadir comentarios. Fuente: Ferre, Sánchez.

Una nota se representa como un rectángulo con una esquina doblada con texto en su interior. Puede aparecer en un diagrama tanto solo como unido a un elemento por medio de una línea discontinua. Puede contener restricciones, comentarios, el cuerpo de un procedimiento, etc.

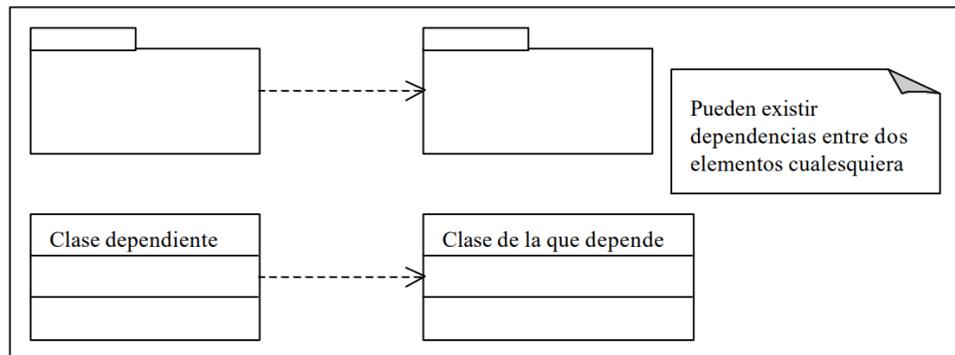
2.3.2.2. Dependencias

La relación de dependencia entre dos elementos de un diagrama significa que un cambio en el elemento destino puede implicar un cambio en el elemento origen (por tanto, si cambia el elemento destino habría que revisar el elemento origen).

Una dependencia se representa por medio de una línea de trazo discontinuo entre los dos elementos con una flecha en su extremo. El elemento dependiente es el origen de la flecha y el elemento del que depende es el destino (junto a él está la flecha).

Figura 9

Dependencias



Nota. En la figura se representa las dependencias entre dos elementos. Fuente: Ferre, Sánchez

2.3.3. Diagramas de Estructura Estática

Con el nombre de Diagramas de Estructura Estática se engloba tanto al Modelo Conceptual de la fase de Análisis como al Diagrama de Clases de la fase de Diseño.

Ambos son distintos conceptualmente, mientras el primero modela elementos del dominio el segundo presenta los elementos de la solución software. Sin embargo,

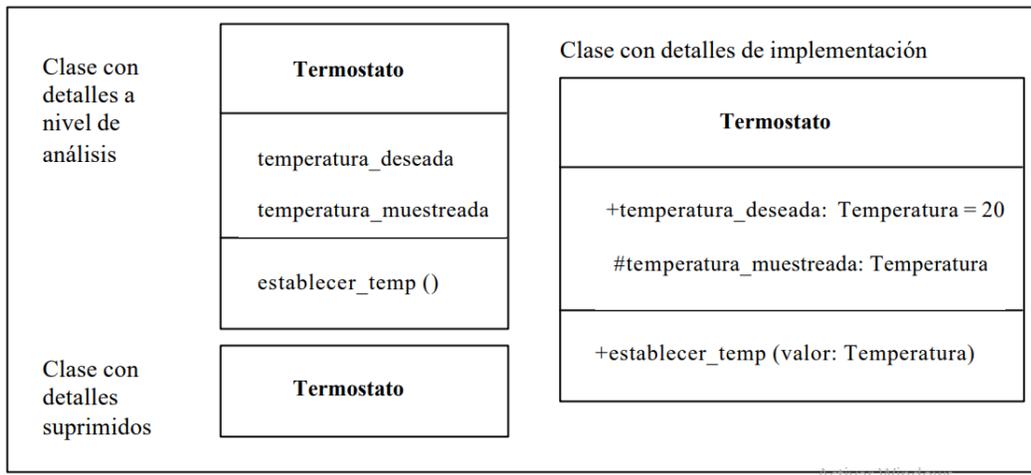
ambos comparten la misma notación para los elementos que los forman (clases y objetos) y las relaciones que existen entre los mismos (asociaciones).

2.3.3.1. Clases

Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Una clase puede representarse de forma esquemática (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase.

Figura 10

Notación para clases



Nota. La figura muestra la notación para clases a distintos niveles de detalle. Fuente: Ferre, Sánchez.

En la Figura se ve cómo una misma clase puede representarse a distinto nivel de detalle según interese, y según la fase en la que se esté.

2.3.3.2. Objetos

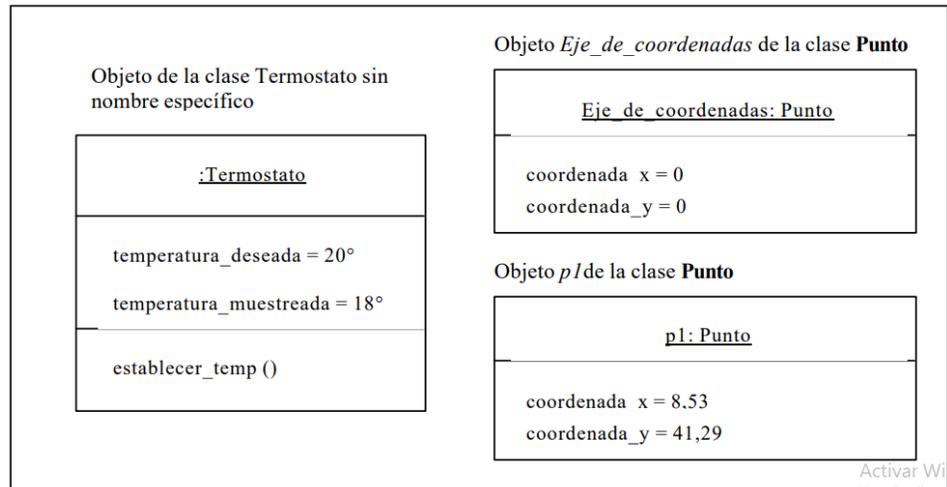
Un objeto se representa de la misma forma que una clase. En el compartimento superior aparece el nombre del objeto junto con el nombre de la clase subrayado, según la siguiente sintaxis:

nombre_del_objeto: nombre_de_la_clase

Puede representarse un objeto sin un nombre específico, entonces sólo aparece el nombre de la clase.

Figura 11

Objetos



Nota. En la figura se muestra el ejemplo de objetos. Fuente: Ferre, Sánchez

2.3.3.3. Asociaciones

Las asociaciones entre dos clases se representan mediante una línea que las une. La línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación. A continuación, se verán los más importantes de entre dichos elementos gráficos.

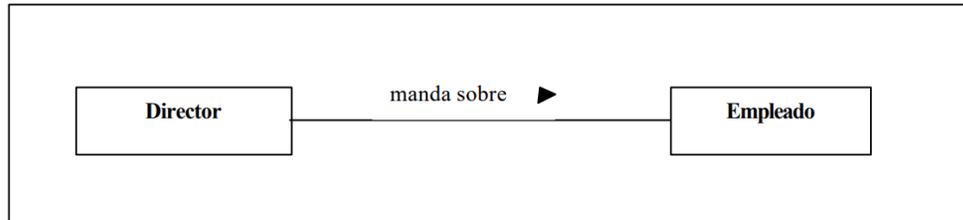
2.3.3.3.1. Nombre de la Asociación y Dirección

El nombre de la asociación es opcional y se muestra como un texto que está próximo a la línea. Se puede añadir un pequeño triángulo negro sólido que indique la dirección en la cual leer el nombre de la asociación.

En el ejemplo, se puede leer la asociación como “Director manda sobre Empleado”.

Figura 12

Asociación



Nota. La figura muestra un ejemplo de asociación con nombre y dirección. Fuente: Ferre, Sánchez

Los nombres de las asociaciones normalmente se incluyen en los modelos para aumentar la legibilidad. Sin embargo, en ocasiones pueden hacer demasiado abundante la información que se presenta, con el consiguiente riesgo de saturación.

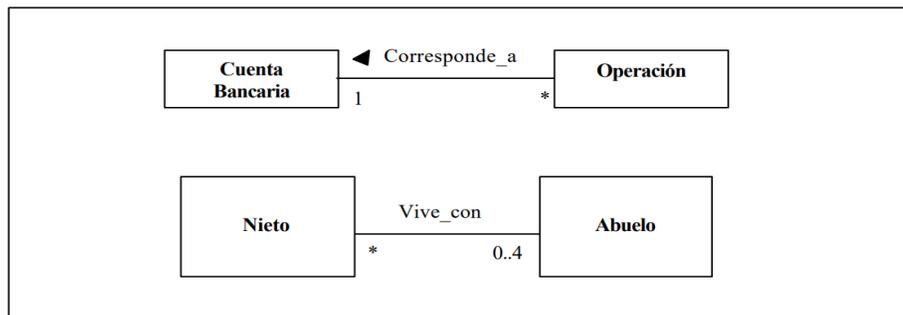
En ese caso se puede suprimir el nombre de las asociaciones consideradas como suficientemente conocidas. En las asociaciones de tipo agregación y de herencia no se suele poner el nombre.

2.3.3.3.2. Multiplicidad

La multiplicidad es una restricción que se pone a una asociación, que limita el número de instancias de una clase que pueden tener esa asociación con una instancia de la otra clase. Puede expresarse de las siguientes formas:

Figura 13

Multiplicidad



Nota. En la figura muestra un ejemplo de multiplicidad en asociaciones. Fuente: Ferre, Sánchez.

La multiplicidad es una restricción que se pone a una asociación, que limita el número de instancias de una clase que pueden tener esa asociación con una instancia de la otra clase. Puede expresarse de las siguientes formas:

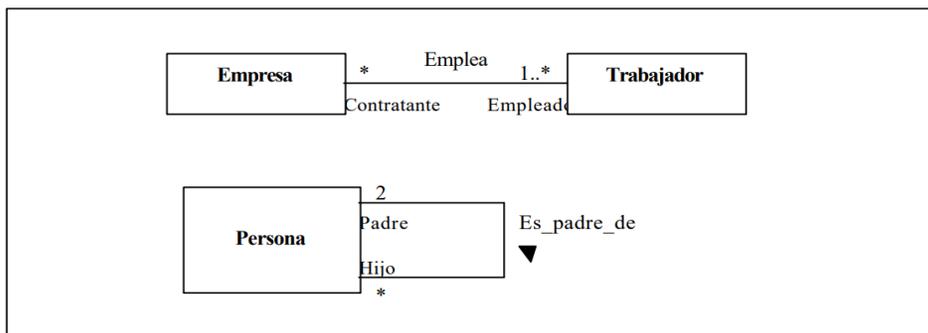
- Con un número fijo: 1.
- Con un intervalo de valores: 2..5.
- Con un rango en el cual uno de los extremos es un asterisco. Significa que es un intervalo abierto. Por ejemplo, 2..* significa 2 o más.
- Con una combinación de elementos como los anteriores separados por comas: 1, 3..5, 7, 15..*.
- Con un asterisco: *. En este caso indica que puede tomar cualquier valor (cero o más).

2.3.3.3. Roles

Para indicar el papel que juega una clase en una asociación se puede especificar un nombre de rol. Se representa en el extremo de la asociación junto a la clase que desempeña dicho rol.

Figura 14

Roles de una asociación

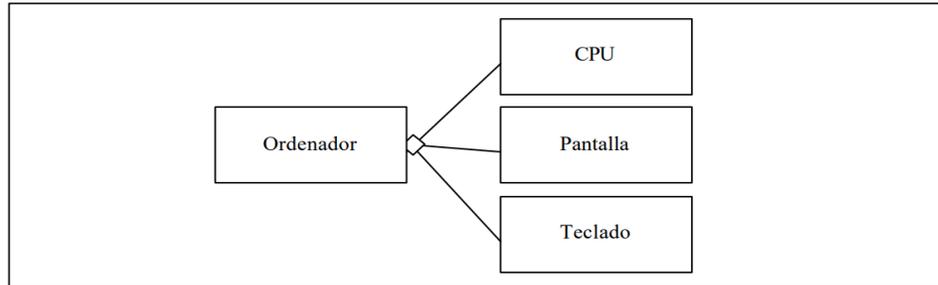


Nota. La figura muestra el ejemplo de roles en una asociación. Fuente: Ferre, Sánchez.

2.3.3.4. Agregación

El símbolo de agregación es un diamante colocado en el extremo en el que está la clase que representa el “todo”.

Figura 15
Agregación



Nota. En la figura se muestra un ejemplo de agregación. Fuente: Ferre, Sánchez.

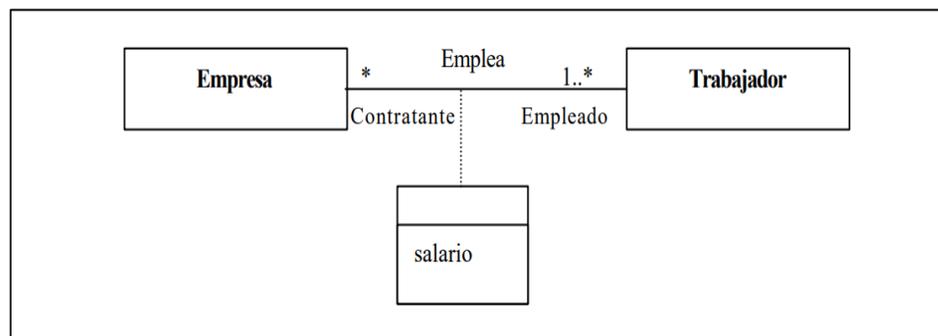
2.3.3.3.5. Clases Asociación

Cuando una asociación tiene propiedades propias se representa como una clase unida a la línea de la asociación por medio de una línea a trazos. Tanto la línea como el rectángulo de clase representan el mismo elemento conceptual: la asociación.

Por tanto, ambos tienen el mismo nombre, el de la asociación. Cuando la clase asociación sólo tiene atributos el nombre suele ponerse sobre la línea (como ocurre en el ejemplo de la Figura 15).

Por el contrario, cuando la clase asociación tiene alguna operación o asociación propia, entonces se pone el nombre en la clase asociación y se puede quitar de la línea.

Figura 16
Clase asociación



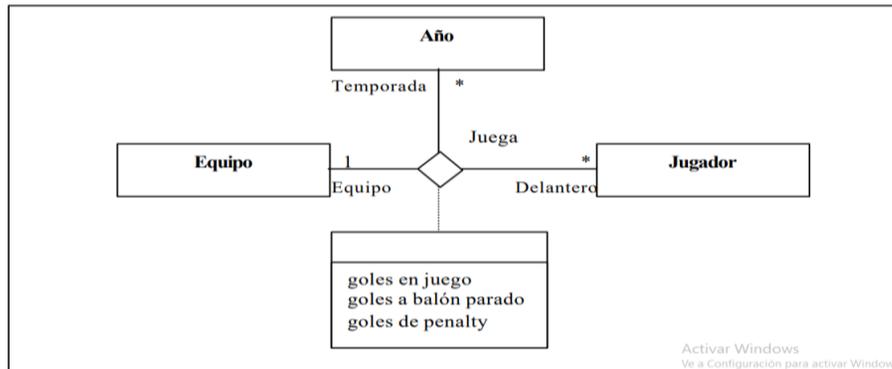
Nota. En la figura se muestra un ejemplo de clase asociación. Fuente: Ferre, Sánchez.

2.3.3.3.6. Asociaciones N-Alias

En el caso de una asociación en la que participan más de dos clases, las clases se unen con una línea a un diamante central. Si se muestra multiplicidad en un rol, representa el número potencial de tuplas de instancias en la asociación cuando el resto de los N-1 valores están fijos.

Figura 17

Asociación ternaria



Nota. En la figura se muestra un ejemplo de asociación ternaria. Fuente: Ferre, Sánchez.

En la Figura se ha impuesto la restricción de que un jugador no puede jugar en dos equipos distintos a lo largo de una temporada, porque la multiplicidad de "Equipo" es 1 en la asociación ternaria.

2.3.3.3.7. Navegabilidad

En un extremo de una asociación se puede indicar la navegabilidad mediante una flecha. Significa que es posible "navegar" desde el objeto de la clase origen hasta el objeto de la clase destino.

Se trata de un concepto de diseño, que indica que un objeto de la clase origen conoce al(los) objeto(s) de la clase destino, y por tanto puede llamar a alguna de sus operaciones.

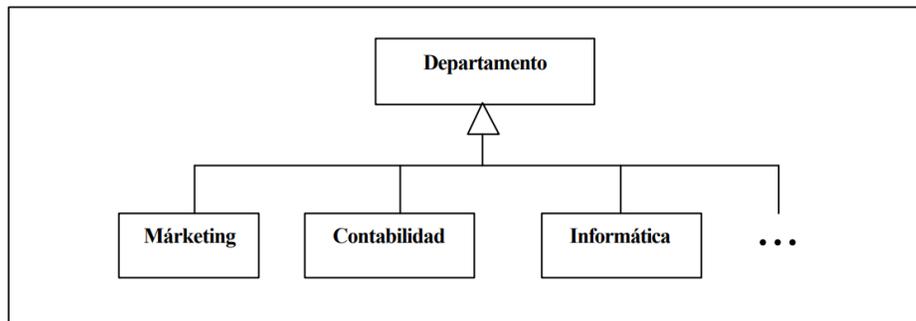
2.3.3.4. Herencia

La relación de herencia se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o clase “padre”.

Si se tiene una relación de herencia con varias clases subordinadas, pero en un diagrama concreto no se quieren poner todas, esto se representa mediante puntos suspensivos.

Figura 18

Herencia



Nota. En la figura se muestra un ejemplo de herencia. Fuente: Ferre, Sánchez.

En el ejemplo de la Figura, sólo aparecen en el diagrama 3 tipos de departamentos, pero con los puntos suspensivos se indica que en el modelo completo (el formado por todos los diagramas) la clase “Departamento” tiene subclases adicionales, como podrían ser “Recursos Humanos” y “Producción”.

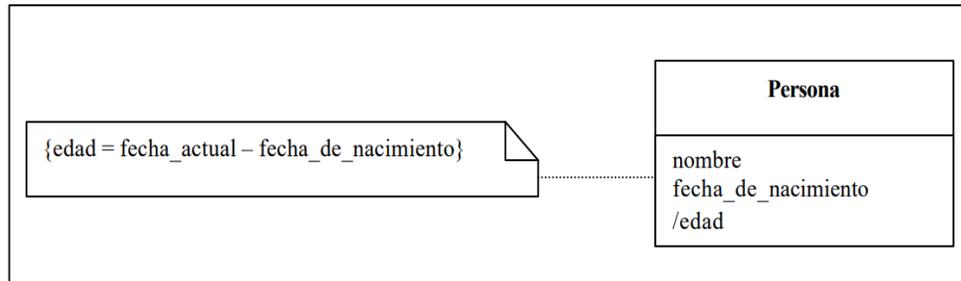
2.3.3.5. Elementos Derivados

Un elemento derivado es aquel cuyo valor se puede calcular a partir de otros elementos presentes en el modelo, pero que se incluye en el modelo por motivos de claridad o como decisión de diseño.

Se representa con una barra “/” precediendo al nombre del elemento derivado.

Figura 19

Atributo derivado



Nota. En la figura se muestra un ejemplo de atributo derivado. Fuente: Ferre, Sánchez.

2.3.4. Diagrama de Casos de Uso

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

2.3.4.1. Elementos

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

2.3.4.1.1. Actores

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).

2.3.4.1.2. Casos de Uso

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

2.3.4.1.3. Relaciones entre Casos de Uso

Entre dos casos de uso puede haber las siguientes relaciones:

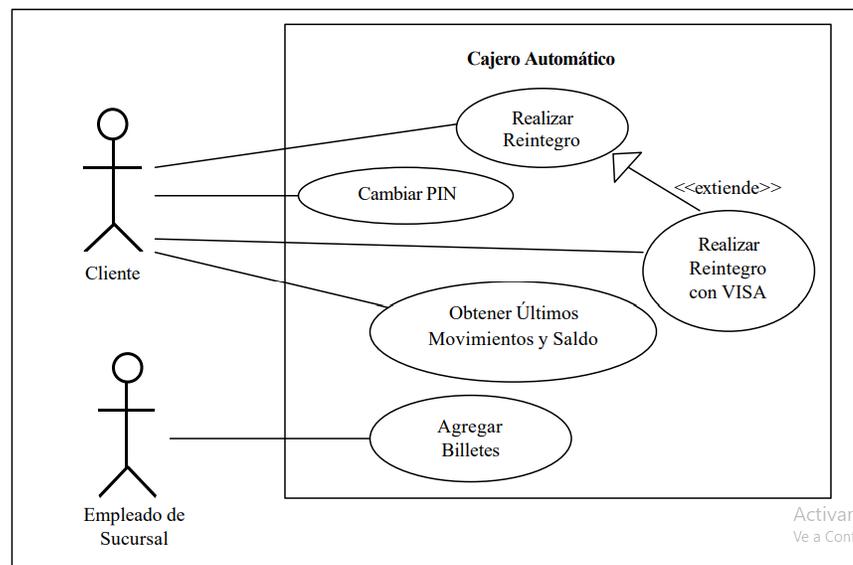
- **Extiende:** Cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- **Usa:** Cuando un caso de uso utiliza a otro.

Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<usa>> según sea el tipo de relación.

En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

Figura 20

Diagrama de Casos de Uso



Nota. En la figura se muestra un ejemplo de Diagrama de Casos de Uso para un cajero automático.

Fuente: Ferre, Sánchez.

2.3.5. Diagramas de Interacción

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular:

- Diagramas de Secuencia
- Diagramas de Colaboración.

2.3.5.1. Diagrama de Secuencia

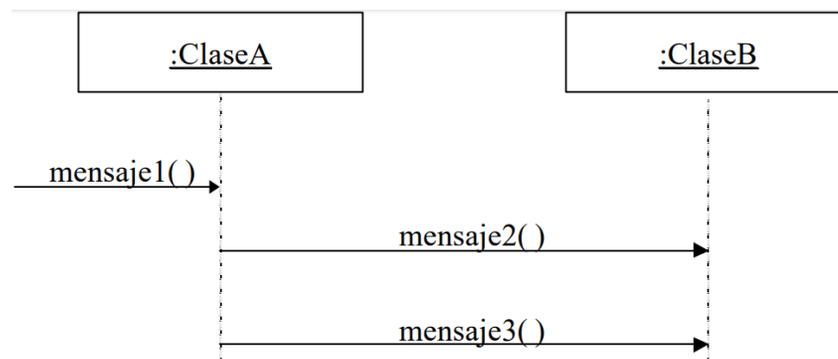
Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo.

Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

Figura 21

Diagrama de Secuencia



Nota. En la figura se muestra la interacción del diagrama de secuencia. Fuente: Ferre, Sánchez.

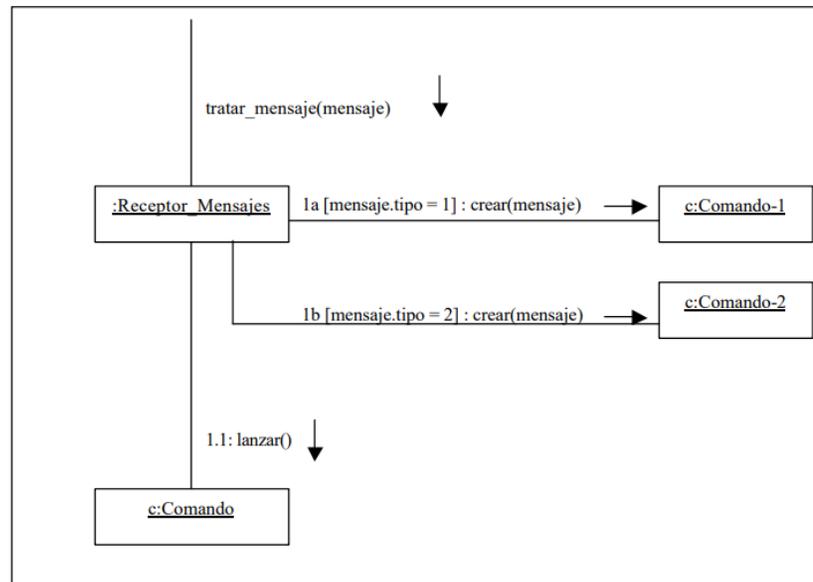
2.3.5.2. Diagrama de Colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere).

A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

Figura 22

Diagrama de Colaboración



Nota. En la figura se muestran las relaciones entre los roles de los objetos del diagrama de colaboración.

Fuente: Ferre, Sánchez.

En cuanto a la representación, un Diagrama de Colaboración muestra a una serie de objetos con los enlaces entre los mismos, y con los mensajes que se intercambian dichos objetos. Los mensajes son flechas que van junto al enlace por el que “circulan”, y con el nombre del mensaje y los parámetros (si los tiene) entre paréntesis.

Cada mensaje lleva un número de secuencia que denota cuál es el mensaje que le precede, excepto el mensaje que inicia el diagrama, que no lleva número de secuencia. Se pueden indicar alternativas con condiciones entre corchetes (por ejemplo 3 [*condición_de_test*] : *nombre_de_método* ()), tal y como aparece en el ejemplo de la Figura 22.

También se puede mostrar el anidamiento de mensajes con números de secuencia como 2.1, que significa que el mensaje con número de secuencia 2 no acaba de ejecutarse hasta que no se han ejecutado todos los 2. x.

2.3.6. Diagrama de Estados

Un Diagrama de Estados muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos. Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino.

La caja de un estado puede tener 1 o 2 compartimentos. En el primer compartimento aparece el nombre del estado. El segundo compartimento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas.

