

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



PROYECTO DE GRADO

**“SISTEMA ACADÉMICO INTEGRAL PARA EL INSTITUTO
TECNOLÓGICO AYACUCHO”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: EDDY RICHARD AGNE QUISPE
TUTOR METODOLÓGICO: M. Sc. ALDO RAMIRO VALDEZ ALVARADO
ASESOR: M. Sc. JOSÉ MARÍA TAPIA BALTAZAR

LA PAZ – BOLIVIA
2015



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



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria:

El presente proyecto está dedicado a Dios y al señor Jesucristo, los cuales son mi razón de mi existencia, por darme siempre fortaleza, y estar siempre conmigo, por permitirme alcanzar una de mis metas.

A mi padre Heriberto Agne, por ser un padre ejemplar, dándome consejos y por hacer que jamás nada nos hiciera falta.

A mi madre Irene Quispe por su amor, cuidado apoyo y comprensión que me brindaba.

A mis hermanas Silvia y Jimena con quienes pase momentos inolvidables y apoyándome siempre que las necesitaba

Con mucho cariño

Eddy Agne

Agradecimientos:

A mi docente y tutor metodológico M. Sc. Aldo Ramiro Valdez Alvarado por su valiosa colaboración, paciencia, por ser explícito como pocos al corregir algún detalle, por su valioso tiempo, conocimiento y consejo muchas gracias de verdad.

A mi docente y asesor M. Sc. José María Tapia Baltazar por su impulso y comprensión en el desarrollo del presente trabajo, muchas gracias.

Al Lic. Rolando Huarachi Quispe, por su gran ayuda y colaboración y sus ánimos a continuar en la realización del presente proyecto

A mi esposa por la paciencia comprensión y ser de gran ayuda para la realización de mi proyecto.

A mi hijo Ademar Joel quien a cada instante fue mi inspiración para continuar.

A los distintos docente de la carrera de Informática los cuales fueron partícipes de mi formación académica.

A las amistades que siempre me brindaron su amistad y apoyo, animándome para seguir adelante.

A todos los mencionados anteriormente de corazón mis más sinceros agradecimientos

RESUMEN

En esta nueva era de revolución de las tecnologías de información, los sistemas que realizan el tratamiento de información se van haciendo más necesarios e imprescindibles para toda Institución, es así que Instituciones dedicadas al área de la educación superior, manejan cada vez más grandes volúmenes de información, tal es el caso del Instituto Tecnológico Ayacucho, encargado de la formación de profesionales a nivel de Técnico Medio y Técnico Superior, que debe adecuarse a estas nuevas tecnologías para mejorar sus servicios.

En las últimas gestiones la cantidad de estudiantes del Instituto Tecnológico Ayacucho ha tenido un crecimiento considerable, esto debido a la gran demanda que esta Institución tiene, actualmente el Instituto Tecnológico Ayacucho no cuenta con un Sistema de Información que ayude a realizar procesos fundamentales del área académica como ser: Registro de Estudiantes Nuevos, Matriculación, Inscripción, Registro y Control de notas, Seguimiento Académico, Emisión de Documentos de estudiantes, entre otros, estos procesos son realizados en forma manual produciendo demora en la realización de las diferentes labores que realiza el personal administrativo del Área Académica, sin embargo en el presente proyecto se pretende afrontar los inconvenientes anteriormente mencionados implementando el Sistema Web Académico Integral para el Instituto Ayacucho, que coadyuve en los procesos del área académica, para proporcionar información confiable, segura y oportuna, disminuyendo el trabajo excesivo del personal administrativo, beneficiando a la Institución, realizando su prestigio y proporcionando credibilidad en el manejo de la información de sus estudiantes, también beneficiará al personal administrativo, ayudando en los procesos del área académica, reduciendo el trabajo excesivo y moroso, para una mejor atención a la comunidad normalista y usuarios en general.

ÍNDICE

1. MARCO INTRODUCTORIO

1.1.INTRODUCCIÓN.....	1
1.2.ANTECEDENTES.....	2
1.2.1. ANTECEDENTES DE LA INSTITUCIÓN.....	2
1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES.....	6
1.2.3. FORMULACIÓN DEL PROBLEMA.....	7
1.3.DEFINICIÓN DE OBJETIVOS.....	7
1.3.1. OBJETIVO GENERAL.....	7
1.3.2. OBJETIVOS ESPECÍFICOS.....	7
1.4.JUSTIFICACIÓN.....	8
1.4.1. JUSTIFICACIÓN TÉCNICA.....	8
1.4.2. JUSTIFICACIÓN ECONÓMICA.....	9
1.4.3. JUSTIFICACIÓN SOCIAL.....	10
1.4.4. ALCANCES Y LIMITES.....	10
1.4.5. METODOLOGÍAS.....	12

2. MARCO TEÓRICO

2.1.INTRODUCCIÓN.....	16
2.2.INGENIERÍA DE SOFTWARE.....	16
2.2.1. DEFINICIÓN DE INGENIERÍA DE SOFTWARE.....	16
2.2.2. CAPAS DE LA INGENIERÍA DE SOFTWARE.....	17
2.2.3. MODELOS DEL PROCESO DE SOFTWARE.....	18
2.2.3.1.MODELO ITERATIVO.....	18
2.2.3.2.MODELO EVOLUTIVO.....	19
2.2.3.3.MODELO CASCADA.....	19
2.2.3.4.MODELO EN V.....	20
2.2.3.5.MODELO DE DESARROLLO INCREMENTAL.....	20
2.2.3.6.MODELO DE PROTOTIPOS.....	21
2.3.METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	21
2.3.1. METODOLOGÍAS AGILES.....	22
2.3.2. PROCESO UNIFICADO ÁGIL (AUP).....	23
2.3.3. PRINCIPIOS DEL AUP.....	24