Una acción de entrada aparece en la forma *entrada/acción_asociada* donde *acción_asociada* es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta.

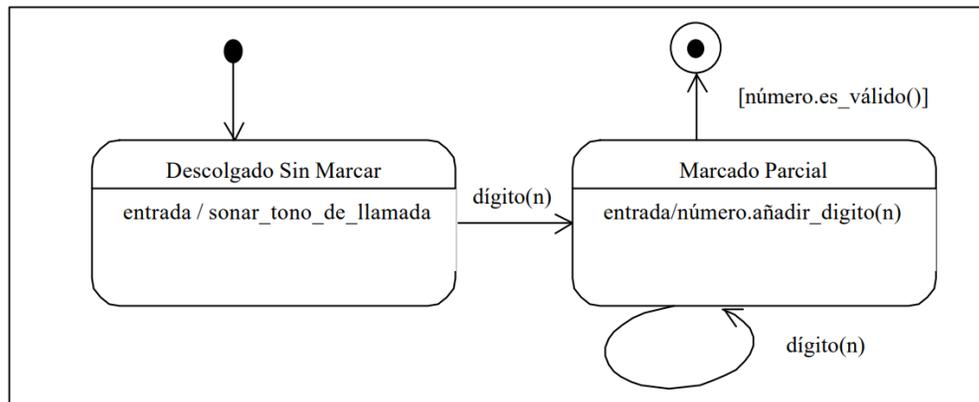
Una acción de salida aparece en la forma *salida/acción_asociada*. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta. Una

acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma:

nombre_de_evento / acción_asociada.

Figura 23

Diagrama de Estados



Nota. En la figura se muestra el diagrama de estado que indica, qué eventos hacen que se pase de un estado a otro Fuente: Ferre, Sánchez.

Un diagrama de estados puede representar ciclos continuos o bien una vida finita, en la que hay un estado inicial de creación y un estado final de destrucción (del caso de uso o del objeto). El estado inicial se muestra como un círculo sólido y el estado final como un círculo sólido rodeado de otro círculo. En realidad, los estados inicial y final son pseudoestados, pues un objeto no se puede incluir en esos estados, pero nos sirven para saber cuáles son la transición inicial y la final(es).

2.4. CALIDAD DE SOFTWARE

La calidad de software en el sentido más general se la define como “Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan”. Pressman (2010).

Un proceso eficaz de software establece la infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. Los aspectos de administración del proceso generan las verificaciones y equilibrios que ayudan evitar que el proyecto caiga en el caos, contribuyente clave de la mala calidad.

Un producto útil entrega contenido, funciones y características que el usuario final desea; sin embargo, de igual importancia es que entrega estos activos en forma confiable y libre de errores.

Al agregar valor para el productor y para el usuario de un producto, el software de alta calidad proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente.

2.4.1. ISO/IEC 9126

ISO 9126 es un estándar internacional para la evaluación del Software. Está supervisado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos. El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente:

- Modelo de calidad
- Métricas externas
- Métricas internas y
- Calidad en las métricas de uso.

Solo la parte primera, modelo de calidad, es un estándar aprobado y publicado siendo el resto de partes de la norma informes que se encuentran en la fase denominada Technical Report (TR).

En la actualidad el uso de las métricas se está poniendo en práctica con éxito en el amplio mercado del software pues las empresas productoras están reconociendo la importancia que tienen las mediciones para cuantificar y por consiguiente gestionar de forma más efectiva la calidad de los procesos y productos de software.

En empresas que se dedican exclusivamente a la informática, se tiene noción de la necesidad de formalizar los mecanismos de estimación, comprendiendo que los registros históricos de antiguos proyectos realizados pueden ayudar a estimar con

mayor exactitud el esfuerzo, tiempo de desarrollo, costo, posibles errores, recursos y tamaño para los nuevos proyectos.

Es válido aclarar que en ocasiones los resultados de los procesos de medición no son interpretados de la mejor manera, pues aún existen compañías que no tienen una cultura adecuada sobre la medición, desconociendo el alcance de madurez y calidad que pudiera alcanzar el producto final.

2.4.1.1. Características de ISO 9126

El modelo establece diez características, seis que son comunes a la vista interna y externa y cuatro que son propias de la vista en uso. Las características que definen las vistas interna y externa, y son:

Figura 24

Características de la ISO/IEC 9126.



Nota. La figura, muestra las características de la Calidad según la ISO/IEC 9126. Fuente: Figueroa, F. (2009).

- **Funcionalidad**, capacidad del software de proveer los servicios necesarios para cumplir con los requisitos funcionales.
- **Fiabilidad**, capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.
- **Usabilidad**, esfuerzo requerido por el usuario para utilizar el producto satisfactoriamente.
- **Eficiencia**, relación entre las prestaciones del software y los requisitos necesarios para su utilización.
- **Mantenibilidad**, esfuerzo necesario para adaptarse a las nuevas especificaciones y requisitos del software.
- **Portabilidad**, capacidad del software ser transferido de un entorno a otro.

Mientras que las características propias de la vista en uso, se muestran a continuación:

Figura 25

Características de la vista en uso



Nota. En la figura se muestra las Características de la vista en uso. Fuente: Figueroa, F. (2009).

- **Efectividad**, capacidad del software de facilitar al usuario alcanzar objetivos con precisión y completitud.
- **Productividad**, capacidad del software de permitir a los usuarios gastar la cantidad apropiada de recursos en relación a la efectividad obtenida.

- **Seguridad**, capacidad del software para cumplir con los niveles de riesgo permitidos tanto para posibles daños físicos como para posibles riesgos de datos.
- **Satisfacción**, capacidad del software de cumplir con las expectativas de los usuarios en un contexto determinado.

ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.

2.5. PRUEBAS DE SOFTWARE

Una prueba debe mostrar un conjunto de características que logren la meta de encontrar la mayor cantidad de errores con el mismo esfuerzo.

2.5.1. Método de la Caja Blanca

Se basa en una visión interna, que usa la estructura de control descrita como parte del diseño a nivel de componentes. Al usar este método, puede derivar casos de prueba que:

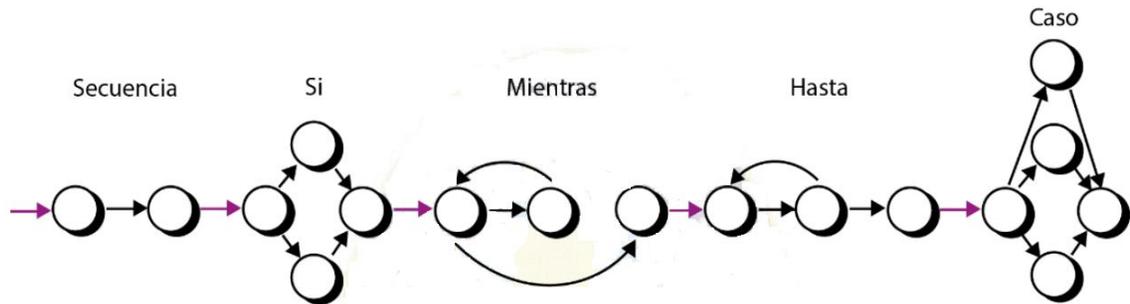
- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras operativas
- Revisen estructuras de datos internas para garantizar su validez.

2.5.1.1. Prueba de ruta básica

Es una técnica que permite derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida como guía para.

Figura 26

Notación de tráfico de flujo



Nota. La figura muestra el flujo de control lógico que se muestra en un conjunto básico de rutas de ejecución. Fuente: Pressman (2010)

El grafico de flujo muestra la Figura 26, donde cada círculo es llamado nodo de gráfico de flujo, representa uno o más enunciados de procedimientos. Las flechas son llamadas aristas o enlaces, representan flujo de control. Una arista debe terminar en un nodo. Las áreas acotadas por aristas y nodos se llaman regiones. A continuación, se presenta una serie de pasos para derivar la ruta básica:

- Dibujar el grafico de flujo a partir del diseño o código.
- Determinar la complejidad ciclomática del grafico del flujo resultante.
- Determinar un conjunto básico de rutas linealmente independientes.
- Preparar casos de prueba para forzar la ejecución de cada ruta en el conjunto básico.

2.5.2. Método de la Caja Negra

Realizan pruebas sobre la interfaz del programa, se entendiendo por interfaz las entradas y salidas de dicho programa.

Las pruebas de caja negra consisten en probar un sistema o programa informático sin tener conocimiento previo de su funcionamiento interno. Esto no sólo se refiere a no conocer el código fuente en sí, sino que implica no haber visto ninguna de las documentaciones de diseño que rodean al software. Los probadores se limitan a dar entrada y recibir salida como lo haría un usuario final. Aunque se trata de una simple definición de prueba de caja negra, establece el sistema general.

El objetivo de las pruebas de caja negra es conseguir que los usuarios interactúen con el software de una forma más natural de lo normal, sin tener ningún prejuicio existente derivado de conocer ya el software.

En esta metodología, los responsables de realizar las pruebas son distintos de los que han desarrollado el software, lo que crea una separación entre ambos equipos. Estas pruebas intentan encontrar errores en las siguientes categorías:

- Funciones incorrecta o faltante.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a bases de datos externos.
- Errores de comportamiento o rendimiento
- Errores de inicialización y terminación

Durante el proceso de desarrollo, es fundamental garantizar que el software funcione como se espera antes de su lanzamiento.

Para hacerlo, debe pasar por procesos de prueba extremadamente exhaustivos durante todo el período de desarrollo, incluido asegurarse de que su producto sea adecuado para el usuario.

Es donde entran en juego las Pruebas de Aceptación del Usuario (UAT). La prueba UAT significa prueba de aceptación del usuario y es el paso final en el proceso de desarrollo de software.

En esta etapa del proceso, se compila un producto final y se envía a una variedad de usuarios y clientes de software del mundo real para recibir comentarios. Esto garantiza que el software pueda manejar escenarios del mundo real dentro de sus especificaciones de diseño iniciales y establece si los clientes están satisfechos o no con el producto que usted crea para ellos.

2.6. MODELO PARA EL ANÁLISIS DE COSTOS

2.6.1. Cocomo II

El modelo COCOMO original, evoluciono hacia un modelo más exhaustivo, llamado COCOMO II, el cual permite realizar estimaciones en función del tamaño de software, y de conjunto de factores de costo y de escala. Los factores de costo incluyen aspectos relacionados con la naturaleza del sistema, equipo, y características propias del proyecto, los factores de escala incluyen la parte de escala producida a medida que un proyecto de software incrementa su tamaño.

Los objetivos que se tomaron en cuenta para construir el modelo COCOMO II fueron:

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

En respuesta a la diversidad del mercado actual y futuro de desarrollo de software, surgen sub modelos, cada uno que ofrece mayor fidelidad

2.6.1.1. Modelo de Diseño Temprano

El modelo se utiliza en las primeras etapas del desarrollo en la cuales se evalúan las alternativas de hardware y software. Al tener poca información en estas etapas, hace que concuerde el uso de Puntos Función, para estimar tamaño y el uso de un número reducido de factores de costo.

2.6.1.2. Puntos de Función no Ajustados

Tabla 3

Puntos función sin ajustar

TIPO DE ELEMENTOS	DIFICULTAD	PESO	CANTIDAD	TOTAL PUNTOS	TOTAL ELEMENTOS
ENTRADAS	Simple	3	E1	Peso * E1	$\sum(\text{Peso} * E_i)$
	Media	4	E2	Peso * E2	
	Compleja	6	E3	Peso * E3	
Total puntos de función entrada:					
SALIDAS	Simple	4	S1	Peso * S1	$\sum(\text{Peso} * S_i)$
	Media	5	S2	Peso * S2	
	Compleja	7	S3	Peso * S3	
Total puntos de función salida:					
CONSULTAS	Simple	3	C1	Peso * C1	$\sum(\text{Peso} * C_i)$
	Media	4	C2	Peso * C2	
	Compleja	6	C3	Peso * C3	
Total puntos de función consulta:					
ARCHIVOS	Simple	7	A1	Peso * A1	$\sum(\text{Peso} * A_i)$
	Media	10	A2	Peso * A2	
	Compleja	15	A3	Peso * A3	
Total puntos de función archivos:					
TOTAL PUNTOS DE FUNCIÓN NO AJUSTADOS					$\sum \sum (\text{Peso} * X_i) j$

Nota. La tabla, estima el tamaño y el uso de un número reducido de factores de costo. Fuente: Pressman (2010).

Dónde:

PFNA son los puntos función no ajustado ($\text{PFNA} = \sum \sum (\text{Peso} * X_i) j$).

2.6.1.3. Puntos de Función Ajustados

El modelo utiliza puntos función ajustado (**PFA**) como base para medir el tamaño, y con calculados en la ecuación siguiente:

$$\text{PFA} = \text{PFNA} * \text{FA}$$

$$\text{FA} = 0.65 + 0.01 * \sum F_i$$

Dónde:

FA es el factor de ajuste.

Fi corresponde a los pesos asignados a los siguientes factores:

Tabla 4

Factores de complejidad

CARACTERÍSTICAS	PESO
Comunicación de datos	F₁
Procesamiento distribuido de datos	F₂
Rendimiento	F₃
Configuraciones fuertemente utilizadas	F₄
Frecuencia de transacciones	F₅
Entrada de datos on-line	F₆
Eficiencia del usuario final	F₇
Actualizaciones Online	F₈
Procesamiento complejo	F₉
Reusabilidad	F₁₀
Facilidad de Instalación	F₁₁
Facilidad de operación	F₁₂
Instalación de distintos lugares	F₁₃
Facilidad de cambio	F₁₄
TOTAL	∑Fi

Nota. Modelo de puntos fusión ajustado. Fuente: Pressman (2010).

Los pesos se consideran dentro de una escala de 0 a 5, donde: 0 es no influencia, 1 es incidental, 2 es moderado, 3 es medio, 4 es significativo y 5 es esencial.

Los puntos función tienen que ser convertidos a líneas de código fuente (**KSLOC**) considerando el lenguaje de implementación.

$$\text{KSLOC} = \frac{\text{PF} * \text{LLP}}{1000}$$

LLP: Líneas de código requeridos por puntos de función en un lenguaje.

Tabla 5*Líneas de código por cada punto de función*

LENGUAJE	LDC/PF
Access	38
Ada 95	49
Al Shell	49
APL	32
Assembly - Basic	320
Basic – ANSI	64
Basic – Compiled	91
Basic – Visual	32
C	128
C++	55
Cobol (ANSI 85)	91
Base de Datos	40
4ta Generación de Lenguaje	20
Nivelar Lenguaje	64
Jovial	107
Machine Code	640
Modula 2	80
PHP	29
PERL	27
Prolog	64
Query – Default	13
Report Generator	80
Second Generation Language	107
Spreadsheet	46
Third Generation Language	80
USR 5	1
Visual Basic 5.0	29

Nota. Escala de líneas de código por cada punto de función. Fuente: Pressman (2010).

2.6.1.4. Estimación del Esfuerzo

El modelo ajusta el esfuerzo nominal usando siete factores de costo. La ecuación para el cálculo del esfuerzo es la siguiente:

$$PM_{ajustado} = PM_{nominal} * \prod E_{Mi}$$

$$PM_{Nominal} = A * SIZE^B$$

$$B = 0.91 + 0.01 * \sum W_i$$

Dónde:

PM_{nominal}, es el esfuerzo nominal ajustado por 7 factores, que reflejan aspectos propios del proyecto.

SIZE, es el tamaño de software expresado en miles de líneas de código fuente (**KSLOC**).

A, es una constante que captura los efectos lineales sobre el esfuerzo (**A = 2.45**).

B, es el factor de ahorro o gasto relativo de escala.

W_i, corresponde a los valores de los siguientes factores de escala.

EM_i Corresponde a los factores de ajuste que tienen un efecto multiplicativo sobre el esfuerzo.

Tabla 6

Factores de escala

FACTOR ES DE ESCALA	MUY ABAJO	BAJO	NOMINAL	ALTO	MUY ALTO	EXTRA ALTO
PREC	Completamente sin precedentes 6.20	Prácticamente sin precedentes 4.96	Casi sin precedentes 3.72	Algo familiar 2.48	Muy familiar 1.24	Completamente Familiar 0.00
FLEX	Riguroso 5.07	Relajación ocasional 4.05	Algo de relajación 3.04	Conformidad general 2.03	Algo de conformidad 1.01	Metas generales 0.00
RESL	Poco (20%) 7.07	Poco (40%) 5.65	A menudo (60%) 2.24	Generalmente (75%) 2.83	En su mayor parte (90%) 1.41	Por completo (100%) 0.00
TEAM	Interacciones muy difíciles 5.48	Algo de dificultad en las interacciones 4.38	Interacciones básicamente cooperativas 3.29	Bastante cooperativo 2.19	Altamente cooperativo 1.10	Completas interacciones 0.00
PMAT	SW-CMM Nivel 1 7.80	SW-CMM Nivel 1 Alto 6.24	SW-CMM Nivel 2 4.68	SW-CMM Nivel 3 3.12	SW-CMM Nivel 4 1.56	SW-CMM Nivel 5 0.00
Peso medio de respuestas "Si" para el cuestionario de Madurez CMM						

Nota. Los factores de escala muestran el peso medio para el cuestionario de madurez. Fuente: Pressman (2010)

Tabla 7

Valores de los factores de escala

FACTORES DE ESCALA	NIVEL	VALOR
PREC		W_1
FLEX		W_2
RESL		W_3
TEAM		W_4
PMAT		W_5
Resultado		$\sum W_i$

Nota. Los factores de escala reflejan el nivel de flexibilidad, amplitud, relación y madurez del proyecto.

Fuente: Pressman (2010)

- **PREC:** Experiencia previa de la organización con este tipo de proyectos.
- **FLEX:** Refleja el grado de flexibilidad en el proceso de desarrollo.
- **RESL:** Refleja la amplitud de análisis de riesgo que se lleva a cabo.
- **TEAM:** Refleja la relación entre los miembros del equipo de desarrollo.
- **PMAT:** Refleja la madurez del proceso de la organización.

Los siguientes puntos describen los factores de ajuste en categorías:

Del producto

- **RCPX:** Fiabilidad y confiabilidad del producto.
- **RUSE:** Requerimiento de reusabilidad.

De la plataforma

- **PDIF:** Dificultad de la plataforma.

Del personal

- **PERS:** Capacidad del personal.
- **PREX:** Experiencia del personal.

Del proyecto

- **FCIL:** Facilidades para el desarrollo.
- **SCED:** Esfuerzo del calendario.

Tabla 8*Factores de ajuste*

DRIVERS	EXTRA ABAJO	MUY BAJO	BAJO	NOMINAL	ALTO	MUY ALTO	EXTRA ALTO
RCPX	0.73	0.81	0.98	1	1.30	1.74	2.38
RUSE	-	-	0.95	1	1.07	1.15	1.24
PDIF	-	-	0.87	1	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1	0.87	0.73	0.62
SCED	-	1.43	1.14	1	1		

Nota. En la tabla se describe los factores de ajuste en categorías. Fuente: Pressman (2010).

Tabla 9*Valores de los factores de ajuste*

DRIVERS	NIVEL	VALOR
RCPX		EM ₁
RUSE		EM ₂
PDIF		EM ₃
PERS		EM ₄
PREX		EM ₅
FCIL		EM ₆
SCED		EM ₇
Resultado		

Nota. Resumen de los factores de ajustes obtenidos. Fuente: Pressman (2010)

2.6.1.5. Estimación del Cronograma

La ecuación inicial es:

$$\text{TDEV} = [3.67 * \text{PM}^{(0.28+0.2*(B-1.01))}] * [\text{SCED}\% / 100]$$

Dónde:

TDEV es el tiempo calendario en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad.

PM es el esfuerzo expresado en meses persona.

B es el factor de escala.

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

Para el número de personas (**NP**) requeridas se utiliza la ecuación siguiente:

$$NP = \frac{PM_{ajustado}}{TDEV}$$

2.6.1.6. Costo Total del Sistema

El costo de desarrollo (**CD**) se calcula con la ecuación siguiente:

$$CD = TDEV * PS * NP$$

Dónde:

PS es el sueldo promedio de un desarrollador de software.

2.7. HERRAMIENTAS DE APLICACIÓN WEB

Las herramientas de desarrollo ayudan al Desarrollo de Sistemas de Información, surgieron para intentar dar solución a los problemas inherentes a los proyectos de generación de aplicaciones informáticas. Se da uso de distintas herramientas, mismos que colaboran con lo que se efectiviza el producto de software. Para esto es necesario mínimamente un lenguaje de programación, un servidor de aplicaciones, un servidor de base de datos, donde se será guardada la información.

2.7.1. Servidor HTTP Apache

El servidor HTTP Apache es un servidor web de código abierto, uso gratuito y multiplataforma, eso quiere decir que cuenta con versiones para todos los sistemas operativos modernos, como ser UNIX, Windows, Macintosh y otros. El servidor Apache se desarrolla dentro del proyecto HTTP server (HTTPD) de la Apache Software Foundation y existe para proporcionar implementaciones robustas y confiables.

Apache es un servidor web que se encarga de almacenar, procesar y servir las páginas web a los usuarios de las mismas. El servidor Apache, no es un servidor físico, sino un software que se ejecuta en un servidor HTTP.

La función de Apache es la de servir de enlace entre el servidor y los navegadores de los usuarios del sitio web, como Firefox, Google Chrome o Safari, mientras intercambia información entre ellos. Este proceso es lo que se conoce como la arquitectura cliente-servidor, que se caracteriza, entre otras cosas, por su centralización y escalabilidad.

2.7.1.1. Apache Maven

Apache Maven es una herramienta de gestión y comprensión de proyectos, así como Apache Ant (ambas en Java) pero tiene diferentes conceptos y funcionalidades, en especial podemos hablar de la base de Maven que el Modelo de Objeto de Proyecto, el modelo organiza toda la información del proyecto en un solo archivo: pom.xml.

Esta es la manera en que Maven realiza la construcción del proyecto, es decir, como dependencias son requeridas, POM se actualiza. El proyecto Maven puede poseer módulos y cada módulo puede tener su respectivo POM sin perder la organización y jerarquía del proyecto principal. También es posible realizar la construcción de informes y documentación con la herramienta.

2.7.1.2. Instalación de Maven

En principio tenemos como requisito previo la instalación de JDK, especialmente en la versión 1.7 o superior. Después de obtener el JDK, independientemente del sistema operativo que esté utilizando, podemos continuar con la instalación de Maven.

2.7.1.3. Apache Tomcat

Apache Tomcat, se compara con el servidor Apache. Apache Tomcat, también conocido como Servidor Tomcat, demuestra ser una opción popular para los desarrolladores web que crean y mantienen sitios web dinámicos y aplicaciones basadas en la plataforma de software Java.

2.7.2. DSpace

DSpace es un software de código abierto para la creación de Repositorios y Bibliotecas Digitales que provee herramientas para la administración de colecciones digitales. Soporta una gran variedad de datos, incluyendo libros, tesis, fotografías, videos, datos de investigación y otras formas de contenido.

DSpace está organizando en comunidades, asignándoles metadatos y permitiendo su difusión en recolectores o agregadores.

DSpace tiene funciones específicas como gestores de la información y la documentación, centrado en aquellas funciones del programa que se encuentran más directamente relacionadas con nuestra profesión (esquema de metadatos, vocabularios controlados, interfaces de consulta, herramientas de difusión, estadísticas, etc.)

2.7.3. Lenguaje de Programación

Un lenguaje de programación consiste en todos los símbolos, caracteres y sintaxis o reglas de uso que permiten a las personas comunicarse con los computadores. Existen varios, algunos se crean para una aplicación especial, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones.

Los lenguajes de programación intentan conservar una similitud con el lenguaje humano, con la finalidad de que sean más naturales a quienes los usan. Establecen un conjunto de reglas sintácticas y semánticas, las cuales rigen la estructura del programa de computación que se escribe o edita. De esta forma, permiten a los

programadores o desarrolladores, poder especificar de forma precisa los datos sobre los que se va a actuar, su almacenamiento, transmisión y demás acciones a realizar bajo las distintas circunstancias consideradas (Pressman, 2010).

2.7.3.1. HTML

El HTML es el núcleo de cada página web, independientemente de la complejidad de un sitio o la cantidad de tecnologías involucradas. Es el punto de partida para cualquier persona que esté aprendiendo a crear contenido web. Y, por suerte para nosotros, es sorprendentemente fácil de aprender.

HTML significa lenguaje de marcado de hipertexto. "Lenguaje de marcado" significa que, en lugar de usar un lenguaje de programación para ordenar funciones, el HTML utiliza etiquetas para identificar diferentes tipos de contenido y los fines de cada uno para la página web.

El HTML usa etiquetas o elementos para identificar los distintos tipos de contenido. Estas etiquetas tienen nombres bastante intuitivos: etiquetas de encabezado, etiquetas de párrafo, etiquetas de imagen, etc. Cada página web se compone de un montón de estas etiquetas HTML que identifican cada tipo de contenido en la página.

2.7.3.2. JavaScript

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web.

Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable "click-to-show" (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript

Todos los lenguajes de programación funcionan mediante la traducción de sintaxis similar a la del inglés a código de máquina, que posteriormente el sistema operativo se encarga de ejecutar. JavaScript se clasifica principalmente como un lenguaje de scripting o interpretado.

El código JavaScript es interpretado y directamente traducido a código de lenguaje de máquina subyacente mediante un motor de JavaScript. En el caso de otros lenguajes de programación, un compilador se encarga de compilar todo el código en código de máquina en un paso diferente. En consecuencia, todos los lenguajes de scripts son lenguajes de programación, pero no todos los lenguajes de programación son lenguajes de scripts.

2.7.3.3. CSS

CSS no son más que hojas de estilo en cascada. Este lenguaje de programación determina cómo deben aparecer los elementos HTML de un sitio web en la parte frontal de la página.

2.7.4. Gestor de base de Datos

Un sistema gestor de base de datos (SGBD), consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos, la colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa, el objetivo principal de un SGDB es proporcionar una forma de almacenar y recuperar información de una base de datos de manera que sea practica como eficiente.

Los sistemas de base de datos se diseñan para gestionar grandes cantidades de información. La gestión de datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismo para la manipulación de la información.

Además, los sistemas de base de datos tienen que proporcionar fiabilidad de la información almacenada a pasar a las caídas del sistema o los intentos de acceder sin

autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos. Sudarshan (2002).

2.7.4.1. PostgreSQL

PostgreSQL es un sistema de bases de datos de código abierto que se destaca por su capacidad de adaptación y su cumplimiento con el estándar SQL. Actualmente se considera uno de los motores de base de datos más avanzados en la industria.

El proyecto es gratuito y libre, al ser creado por la comunidad de desarrolladores PGDG, además ofrece una gran cantidad de opciones avanzadas para gestionar bases de datos relacionales en programación. Esta solución tiene una característica llamada control de concurrencia multiversión (MVCC), que proporciona una imagen del estado de la base de datos a cada transacción.

Sus posibilidades brindan notables ventajas en términos de rendimiento y manejo de datos, pues al aprovechar el MVCC, PostgreSQL permite realizar transacciones de manera eficiente, optimizando el rendimiento general del sistema y mejorando la experiencia del usuario.

2.7.4.2. Funciones de PostgreSQL

PostgreSQL es una poderosa herramienta para gestionar grandes volúmenes de datos y la aplican sectores como gobierno, industria, educación y comercio. Estas son sus funciones principales:

- **Almacenamiento eficiente de datos:** organiza y almacena datos garantizando su integridad y consistencia.
- **Consultas avanzadas:** permite realizar consultas complejas y analizar datos para obtener información valiosa con lenguaje SQL.
- **Backend confiable para aplicaciones:** empleado principalmente en el desarrollo de aplicaciones web y móviles, pues ofrece un almacenamiento seguro y escalable para los datos de la aplicación.

- **Soporte geoespacial:** cuenta con funciones avanzadas para manejar datos geoespaciales, siendo ideal para apps que requieren análisis y visualización de mapas.

2.7.4.3. Ventajas de PostgreSQL

Adicionalmente, este software ofrece una serie de ventajas que lo convierten en una elección popular para muchos proyectos y organizaciones. exploremos algunas ventajas destacadas de PostgreSQL.

- **Escalabilidad y rendimiento**

PostgreSQL demuestra su capacidad para manejar grandes volúmenes de datos y cargas de trabajo intensivas. Su arquitectura sólida y su enfoque en la concurrencia y optimización de consultas garantizan un rendimiento óptimo incluso en entornos con alta demanda.

- **Flexibilidad y extensibilidad**

Una de las principales fortalezas de PostgreSQL es su capacidad para adaptarse a diferentes necesidades y requisitos. Permite la creación de funciones personalizadas en varios lenguajes de programación, lo que brinda a los desarrolladores la posibilidad de ampliar su funcionalidad según sus necesidades específicas.

- **Soporte de estándares**

PostgreSQL se adhiere a los estándares de SQL y ofrece un amplio conjunto de características que cumplen con las especificaciones del lenguaje. Esto garantiza la portabilidad de las aplicaciones y facilita la migración desde otros sistemas de gestión de bases de datos.

- **Seguridad y confiabilidad**

Como último punto, prioriza la seguridad de los datos almacenados al ofrecer medidas sólidas, como autenticación avanzada, encriptación de datos, control

de acceso a nivel de columna y capacidades de auditoría. Adicionalmente, este motor ha ganado una reputación como un sistema confiable y estable, utilizado por organizaciones y empresas en todo el mundo.

2.7.5. Patrón de Arquitectura Modelo Vista Controlador (MVC)

El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del workflow de la aplicación).

De esta forma, dividimos el sistema en tres capas donde, como explicaremos más adelante, tenemos la encapsulación de los datos, la interfaz o vista por otro y por último la lógica interna o controlador. MVC (2016).

El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:

Modelo

- Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
- Encapsula el estado de la aplicación.
- No sabe nada / independiente del Controlador y la Vista.

Vista

- Es la presentación del Modelo.
- Puede acceder al Modelo, pero nunca cambiar su estado.
- Puede ser notificada cuando hay un cambio de estado en el Modelo.

Controlador

- Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente.

Para entender cómo funciona nuestro patrón Modelo vista controlador, se debe entender la división a través del conjunto de estos tres elementos y como estos componentes se comunican unos con los otros y con otras vistas y controladores externos al modelo principal. Para ello, es importante saber que el controlador interpreta las entradas del usuario (tanto teclado como el ratón), enviado el mensaje de acción al modelo y a la vista para que se proceda con los cambios que se consideren adecuados. Burbeck (1992).

CAPITULO III

3. MARCO APLICATIVO

En este capítulo se realizará el análisis de las fases de la Metodología OOHDM que son: obtención de requerimientos, diseño conceptual, diseño navegacional, diseño de interfaz abstracta e implementación; nos presentan diversos esquemas en un proceso iterativo e incremental dando apoyo al modelado de la aplicación.

En la implementación es necesario estar en constante comunicación con los usuarios que participan en el uso y administración del sistema, con el fin de identificar las tareas, escenarios, casos de uso, diagramas de interacción de usuario.

Se continua el desarrollo de la metodología utilizando modelos de UML, diagramas de contexto en el diseño navegacional, vista de datos abstracto en las interfaces y finalmente se desarrolla una interfaz que muestra los datos que maneja la base de datos, para pasar a utilizar las herramientas de programación e implementar el sistema académico.

3.1. DESARROLLO DE LA METODOLOGÍA OOHDM

3.1.1. Obtención de Requerimientos

3.1.1.1. Identificación de Roles y Tareas

La siguiente descripción muestra los usuarios y las tareas definidas para el sistema.

Rol: Administrador General (Director General).

Tareas:

- El administrador general accede al sitio web.
- El administrador general introduce su identificación de usuario, y luego ingresa al sistema.

- El administrador general registra al administrador (Secretaria) del sistema y puede borrar y editar la información registrada.

Rol: Administrador (Biblioteca).

Tareas:

- El administrador evalúa la documentación de trabajo de grado de cada carrera.
- El administrador accede al sitio web.
- El administrador introduce su identificación de usuario, y luego ingresa al sistema.
- El administrador sube documento de trabajo de grado al sistema.
- El administrador publica el documento de trabajo de grado.

3.1.2. Especificación de Escenarios

Los siguientes escenarios muestran textualmente como será utilizado el sistema con respecto a los roles y tareas ya descritos anteriormente.

Tabla 10

Definición y especificación de escenarios

ESCENARIO	TAREAS Y/O FUNCIONES EN EL SISTEMA
Identificación de usuario	El Administrador General , tiene un nombre de usuario y contraseña para ingresar al sistema académico, estos datos serán generados automáticamente por el sistema al realizar su registro.
Registro de Administrador	El Administrador es registrado en el sistema para su identificación, por el Administrador General a previa entrega de sus documentos personales. El sistema genera automáticamente el nombre de usuario y contraseña.
Evaluación de Trabajo de Grado	El siguiente paso es verificar, evaluar los trabajos de grado según la modalidad de cada carrera.
Registro de Trabajo de Grado	El siguiente paso después de evaluar el Trabajo de Grado, es cargar y registrar los trabajos de grado en el sistema.
Publicación de Trabajo de Grado	El Administrado, dentro del sistema puede publicar los Trabajos de Grado de cada carrera: Electrónica y Electricidad, Electromecánica, Mecánica Automotriz, Transformación de Alimentos y Gastronomía.

Administración de Administrador	El Administrador dentro del sistema, tendrá la opción de modificar su contraseña y será encargado de administrar la publicación o la eliminación del Trabajo de Grado.
Administración de Administrador General	El Administrador General dentro del sistema, tendrá la opción de modificar su contraseña, habilitar y eliminar usuarios y contraseñas, Además tendrá el acceso a la administración de Trabajos de Grado.

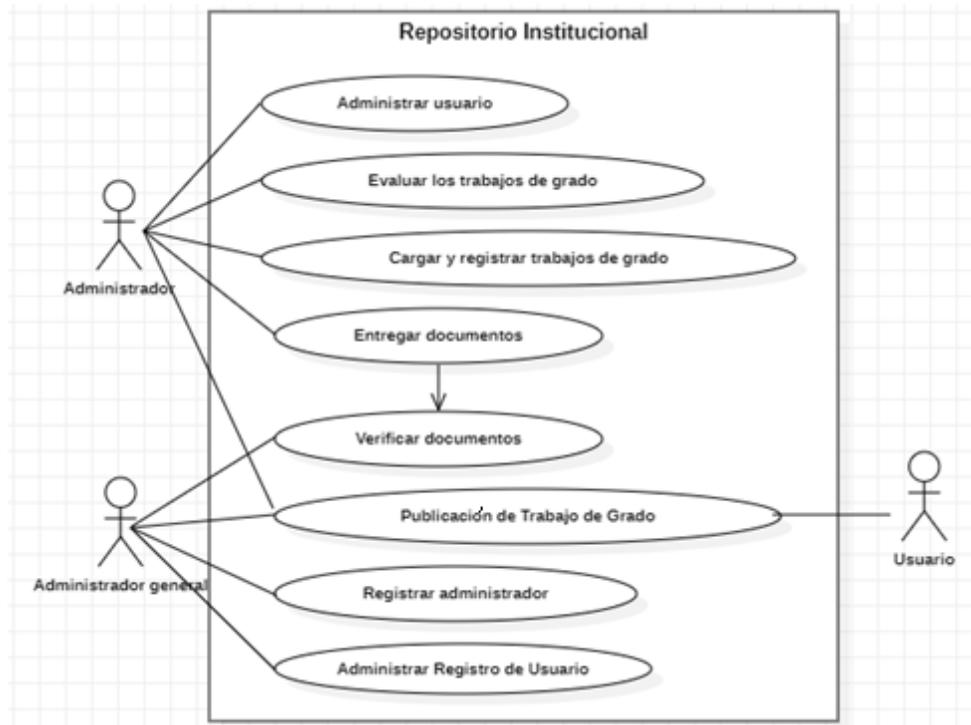
Nota. En la siguiente tabla se describe los roles según el escenario existentes dentro del sistema.

3.1.3. Especificación de Casos de Uso

En esta sección se presentan los casos de uso aplicando UML. En la Figura se muestra el caso de uso general del sistema académico, donde se identifican a los usuarios (administradores) y las acciones con respecto al sistema.

Figura 27

Caso de uso administración



Nota. En la figura se muestra el caso de uso de los roles que cumple el repositorio institucional. Fuente: Rumbaugh, Jacobson & Booch (2007).

En los siguientes casos de uso dividimos el caso de uso general, para mostrar las tareas específicas con respecto a cada administrador. Además, entendemos que

las tareas realizadas por el administrador (biblioteca), también pueden ser hechas por un administrador general (director general), la única diferencia está en el registro que uno hace del otro.

Figura 28

Caso de uso registro de administrador



Nota. En la figura muestra al administrador general y la interacción con el sistema, respecto al registro de los administradores. Fuente: Rumbaugh, Jacobson & Booch. (2007).

Tabla 11

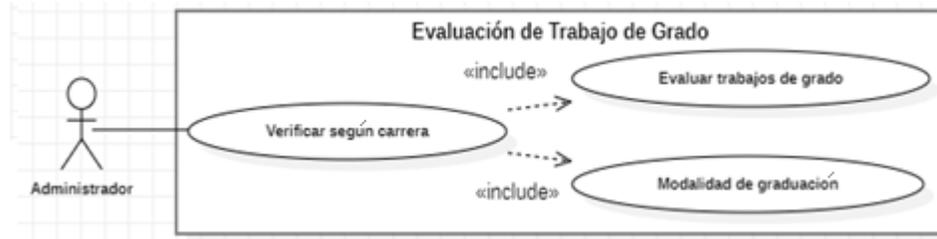
Caso de uso registro de administrador

Caso de uso	Registro de administrador
Actores	Administrador general y administrador
Propósito	Registrar a nuevos administradores para administrar el sitio web y el sistema académico.
Tipo	Primario
Resumen	El administrador (biblioteca) entrega sus documentos y el administrador general (director general) verifica los datos para registrarlo en el sistema.
Precondiciones	No estar registrado en el sistema académico.
Postcondición	El administrador es registrado y el sistema le proporciona un nombre de usuario y una contraseña de administrador.

Nota. En la tabla siguiente hacemos una descripción detallada del caso de uso registro de administrador. Fuente: Larman (2003).

Figura 29

Caso de uso evaluación de trabajo de grado



Nota. La Figura muestra la evaluación del trabajo de grado antes de ser publicado. Fuente. Rumbaugh, Jacobson, & Booch, 2007.

Tabla 12

Caso de uso evaluación de trabajo de grado

Caso de uso	Evaluación de Trabajo de Grado
Actores	Administrador
Propósito	Verificar, revisar y evaluar el trabajo de grado por carreras.
Tipo	Primario
Resumen	El administrador (biblioteca) revisa los documentos y lo selecciona según la carrera a la que pertenece.
Precondiciones	El trabajo de grado no debe presentar copias ni plagio de otros documentos.
Postcondición	El administrador es responsable de verificar, cuidar y resguardar los documentos de trabajo de grado.

Nota. La tabla siguiente presenta una descripción detallada del caso de uso evaluación de trabajo de grado. Fuente: Larman (2003).

Figura 30

Caso de uso registro de trabajo de grado



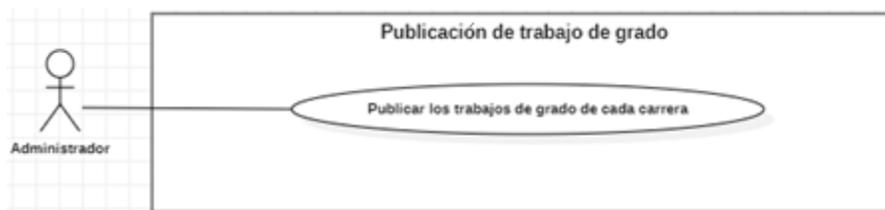
Nota. La Figura muestra el registro de trabajo de grado que el administrador realiza. Fuente: Rumbaugh, Jacobson & Booch (2007).

Tabla 13*Caso de uso registro de trabajo de grado*

Caso de uso	Registro de Trabajo de Grado
Actores	Administrador
Propósito	Cargar y registrar los trabajos de grado en el sistema.
Tipo	Primario
Resumen	El administrador (biblioteca) carga en el sistema los documentos de trabajo de grado seleccionados según la carrera a la que pertenece.
Precondiciones	Haber sido revisado y evaluado por el administrador (biblioteca).
Postcondición	Será publicado.

Nota. La siguiente tabla presenta una descripción detallada del caso de uso registro de trabajo de grado.

Fuente: Larman (2003).

Figura 31*Caso de uso publicación de trabajo de grado.*

Nota. La Figura muestra al administrador que realiza la publicación de diferentes registros de trabajos de grado. Fuente: Rumbaugh, Jacobson & Booch (2007)

Tabla 14*Caso de uso publicación de trabajo de grado*

Caso de uso	Publicación de Trabajo de Grado
Actores	Administrador
Propósito	Publicar en el sitio web del sistema académico los trabajos de grado registrados.
Tipo	Primario
Resumen	El administrador (biblioteca) publica los trabajos de grado de cada carrera.
Precondiciones	Tener el documento registrado en la base de datos.

Flujo normal	Los trabajos de grado son publicados para los usuarios investigadores.
--------------	--

Nota. La siguiente tabla describe la publicación de trabajos de grado. Fuente. Larman (2003).

Figura 32

Caso de uso Administración del administrador



Nota. La Figura, muestra las diferentes acciones que tiene el administrador con respecto al sistema. Fuente: Rumbaugh, Jacobson & Booch (2007).

Tabla 15

Caso de uso administración de administrador

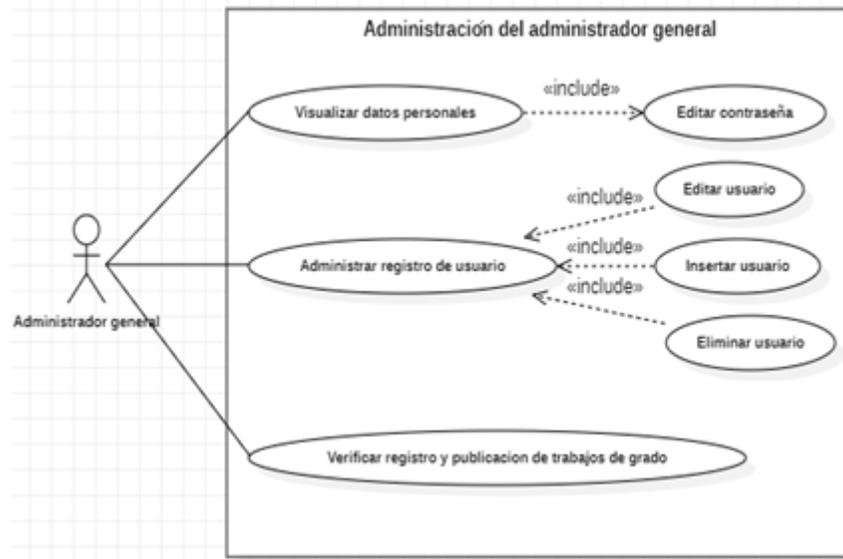
Caso de uso	Administración del administrador
Actores	Administrador
Propósito	Administrar la información.
Tipo	Opcional.
Resumen	El administrador (biblioteca) visualiza sus datos personales y puede modificar su contraseña. El administrador puede administrar los registros de los trabajos de grado publicar y eliminar.
Precondiciones	Estar registrado en el sistema como administrador y tener un nombre de usuario y contraseña.
Flujo normal	El sistema guarda los registros de trabajo de grado.

Nota. La siguiente tabla muestra las acciones que toma el administrador del administrador del sistema.

Fuente: Larman (2003).

Figura 33

Caso de uso Administración del administrador general



Nota. La Figura muestra las diferentes acciones que tiene el administrador general con respecto al sistema. Fuente: Rumbaugh, Jacobson & Booch (2007).

Tabla 16

Caso de uso administración de administrador general

Caso de uso	Administración del administrador general
Actores	Administrador general
Propósito	Administrar la información.
Tipo	Primario
Resumen	El administrador general visualiza sus datos personales y puede modificar su contraseña. El administrador general puede administrar los registros de usuario, insertar, editar y eliminar. El administrador general verifica los registros y publicaciones realizadas.
Precondiciones	Estar registrado en el sistema como administrador general.
Flujo normal	El sistema guarda los registros.

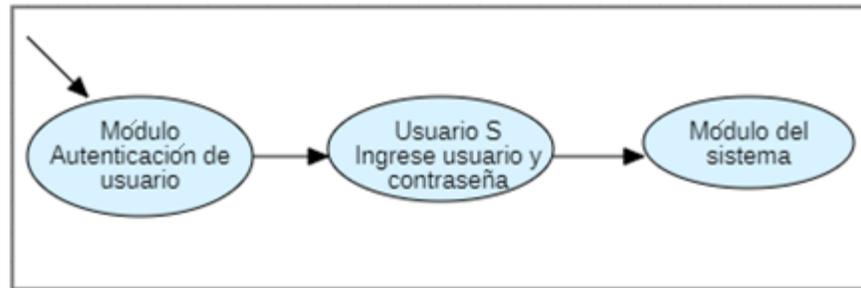
Nota. La siguiente tabla describe las acciones que toma el administrador general en el sistema. Fuente: Larman (2003).

3.1.4. Especificación de UIDS (User Integration Diagram)

Si bien la especificación de casos de uso se la realiza con UML e historias de usuario, OOADM propone utilizar los diagramas de interacción de usuarios (UIDs) para mostrar el intercambio de información entre los usuarios (administrador general, administrador y estudiantes) y el sistema académico.

Figura 34

UID Autenticación de usuario

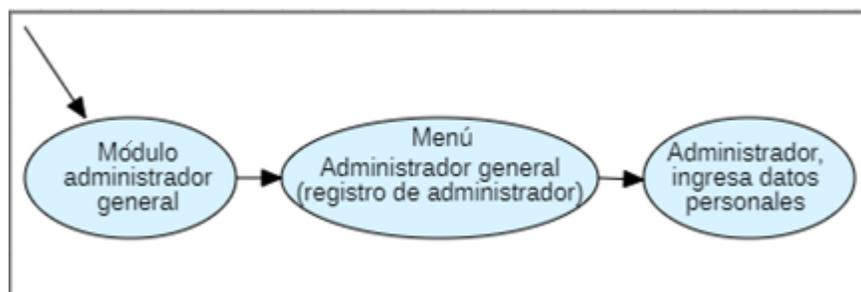


Nota. El UID de la figura, presenta en la primera interacción de autenticación de usuario. Fuente: Guell, Schawabe & Patricia.

El módulo autenticación de usuario, y la siguiente interacción presenta el ingreso de los datos de Autenticación del Usuario S, y la última interacción muestra los módulos del sistema.

Figura 35

UID Registro de administrador

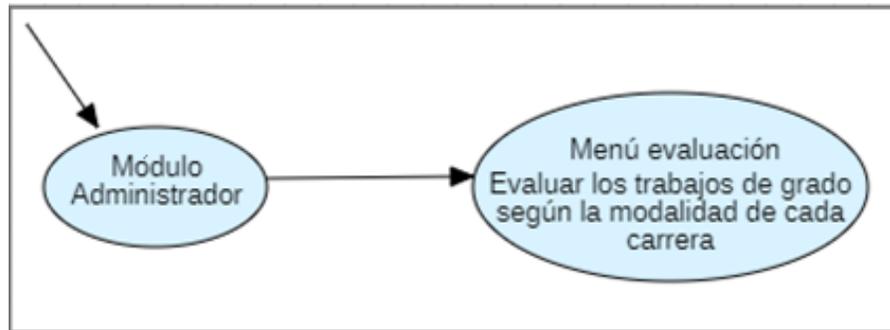


Nota. El UID de la Figura, corresponde al caso de uso registro de administrador (ver figura 30). Fuente: Guell, Schawabe & Patricia.

La primera interacción presenta el Módulo de administrador general, la segunda interacción presenta el Menú Administrador general, y la tercera interacción presenta el objeto administrador que contiene un conjunto de atributos.

Figura 36

UID Evaluación de trabajo de grado



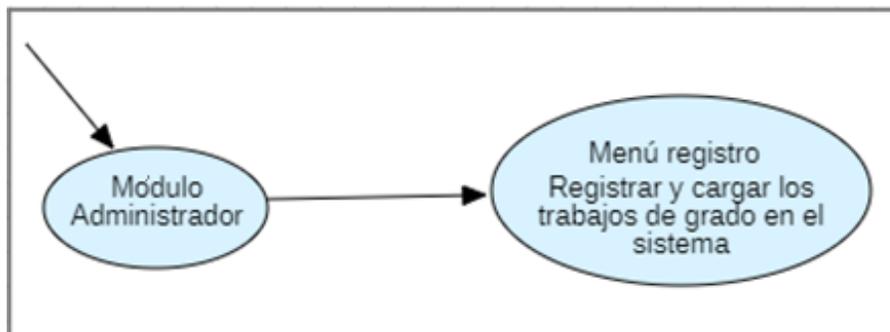
Nota. El UID de la Figura, corresponde al caso de uso evaluación de trabajo de grado (ver Figura 31).

Fuente: Guell, Schawabe & Patricia.

La primera interacción presenta el módulo administrador y la siguiente presenta el Menú evaluación, de trabajos de grado según la modalidad de cada carrera.

Figura 37

UID Registro de trabajo de grado



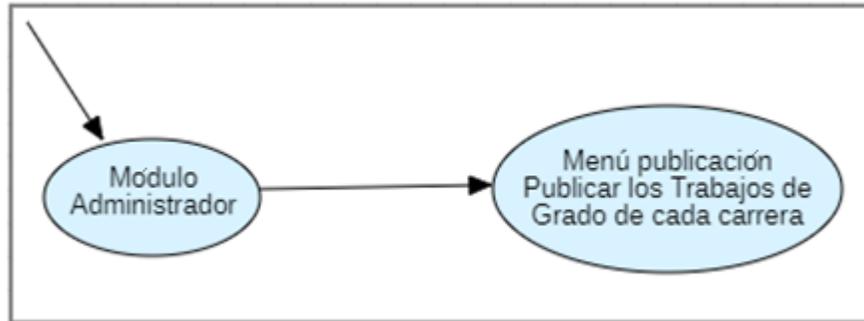
Nota. El UID de la Figura corresponde al caso de uso registro de trabajo de grado (ver Figura 32).

Fuente: Guell, Schawabe & Patricia.

La primera interacción presenta el módulo administrador y la siguiente presenta el menú registro, registra y carga los trabajos de grado en el sistema.