2.3.4. CARACTERÍSTICAS DE LA AUP.....	25
2.3.5. FASES DEL AUP.....	26
2.3.5.1.PRIMERA FASE: CONCEPCIÓN.....	26
2.3.5.2.SEGUNDA FASE: ELABORACIÓN.....	27
2.3.5.3.TERCERA FASE: CONSTRUCCIÓN.....	27
2.3.5.4.CUARTA FASE: TRANSICIÓN.....	28
2.3.6. DISCIPLINAS DEL AUP.....	28
2.3.6.1.DISCIPLINA DEL MODELADO.....	28
2.3.6.2.DISCIPLINA DE LA IMPLEMENTACIÓN.....	31
2.3.6.3.DISCIPLINA DE PRUEBA.....	32
2.3.6.4.DISCIPLINA DE DESPLIEGUE.....	34
2.3.6.5.DISCIPLINA DE ADMINISTRACIÓN DE LA CONFIGURACIÓN.....	37
2.3.6.6.DISCIPLINA DE LA ADMINISTRACIÓN DEL PROYECTO.....	38
2.3.6.7.DISCIPLINA DEL AMBIENTE.....	40
2.3.6.8.ITERACIONES E INCREMENTOS DE TIEMPO.....	41
2.3.6.9.ROLES DEL EQUIPO.....	42
2.3.6.10. ENTREGABLES.....	44
2.4.INGENIERÍA WEB.....	47
2.4.1. APLICACIONES WEB.....	47
2.4.2. MODELO DE PROCESO DE INGENIERÍA WEB.....	48
2.5.METODOLOGÍA DE MODELADO WEB UWE.....	51
2.5.1. ACTIVIDADES DE MODELADO DE UWE.....	52
2.5.2. ETAPAS O SUB-MODELOS DE UWE.....	63
2.5.3. FASES Y ETAPAS DE LA UWE.....	64
2.6.MODELO VISTA CONTROLADOR.....	65
3. MARCO APLICATIVO	
3.1.INTRODUCCIÓN.....	67
3.2.FASE DE INICIO.....	69
3.2.1. MODELADO DEL NEGOCIO.....	69
3.2.1.1.MODELADO DE CASOS DE USO DEL NEGOCIO.....	69
3.2.1.2.DESCRIPCIÓN DE ACTORES DEL CASO DE USO DE NEGOCIO....	71
3.2.2. MODELO DE REQUERIMIENTOS.....	72

3.2.2.1.DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL DE NEGOCIOS ...	72
3.2.2.2.DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL DE USUARIO.....	72
3.2.2.3.DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL DE SISTEMA.....	72
3.2.2.4.DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL TÉCNICOS.....	73
3.3.FASE DE ELABORACIÓN.....	73
3.3.1. MODELO DE ANÁLISIS.....	73
3.3.1.1.ANÁLISIS DE REQUISITOS.....	73
3.3.2. MODELO DE DISEÑO.....	88
3.3.2.1.DISEÑO	88
CONCEPTUAL.....	92
3.3.2.2.DISEÑO NAVEGACIONAL.....	93
3.4.FASE DE CONSTRUCCIÓN.....	94
3.4.1. DISEÑO DE INTERFACES.....	94
3.4.2. DISEÑO DE PRESENTACIÓN.....	103
3.5.FASE DE TRANSICIÓN.....	103
3.5.1. IMPLEMENTACIÓN.....	103
3.5.1.1.PRUEBAS DEL SISTEMA.....	104
3.5.1.2.PRUEBAS DE ESTRÉS.....	107
3.5.1.3.PRUEBAS DE SEGURIDAD.....	
4. CALIDAD DE SOFTWARE	
4.1.INTRODUCCIÓN.....	109
4.2.CARACTERÍSTICAS PROPUESTAS POR ISO-9126.....	109
4.2.1. CONFIABILIDAD.....	111
4.2.2. FUNCIONALIDAD.....	113
4.2.2.1.NUMERO DE ENTRADAS DE USUARIO.....	114
4.2.2.2.NUMERO DE SALIDAS DE USUARIO.....	114
4.2.2.3.NÚMERO DE PETICIONES DE USUARIO.....	115
4.2.2.4.NÚMERO DE ARCHIVOS.....	115
4.2.2.5.NUMERO DE INTERFACES EXTERNAS.....	116
4.2.2.6.PONDERACIÓN.....	116
4.2.3. MANTENIBILIDAD.....	119
4.2.4. PORTABILIDAD.....	120
4.2.4.1.NIVEL DE APLICACIONES.....	121