Figura 38

UID Publicación de trabajo de grado



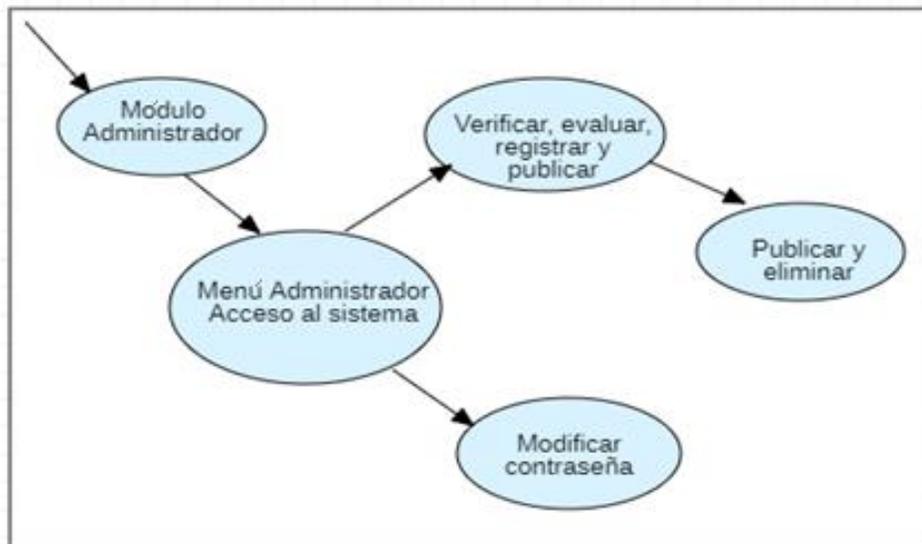
Nota. El UID de la Figura, corresponde al caso de uso publicación de trabajo de grado (ver Figura 33).

Fuente: Guell, Schawabe & Patricia.

La primera interacción presenta el Módulo administrador y la siguiente presenta el Menú publicación, publicar los trabajos de grado de cada carrera.

Figura 39

UID Administración de administrador



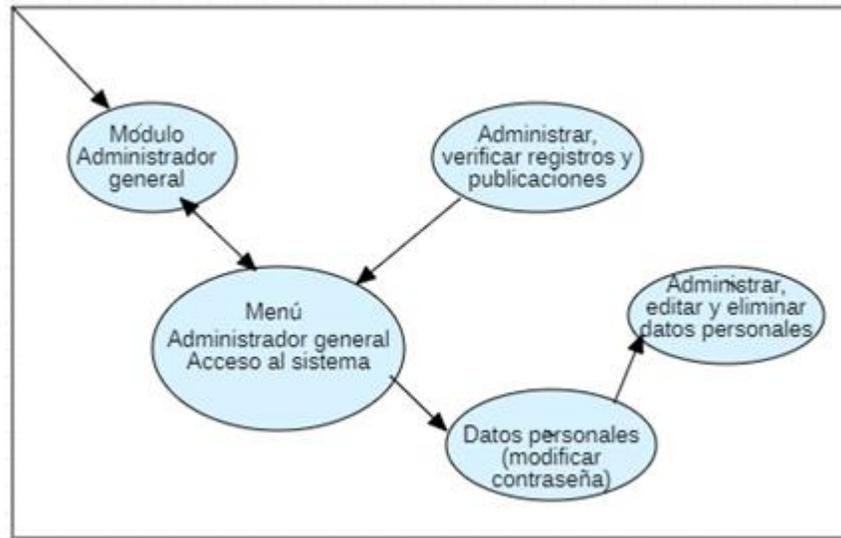
Nota. El UID de la Figura, corresponde al caso de uso administración de administrador (ver Figura 34).

Fuente: Guell, Schawabe & Patricia.

La primera interacción presenta el módulo administrador y la siguiente es el menú administrador, a este último le siguen dos interacciones, la primera es modificar contraseñas y la segunda verificar y publicar seguido de la interacción publicar y eliminar.

Figura 40

UID Administración de administrador general



Nota. El UID de la Figura, corresponde al caso de uso administración de administrador general (ver Figura 35). **Fuente:** Guell, Schawabe & Patricia.

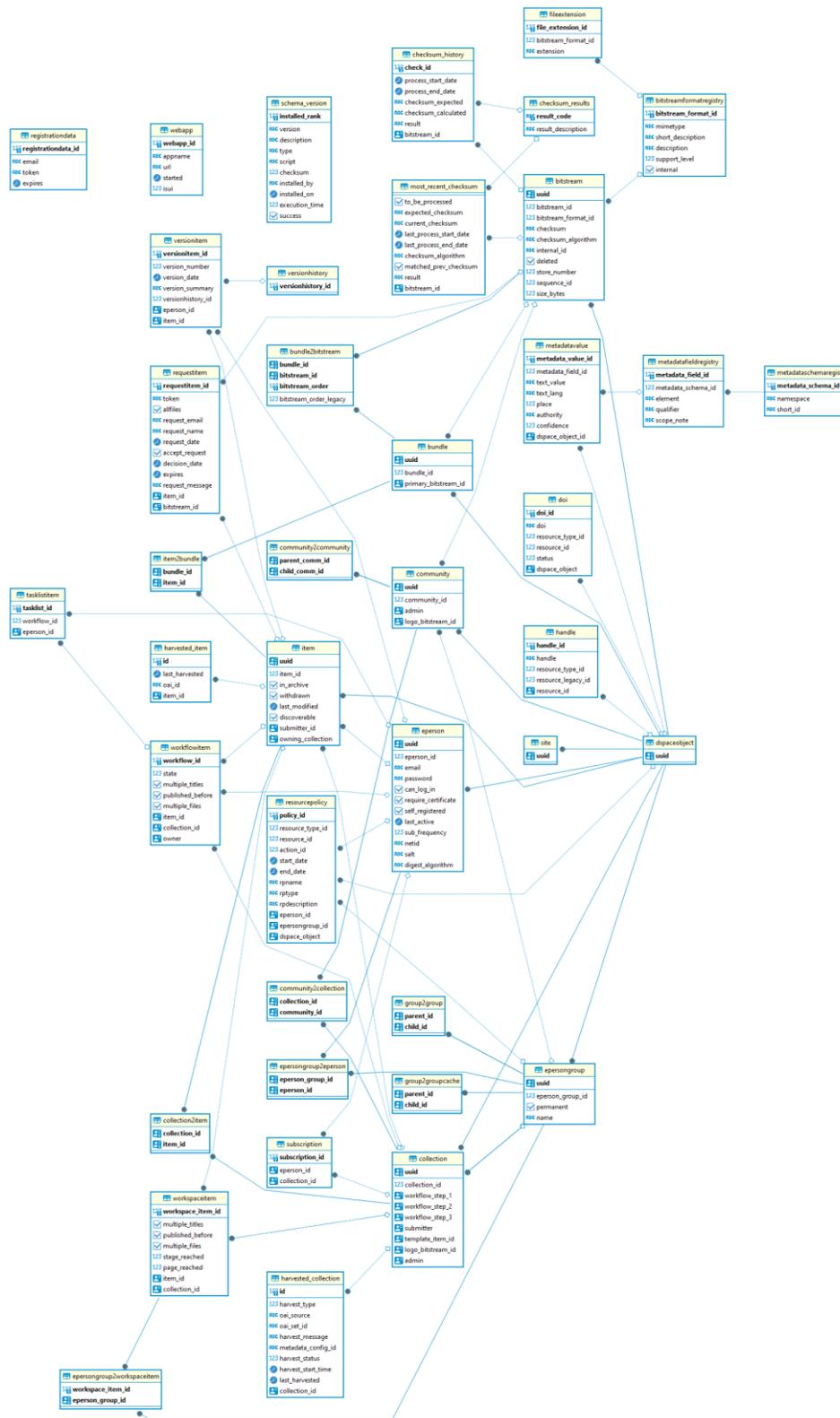
La primera interacción presenta el Módulo administrador y la siguiente es el menú administrador, a este último le siguen dos interacciones, la primera son los datos personales y la segunda el menú usuario que tiene una interacción Usuario X.

3.2. DISEÑO CONCEPTUAL

En esta parte se presentan los objetos que se manipularán en el sistema, las operaciones (métodos) que se aplicarán a los objetos, las relaciones (algunas de ellas jerárquicas) entre los objetos. Una de las principales diferencias con UML que plantea el modelo, es el uso de instrucciones de manera explícita en las relaciones.

Figura 41

Diagrama de clases del Repositorio Institucional



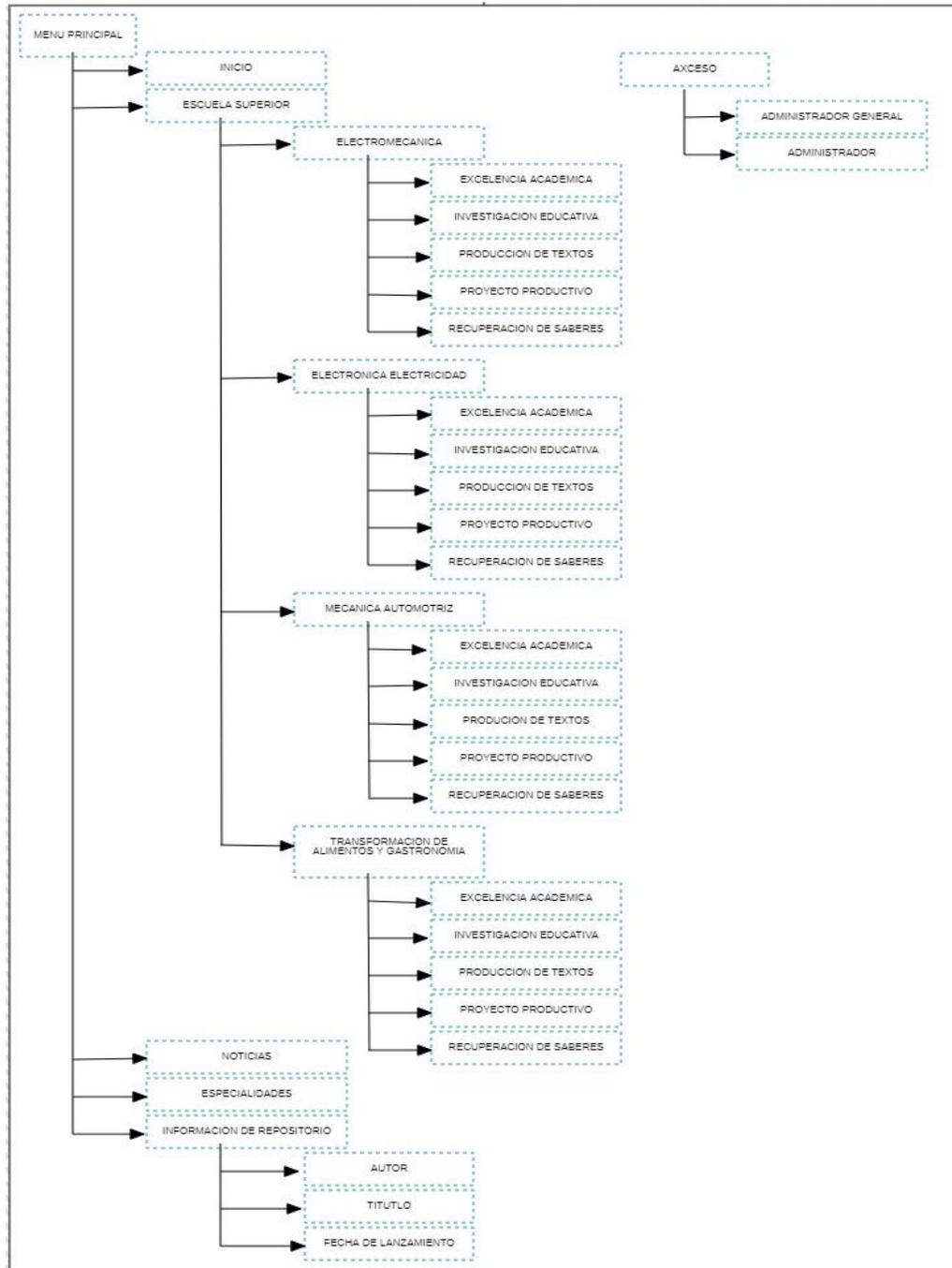
Nota. La siguiente figura muestra el diagrama de clases del Repositorio Institucional.

3.3. DISEÑO DE NAVEGACIÓN

El diseño de navegación, muestra todas las estructuras de acceso que la componen, siendo así el menú, sidebar y los accesos para usuarios.

Figura 42

Diseño de navegación de la página principal del sistema

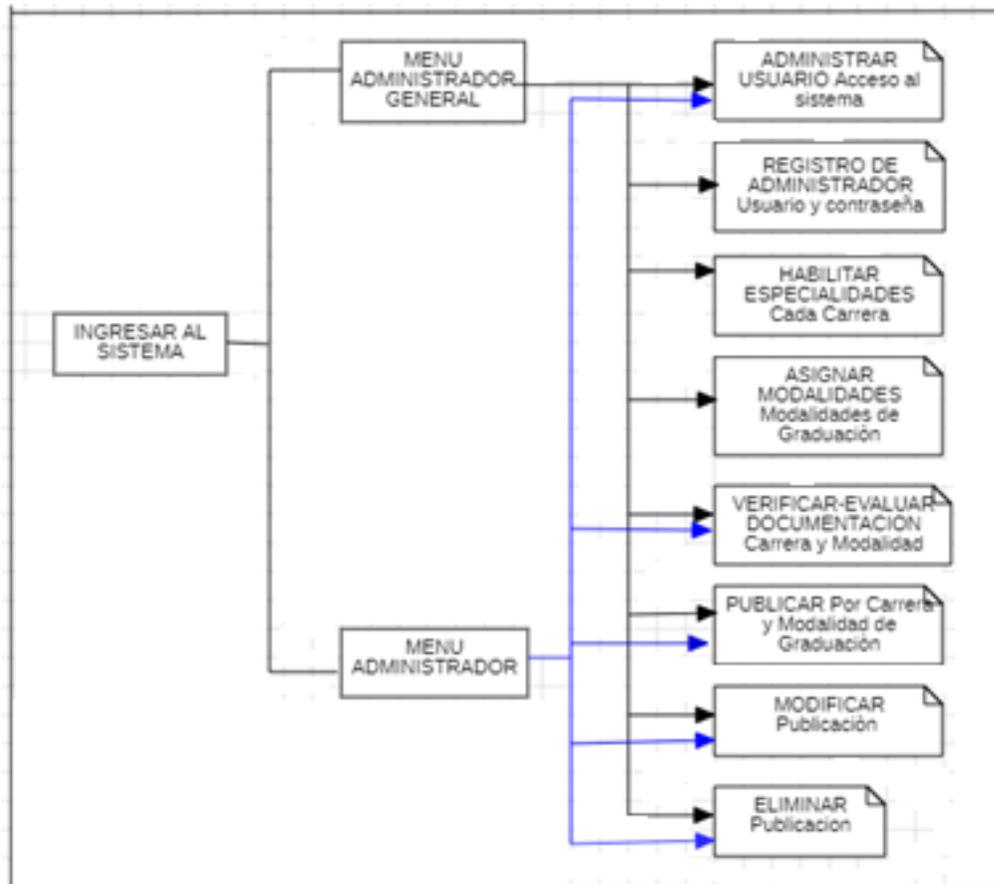


Nota. La figura, representa el diseño de navegación de la página principal del sitio web.

El siguiente diagrama, muestra dos estructuras de acceso “Ingresar al sistema”, “Menú administrador general”, “Menú administrador”, y 8 contextos enlazados a las estructuras.

Figura 43

Diseño de navegación del sistema académico



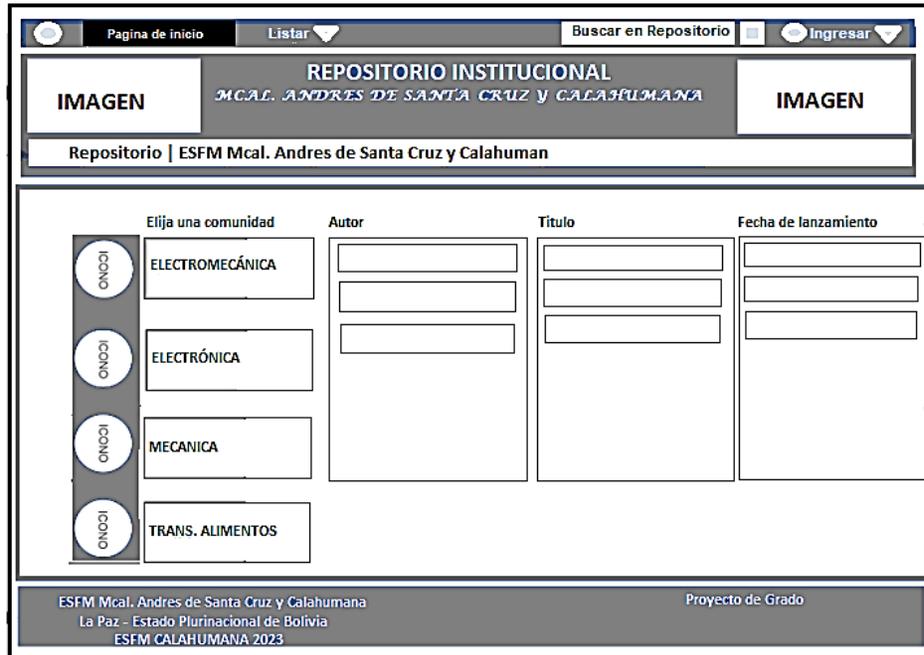
Nota. En la figura, se presenta el diseño de navegación del sistema de repositorio.

3.4. DISEÑO DE INTERFAZ ABSTRACTA

Para representar la interfaz del usuario, se utiliza el modelo de Vista de datos Abstracta (ADVs), donde se especifica la organización de todos los elementos que compone el sistema. Esto significa definir la manera en que diferentes objetos de navegación aparecerán, qué objetos de navegación de la interfaz se activara y otra funcionalidad de aplicación, y qué transformaciones de la interfaz tendrán lugar y cuando.

Figura 44

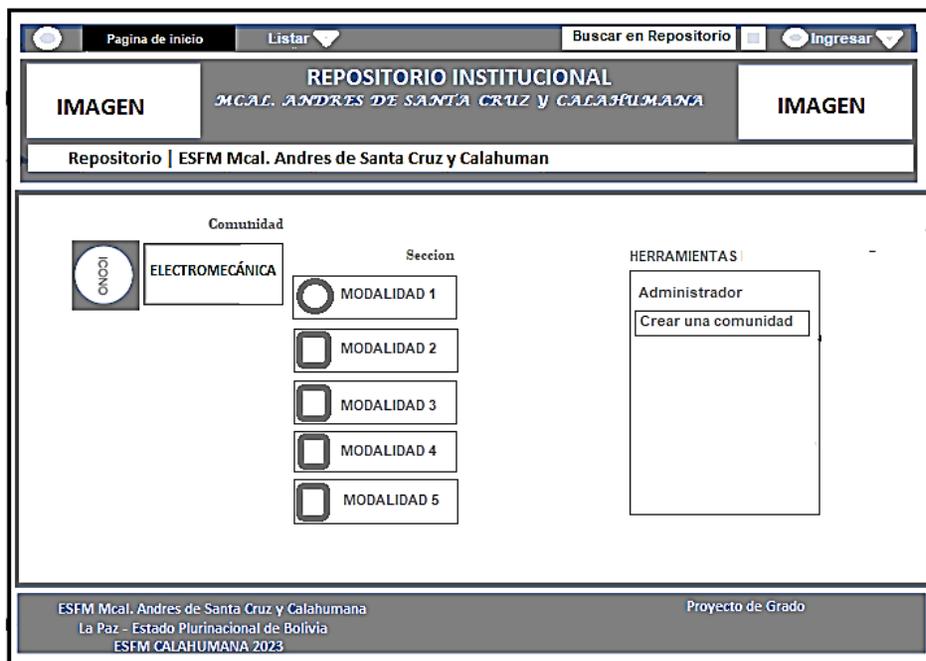
Página principal del sistema



Nota. La Figura, muestra la página principal del sistema que contiene un encabezado, presentación de imágenes, buscador de información, publicación de noticias y por debajo el pie de página.

Figura 45

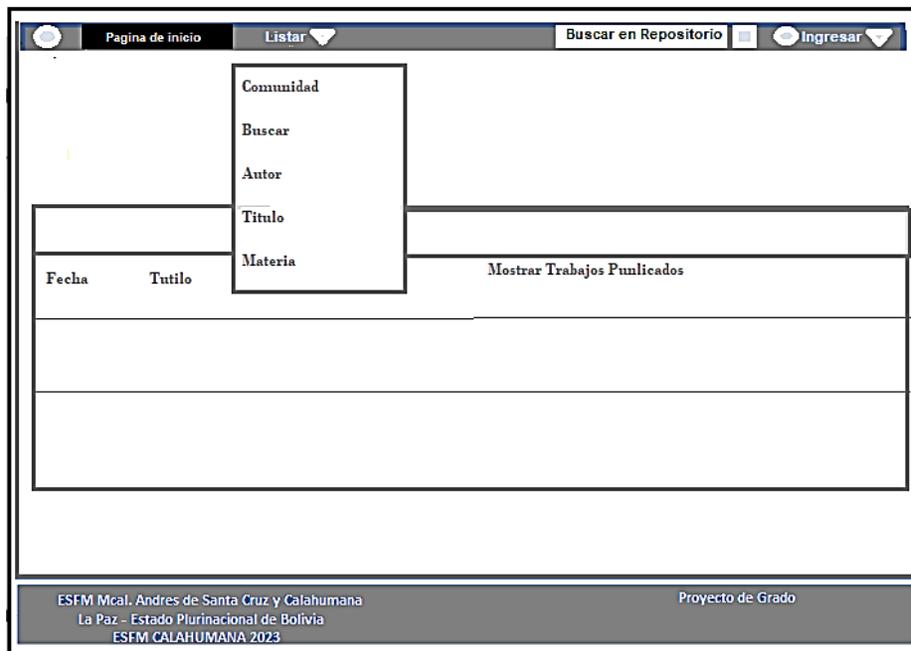
Página de modalidades de graduación



Nota. La figura muestra la carrera y las modalidades de graduación que oferta la institución.

Figura 46

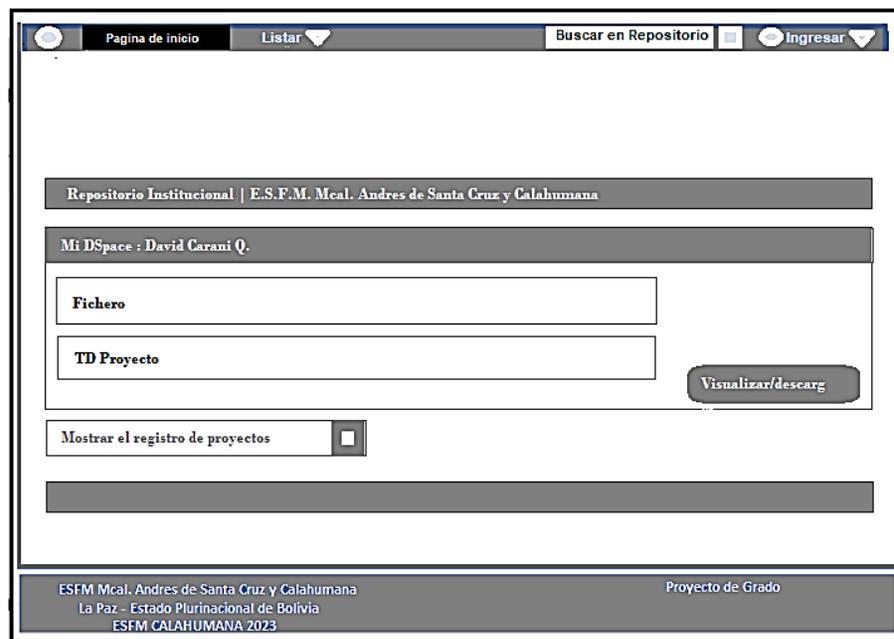
Búsqueda de información



Nota. La figura, muestra la búsqueda de información que se realiza en el sistema; mediante autor, titulo, materia, o mostrar todas más modalidades de graduación.

Figura 47

Visualizar y descargar archivos



Nota. La figura, muestra la forma en la que podremos descargar los archivos del repositorio.

Figura 48

Autenticación y acceso de administrador

The screenshot shows the administrator login interface. At the top, there is a navigation bar with 'Pagina de inicio', 'Listar', 'Buscar en Repositorio', and 'Ingresar'. Below this is a header section with 'IMAGEN' on the left and right, and 'REPOSITORIO INSTITUCIONAL' in the center, with the subtitle 'M.CAL. ANDRES DE SANTA CRUZ Y CALAHUMANA'. A sub-header reads 'Repositorio | ESFM Mcal. Andres de Santa Cruz y Calahuman'. The main content area is titled 'DSpace' and contains the instruction 'Introduzca su direccion de correo electronica,'. It features two input fields: 'Direccion de correo electronico' and 'Password', followed by an 'Enviar' button. The footer contains the text 'ESFM Mcal. Andres de Santa Cruz y Calahumana', 'La Paz - Estado Plurinacional de Bolivia', 'ESFM CALAHUMANA 2023', and 'Proyecto de Grado'.

Nota. La figura, muestra la autenticación para el ingreso al sistema en modo administrador.

Figura 49

Envío y publicación de documentos

The screenshot shows the administrator dashboard. At the top, there is a navigation bar with 'Pagina de inicio', 'Listar', 'Buscar en Repositorio', and 'Ingresar'. Below this is a header section with 'Repositorio Institucional | E.S.F.M. Mcal. Andres de Santa Cruz y Calahumana'. The main content area is titled 'Mi DSpace : David Carani Q.' and contains two buttons: 'Realizar Nuevo Envio' and 'Ver los copumentos publicados'. The footer contains the text 'ESFM Mcal. Andres de Santa Cruz y Calahumana', 'La Paz - Estado Plurinacional de Bolivia', 'ESFM CALAHUMANA 2023', and 'Proyecto de Grado'.

Nota. La figura, muestra la forma de publicar los documentos digitales en el repositorio institucional.

3.5. IMPLEMENTACIÓN

En la implementación del sistema, se muestra las interfaces del sitio web desarrollado con las herramientas ya mencionadas; a continuación, se muestra el sistema en dos distintos soportes.

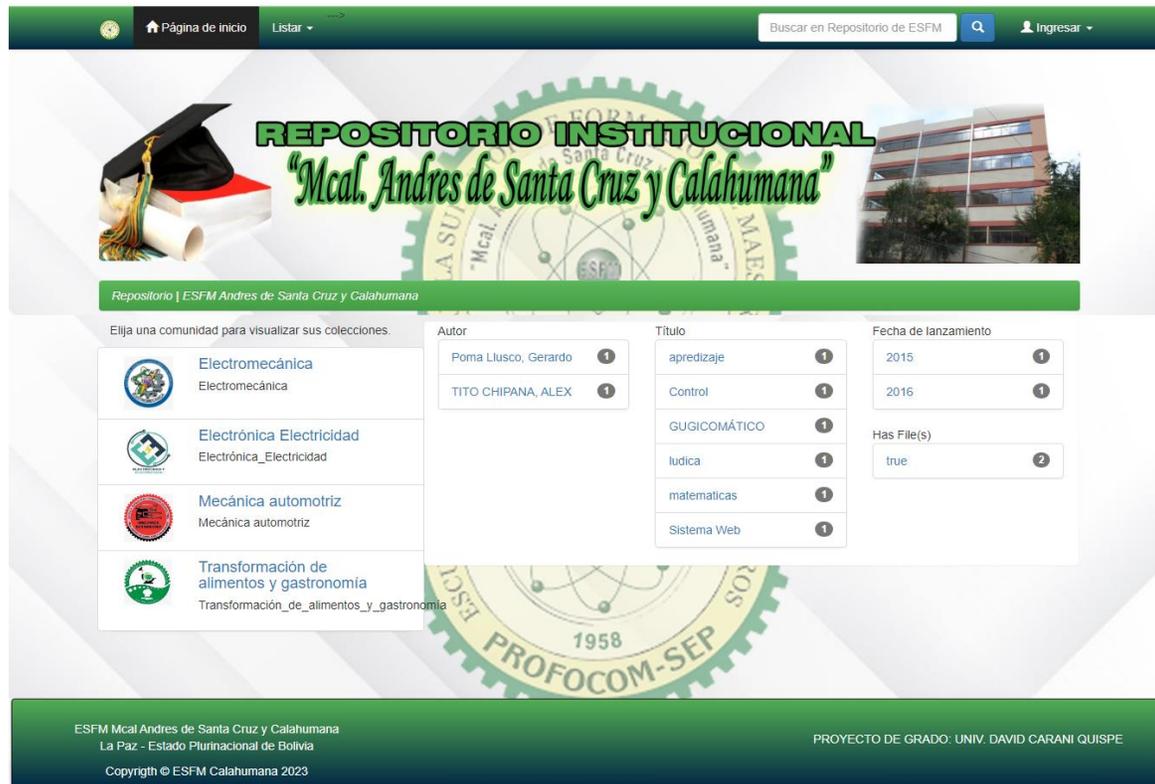
Figura 50
Interfaz de usuario



Nota. En las figuras, se muestra los accesos al sistema, desde un PC o Smartphone.

Figura 51

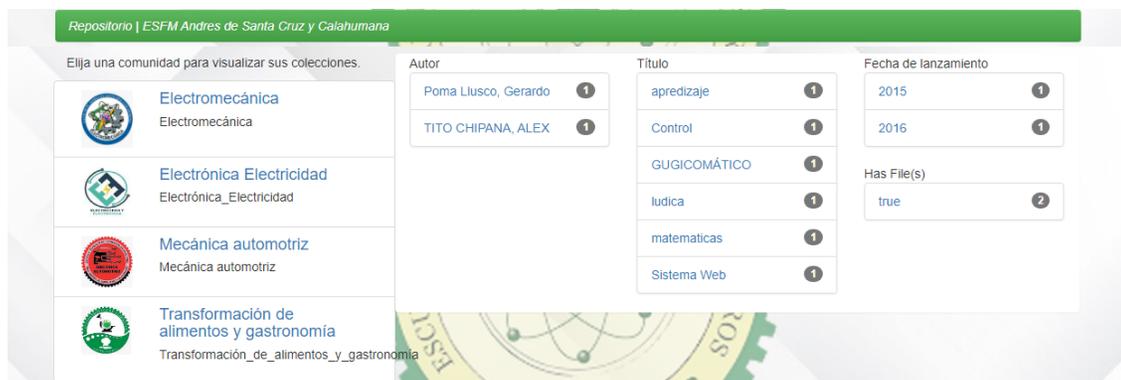
Interfaz de página principal



Nota. La figura, muestra la interfaz de la página principal del sitio web, utilizado en computadoras de escritorio.

Figura 52

Interfaz de comunidades y carreras



Nota. En la figura, se muestra la interfaz de las comunidades o carreras que oferta la institución.

La página también muestra los autores de algunos proyectos publicados, se da prioridad a los títulos de proyectos y la fecha de lanzamiento de proyecto.

Figura 53

Interfaz de subcomunidades y modalidades de graduación

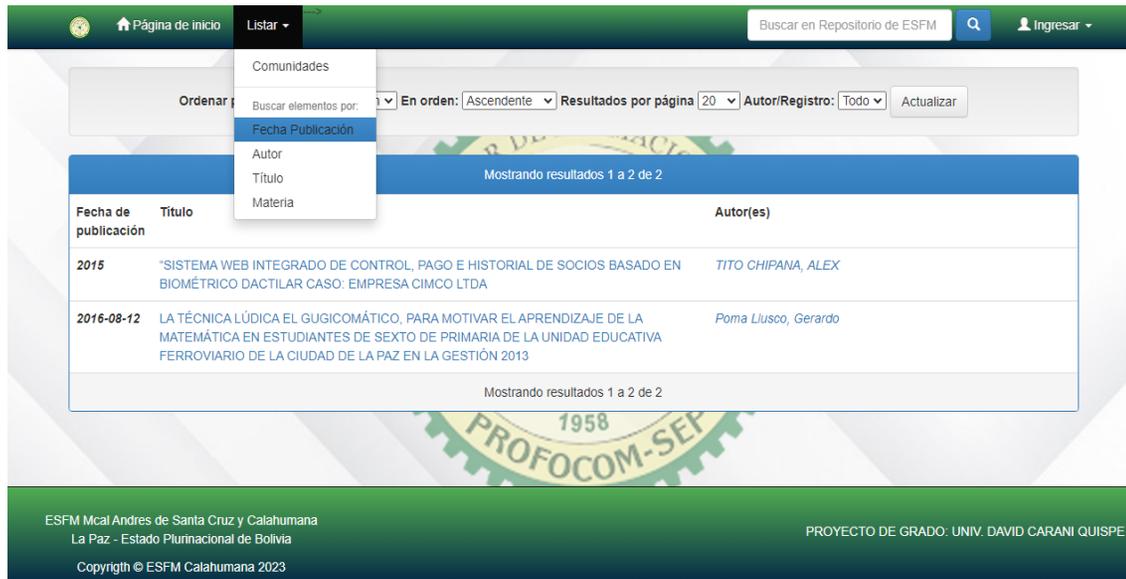


Nota. La figura, muestra la interfaz de las sub comunidades o modalidades de graduación de la institución superior.

Todas las carreras tienen las mismas modalidades de graduación según reglamentación.

Figura 54

Interfaz publicación de documentos



Nota. La Figura, muestra la interfaz de publicación de todos los documentos académicos.

Esta consulta se puede realizar por carrera, modalidad de graduación, autor, titulo o listar todo el conjunto de publicaciones realizadas en el sistema.

Figura 55

Interfaz descarga de documentos



Nota. La figura, muestra la interfaz de descarga de documentos digitales en formato pdf.

El usuario, estudiante e investigador académico podrá acceder a toda la página del sistema Repositorio Académico y descargar los archivos necesarios o de su interés personal.

Figura 56

Interfaz de acceso y autenticación de administrador



Nota. La figura, muestra la interfaz de acceso; la pantalla muestra una página modal de ingreso al sistema para administradores.

Para el acceso al sistema, es necesario completar los campos con el nombre de usuario y la contraseña, que son asignados por el administrador general.

Figura 57

Interfaz de administrador

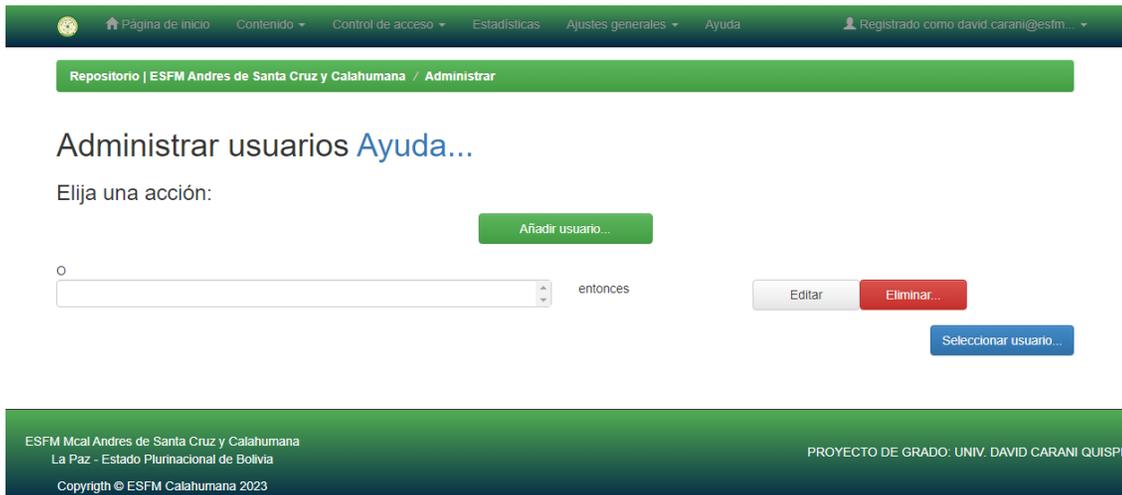


Nota. La figura muestra el ingreso del administrador en el sistema.

Si el administrador no está registrado el administrador general deberá registrar un nuevo administrador para el sistema.

Figura 58

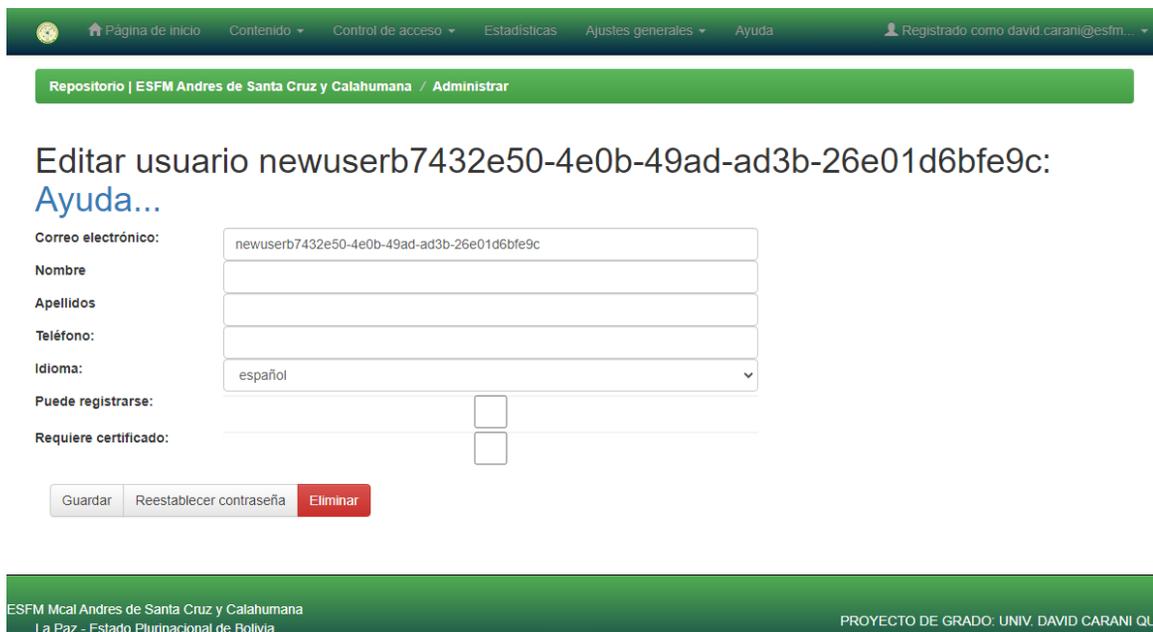
Interfaz de nuevo registro



Nota. La Figura muestra la interfaz de agregar nuevo usuario administrador.

Figura 59

Interfaz de registro de usuario administrador



Nota. En la figura se muestra el registro de nuevo usuario administrador.

Los datos del nuevo usuario y la contraseña para la autenticación deben ser registrado por el administrador general. Los datos del nuevo usuario deben ser registrados de forma correcta.

Figura 60

Interfaz herramientas de administrador



Nota. La Figura, muestra la interfaz de herramientas de administración, el administrador puede realizar todas las acciones según los privilegios autorizados.

Figura 61

Interfaz crear nueva comunidad



Nota. La figura, muestra la interfaz crear nueva comunidad, se podrá adicionar nueva comunidad o carrera según la oferta académica de la institución.

Figura 62

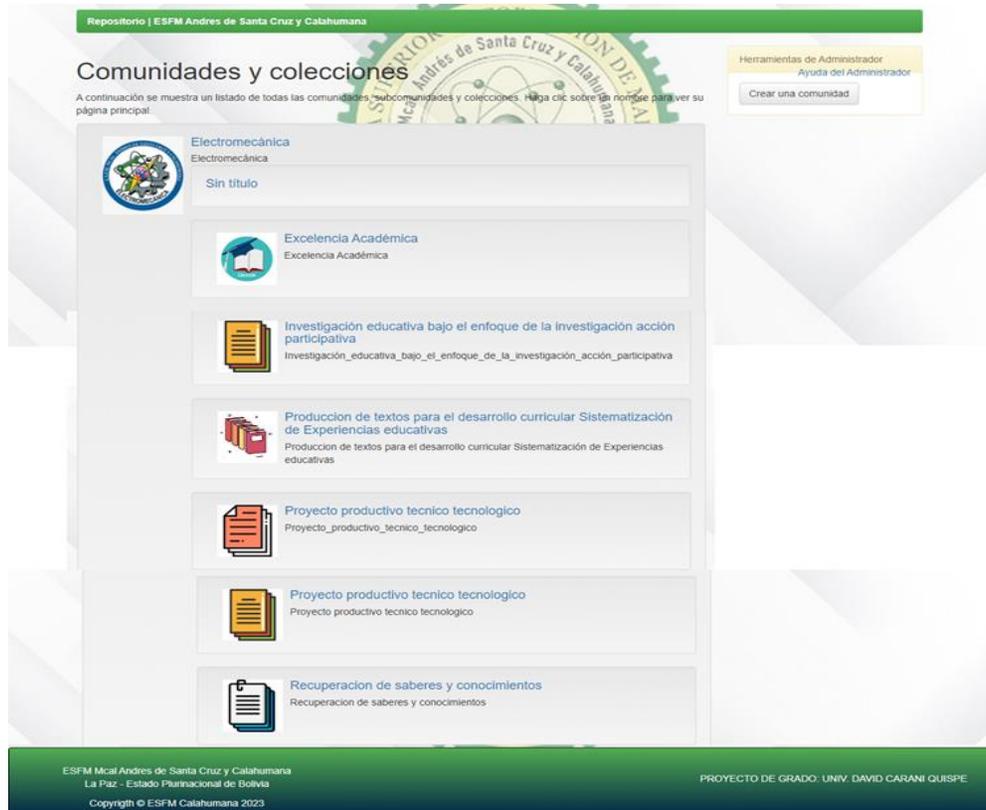
Interfaz crear una colección



Nota. La Figura, muestra la interfaz crear sub comunidad o colección, donde se podrá adicionar las modalidades de graduación de la institución académica.

Figura 63

Interfaz listado de colecciones



Nota. En la figura, se muestra las colecciones o modalidades de graduación habilitadas.

Figura 64

Interfaz enviar documentos



Nota. La figura, muestra la interfaz enviar y publicar nuevo documento.

Figura 65

Interfaz publicar documentos



Nota. En la figura, se muestra la publicación de nuevos documentos en el sistema.

El administrador podrá adicionar y publicar nuevos documentos según la modalidad de graduación y la carrera a la cual corresponde.

Figura 66

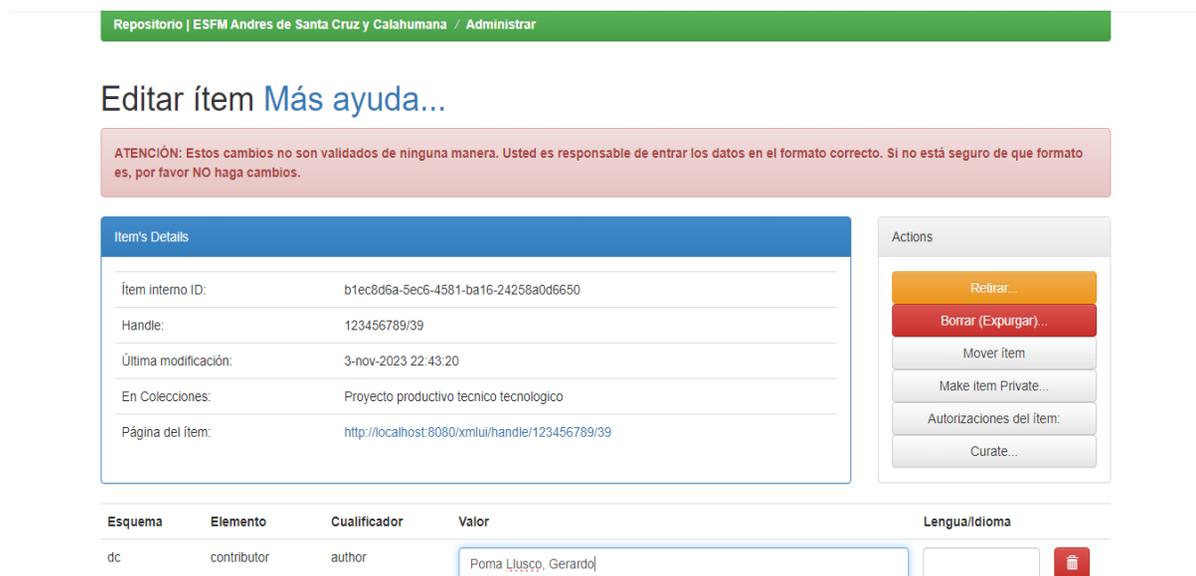
Interfaz editar documentos



Nota. La figura, muestra la interfaz editar y actualizar los documentos publicados en el sistema.

Figura 67

Interfaz retirar borrar y actualizar documentos



Nota. La figura, muestra las interfaces de retirar, borrar y actualizar los documentos publicados en el sistema.

Solo los administradores son responsables de realizar estas acciones según los privilegios designados.

Figura 68

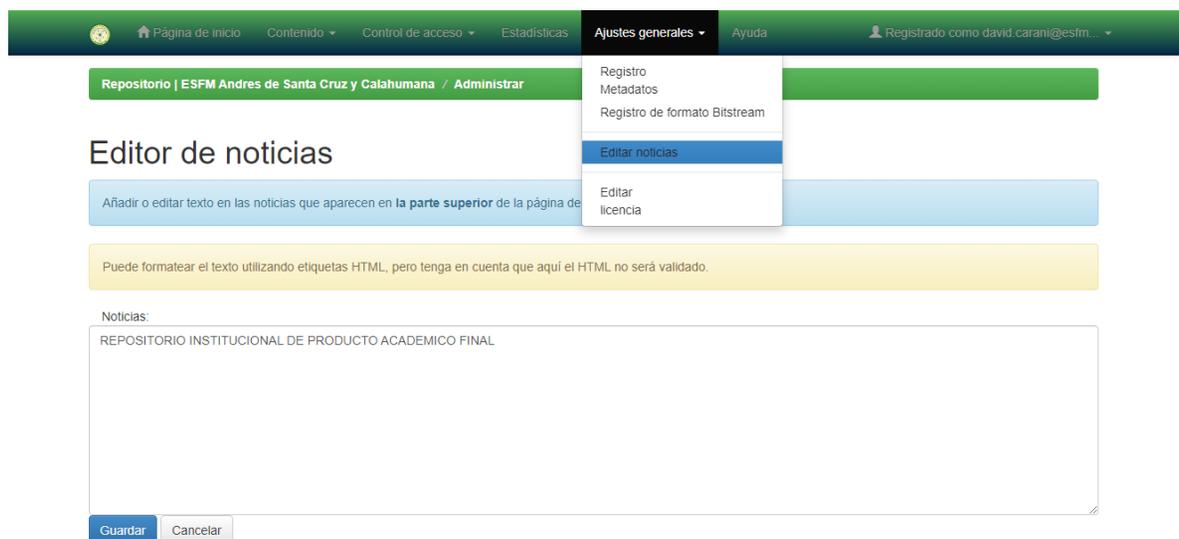
Interfaz publicar noticias



Nota. La figura muestra la interfaz publicar noticias, donde se publica alguna información importante que refiera al sistema.

Figura 69

Interfaz editar noticias



Nota. La figura, muestra la interfaz editar noticias, donde se puede modificar y actualizar las noticias relacionadas con el sistema.

CAPÍTULO IV

4. CALIDAD, COSTOS, SEGURIDAD Y PRUEBAS

En este capítulo se determina la calidad que posee el sistema de repositorio, los procedimientos que se tiene que tomar en cuenta para temas de seguridad y la gestión de riesgos. Debido a que el sistema ha sido desarrollado con una visión de calidad en el diseño y funcionalidad, ese es uno de los aspectos más importantes a tomar en cuenta, esta medición se la realizo bajo las métricas ISO 9126 detalladas en el capítulo II.

El objetivo fue alcanzar un nivel de calidad necesario para satisfacer los objetivos propuestos. Se detalla la gestión de riesgos, los procedimientos más imprescindibles de seguridad que posee el sistema. También se detalla algunas políticas de seguridad.

4.1. MÉTRICAS DE CALIDAD

Es importante recordar que un nivel de aceptabilidad satisfactorio está en el rango de 60 a 100%, un nivel de aceptabilidad Marginal está en el rango de 40 a 60%. Finalmente, un valor de preferencia Insatisfactorio está en el rango de 0 a 40%.

4.1.1. NORMA ISO 9126

El modelo establece seis características más comunes para verificar la calidad del software a la vista interna y externa, cuatro que son propias de la vista en uso.

4.1.1.1. Funcionalidad

La funcionalidad mide el grado en el que el software satisface las necesidades indicadas por la adecuación, exactitud, interoperabilidad y seguridad de acceso.

La funcionalidad no puede ser medida directamente, ya que corresponde derivar medidas directas, como los puntos de función que cuantifica el tamaño y la

complejidad del sistema en términos de las funciones del usuario, determinando las características del dominio.

Para realizar el cálculo de la cuenta total con factores de ponderación se debe tomar en cuenta lo siguiente:

Tabla 17

Factores de ponderación

PARÁMETRO DE MEDICIÓN	CUENTA	FACTOR MEDIO	TOTAL
Número de Entradas del Usuario	45	4	180
Número de salidas del Usuario	20	5	100
Número de Peticiones del Usuario	7	4	28
Numero de Archivos	22	10	220
Numero de Interfaces externas	0	7	0
Cuenta Total			528

Nota. En la tabla, se recopila los factores de ponderación para realizar el cálculo de la cuenta total del sistema.

Los valores de ajuste de complejidad son tomadas en cuenta a partir de la siguiente valoración, respondiendo las siguientes interrogantes:

Tabla 18

Ajuste de complejidad

FACTOR DE COMPLEJIDAD	SIN INFLUENCIA	INCIDENTAL	MODERADO	MEDIO	SIGNIFICATIVA	ESENCIAL	<i>t</i>
Factor	0	1	2	3	4	5	
¿Requiere el sistema copias de seguridad y recuperación fiable?						X	5
¿Se requiere comunicación de datos?						X	5
¿Existen funciones de procesamiento distribuido?						X	3
¿Es crítico el rendimiento?						X	4
¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?						X	3
¿Requiere el sistema entrada de datos interactiva?						X	5
¿Requiere el sistema entrada de datos interactivos que las transacciones de entrada se lleven a cabo sobre múltiples entradas u operaciones?						X	5
¿Se actualizan los archivos maestros de forma interactiva?						X	5

¿Son complejas las entradas, las salidas, los archivos o peticiones?	X	3
¿Es complejo el procesamiento interno?	X	3
¿Se ha diseñado código para ser reutilizable?	X	5
¿Están incluidas en el diseño la conversión y la instalación?	X	3
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	X	2
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizado por el usuario?	X	5
FACTOR DE COMPLEJIDAD		56

Nota. Los ajustes de nivel de complejidad son tomados de la tabla mostrada.