4.2.4.2.NIVEL DEL SISTEMA OPERATIVO.....	121
4.2.4.3.NIVEL DE HARDWARE.....	121
4.2.5. USABILIDAD.....	121
4.2.5.1.LA COMPLEJIDAD DE LA DESCRIPCIÓN.....	121
4.2.5.2.CONSISTENCIA OPERACIONAL.....	122
4.2.5.3.CONSISTENCIA OPERACIONAL EN EL USO.....	122
4.2.5.4.RESULTADO FINAL.....	123
5. PRUEBAS Y SEGURIDAD	
5.1.INTRODUCCIÓN.....	124
5.2.PRUEBAS DEL DISEÑO ARQUITECTÓNICO.....	124
5.3.SEGURIDAD.....	125
5.3.1. SEGURIDAD FÍSICA.....	125
5.3.2. SEGURIDAD DE BASE DE DATOS.....	127
5.3.3. SEGURIDAD EN LA WEB.....	128
6. ANÁLISIS DE COSTO/BENEFICIO	
6.1.INTRODUCCIÓN.....	129
6.2.CARACTERÍSTICAS PROPUESTAS POR ISO-9126.....	129
6.3.ANÁLISIS DE BENEFICIOS.....	133
7. CONCLUSIONES Y RECOMENDACIONES	
7.1.CONCLUSIONES.....	136
7.2.RECOMENDACIONES.....	136

ÍNDICE DE FIGURAS

Figura 1.1. Estructura organizacional I.T.A.....	3
Figura 2.1. Capas de la ingeniería de software.....	18
Figura 2.2. Modelo iterativo.....	18
Figura 2.3. Modelo evolutivo.....	19
Figura 2.4. Modelo cascada.....	19
Figura 2.5. Modelo en v.....	20
Figura 2.6. Modelo en incremental.....	20
Figura 2.7. Modelo de prototipos.....	21
Figura 2.8. Fases y Disciplinas del AUP.....	24
Figura 2.9. Dirigido por Casos de Uso.....	25
Figura 2.10. Dinámica del proceso.....	26
Figura 2.11. Desarrollo Iterativo.....	26
Figura 2.12. Flujo de trabajo de la Disciplina del modelado.....	29
Figura 2.13. Flujo de Trabajo de la Disciplina de la implementación.....	31
Figura 2.14 Flujo de Trabajo de la disciplina de Prueba.....	33
Figura 2.15. Flujo de trabajo de la disciplina de Despliegue.....	35
Figura 2.16. Flujo de trabajo de la disciplina de Administración de la Configuración.....	37
Figura 2.17. Flujo de trabajo de la disciplina de Administración del Proyecto.....	38
Figura 2.18. Flujo de trabajo de la disciplina del Ambiente.....	40

Figura 2.19. Iteraciones Production Release.....	41
Figura 2.20. Actividades del modelado UWE.....	52
Figura 2.21. Casos de Uso requisitos en UWE.....	53
Figura 2.22. Modelo de contenido.....	55
Figura 2.23. Estereotipos de estructura de navegación.....	56
Figura 2.24. Clases de navegación.	57
Figura 2.25. Página de presentación: inicio.	59
Figura 2.25. Página de presentación: búsqueda.	60
Figura 2.26. Estructura del proceso.	62
Figura 2.27. Flujo del proceso: inicio.	62
Figura 2.28. Flujo del proceso: buscar.....	63
Figura 2.29. Etapas del modelado UWE.....	65
Figura 3.1. Modelos AUP y UWE.....	67
Figura 3.2. Combinación de modelos AUP y UWE.....	68
Figura 3.3: Diagrama de casos de uso del negocio.....	70
Figura 3.4. Diagrama de casos de uso del sistema de registro y administración.....	74
Figura 3.5. Casos de uso de administrar usuario.....	75
Figura 3.6. Diagrama de casos de uso de administrar estudiantes.....	77
Figura 3.7. Diagrama de casos de uso administrar docente.....	78
Figura 3.8. Diagrama de casos de uso administrar historial.....	80
Figura 3.9. Casos de uso de reportes.....	81

Figura 3.10 Diagrama de paquetes del sistema de registro y administración de documentos.....	83
Fuente 3.11. Diagrama de despliegue.....	84
Figura 3.12. Diagrama de secuencia: autenticación.....	85
Figura 3.13. Diagrama de secuencia: administración usuarios.....	85
Figura 3.14. Diagrama de secuencia: administrar estudiante.....	86
Figura 3.15. Diagrama de secuencia: administración historial.....	87
Figura 3.16. Diagrama de secuencia: elaboración de reportes.....	88
Figura 3.17. Diagrama de clases del sistema.....	89
Figura 3.18. Modelo entidad relación.....	90
Figura 3.19. Modelo físico de base de datos.....	91
Figura 3.20. Modelo de espacio de navegación.....	92
Figura 3.21. Modelo de navegación acceso de usuarios.....	93
Figura 3.22. Diseño de presentación.....	94
Figura 3.23. Autenticación del sistema.....	95
Figura 3.24. Pantalla de inicio del sistema.....	96
Figura 3.25. Pantalla administración de usuario.....	97
Figura 3.26. Pantalla registro de usuario.....	98
Figura 3.27. Modificación de usuario.....	99
Figura 3.28. Eliminación de usuario.....	100
Figura 3.29. Búsqueda y modificación de historial.....	101

Figura 3.30. Reporte historial.....	102
Figura 3.31. Prueba de caja negra.....	104
Figura 3.31. Prueba de caja negra.....	105
Figura 3.33. Apache jmeter: grafico de barras.....	106
Figura 4.1. Norma ISO-9126.....	110
Figura 4.2. Modelo del Sistema Web Académico Integral.....	111
Figura 5.1. Grafio del sistema.....	124

ÍNDICE DE TABLAS

Tabla 2.1. Convergencia y Divergencia entre las principales metodologías ágiles.....	23
Tabla 2.2. Fases de la disciplina del modelado.....	30
Tabla 2.3. Fases de la disciplina de implementación.....	32
Tabla 2.4. Fases de la disciplina de prueba.....	34
Tabla 2.5. Fases de la disciplina de despliegue.....	35
Tabla 2.6. Fases de la disciplina de Administración de la Configuración.....	37
Tabla 2.7. Fases de la disciplina de Administración del Proyecto.....	39
Tabla 2.8. Fases de la disciplina del ambiente.....	40
Tabla 2.9. Descripción de roles del equipo.....	42
Tabla 2.10. Productos mínimos a entrega.....	45
Tabla 2.11. Requisitos de cada propuesta.....	50
Tabla 2.12. Orientación de cada propuesta.....	50
Tabla 3.1. Descripción de actores de casos de uso del negocio.....	71
Tabla 3.2. Descripción de caos de uso administrar usuario.....	76
Tabla 3.3. Descripción de casos de uso administrar Estudiantes.....	78
Tabla 3.4. Descripción de casos de uso administrar Docente.....	79
Tabla 3.5. Descripción de casos de uso registrar historial.....	81
Tabla 3.6 Descripción de caos de uso de reportes.....	82
Tabla 4.1: Calculo de Confiabilidad	112
Tabla 4.2. Entrada de Usuarios.....	114

Tabla 4.3. Salidas de usuario.....	114
Tabla 4.4. Petición de Usuario.....	115
Tabla 4.5. Archivos.....	115
Tabla 4.6. Interfaces Externas.....	116
Tabla 4.7. Cálculos de puntos de Función no ajustados.....	116
Tabla 4.8. Ponderaciones.....	117
Tabla 4.9. Valores de Ajuste de Complejidad.....	117
Tabla 4.10. Resultado total.....	123
Tabla 5.1. Calculo de punto función.....	129
Tabla 5.2. Calculo de punto de función ajustada.....	130
Tabla 5.3. Conversión de puntos de función.	131
Tabla 5.4. Relación de valores del modelo COCOMO.....	132

1.1.INTRODUCCIÓN

El mundo de la Internet y la tecnología, son posiblemente los progresos más importantes en la historia de la computación. Estas tecnologías han llevado a todos (con cientos de millones más que eventualmente seguirán) a la era de la informática; además, se han convertido en parte integral de la vida diaria en la primera década del siglo XXI. [PRESSMAN, 2005]

En esta nueva era de las tecnologías de información, los sistemas que realizan el tratamiento de información se van haciendo más necesarios e imprescindibles, más aún si estos están interconectados vía intranet o Internet, ya que facilitan la comunicación y la integración de los usuarios, brindando la capacidad de compartir dinámicamente los recursos, disminuyendo el trabajo redundante y optimizando el flujo de la información.

Hoy en día la información es un recurso muy importante para toda Institución que trabaja con cantidades de datos enormes; es así que Instituciones dedicadas al área de la enseñanza, manejan cada vez más grandes volúmenes de información, ya que la demanda de alumnos que ingresan a estas instituciones va creciendo continuamente, tal es el caso del **“INSTITUTO TECNOLÓGICO AYACUCHO”**, encargado de la formación de profesionales a nivel técnico medio y técnico superior, que actualmente cuenta con una cantidad considerable de estudiantes, en las distintas carreras; la información de todas estas personas es manipulada y procesada en forma manual, tanto en el registro de estudiantes nuevos, matriculación e inscripción de estudiantes regulares, registro y control de notas y la emisión de documentos (matricula, boleta de inscripción, certificado de notas, historial de notas y otros) que son realizadas al inicio y culminación de cada gestión. Al ser un Instituto de Educación Superior, los procesos del área académica, deben funcionar eficientemente, para asegurar el prestigio y el éxito de dicha entidad, cabe resaltar que las inscripciones y matriculaciones son semestrales y anuales, por lo que cada gestión se tropieza con procesos manuales que retardan el proceso de inscripción y matriculación de estudiantes, ocasionando grandes filas y mucho trabajo para el personal administrativo. Por otra parte los documentos que se exigen producen una gran cantidad de papeles que sencillamente se podrían suprimir

sistematizando y almacenando esta información en una Base de Datos centralizada, que proporcione información eficiente, precisa y en el menor tiempo posible.

En el presente proyecto se pretende afrontar los inconvenientes anteriormente mencionados implementando el **“SISTEMA WEB ACADÉMICO INTEGRAL PARA EL INSTITUTO TECNOLÓGICO AYACUCHO”**, que coadyuve en los procesos del área académica, para proporcionar información eficiente, segura y oportuna, disminuyendo así el trabajo excesivo del personal administrativo del área académica.

1.2.ANTECEDENTES

1.2.1.ANTECEDENTES DE LA INSTITUCIÓN

El Instituto Tecnológico “Ayacucho” inicio sus actividades bajo la Resolución Ministerial No 960/1981, el Instituto cuenta con una cantidad considerable de estudiantes de nivel técnico medio y técnico superior, actualmente se encuentra ubicado en la Av. Busch esquina plaza Villarroel zona Miraflores, dirigida por el Rector de la institución Lic. José A. Apala Bello, dicha institución tiene como misión formar jóvenes y señoritas profesionales en la modalidad comercial e industrial en los niveles técnico medio y técnico superior, idóneos de reconocida excelencia académica con valores éticos para su incorporación en la actividad social productiva del país en base a una estructura curricular. El instituto tiene como visión el ser un centro de formación profesional que logre liderazgo y reconocimiento nacional a partir de una educación personalizada Técnica y Tecnológica, para la innovación productiva integral.