Para el cálculo de los puntos de función (PF) se toma la siguiente relación:

$$PF = cuentaTotal * [0,65 + 0,01 * \Sigma(Fi)]$$

Donde Cuenta Total, es la suma de todas las entradas de los factores de ponderación obtenidas en la tabla anterior. Entonces se tiene.

$$PF = 528 * [0,65 + 0,01 * 56]$$

$$PF = 638,88$$

$$\therefore PF = 639$$

Entonces considerando al 100%, el nivel de confianza con la sumatoria de $Fi=70$ como valor máximo de ajuste de complejidad entonces se tiene lo siguiente:

$$PF_{maximo} = 528 * [0,65 + 0,01 * 70]$$

$$PF_{maxima} = 712,8$$

$$\therefore PF_{maximo} = 713$$

Por lo tanto, la funcionalidad tomara el siguiente valor:

$$Funcionalidad = \frac{PF}{PF_{maxima}}$$

$$\therefore Funcionalidad = \frac{639}{713} = 0,8962$$

La funcionalidad del sistema se pondera en un 89% sin riesgo a fallas y operatividad y un 11% susceptible a fallas o colapso del sistema.

4.1.1.2. Confiabilidad

La confiabilidad del sistema define la probabilidad de operación libre de fallos en un entorno determinado y durante un tiempo específico. Para determinar la confiabilidad de un software especificamos desde el instante que empieza a funcionar es decir $t=0$, a partir de ese momento se realiza las observaciones pertinentes hasta un $t=n$.

- $P(T \leq t)$ Probabilidad de fallas (termino en el cual el sistema trabaja sin fallas)
- $P(T \leq t) = 1 - F(t)$, probabilidad de trabajo sin fallas (Tiempo en el cual no ocurren fallas en el sistema)

Para calcular la confiabilidad del sistema se debe tomar en cuenta el periodo de tiempo en el que se ejecuta el sistema, a partir de ello se irán obteniendo las muestras respectivas.

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

Donde:

f : Funcionalidad del sistema.

μ : Es la probabilidad de error que puede tener el sistema.

t : Tiempo de duración de gestión en el sistema

Entonces, se considerará un periodo de 20 días como tiempo de prueba donde se define que de cada diez ejecuciones se presenta un fallo con el sistema. Procedemos a realizar los cálculos respectivos:

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

$$F(t) = 0,8962 * e^{-110 * 20}$$

$$F(t) = 0,1212$$

$$\therefore F(t) = 12,12 \%$$

Tomando en cuenta el resultado anterior y reemplazando el resultado en las fórmulas de probabilidades se tiene lo siguiente:

$$P(T \leq t) = F(t) \Rightarrow P(T \leq t) = 0,1212 = 12,12\%$$

$$P(T \leq t) = 1 - F(t) \Rightarrow P(T \leq t) = 1 - 0,1212$$

$$\therefore P(T \leq t) = 0,8788 = 87,89\%$$

Entonces, la confiabilidad del sistema es del **88%** en un periodo de 20 días como tiempo de prueba.

4.1.1.3. Mantenibilidad

Se refiere al conjunto de atributos relacionados con el esfuerzo necesario para realizar determinadas modificaciones en el producto. Los atributos son la capacidad de ser analizado, capacidad para ser modificado, estabilidad y capacidad para ser probado.

El estándar IEEE-982.1 sugiere un índice de madurez del software (IMS) que proporcione una indicación de la estabilidad del producto de software, se determina de la siguiente relación:

$$IMS = [Mt - (Fc + Fa + Fd)] / Mt$$

Donde:

Mt: Número de módulos en la versión actual.

Fc: Número de módulos en la versión actual que se han cambiado.

Fa: Numero de módulos en la versión actual que se han añadido.

Fd: Numero de módulos en la versión anterior que se han borrado en la versión actual.

Entonces se procede al cálculo del IMS:

$$IMS = [10 - (0 + 1 + 0)] / 10$$

$$IMS = 0,9 \cong 90\%$$

Entonces se puede indicar que el software tiene una estabilidad del 90% para su mantenibilidad y un 10% restante sería el margen de error por los cambios o modificaciones que sufriera el sistema.

4.1.1.4. Usabilidad

La usabilidad se refiere al conjunto de atributos que miden el esfuerzo cognitivo necesario que deben realizar los usuarios para utilizar el sistema de software, es decir nos indica la facilidad de uso. Mismo que presenta la siguiente ecuación para su respectivo calculo:

$$U = [(Sum(xi)/n)*100]$$

Donde:

X_i : Es la sumatoria de los valores.

n : Es el número de preguntas.

Para responder las preguntas, se deben considerar los siguientes valores

Tabla 19

Ajustes de usabilidad

PREGUNTAS	RESPUESTAS	PONDERACIÓN
¿Puede utilizar con facilidad el sistema?	6 1	0,86
¿Puede controlar operaciones que el sistema solicita?	6 1	0,86
¿Las respuestas del sistema son complicadas?	1 6	0,86
¿El sistema permitió la retroalimentación de información?	5 2	0,71
¿El sistema cuenta con interface agradable a la vista?	6 1	0,86
¿La respuesta del sistema es satisfactoria?	6 1	0,86
¿Los resultados que proporciona el sistema facilitan el trabajo?	6 1	0,86
USABILIDAD		0,84

Nota. El resultado obtenido, concluye que sistema cuenta con una usabilidad del 84% de acuerdo a los datos obtenidos.

4.1.1.5. Portabilidad

La portabilidad del sistema toma en cuenta dos aspectos importantes, el nivel de aplicación y nivel de hardware.

El software desarrollado se trata de un sistema con tecnologías web; es un aplicativo de fácil implementación en cualquier plataforma con servidor web y un gestor de base de datos previamente instalado en nuestro caso PostgreSQL.

Esta tecnología web puede ser ejecutado desde cualquier ordenador o dispositivo móvil con acceso a internet que posea un navegador instalado: Internet Explorer, FireFox Mozilla, Opera, Chrome y otros.

4.1.1.6. Nivel de Aceptabilidad

Tabla 20

Resultados obtenidos

CARACTERÍSTICAS	%
Funcionalidad	89
Confiabilidad	88
Mantenibilidad	90
Usabilidad	84
Portabilidad	99

Nota. Se muestra un resumen de los niveles de aceptabilidad del sistema.

4.1.2. Pruebas de software

Para las pruebas de software, se llevan a cabo una serie de procedimientos:

- Primero es la prueba de caja blanca, que tiene una visión interna, el cual permite garantizar que los elementos del sistema se hayan integrado de manera adecuada.
- Segundo es la prueba de caja negra, con un enfoque externo, este permite verificar que los procesos funcionen y al mismo tiempo se busquen errores.

- Tercero se hace la evaluación de calidad del sistema. Se realiza la definición y especificación de los requerimientos, seguido de la evaluación elemental y global. Al final, se muestra el nivel de aceptabilidad que tiene el sistema.

4.1.2.1. Pruebas de Caja Negra

En la evaluación fundamental del sistema se observaron aspectos de funcionalidad, operativa, aceptación de entradas, resultado, etc. Se procede a realizar una evaluación de acuerdo a los módulos para cotejar los resultados que entrega el sistema ya sea mediante reportes o interfaces de resultados.

Registro de Administrador

Datos de Ingreso: Se ingresa el usuario validado en diferentes caracteres, y se ingresa un password en formato contraseña.

Prueba: El sistema realice las respectivas validaciones para el ingreso de administradores.

Resultado: El sistema no permite el ingreso a usuarios no registrados en el sistema como lo muestra la figura 66.

Figura 70

Interfaz inicio de sesión de sistema

The screenshot shows the DSpace login page. At the top, there is a navigation bar with a home icon, 'Página de inicio', 'Listar', a search bar containing 'Buscar en Repositorio de ESFM', and an 'Ingresar' button. Below this is a green banner with the text 'Repositorio | ESFM Andres de Santa Cruz y Calahumana'. The main heading is 'Entrar en DSpace Ayuda...'. A yellow error message box states: 'La dirección de correo electrónico y la contraseña introducida no es válida. Por favor, inténtelo de nuevo o ¿ha olvidado su contraseña?'. Below the error message is a blue-bordered login form titled 'Entrar en DSpace' with an 'Ayuda...' link. The form contains the text '¿Usuario nuevo? haga clic para registrarse.' and 'Introduzca su dirección de correo electrónico o nombre de usuario y su contraseña:'. It has two input fields: 'Dirección de correo electrónico:' and 'Contraseña:'. A green 'Entrar' button is positioned below the password field. At the bottom left of the form, there is a link '¿Olvidó su contraseña?'

Nota. El inicio de sesión, no permite el ingreso a usuarios no registrados.

Cambio de contraseña

Datos de Ingreso: Se ingresa caracteres de nueva contraseña, se ingresa los mismos datos para confirmar la nueva contraseña.

Prueba: El sistema valida si los datos ingresados dos veces coinciden o no en el formato establecido por el sistema.

Resultado: Como se muestra en la figura 67, el sistema valida si la contraseña ingresado no coincide con el anterior.

Figura 71

Interfaz modificar contraseña

Las contraseñas que ha introducido deben coincidir, y han de tener al menos 6 caracteres.

Por favor, introduzca o corrija la siguiente información. Los campos marcados con * son obligatorios.

Nombre*: DAVID

Apellido*: CARANI

Teléfono de contacto:

Idioma: español

Opcionalmente, puede elegir una nueva contraseña, introduciéndola en el recuadro de abajo y confirmarla escribiéndola de nuevo en el segundo recuadro para verificarla. Debe tener al menos 6 caracteres.

Nueva contraseña:

Confirme la contraseña:

Actualizar perfil

Nota. El sistema verifica si la contraseña ingresado no coincide con el anterior.

Enlaces

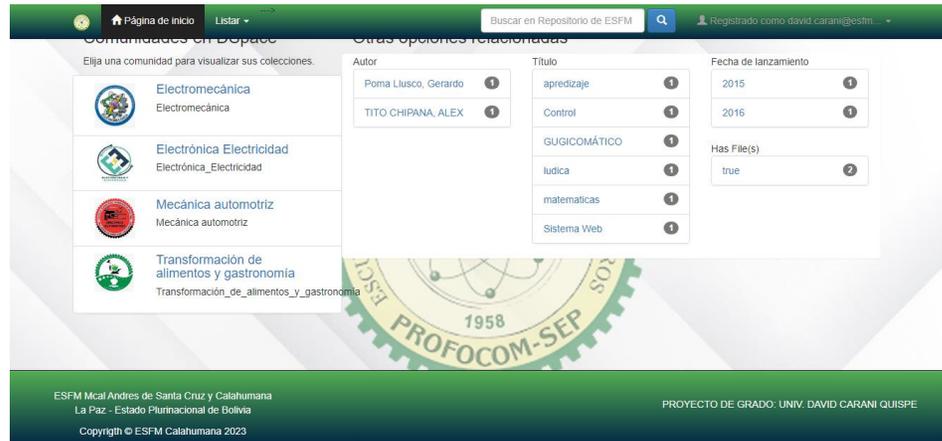
Datos de Ingreso: Ingresar al módulo de enlaces.

Prueba: Se espera que el sistema despliegue la información de las páginas más visitadas por los investigadores académicos.

Resultado: El sistema despliega enlaces de todas las páginas como se muestra en la figura 68.

Figura 72

Interfaz módulos enlaces del sistema



Nota. El módulo de enlace del sistema, muestra todas las utilidades que tiene el sistema.

Búsqueda de Información

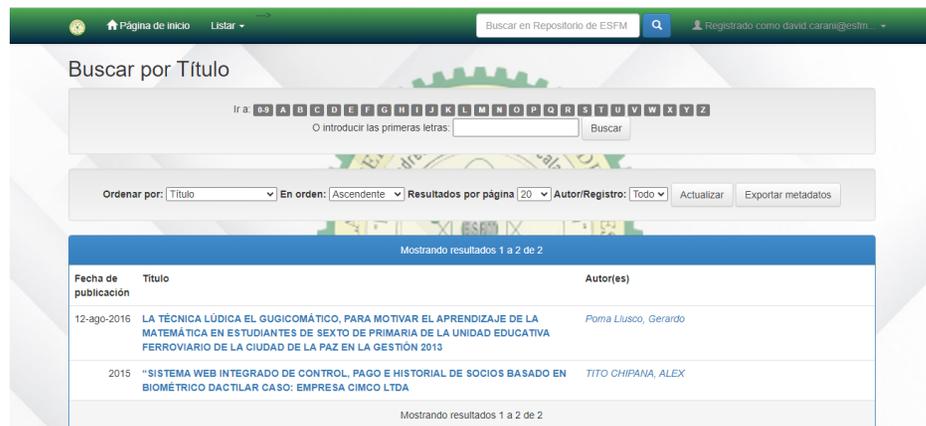
Datos de Ingreso: Se ingresa el autor, titulo o materia para iniciar la búsqueda de información.

Prueba: Se espera que el sistema ejecute la opción seleccionada, el sistema despliega todos los internos o nombres relacionados con los caracteres ingresados.

Resultado: Como se muestra en la figura 69, el sistema despliega todos los datos encontrados y relacionados con los datos de búsqueda ingresados.

Figura 73

Interfaz búsqueda de información



Nota. El sistema despliega todos los datos encontrados y relacionados con los datos de búsqueda.

Información Académica

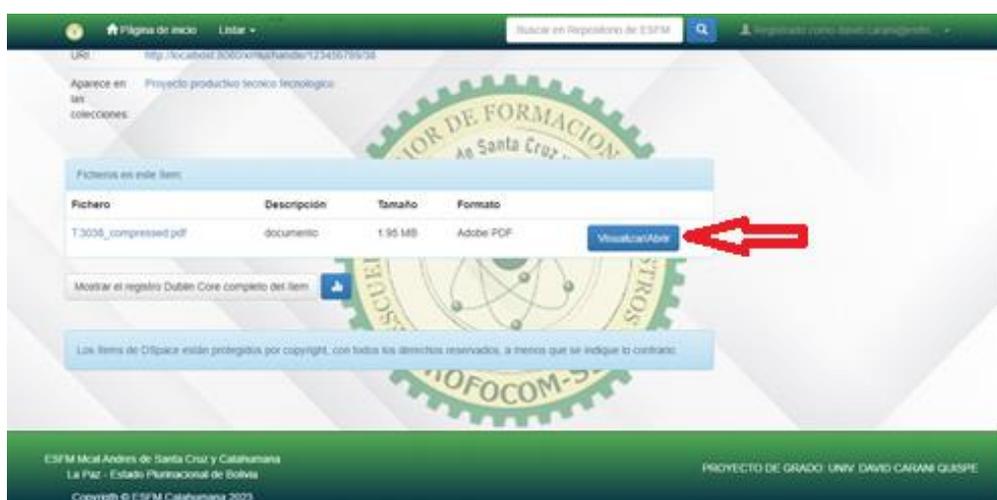
Datos de Ingreso: Seleccione el archivo o información requerida en el despliegue de datos que realiza el sistema.

Prueba: Espere a que el sistema extraiga el archivo seleccionado.

Resultado: Como se muestra en la figura 70, el sistema visualiza el archivo seleccionado para luego ser descargado en formato PDF (figura 71).

Figura 74

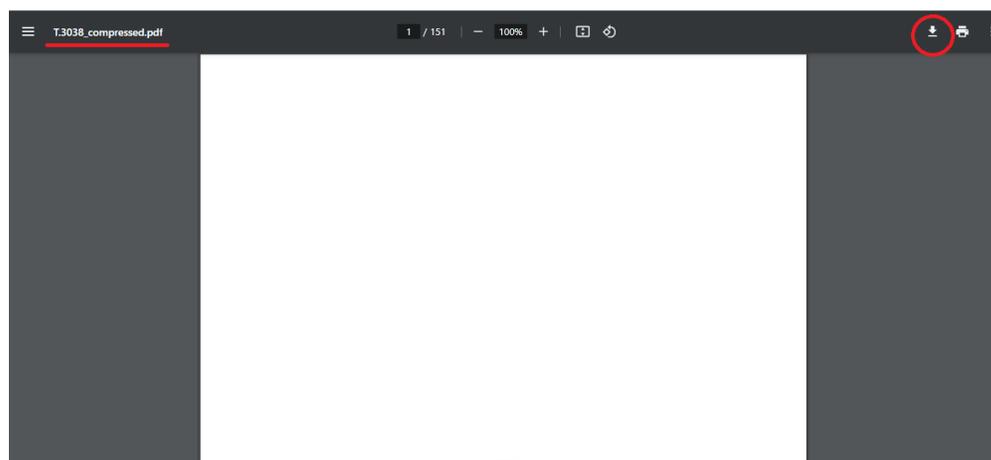
Interfaz información académica



Nota. En la Figura se muestra la información obtenida del repositorio.

Figura 75

Interfaz descargar archivo



Nota. La figura muestra la forma de descarga de archivos digitales.

4.1.2.2. Prueba de Caja Blanca

Para las pruebas de caja blanca se proporciona una medición cuantitativa de la complejidad lógica del programa. La Complejidad ciclomática de un grado de flujo $V(G)$ establece el número de caminos independientes:

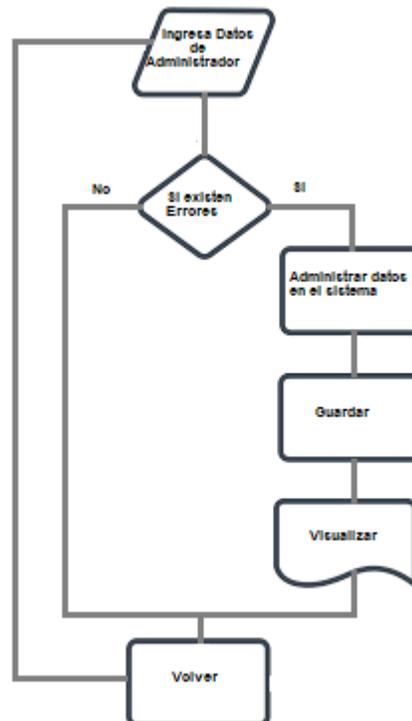
- ✓ El número de regiones del gráfico de flujo.
- ✓ $V(G) = A - N + 2$, donde A es el número de aristas y N es el número de nodos.
- ✓ $V(G) = P + 1$, donde P es el número de nodos predicado.

Gestión de Usuarios

Es necesario realizar pruebas de caja blanca para el módulo de administración del sistema debido al flujograma que presenta.

Figura 76

Pruebas de caja blanca



Nota. En la figura se muestra el flujo grama para el registro de administrador del sistema.

Para ilustrar el flujograma se puede evidenciar el procedimiento en las figuras 71 y 72 del módulo de administración de administrador.

Figura 77

Alta de administrador

Repositorio | ESFM Andres de Santa Cruz y Calahumana / Administrar

Editar grupo : Administrator (id: f463fc47-e01e-4948-a597-4fdcd9dcde8f) Ayuda...

Nombre: Administrator

Miembros actuales del grupo

Miembros EPersona

- DAVID CARANI (david.carani@esfm.gob.bo)
- MARISEL YARARI (marisel.yarari@esfm.gob.bo)

Miembros del grupo

Borrar selección Sacar selección Seleccionar usuarios...

Actualizar grupo

Nota. En la Figura, se muestra la habilitación y roles de los administradores.

Figura 78

Reporte de alta de administrador

Repositorio | ESFM Andres de Santa Cruz y Calahumana / Administrar

Editar usuario newuser9d0b0cce-afeb-4363-ae34-71a24f82f3b6: Ayuda...

Correo electrónico: newuser9d0b0cce-afeb-4363-ae34-71a24f82f3b6

Nombre: Marisel

Apellidos: Yarari

Teléfono:

Idioma: español

Puede registrarse:

Requiere certificado:

Guardar Reestablecer contraseña Eliminar

ESFM Mcal Andres de Santa Cruz y Calahumana
La Paz - Estado Plurinacional de Bolivia

PROYECTO DE GRADO. UNIV. DAVID CARANI QUISPE

Nota. En la figura, se verifica el registro del nuevo administrador.

La figura 76 muestra la complejidad ciclomatica de un módulo de administración, por lo que reemplazando el número de predicados en la fórmula se obtiene:

$$V(G) = 1 + 1 = 2$$

Este valor determina dos caminos independientes, dos casos de prueba, para el módulo de administración. Por lo tanto, se realizará las pruebas necesarias para cada uno de los caminos como se muestra en la tabla 27.

Tabla 21*Casos de Prueba*

CAMINO 1	CAMINO 2
1. Ingresar datos en el formulario de administración.	1. Ingresar datos en el formulario de Administración
2. Vuelve a ingresar datos para un nuevo registro de administración.	2. Verifica y valida si existen datos incorrectos en el formulario.
	3. Guarda información del administrador en el sistema.
	4. Visualizar reportes de administrador.
	5. Vuelve a ingresar datos para verificar la validación de los nuevos datos en el sistema.

Nota. La tabla, identifica los procesos de validación de nuevos administradores del sistema.

4.2. ESTIMACIÓN DE COSTOS

La estimación del tamaño de software se puede realizar en diferentes etapas del proyecto, en este caso se la realiza en la etapa temprana del proyecto. Para esta actividad, se utiliza el Modelo Constructivo de Costos COCOMO II, que es la evolución de COCOMO, permitiendo realizar las estimaciones en función del tamaño del software y de un conjunto de factores de coste y de escala. Al final se encuentra el costo total de desarrollo del sistema.

4.2.1. Modelo de Diseño Temprano

En esta etapa se realiza la estimación de los costos mediante el COCOMO II y se utiliza el modelo de diseño temprano.

4.2.1.1. Calculo de los Puntos de Función no Ajustados

Tabla 22*Puntos función sin ajustar*

TIPO DE ELEMENTOS	DIFICULTAD	PESO	CANTIDAD	TOTAL PUNTOS	TOTAL ELEMENTOS
ENTRADAS	Simple	3	4	12	
	Media	4	8	32	
	Compleja	6	2	21	
Total puntos de función entrada:					65
	Simple	4	2	8	

SALIDAS	Media	5	3	15	
	Compleja	7	1	7	
Total puntos de función salida:					30
CONSULTAS	Simple	3	4	12	
	Media	4	6	24	
	Compleja	6	3	18	
Total puntos de función consulta:					54
ARCHIVOS	Simple	7	5	35	
	Media	10	6	60	
	Compleja	15	2	30	
Total puntos de función archivos:					125
TOTAL PUNTOS DE FUNCIÓN NO AJUSTADOS					274

Nota. El proceso inicia calculando los puntos de función no ajustados: PFNA = 274

4.2.1.2. Calculo de los Puntos de Función Ajustados

En esta fase se valora una serie de factores de complejidad. Es necesario responder 14 preguntas y evaluarlos en una escala del 0 al 5, donde: 0 es no influencia, 1 es incidental, 2 es moderado, 3 es medio, 4 es significativo y 5 es esencial.

Tabla 23

Factores de complejidad

CARACTERÍSTICAS	ESO
Comunicación de datos	5
Procesamiento distribuido de datos	5
Rendimiento	3
Configuraciones fuertemente utilizadas	3
Frecuencia de transacciones	5
Entrada de datos on-line	5
Eficiencia del usuario final	4
Actualizaciones Online	0
Procesamiento complejo	4
Reusabilidad	3
Facilidad de Instalación	2
Facilidad de operación	2
Instalación de distintos lugares	0
Facilidad de cambio	2
TOTAL Fi	43

Nota. Los factores de complejidad se calculan según la tabla de datos.

El factor de ajuste (FA) se calcula de la siguiente ecuación:

$$FA = 0.65 + 0.01 * \sum Fi$$

Dónde:

Fi corresponde a los pesos asignados a los siguientes factores:

Reemplazando en la ecuación, se tiene:

$$FA = 0.65 + 0.01 * 43$$

$$FA = 1.08$$

Los puntos función ajustados (PFA) se obtienen calculando:

$$PFA = PFNA * FA$$

Reemplazando en la ecuación se tiene:

$$PFA = 274 * 1.08$$

$$PFA = 295.92$$

Tabla 24

Líneas de código por cada punto de función

LENGUAJE	LDC/PF
Access	38
Ada 95	49
AI Shell	49
APL	32
Assembly - Basic	320
Basic – ANSI	64
Basic – Compiled	91
Basic – Visual	32
C	128
C++	55
Cobol (ANSI 85)	91
Base de Datos	40
4ta Generación de Lenguaje	20
Nivelar Lenguaje	64
Jovial	107

Machine Code	640
Modula 2	80
PHP	29
PERL	27
Prolog	64
Query – Default	13
Report Generator	80
Second Generation Language	107
Spreadsheets	46
Third Generation Language	80
USR 5	1
Visual Basic 5.0	29

Nota. La Tabla, muestra la equivalencia de líneas de código y puntos de función.

Los puntos función convertidos a miles de códigos (**KSLOC**) se calculan de la siguiente ecuación:

$$\mathbf{KSLOC = \frac{PF * LLP}{1000}}$$

LLP: Líneas de código requeridos por puntos de función en un lenguaje LLP = 29.

Remplazando en la ecuación se tiene:

$$\mathbf{KSLOC = \frac{295.92 * 29}{1000}}$$

$$\mathbf{KSLOC = 8.58168}$$

4.2.1.3. Estimación del Esfuerzo

Para la estimación del esfuerzo nominal (**PM_{Nominal}**) se utiliza la siguiente ecuación:

$$\mathbf{CPM_{Nominal} = A * SIZE^B}$$

Dónde:

A: Constante de calibración, **A = 2.45**

SIZE (tamaño) = KSLOC

B: es el factor de ahorro o gasto relativo de escala.

$$B = 0.91 + 0.01 * \sum W_i$$

Donde:

W_i: Es el factor de escala.

Tabla 25

Factores de escala

FACTORES DE ESCALA	MUY ABAJO	BAJO	NOMINAL	ALTO	MUY ALTO	EXTRA ALTO
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	2.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

Nota. En la tabla se detalla los valores predeterminados para calcular la estimación de esfuerzo.