La Unidad de Archivo y Kardex, según el Manual de Funciones, está encargada de organizar y controlar el Archivo y Kardex a fin de dar celeridad y fluidez a los diferentes requerimientos de parte de los estudiantes, registra y archiva los datos de las actas oficiales de cada gestión académica, planifica las inscripciones de alumnos nuevos y antiguos, atiende a los estudiantes en la recepción de consultas académicas. Sin embargo, actualmente el **INSTITUTO TECNOLÓGICO AYACUCHO**, no cuenta con un Sistema de Información que ayude a realizar procesos fundamentales como ser: Registro de Estudiantes Nuevos, Matriculación, Inscripción, Registro y Control de notas, Emisión de Documentos del

estudiantes entre otros, todas estas tareas son realizadas en forma manual por el personal administrativo de esta Unidad.

La siguiente figura muestra la estructura organizacional del instituto (ver figura 1)

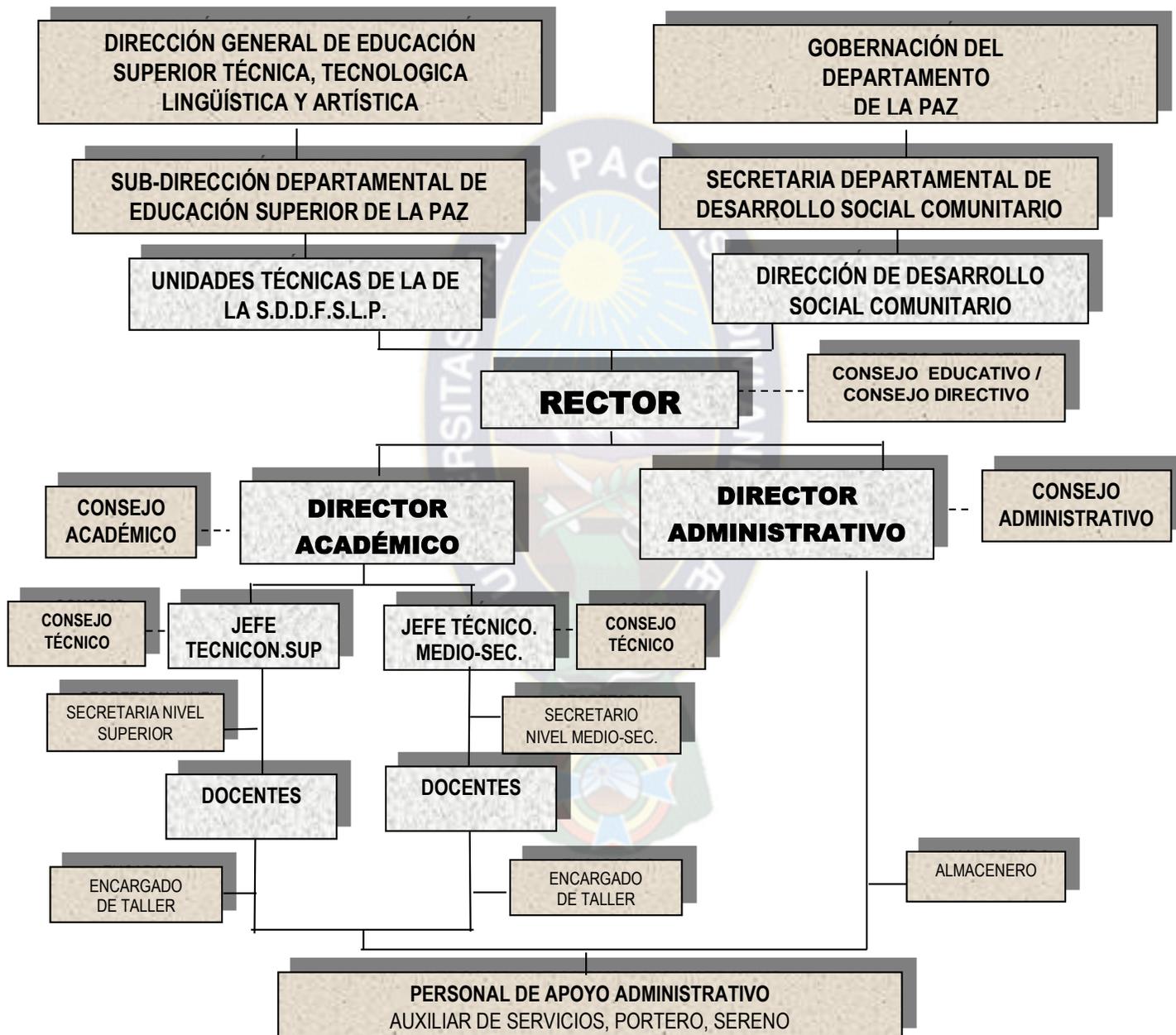


FIGURA 1.1. Estructura organizacional I.T.A.
FUENTE: Apala, 2015

De acuerdo a la descripción del punto anterior, el incremento de la población estudiantil del INSTITUTO TECNOLÓGICO AYACUCHO (ITA) hace que la información sea excesiva y difícil de manejarla adecuadamente, además de que este arduo trabajo se realiza de forma manual y está concentrado en pocas personas, haciendo dificultosa y morosa estas tareas.