Tabla 26

Valores de los factores de escala

FACTORES DE ESCALA	NIVEL	VALOR
PREC	Bajo	4.96
FLEX	Alto	2.03
RESL	Nominal	4.24
TEAM	Alto	2.19
PMAT	Nominal	4.98
Resultado		18.1

Nota. La tabla muestra los valores obtenidos.

El valor de $\sum W_i = 18.1$, se reemplaza en la ecuación de factor de ahorro.

$$B = 0.91 + 0.01 * 18.1$$

$$B = 1.091$$

Con el valor obtenido de factor de ahorro, se procede a calcular el esfuerzo nominal.

$$PM_{\text{nominal}} = 2.45 * 8.58168^{1.091}$$

$$PM_{\text{nominal}} = 25.56781$$

Para el esfuerzo ajustado ($PM_{ajustado}$) en Personas/Mes se utiliza la siguiente ecuación:

$$PM_{ajustado} = PM_{nominal} * \prod EMI$$

EMI: Son los valores que toman los drivers de coste, los cuales serán tomados de la siguiente tabla:

Tabla 27

Factores de ajuste

DRIVERS	EXTRA ABAJO	MUY BAJO	BAJO	NOMINA L	ALTO	MUY ALTO	EXTRA ALTO
RCPX	0.73	0.81	0.98	1	1.30	1.74	2.38
RUSE	-	-	0.95	1	1.07	1.15	1.24
PDIF	-	-	0.87	1	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1	0.87	0.73	0.62
SCED	-	1.43	1.14	1	1	1	

Nota. En la tabla se detalla los valores predeterminados de los factores de ajuste.

Tabla 28

Valores de los factores de ajuste

DRIVERS	NIVEL	VALOR
RCPX	Alto	1.30
RUSE	Bajo	0.95
PDIF	Alto	1.29
PERS	Nominal	1
PREX	Nominal	1
FCIL	Alto	0.87
SCED	Nominal	1
Resultado		1.05

Nota. La tabla muestra los valores obtenidos.

Remplazando en la ecuación de esfuerzo ajustado se tiene:

$$PM_{ajustado} = 25.56781 * 1.05$$

$$PM_{ajustado} = 26,85$$

$$PM_{ajustado} = 27 \text{ Personas/mes}$$

4.2.1.4. Estimación del Cronograma

El tiempo de desarrollo (TDEV) se calcula de la siguiente ecuación:

$$DEV = [3.67 * PM^{(0.28+0.2*(B-1.01))}] * [SCED\% / 100]$$

Donde:

SCED%: Es el porcentaje en que se calcula podrá aumentar o disminuir el calendario (SCED% = 100)

Remplazando en la ecuación se tiene:

$$TDEV = [3.67 * 27.06^{(0.28+0.2*(1.091-1.01))}] * [100/100]$$

$$TDEV = 9.75 T$$

$$TDEV = 8 \text{ meses}$$

La cantidad de personal (NP) se calcula de la siguiente ecuación:

$$NP = \frac{PM_{\text{ajustado}}}{TDEV}$$

Remplazando en la ecuación se tiene:

$$NP = 27 / 8$$

$$NP = 3.37$$

$$NP = 3 \text{ personas}$$

4.2.1.5. Costo Total del Sistema

El costo de desarrollo (**CD**) se calcula con la ecuación siguiente:

$$CD = TDEV * PS * NP$$

Remplazando en la ecuación se tiene:

$$CD = 8 * 2709 * 3$$

$$CD = 65016 \text{ Bs}$$

El proyecto tendrá un costo de Bs. 65016, será desarrollado aproximadamente en 8 meses y será necesario 3 personas, que tendrán un sueldo promedio de 2709 Bs.

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

Las conclusiones permiten establecer, hasta qué punto fueron alcanzados los objetivos propuestos en el presente proyecto, también las recomendaciones constituyen un nuevo enfoque que se podría tomar en cuenta en el desarrollo de trabajos similares o mejoras en este mismo proyecto.

5.1. CONCLUSIONES

Cumplida la implementación del sistema, se concluye que:

Se implementó el sistema Repositorio Institucional de Producto Académico Final, que mejorara la difusión de información y gestión académica en la E.S.F.M. “Mcal. Andrés de Santa Cruz y Calahumana”

De acuerdo a los objetivos específicos planteados se llega a las siguientes conclusiones:

- Se desarrolló una plataforma sencilla e intuitiva con autenticación única que permite la integración de sistemas con los administradores.
- Se desarrolló una herramienta que brinda información a los estudiantes y a toda la comunidad académica investigadora.
- Se desarrolló una herramienta en la cual los administradores pueden tener un control sobre sus documentos académicos de la institución.
- Se integró todas estas herramientas en una plataforma colaborativa.

Como resultado de las evaluaciones realizadas al sistema se han alcanzado niveles altos de calidad y usabilidad, el sistema hace énfasis en el diseño y se cuenta con ayudas integradas al sistema alcanzado un sistema intuitivo, en el que los usuarios y administradores no necesitan una capacitación previa.

5.2. RECOMENDACIONES

Para finalizar el proyecto se manifiesta las siguientes recomendaciones:

Promover el desarrollo de herramientas libres visualizando a los navegadores de los componentes que necesitan para sus procedimientos.

Que un sistema sea compatible con distintos navegadores sin necesidad de rediseñar y adecuarla para cada uno.

Se recomienda tener de dos a tres administradores del sistema como máximo, para evitar algún tipo de mala manipulación del mismo.

BIBLIOGRAFÍA

LIBROS CONSULTADOS

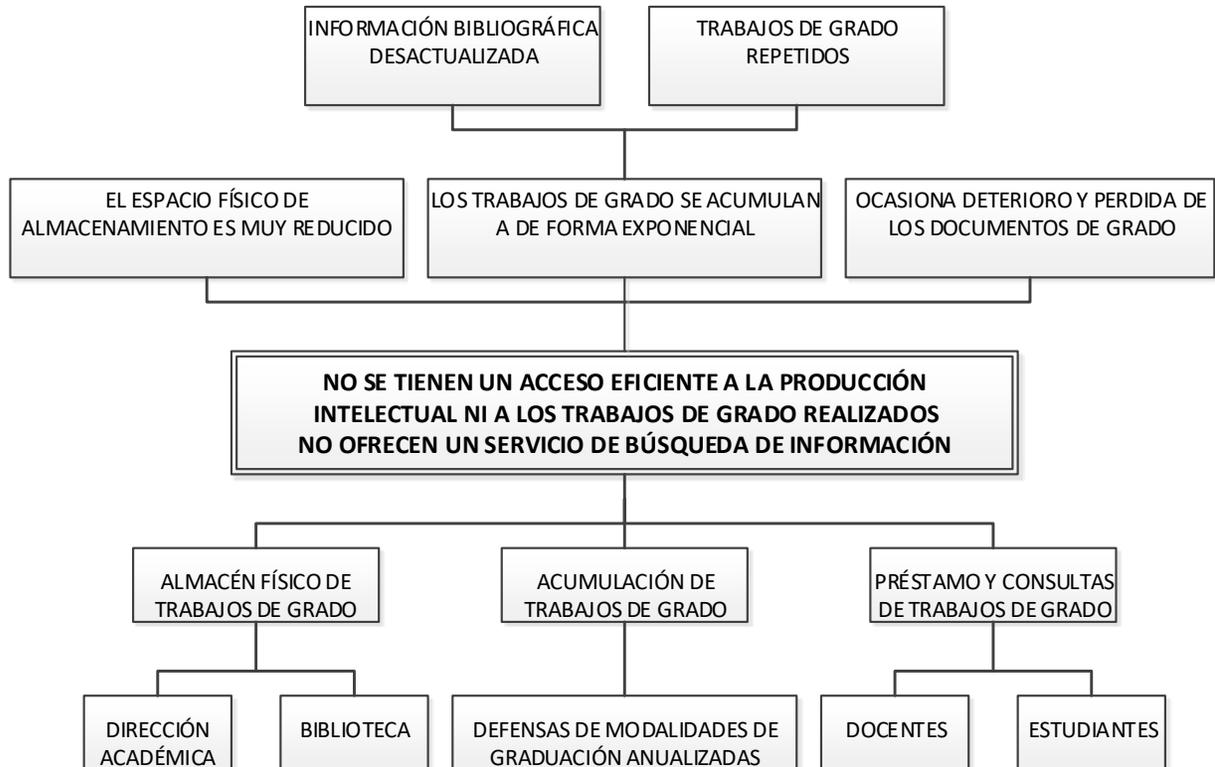
- Calero, 2003, CALERO SOLIS Celso. Una Explicación de la Programación Extrema (XP). Tercera Edición.
- Chaves, 2006, CHAVES Norberto. El Diseño Invisible: Siete lecciones sobre la intervención culta en el hábitat humano. Paidós Iberica.
- Gauchat, 2012, GAUCHAT, Juan Diego. El gran libro de HTML5, CSS3 y JAVASCRIPT
- Ramos, 2008, RAMOS Efraim Wilfredo. Sistema de Control y Seguimiento Administrativo.
- Sommerville, 2005 SOMERVILLE, Ian . Ingeniería del Software. Séptima Edición. Pearson Education Limited.
- Guell, N., Schawabe, D., & Patricia, V. (s.f.). Modeling Interactions and Navigation in Web Applications. PUC-Rio. Rua M. de São Vicente, 225. Rio de Janeiro.
- Kendall, K. E., & Kendall, J. E. (2011). Análisis y Diseño de Sistemas, Octava Edición. Naucalpan de Juárez, Estado de México: PEARSON EDUCATION, INC.
- Larman, G. (2003). UML y Patrones, Introducción al Análisis y Diseño Orientado a Objetos, Segunda Edición. México: PEARSON EDUCACIÓN, S.A.
- Pressman, R. S. (2010). Ingeniería del Software, Un Enfoque Práctico, Séptima Edición. México, D. F.: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Ramez, E., & Shamkant B., N. (2007). Fundamentos de Sistemas de Bases de Datos. Madrid (España): PEARSON EDUCACIÓN S.A.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2007). El Lenguaje de Modelado Unificado, Manual de Referencia, Segunda Edición. Madrid (España): PEARSON EDUCACIÓN, S. A.

- Schawabe, D., & Rossi, G. (s.f.). An Object Oriented Approach to Web-Based Application Design. Departamento de Informática. PUC-Rio, Brazil.
- Schwabe, D., & Rossi, G. (1998). Developing Hypermedia Applications using OOHDM. Rio (Brazil).
- V. Mannino, M. (2007). Administración de Bases de datos, Diseño y Desarrollo de Aplicaciones, Tercera Edición. México, D.F.: McGRAW-HILL / INTERAMERICAN EDITORES, S.A. DE C.V.

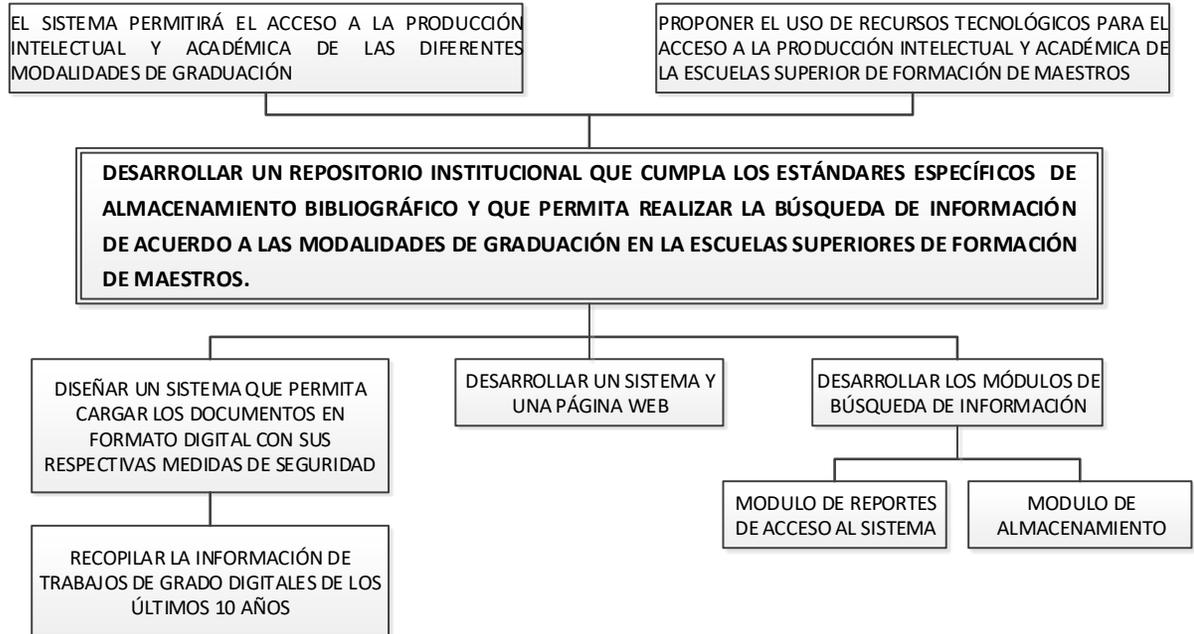
ANEXOS

ANEXO 1

ÁRBOL DE PROBLEMAS



ÁRBOL DE OBJETIVOS





MINISTERIO DE EDUCACIÓN
DIRECCIÓN GENERAL DE FORMACIÓN DE MAESTROS
ESCUELA SUPERIOR DE FORMACIÓN DE MAESTROS
"MCAL. ANDRÉS DE SANTA CRUZ Y CALAHUMANA"
FUNDADO - 1958

La Paz, noviembre 27 de 2023

NE/ESFM MASCYC/DG No. 0114/2023

Señor (a):

M. Sc. Lic. Ing. Maricel Yarari Mamani

TUTOR METODOLOGICO

TALLER DE GRADO II

Presente.-

Ref.: **AVAL DE CONFORMIDAD**

La Dirección General de la Escuela Superior de Formación de Maestros "Mcal. Andres de Santa Cruz y Calahumana", hace llegar un saludo cordial y deseos de éxitos en sus actividades.

A través de la presente tengo a bien hacer conocer el aval de conformidad del Trabajo de Grado:

Título: **REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL**

Caso: ESFM "Mcal. Andrés de Santa Cruz y Calahumana".

Modalidad: Proyecto de Grado.

Universitario: David Carani Quispe.

Carnet de Identidad: 6001668 LP.

Registro Universitario: 200477.

Por otro lado, hago conocer que el sistema implementado satisface los requerimientos de la institución, el mismo que avala al universitario David Carani Quispe para continuar a la defensa pública, evaluación correspondiente a la materia de Taller de Grado II y otros.

Agradeciendo su atención me despido con las consideraciones del caso.

Atentamente,




Lic. **Edy Lupa Juarez**
DIRECTOR GENERAL
ESCUELA SUPERIOR DE FORMACION DE MAESTROS
"MCAL. ANDRES DE STA. CRUZ Y CALAHUMANA"

La Paz, noviembre de 2023

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:

TÍTULO: "REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL"

CASO: ESCUELA SUPERIOR DE FORMACIÓN DE MAESTROS "MCAL. ANDRÉS DE
SANTA CRUZ Y CALAHUMANA"

MODALIDAD: PROYECTO DE GRADO

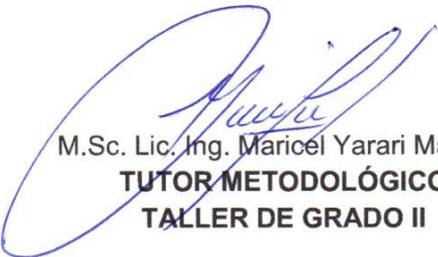
UNIVERSITARIO: DAVID CARANI QUISPE

REGISTRO UNIVERSITARIO: 200477

CEDULA DE IDENTIDAD: 6001668 LP

Para su Defensa Pública y valuación 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. Maricel Yarari Mamani

**TUTOR METODOLÓGICO
TALLER DE GRADO II**

La Paz, noviembre de 2023

Señor(a):

M. Sc. Lic. Ing. Maricel Yarari Mamani

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:

TÍTULO: "REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL"

CASO: ESCUELA SUPERIOR DE FORMACIÓN DE MAESTROS "MCAL. ANDRÉS DE SANTA CRUZ Y CALAHUMANA"

MODALIDAD: PROYECTO DE GRADO

UNIVERSITARIO: DAVID CARANI QUISPE

REGISTRO UNIVERSITARIO: 200477

CEDULA DE IDENTIDAD: 6001668 LP

En cuanto certifico, en honor a la verdad para su Defensa Publica 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.



Ing. Adelaida Ximena Pastrana Arcani

TUTOR REVISOR

La Paz, noviembre de 2023

Señor(a):

M. Sc. Lic. Ing. Maricel Yarari Mamani

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:

TÍTULO: "REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL"

CASO: ESCUELA SUPERIOR DE FORMACIÓN DE MAESTROS "MCAL. ANDRÉS DE SANTA CRUZ Y CALAHUMANA"

MODALIDAD: PROYECTO DE GRADO

UNIVERSITARIO: DAVID CARANI QUISPE

REGISTRO UNIVERSITARIO: 200477

CEDULA DE IDENTIDAD: 6001668 LP

En cuanto certifico, en honor a la verdad 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 de Sistemas de la Universidad Pública de El Alto.

Atentamente.



M. Sc. Lic. Juan Carlos Sarzuri Patzi
TUTOR ESPECIALISTA



REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL

**ESCUELA SUPERIOR DE FORMACIÓN DE MAESTROS
“MCAL. ANDRÉS DE SANTA CRUZ Y CALAHUMANA”**



Elaborado por: Univ. David Carani Quispe

DSPACE

2023



REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADÉMICO FINAL

INTRODUCCIÓN.

El Sistema Repositorio institucional es una herramienta desarrollada para facilitar el acceso a las publicaciones de documentos de grado realizados por la comunidad de investigadores, colaboradores y estudiantes que forman parte de la Escuela Superior de Formación de Maestros “Mcal. Andrés de Santa Cruz y Calahumana”.

El mismo constituye la forma más apropiada para proporcionar acceso abierto a textos completos y diversos recursos de información producida por la institución académica para optar el título de maestro/a con grado de licenciatura.

El repositorio busca mantener, preservar y diseminar la información académica, tecnológica y de innovación derivada de las investigaciones, productos educativos y académicos de la institución.

Desde el sistema, se puede descargar los trabajos de investigación realizados dentro de la institución.

A continuación, se detalla los manuales necesarios para un buen uso y administración del Repositorio Institucional.



MANUAL DE USUARIO



MANUAL DE USUARIO

El objetivo del repositorio es el resguardo y organización de la producción intelectual, la misma describirá y preservará la documentación de forma digital.

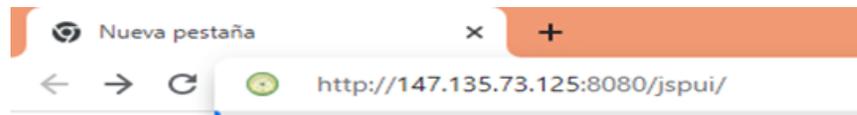
1. Acceso al Repositorio Institucional

Para poder acceder al repositorio institucional, es necesario utilizar un navegador web. Usted puede acceder desde el navegador de su preferencia:



En el navegador web, escribir la siguiente dirección URL en la barra de direcciones:

<http://147.135.73.125:8080/jspui/>



Después de cargar la página web, encontrará el link "Repositorio Institucional" en la parte inferior de la pantalla. Haciendo clic sobre el mismo se accede al repositorio.



REPOSITORIO INSTITUCIONAL
"Mcal. Andrés de Santa Cruz y Calahumana"

Repositorio | ESFM Andres de Santa Cruz y Calahumana

REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADEMICO FINAL

NOTICIA DE BARRA LATERAL

Comunidades en DSpace
Elija una comunidad para visualizar sus colecciones.

Otras opciones relacionadas

Autor	Título	Fecha de lanzamiento
Alanoca Hidalgo, Ovidio	DIDÁCTICO	2022

2. Usuarios

Usuarios o visitantes, son usuarios que pueden revisar el material publicado en el repositorio, con la opción de descarga.

Para realizar la búsqueda de información, lo pueden realizar a través de las siguientes formas:

- Comunidades
- Autor
- Título
- Fecha de lanzamiento
- Has file(s)

Comunidades en DSpace
Elija una comunidad para visualizar sus colecciones.

ELECTRICIDAD - ELECTRONICA Electricidad - Electrónica	Alanoca Hidalgo, Ovidio	DIDÁCTICO	2022
ELECTROMECAÁNICA Electromecánica	Alvarez Condon, Jessica Vianney	TABLEROS	2021
MECÁNICA AUTOMOTRIZ Mecánica automotriz	Apaza Quispe, Andres	CONSTRUCCIÓN DE TABLEROS	Has File(s) true
TRANSFORMACIÓN DE ALIMENTOS Y GASTRONOMÍA Transformación de alimentos y gastronomía	Barríos Tarifa, Adrian Migue	DESTREZAS	
	Cadena Quispe, Lizeth	EMPRENDIMIENTOS PRODUCTIVOS	
	Callizaya Alanoca, Cesar Ricardo	ENTRENAMIENTO	
	Carvajal Yavincha, Ruddy	FORMACIÓN TÉCNICA	
	Chambi Mendoza, Cesar Reynaldo	FORTALECIMIENTO DE HABILIDADES	



3. Visualización de contenido

La visualización del contenido se encuentra estructurado de la siguiente manera:

- a) Parte Superior Letra grande color negro: Título de publicación y a su lado el autor (es).

Fecha de publicación	Título	Autor(es)
2022-11	FORTALECIMIENTO DE HABILIDADES Y DESTREZAS DE LOS ESTUDIANTES, MEDIANTE LA IMPLEMENTACIÓN DE UN BANCO DIDÁCTICO DE PRUEBAS EN UN MOTOR A INYECCIÓN, EN LA U.E. "JOSE LUIS SUÁREZ GUZMÁN"	Alanoca Hidalgo, Ovidio; Callizaya Alanoca, Cesar Ricardo; Quispe Riveros, Juan Deyvis

Ordenar: **Fecha Publicación** (selected), Autor, Título, Materia

Fecha de publicación	Título	Autor(es)
2015	"SISTEMA WEB INTEGRADO DE CONTROL, PAGO E HISTORIAL DE SOCIOS BASADO EN BIOMÉTRICO DACTILAR CASO: EMPRESA CIMCO LTDA	TITO CHIPANA, ALEX
2016-08-12	LA TÉCNICA LÚDICA EL GUGICOMÁTICO, PARA MOTIVAR EL APRENDIZAJE DE LA MATEMÁTICA EN ESTUDIANTES DE SEXTO DE PRIMARIA DE LA UNIDAD EDUCATIVA FERROVIARIO DE LA CIUDAD DE LA PAZ EN LA GESTIÓN 2013	Poma Jusco, Gerardo

- b) Menú Lateral izquierdo: “Ver” Descarga documento original en PDF.

Fichero	Descripción	Tamaño	Formato
Motor a Inyeccion.pdf	PROYECTO PRODUCTIVO TÉCNICO TECNOLÓGICO	9.6 MB	Adobe PDF

Visualizar/Abrir

Mostrar el registro Dublin Core completo del ítem

Los ítems de DSpace están protegidos por copyright, con todos los derechos reservados, a menos que se indique lo contrario.



Hacer clic en, **Visualizar** y **Abrir** para iniciar la descarga y obtener el documento requerido.



c) Menú Lateral: podrá observar, el título, otros títulos, autor, palabra clave, fecha de la publicación (mes y año), citación, el resumen, y la descripción de la publicación del proyecto.

Por favor, use este identificador para citar o enlazar este ítem: <http://localhost:8080/xmlui/handle/123456789/65>

Título :	FORTALECIMIENTO DE HABILIDADES Y DESTREZAS DE LOS ESTUDIANTES, MEDIANTE LA IMPLEMENTACIÓN DE UN BANCO DIDÁCTICO DE PRUEBAS EN UN MOTOR A INYECCIÓN, EN LA U.E. "JOSE LUIS SUAREZ GUZMAN"
Otros títulos :	MOTOR A INYECCIÓN
Autor :	Alanoca Hidalgo, Ovidio Callizaya Alanoca, Cesar Ricardo Quispe Riveros, Juan Deyvis
Palabras clave :	FORTALECIMIENTO DE HABILIDADES MOTOR A INYECCIÓN
Fecha de publicación :	nov-2022
Citación :	PROYECTO PRODUCTIVO TÉCNICO TECNOLÓGICO
Resumen :	En este proyecto se presenta un estudio realizado como Equipo Comunitario de Trabajo de Grado en la Unidad Educativa Técnico Humanístico José Luis Suárez Guzmán para fortalecer las habilidades y destrezas mediante el diseño y elaboración de un banco didáctico de pruebas en un motor a inyección electrónica. Esta necesidad se tomó de la especialidad de mecánica automotriz mediante las técnicas de instrumentos de investigación este proyecto resulto con un diagnóstico socio participativo dentro de la unidad educativa a la cual nos enfocamos a la problematización de la educación BTH.
Descripción :	Ukhamaxa uñajasaxa kunti jani utj mecanica automotriz yatichawirxa, ukatawa imaqaraktanxa aka instrumentos de investigación ukanakampi laqi jaqinakampi qhithinakatixa yatipañanakana utji ukanakana (actores educativos) ukatpi aka jani walt awinakatixa utjki uka pampachañatiki.



MANUAL DE ADMINISTRADOR



MANUAL DE ADMINISTRADOR

El administrador es el responsable de interactuar con el sistema repositorio.