Actualmente la institución no existe un sistema de información para los procesos en el área académica, produciéndose volúmenes considerables de información, ocasionando procesos burocráticos para reducir el trabajo en la parte administrativa lo cual hace notar que en ciertas actividades como ser: inscripciones, calificaciones, record académico entre Otros, el trabajo se acumula y hasta resulta caótico lo que lleva que las tareas no se realicen de manera ordenada e eficiente. (Información Rector I.T.A. Lic. José Apala)

A continuación se presenta procesos realizados manualmente referentes al seguimiento académico:

- Inscripción de estudiantes de las diferentes carreras, el proceso se realiza manualmente mediante el llenado de formularios de inscripción por los estudiantes, posteriormente la secretaria se encarga de ordenar los mismos por materias o semestres.
- Inscripción de los estudiantes a las materias y paralelos respectivos, que implica largas filas para el registro de los mismos.
- Registro de calificaciones de los estudiantes a las diferentes materias de sus carreras.
- Seguimiento académico de los estudiantes, incluye materias de las gestiones correspondientes.

Los problemas que conlleva cada proceso son los siguientes:

- **Existencia de un historial académico y de un historial de docentes**, lo cual dificulta en la obtención de datos personales del estudiante para realizar sus respectivos papeleos e informes, en el caso de docentes la inexistencia del mismo dificulta ubicarlo para realizar determinada consulta, más aun si existe problemas con la perdida de notas de estudiante.

- **Volúmenes considerables de documentación**, los procesos de inscripción a estudiantes, inscripción de materias, registro de calificaciones y otros, acumulan bastante información tangible lo que dificulta para su obtención de manera rápida.
- **Demora en la búsqueda de calificaciones de un estudiante**, se tiene una considerable pérdida de tiempo de búsqueda de sus respectivas calificaciones para la obtención de sus calificaciones de notas como ser su record académico.
- **Atención no apropiada a los estudiantes**, a principio y final de cada semestre, los estudiantes realizan sus inscripciones, solicitud de sus certificados de notas, etc. Los mismos realizan largas filas y deben esperar un tiempo considerable para que se les atienda, por tratarse de procesos manualmente y varios de ellos son realizados al mismo tiempo, considerando también que los documentos están en diferentes archiveros.
- **Demora en la emisión de reportes**, la demora en la búsqueda de información ocasiona que la información no sea oportuna.
- **Listado incorrecto de estudiantes**, se inicia cuando a principio de cada semestre algo más de la mitad de los estudiantes se registra en las boletas de inscripción cancelando algún monto, y la otra cantidad de estudiantes se registra directamente con su respectivo docente sin haber llenado las boletas de inscripción y sin cancelar ningún monto.
- **Inexistencia de copias de seguridad**, los registros de información como ser: calificaciones, kardex de docentes, boletas de inscripción y otros. No tienen sus respectivas copias de seguridad, lo que ocasiona problemas cuando alguno de estos registros desaparece.
- **Desorganización**, en los procesos referentes al seguimiento académico, existe actualmente una confusión en cuanto al manual de funciones de cada administrativo lo que implica desorden en el momento de desempeñar sus funciones.

Gran parte de los problemas existentes se deben a la extensa información y a su tratamiento, ocasionando en los administrativos de la institución un desorden cuanto a las funciones que se desempeñan, no ayudando el que las actividades se desarrollan en forma manual, donde los datos son recopilados y transcritos por empleados de la institución convirtiéndose en procesos tediosos, monótonos y rutinarios, que no ayudan a su desempeño.

1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES

Muchos postulantes realizaron Proyectos de Grado similares al que se está desarrollando, sin embargo cada una presenta características particulares y tienen una aplicación específica ya que fueron desarrollados según los requerimientos de una Institución particular. A continuación se describen los siguientes proyectos:

A. SISTEMAS DESARROLLADOS PARA UNIVERSIDADES BOLIVIANAS

“SISTEMA DE SEGUIMIENTO ACADÉMICO UNIVERSIDAD TECNOLÓGICA BOLIVIANA (U T B)”, del Postulante: Nelson Egberto Tarqui Carpio, cuyo objetivo fue analizar, diseñar, desarrollar e implementar un Sistema de Seguimiento Académico para la U.T.B. que facilite el manejo eficiente y oportuno de la información, para esto desarrollo los módulos de Registro de alumnos nuevos, Inscripciones, Registro y control de notas, Emisión de Historial Académico, Emisión de Certificado de Notas, Registro de Egresados y Reportes. La metodología que planteó para el análisis y desarrollo del Sistema es el Análisis y Diseño Estructurado Moderno de Yourdon, para la programación hizo uso de herramientas visuales. [TARQUI, 2003]

B. SISTEMAS DESARROLLADOS PARA FACULTADES Y CARRERAS DE UNIVERSIDADES

“SISTEMA DE GESTIÓN ACADÉMICA UNIVERSITARIA CASO: FACULTAD DE INGENIERÍA”, de la Postulante: Rocio de la Azucena Sirpa Caceres, cuyo objetivo fue mejorar la gestión de la información en los procesos académicos de la Facultad de Ingeniería, adaptando, desarrollando e implementando el Sistema de Gestión Académica Universitaria en las carreras de Ingeniería Eléctrica, Electromecánica mención Eléctrica, Metalurgia y de

Materiales, Civil, Petrolera y en el departamento de Cursos Básicos. Para tal efecto desarrollo los módulos de Admisión y Matriculación, Planificación de la Gestión Académica, Inscripción, Seguimiento Académico y el acceso seguro, confidencial y restringido a cada usuario. Adoptó la Metodología del Proceso Unificado de Desarrollo de Software, utilizando el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema software. Esta desarrollado en el lenguaje de programación JAVA, su Gestor de Base de Datos es PostgreSQL, y opera en servidores LINUX. [SIRPA, 2005].

C. SISTEMA DESARROLLADO PARA LA CARRERA DE INFORMÁTICA

"SISTEMA DE INFORMACIÓN Y GESTIÓN ACADÉMICA SIGA", del Lic. Mg.Sc. Franz Cuevas Quiroz, desarrollado en el Instituto de Investigación de Informática, el cual realiza el registro de estudiantes nuevos, inscripciones con el control de prerrequisitos, registro de notas. Emite reportes de: historial académico, materias a cursar, materias inscritas y número aleatorio para inscripciones. Cabe mencionar que este sistema estuvo en funcionamiento en la carrera de Informática.

1.2.3. FORMULACIÓN DEL PROBLEMA

¿Cómo se puede mejorar el control en los procesos de Registro de estudiantes nuevos, inscripción, registro y control de notas, seguimiento académico, emisión de documentos, reportes y otros proporcionando información eficiente y oportuna en el Instituto Tecnológico Ayacucho?

1.3.DEFINICIÓN DE OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar e Implementar el **SISTEMA WEB ACADÉMICO WEB INTEGRAL PARA EL INSTITUTO TECNOLÓGICO AYACUCHO**, que coadyuvará en los procesos del área académica, para proporcionar información eficiente y oportuna.

1.3.2. OBJETIVOS ESPECÍFICOS

- Gestionar en forma automática los procesos académicos que permitan agilizar las tareas realizadas en el Área Académica del Instituto Tecnológico Ayacucho.

- Realizar el registro de estudiantes nuevos a la base de datos y emitir la Hoja de registro.
- Realizar la inscripción e impresión de la Boleta de Inscripción de los estudiantes.
- Implementar el módulo Registro y Control de notas que permita realizar el registro de Notas a docentes de manera rápida y confiable.
- Implementar el módulo de Consulta de estudiantes para que realicen su seguimiento académico y puedan acceder a información como ser: materias inscritas, notas del último periodo, historial de notas, pensum y registro personal.
- Emitir documentos: certificado de notas, historial de notas globales, listas oficiales y boletines de notas de forma automática, rápida y confiable.
- Emitir reportes y cuadros comparativos (detalle de estudiantes por especialidad, sexo, edad, idioma, tipo de colegio, etc.) para ayudar a la toma de decisiones a nivel ejecutivo.
- Capacitar al personal encargado que trabajará directamente con el Sistema de Información.

1.4. JUSTIFICACIÓN

1.4.1. JUSTIFICACIÓN TÉCNICA

El continuo desarrollo tecnológico de la ciencia de las computadoras ha permitido masificar el uso de esta herramienta, incrementando cada vez más el manejo de grandes cantidades de información, dando lugar a que Instituciones dedicadas al área de la educación superior, tal es el caso del I.T.A., deban adecuarse a estas nuevas tecnologías para mejorar sus servicios, eliminando los procesos manuales que conducen a la demora en la realización de las diferentes labores que realiza el personal administrativo del Área Académica.

En cuanto a recursos de Hardware y Software, la institución cuenta con equipos suficientes y una red local instalada donde se puede implementar el sistema de información, cuenta con aproximadamente 30 computadoras en los diferentes departamentos (Dirección General, Dirección Académica, Centro Informático, kardex) interconectadas mediante una red de área

local a un servidor ubicado en el Centro informático, también cuenta con un Laboratorio de Computación con aproximadamente 25 computadoras, la mayoría de estos equipos son i3 de tercera generación; en cuanto a Software cuentan con la Licencia del sistema operativo Windows Seven, y utilizan Microsoft Excel 2010 para almacenar los registros de los estudiantes.

Para el desarrollo del Sistema de Información se utilizará como lenguaje de programación PHP , MySQL como Gestor de Base de Datos y Apache como Servidor Web, debido a los muchos beneficios que ofrecen además que son de uso gratuito y no habrá algún costo extra en cuanto a licencias se refiere.

La tecnología Cliente/Servidor utilizada es la topología estrella, para compartir o distribuir la información de la Base de Datos que esté disponible en cualquier computador conectado a la red, utilizando medidas de seguridad para que esta información pueda ser accedida solo por usuarios autorizados, además de usar una interfaz fácil de operar por cualquier usuario.

1.4.2. JUSTIFICACIÓN ECONÓMICA

Actualmente la información es un recurso muy importante para toda Institución que maneja grandes volúmenes de datos, creemos que todas estas entidades deben automatizar su información, ya que esto implica un ahorro de tiempo considerable, eliminando trabajo excesivo, minimizando el uso de material de escritorio, mejorando la productividad y reduciendo los costos relacionados con la recopilación y distribución de información.

Por otra parte el Sistema de Información Integrado se desarrollará utilizando Software Libre, lo que implica total independencia en cuestión a Licencias, es decir el lenguaje de programación (PHP), el gestor de base de datos (MySQL) y el servidor (Apache) son de uso gratuito, no requieren de licencias y no tienen costo alguno. A esto podemos añadir además la plataforma de Windows Seven, la cual cuenta con la Licencia correspondiente, adquirida por el Instituto Tecnológico Ayacucho. Además el I.T.A. cuenta con la instalación de una red local lo cual implica la facilidad en cuanto a la instalación y configuración del Sistema en la Institución.