El administrador (biblioteca) se encarga de verificar y validar los documentos de trabajo de grado que serán publicados.

También tiene la responsabilidad de registrar y carga en el sistema los documentos seleccionados según la carrera y la modalidad de graduación a la que corresponde.

ADMINISTRADOR

1. Acceso al Repositorio Institucional

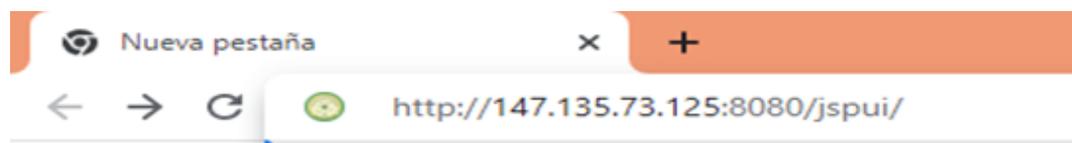
Para acceder al repositorio institucional, no es necesario contar con un equipo de última generación, podrá acceder desde un PC de escritorio, Laptop, Tablet o un Smartphone con acceso a internet.

Es necesario utilizar un navegador web de su preferencia:



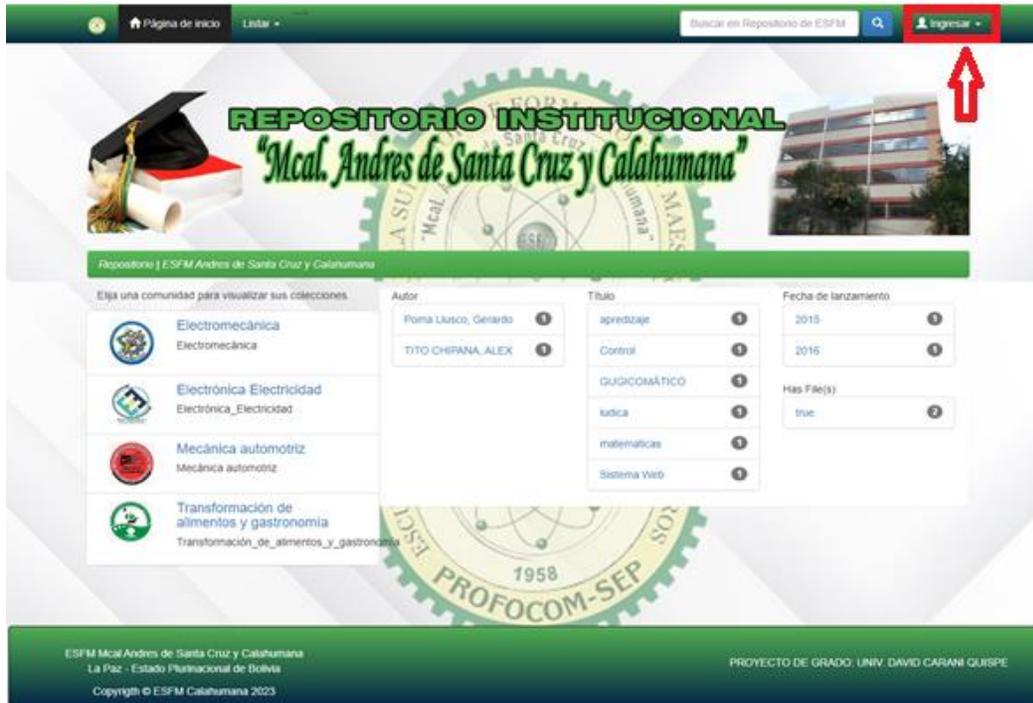
En el navegador web, escribir la siguiente dirección URL en la barra de direcciones:

<http://147.135.73.125:8080/jspui/>





Después de cargar la página web, encontrará la opción de ingresar al “Repositorio Institucional”.



2. Autenticación

La pantalla muestra una página modal de ingreso al sistema para administradores donde es necesario completar los campos con el nombre de usuario y la contraseña, que son asignados por el administrador general.



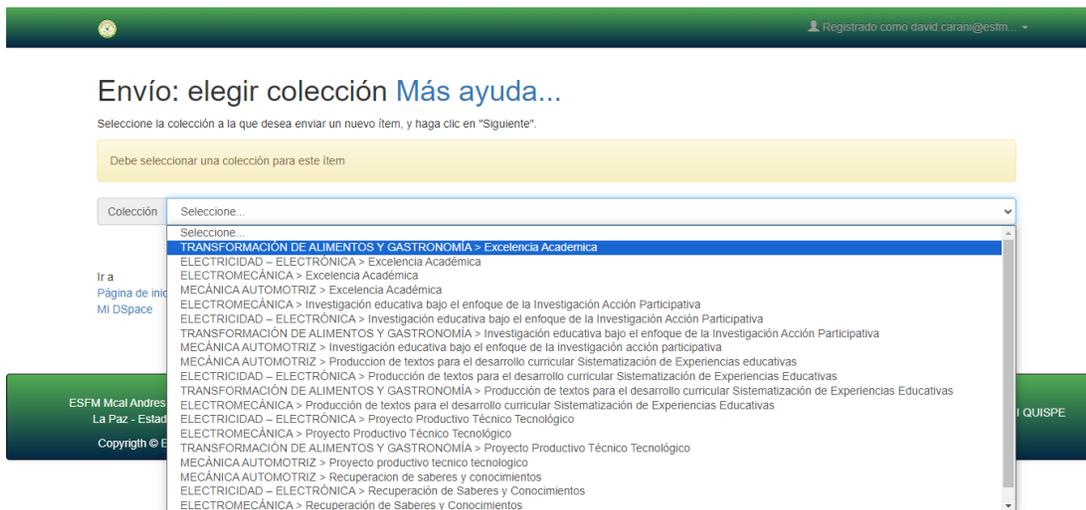


3. Publicar

Seleccione la opción **Nuevo Envió**



Seleccione la carrera y la modalidad de graduación.





Rellene la información requerida para su envío.

Registrado como david.carani@esfm...

Autores [+ Añadir más](#)

Ingrese el título principal del artículo.
Título *

Si el artículo tiene títulos alternativos, ingréselo aquí.
Otros títulos [+ Añadir más](#)

Indique la fecha de la publicación o distribución pública anterior. Puede omitir el día y/o el mes si no son aplicables.
Fecha de emisión * Mes: (sin mes) Día: Año:

Introduzca el nombre del editor de la instancia publicada anteriormente de este elemento.
Editor

Ingrese la cita estándar para la instancia emitida anteriormente de este artículo.
Citación

Identificadores [+ Añadir más](#)

Seleccione el tipo(s) de contenido del artículo. Para seleccionar más de un valor en la lista, es posible que deba mantener presionada la tecla "CTRL" o "Shift".
Tipo

- Animación
- Artículo
- Libro
- Capítulo del libro
- Conjunto de datos
- Objeto de aprendizaje

Seleccione el idioma del contenido principal del artículo. Si el idioma no aparece en la lista, seleccione 'Otro'. Si el contenido realmente no tiene un idioma (por ejemplo, si es un conjunto de datos o una imagen), seleccione 'N/A'.
Idioma

[Cancelar/Guardar](#) [Siguiente >](#)

Introduzca la información requerida para el envío del archivo.

Registrado como david.carani@esfm...

Enter the abstract of the item.
Abstract

Enter the names of any sponsors and/or funding codes in the box.
Sponsors

Enter any other description or comments in this box.
Description

[< Anterior](#) [Cancelar/Guardar](#) [Siguiente >](#)



Seleccione el archivo a ser publicado.

Registrado como david.carani@esfm...

Por favor, entre el nombre del fichero que corresponda al ítem. Si hace clic en "Examinar", aparecerá una nueva ventana en la que podrá localizar y seleccionar el fichero.

Por favor, tenga en cuenta que el sistema DSpace puede preservar el contenido de ciertos tipos de ficheros mejor que otros. La información sobre los tipos de ficheros y los niveles de soporte para cada uno están disponibles.

Select a file or drag & drop files ...

< Anterior Cancelar/Guardar Siguiente >

SHERPA/ROMEO Publisher Policy Database

ESFM Mcal Andres de Santa Cruz y Calahumana
La Paz - Estado Plurinacional de Bolivia
Copyright © ESFM Calahumana 2023

PROYECTO DE GRADO: UNIV. DAVID CARANI QUISPE

Verifique que el fichero haya enviado correctamente el archivo.

Registrado como david.carani@esfm...

Envío: fichero subido correctamente [mas ayuda...](#)

Su fichero fue subido correctamente.

La tabla de abajo muestra los ficheros que ha subido para este ítem.

Bitstream primario	Fichero	Tamaño	Descripción	Formato del fichero
	EYE-0921.pdf	28456535 bytes	PROYECTO PRODUCTIVO TECNICO TECNOLOGICO	Adobe PDF (Conocido)

[Añadir otro fichero](#)

Puede verificar que el fichero(s) ha sido subido correctamente a partir de:

- Haciendo clic en los nombres de los ficheros para comprobar los contenidos.
- El sistema puede calcular un checksum que usted puede verificar. Haga clic aquí para obtener más información. [Mostrar checksums](#)

< Anterior Cancelar/Guardar Siguiente >

ESFM Mcal Andres de Santa Cruz y Calahumana
La Paz - Estado Plurinacional de Bolivia
Copyright © ESFM Calahumana 2023

PROYECTO DE GRADO: UNIV. DAVID CARANI QUISPE



Verifique que todos los datos ingresados sean correctos. Tiene opción a realizar modificaciones antes de ser publicado el trabajo de grado.

Registrado como david.carani@esfm

Type	Book
Language	Spanish

Subject Keywords: ENTRENAMIENTO, FORTALECER [Corregir uno de estos](#)

Abstract: El Ministerio de Educación del Estado Plurinacional de Bolivia mediante el compendio normativo para las Escuelas Superiores de Formación de Maestros y Maestras / Unidades Académicas versión 2018, establece el reglamento de la Investigación Educativa y Producción De Conocimientos – Práctica Educativa Comunitaria (IEPC - PEC)

Sponsors Description: Unidad Educativa Técnico Humanístico "Eufrasio Ibáñez Rivero" Kunjamati Bolivia markanxa Ministerio de Educación uksa tuqinxaxa Escuelas Superiores de Formación de Maestros ukhamaraki Maestras / Unidades Académicas versión 2018 ukanakaxa compendio kamachirjama thakhinchayatarakiwa. ukatawa Investigación Educativa ukhamaraki Producción De Conocimientos – Práctica Educativa Comunitaria (IEPC - PEC) ukanakaxa yabata jisk'a kamachirjama kunkanchayata.

Ficheros subidos: EYE-0921.pdf - Adobe PDF (Conocido) [Añadir o borrar un fichero](#)

< Anterior Cancelar/Guardar Siguiente >

Acepte los términos de publicación.

Registrado como david.carani@esfm

submission to any medium or format for the purpose of preservation.

You also agree that DSU may keep more than one copy of this submission for purposes of security, back-up and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant DSU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN DSU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.

DSU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

[No acepto la licencia](#) [Acepto la licencia](#)



Publicación terminada.

Registrado como david.carani@esfm

Describir Descripción Subir Verificar Licencia Completo

Envío: envío finalizado!

Su envío pasará ahora a un proceso de flujo de trabajo designado para la colección a la que lo está enviando. Recibirá una notificación de correo electrónico tan pronto como su envío pase a formar parte de la colección, o si hubiese algún problema con su envío. También puede comprobar el estado de su envío yendo a la página "Mi DSpace".

Ir a "Mi DSpace"

Comunidades y colecciones

Añadir otro archivo a la colección

ESFM Mcal Andres de Santa Cruz y Calahumana
La Paz - Estado Plurinacional de Bolivia
Copyright © ESFM Calahumana 2023

PROYECTO DE GRADO: UNIV. DAVID CARANI QUISPE

Verificar, que los documentos enviados y cargados se publique de forma correcta. al sistema.

Repositorio | ESFM Andres de Santa Cruz y Calahumana

Comunidades y colecciones

A continuación se muestra un listado de todas las comunidades, subcomunidades y colecciones. Haga clic sobre su nombre para ver su página principal.

Herramientas de Administrador
Ayuda del Administrador
Crear una comunidad

-  **Electromecánica**
Electromecánica
Sin título
-  **Excelencia Académica**
Excelencia Académica
-  **Investigación educativa bajo el enfoque de la investigación acción participativa**
Investigación_educativa_bajo_el_enfoque_de_la_investigación_acción_participativa
-  **Produccion de textos para el desarrollo curricular Sistematización de Experiencias educativas**
Produccion de textos para el desarrollo curricular Sistematización de Experiencias educativas
-  **Proyecto productivo tecnico tecnologico**
Proyecto_productivo_tecnico_tecnologico
-  **Proyecto productivo tecnico tecnologico**
Proyecto productivo tecnico tecnologico
-  **Recuperacion de saberes y conocimientos**
Recuperacion de saberes y conocimientos

ESFM Mcal Andres de Santa Cruz y Calahumana
La Paz - Estado Plurinacional de Bolivia
Copyright © ESFM Calahumana 2023

PROYECTO DE GRADO: UNIV. DAVID CARANI QUISPE



ESFM "Mcal. Andrés de Santa Cruz y Calahumana"



ESTADO PLURINACIONAL DE BOLIVIA

MINISTERIO DE EDUCACIÓN



4. Adicionar nueva comunidad o carrera

The screenshot shows the homepage of the ESFM institutional repository. The header includes navigation links like 'Página de inicio' and 'Listar', a search bar, and a user profile. The main banner features the repository title and a building image. Below the banner, there are several content cards for 'Electrónica Electricidad' with sub-headers like 'Excelencia Académica', 'Investigación educativa...', 'Producción de textos...', 'Proyecto productivo...', and 'Recuperación de saberes...'. On the right side, a 'Herramientas de Administrador' menu is visible, with a red box and arrow pointing to the 'Crear una comunidad' option.

5. Adicionar nueva modalidad de graduación

This screenshot shows the same website but with a specific community selected. The main content area now displays 'Electromecánica' with a star icon and the text 'Página de inicio de la comunidad'. The 'Herramientas de Administrador' menu on the right is expanded, showing options like 'Editar...', 'Crear una colección', 'Crear una subcomunidad', 'Exportar Comunidad', 'Exportar (migrar) comunidad', and 'Exportar metadatos'. A blue box and arrow highlight the 'Crear una subcomunidad' option.



6. Editor de noticias

The screenshot shows a web application interface for editing news. At the top, there is a navigation bar with a home icon, 'Página de inicio', 'Contenido', 'Control de acceso', 'Estadísticas', 'Ajustes generales', and 'Ayuda'. A user profile 'Registrado como david.carami@estm...' is visible on the right. Below the navigation bar, a green header contains 'Repositorio | ESFM Andres de Santa Cruz y Calahumana / Administrar'. A dropdown menu is open under 'Ajustes generales', listing 'Registro', 'Metadatos', 'Registro de formato Bitstream', 'Editar noticias' (highlighted), and 'Editar licencia'. The main content area is titled 'Editor de noticias' and contains a blue instruction box: 'Añadir o editar texto en las noticias que aparecen en la parte superior de la página de...'. Below this is a yellow warning box: 'Puede formatear el texto utilizando etiquetas HTML, pero tenga en cuenta que aquí el HTML no será validado.' A text area labeled 'Noticias:' contains the text 'REPOSITORIO INSTITUCIONAL DE PRODUCTO ACADEMICO FINAL'. At the bottom, there are 'Guardar' and 'Cancelar' buttons.

Un producto útil entrega contenido, funciones y características que el usuario final desea; sin embargo, de igual importancia es que entrega estos activos en forma confiable y libre de errores.



MANUAL TÉCNICO



MANUAL TÉCNICO

El manual técnico proporciona instrucciones detalladas para la configuración del sistema “Repositorio Institucional de Producto Académico Final”.

1. REQUERIMIENTOS TÉCNICOS

Requerimientos de Hardware

- Memoria RAM mínimo 4 gigabits (GB).
- Espacio de Almacenamiento mínimo 320 gigabits (GB).
- Procesador 1vCPU Cores.
- 15 TB de transferencia.
- Conexión a internet estable.

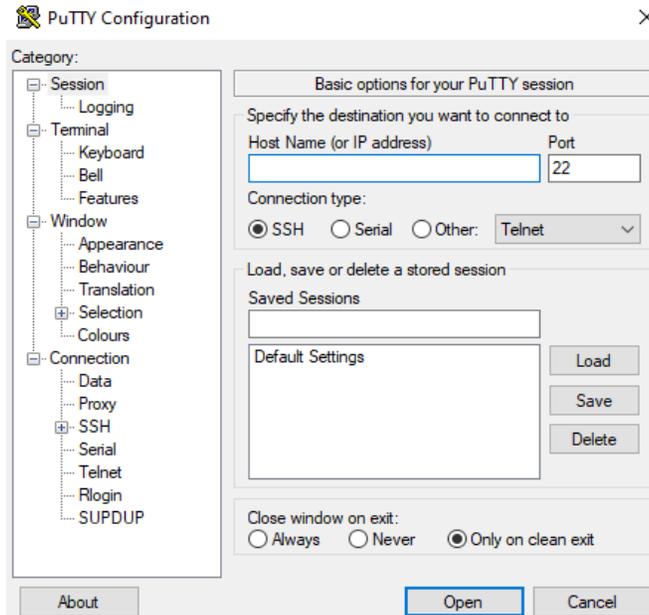
Requerimiento de software

- Sistema Operativo, se recomienda utilizar una distribución de Linux (Ubuntu, Centos).
- Servidor Web, recomendado Apache2.
- Lenguaje de Programación, JavaScript en todo el sistema.

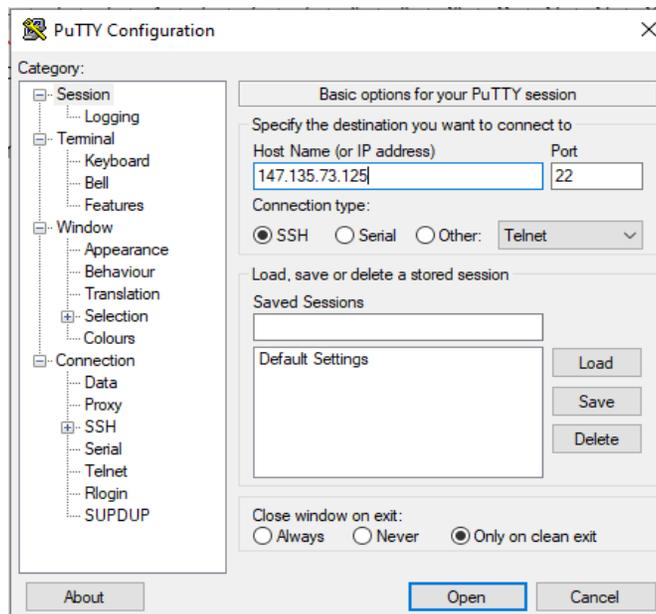
2. HERRAMIENTAS PARA EL ACCESO AL SISTEMA

Para la administración del sistema debemos tener instalado los siguientes softwares:

PuTTY es un cliente SSH y telnet, desarrollado para la plataforma Windows. PuTTY es un software de código abierto que está disponible con código fuente y está desarrollado y respaldado por un grupo de voluntarios.



Ingrese la dirección IP **147.135.73.125** ssh puerto 22





Ingrese al Servidor en modo súper usuario

Usuario : root

Password : +8WVNs14bnsZ#9

```
root@rl-qed: ~  
login as: root  
root@147.135.73.125's password:  
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 6.2.16-14-pve x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
  https://ubuntu.com/livepatch  
You have mail.  
Last login: Mon Nov 20 16:04:03 2023 from 181.115.130.68  
root@rl-qed:~#
```

3. RUTA DE INSTALACIÓN DEL SISTEMA

/dSPACE/webapp

Visualizar grupo de servicios

root@rl-qed:~# cat /etc/group

```
root@rl-qed: ~  
video:x:44:  
sasl:x:45:  
plugdev:x:46:  
staff:x:50:  
games:x:60:  
users:x:100:  
nogroup:x:65534:  
crontab:x:101:  
syslog:x:102:  
ssl-cert:x:103:postgres  
input:x:104:  
systemd-journal:x:105:  
systemd-network:x:106:  
systemd-resolve:x:107:  
messagebus:x:108:  
postfix:x:109:  
postdrop:x:110:  
ssh:x:111:  
mlocate:x:112:  
uidd:x:113:  
tomcat8:x:114:  
postgres:x:115:  
dSPACE:x:1000:  
root@rl-qed:~#
```



Ruta de acceso dspace

cd /dspace

ls

```
root@rl-qed: /dspace
users:x:100:
nogroup:x:65534:
crontab:x:101:
syslog:x:102:
ssl-cert:x:103:postgres
input:x:104:
systemd-journal:x:105:
systemd-network:x:106:
systemd-resolve:x:107:
messagebus:x:108:
postfix:x:109:
postdrop:x:110:
ssh:x:111:
mlocate:x:112:
uidd:x:113:
tomcat8:x:114:
postgres:x:115:
dspace:x:1000:
root@rl-qed:~# cd /dspace
root@rl-qed:/dspace# ls
assetstore  etc          imports      reports      upload
bin         exports     lib          solr         var
config      handle-server log          triplestore webapps
root@rl-qed:/dspace#
```

Ruta de acceso webapps

cd webapps/

ls

```
root@rl-qed: /dspace/webapps
syslog:x:102:
ssl-cert:x:103:postgres
input:x:104:
systemd-journal:x:105:
systemd-network:x:106:
systemd-resolve:x:107:
messagebus:x:108:
postfix:x:109:
postdrop:x:110:
ssh:x:111:
mlocate:x:112:
uidd:x:113:
tomcat8:x:114:
postgres:x:115:
dspace:x:1000:
root@rl-qed:~# cd /dspace
root@rl-qed:/dspace# ls
assetstore  etc          imports      reports      upload
bin         exports     lib          solr         var
config      handle-server log          triplestore webapps
root@rl-qed:/dspace# cd webapps
root@rl-qed:/dspace/webapps# ls
dspace  oai  rdf  rest  solr  sword  swordv2  xmlui
root@rl-qed:/dspace/webapps#
```



Ruta de acceso jspui

```
# cd jspui/
```

```
# ls
```

```
root@rl-qed: /dSPACE/webapps/jspui
tomcat8:x:114:
postgres:x:115:
dSPACE:x:1000:
root@rl-qed:~# cd /dSPACE
root@rl-qed:/dSPACE# ls
assetstore  etc          imports      reports      upload
bin         exports     lib          solr         var
config      handle-server log          triplestore webapps
root@rl-qed:/dSPACE# cd webapps
root@rl-qed:/dSPACE/webapps# ls
jspui  oai  rdf  rest  solr  sword  swordv2  xmlui
root@rl-qed:/dSPACE/webapps# cd jspui
root@rl-qed:/dSPACE/webapps/jspui# ls
META-INF          display-item.jsp          layout                statistics
WEB-INF           display-statistics.jsp   login                 styles.css
browse            dSPACE-admin            mydSPACE              submit
collection-home.jsp error                    print.css             suggest
community-home.jsp favicon.ico              register              tombstone.jsp
community-list.jsp feedback                requestItem           tools
components        help                    robots.txt            utils.js
controlledvocabulary home.jsp                search                workspace
css                image                  sherpa
discovery          index.jsp                static
```

Ruta de acceso layout

```
# cd layout/
```

```
# ls
```

```
root@rl-qed: /dSPACE/webapps/jspui/layout
root@rl-qed:/dSPACE/webapps# ls
jspui  oai  rdf  rest  solr  sword  swordv2  xmlui
root@rl-qed:/dSPACE/webapps# cd jspui
root@rl-qed:/dSPACE/webapps/jspui# ls
META-INF          display-item.jsp          layout                statistics
WEB-INF           display-statistics.jsp   login                 styles.css
browse            dSPACE-admin            mydSPACE              submit
collection-home.jsp error                    print.css             suggest
community-home.jsp favicon.ico              register              tombstone.jsp
community-list.jsp feedback                requestItem           tools
components        help                    robots.txt            utils.js
controlledvocabulary home.jsp                search                workspace
css                image                  sherpa
discovery          index.jsp                static
root@rl-qed:/dSPACE/webapps/jspui# cd layout/
root@rl-qed:/dSPACE/webapps/jspui/layout# ls
footer-default.jsp          legacy
footer-popup.jsp           location-bar.jsp
footer-submission.jsp      navbar-admin.jsp
google-analytics-snippet.jsp navbar-community-or-collection-admin.jsp
header-default.jsp         navbar-default.jsp
header-popup.jsp           navbar-minimal.jsp
header-submission.jsp
root@rl-qed:/dSPACE/webapps/jspui/layout#
```



Desde la carpeta **layout** podemos acceder a las etapas del diseño del sistema.

Ejemplo:

```
router-default.jsp
```

Encabezado del sistema

Para el ingreso a la página del desarrollo se debe ejecutar los comandos:

Ruta de acceso

```
# sudo nano footer-default.jsp
```

```
root@rl-qed: /dSPACE/webapps/jspui/layout
GNU nano 2.9.3 footer-default.jsp
%--
The contents of this file are subject to the license and copyright
detailed in the LICENSE and NOTICE files at the root of the source
tree and available online at

http://www.dSPACE.org/license/

--%>
<%>
<%--
- Footer for home page
--%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>

<%@ page contentType="text/html;charset=UTF-8" %>

<%@ page import="java.net.URLEncoder" %>
<%@ page import="org.dSPACE.app.webui.util.UIUtil" %>

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell
```

Aplique los conocimientos en JavaScript para desarrollar, modificar y mejorar el sistema.