1.4.3. JUSTIFICACIÓN SOCIAL

En las últimas gestiones la cantidad de estudiantes del I.T.A. ha tenido un crecimiento considerable, esto debido a la gran demanda que esta Institución tiene, sin embargo la información de estas personas es manipulada de forma manual, pero con la implementación del Sistema de Información se pretende beneficiar a la Institución, realizando su prestigio y proporcionando credibilidad en el manejo de la información de sus estudiantes, también beneficiará al personal administrativo, ayudando en los procesos del área académica, reduciendo el trabajo excesivo y moroso, para una mejor atención a la comunidad normalista y usuarios en general. Además beneficiará directamente al estudiante, ya que podrá realizar los diferentes procesos académicos de manera rápida y confiable, tendrá la posibilidad de realizar su seguimiento académico mediante consultas a través de Internet, evitando así las aglomeraciones y largas colas existentes en el área de Archivo Kardex.

1.5. ALCANCES Y LIMITES

El alcance del Sistema Web Académico Integral comprende los módulos que se especifican a continuación:

Módulo Registro de Estudiantes Nuevos.

- Registra semestral y anualmente toda la información de los estudiantes nuevos a la base de datos.
- Emite la Hoja de Ingreso Personal, que es archivado en el file del estudiante.
- Realiza el listado de estudiantes registrados (por gestión, carrera y especialidad).
- Incluye una búsqueda de estudiantes registrados por C.I.

Módulo Inscripción.

- Registra la inscripción semestral y anual de los estudiantes en las materias y paralelos correspondientes.
- Emite la Boleta de Inscripción.
- Realiza un listado por gestión, carrera, especialidad y paralelo.

- Realiza una búsqueda de estudiantes inscritos por gestión y C.I.

Módulo Registro y control de notas.

Módulo utilizado uno de los administradores quienes:

- Registran las notas de los estudiantes una vez culminada la gestión.
- Emiten el boletín de notas, una vez registrado las notas.
- Consultan materias a dictar por docentes con sus respectivos horarios.
- Acceden a listas oficiales de estudiantes inscritos en sus respectivas materias.

Módulo Seguimiento Académico de los estudiantes.

Este módulo será para el uso de los estudiantes, quienes podrán realizar consultas y la impresión de algunos documentos, como ser:

- Materias inscritas.
- Horario
- Notas del último periodo.
- Historial de notas.
- Hoja de ingreso personal.
- Pensum de su especialidad.

Módulo de Consultas y Reportes.

Este módulo se encargará de emitir los diferentes documentos y reportes requeridos por la Dirección General y Académica como ser:

- Cuadros comparativos de la cantidad de estudiantes matriculados por carrera, especialidad y gestión.
- Cuadros comparativos de la cantidad de estudiantes inscritos por carrera, especialidad y gestión.
- Cuadros comparativos de la cantidad de estudiantes registrados por carrera, especialidad y gestión.
- Detalle de la cantidad de estudiantes inscritos por paralelo.

- Cantidad de mujeres y varones registrados en carrera y especialidad, por gestión.
- Estado civil (solteros/as, casados/as), de estudiantes registrados por gestión.
- Edad mínima, máxima y promedio de estudiantes registrados por gestión.
- Registro del tipo de colegio (fiscal, particular, convenio) de egreso de los estudiantes, por gestión.
- Ciudad de egreso de los estudiantes registrados (La Paz, El Alto, otros), por gestión.

Módulo de Administración del Sistema.

Este módulo estará encargado de:

- Administrar el diseño académico
- Registro de materias a la base de datos.
- Elaboración del pensum.
- Asignación de docente-materia.
- Habilitación de paralelos.

Administración de usuarios

- Registro de usuarios.
- Asignación de niveles de usuario a personal encargado, (docentes y administrativos)

Administración de procesos académicos.

- Registro de estudiantes nuevos.
- Matriculación.
- Inscripción.
- Registro y control de notas.
- Seguimiento Académico.

1.6. METODOLOGÍAS

El objetivo del análisis es desarrollar en el modelo del funcionamiento del sistema que satisfaga las necesidades de la institución. El modelo se expresa en términos de objetos

relacionados, en el control dinámico de flujo y las transformaciones funcionales, teniendo en cuenta que es una metodología muy organizativa y es adaptable a cualquier tipo de proyecto.

Metodología e investigación científica

Tiene como objetivo el alcance de la efectiva técnica mediante el ajuste de las ideas que van a los hechos, para lo cual utiliza las entrevistas y encuestas.

Tipo de Investigación

El tipo de investigación es descriptiva, ya que conoce la descripción, registro, análisis e interpretación del medio actual, en donde la intención es la de presentar una definición correcta del objeto de estudio.

AUP

AUP (Proceso Unificado Ágil de Scott Ambler) se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados:

- Incepción (Concepción).
- Elaboración.

- Construcción.
- Transición.

Las disciplinas se llevan a cabo de manera sistemática, y estas son:

- Modelo.
- Aplicación
- Prueba
- Despliegue
- Gestión de configuración
- Gestión de proyectos.
- Entorno

UWE

La propuesta de Ingeniería Web basada en UML (UWE (Koch, 2000)) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

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

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

Principios y aspectos de UWE

Los principales de aspectos en los que se fundamenta UWE son los siguientes:

- **Uso de una notación estándar**, para todos los modelos (UML: Lenguaje de modelado unificado).
- **Definición de métodos:** Definición de los pasos para la construcción de los diferentes modelos.
- **Especificación de Restricciones:** Se recomienda el uso de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para aumentar la exactitud de los modelos.

Fases de desarrollo de UWE

Por lo que respecta al proceso de autoría de la aplicación, UWE hace un uso exclusivo de estándares reconocidos como UML y el lenguaje de especificación de restricciones asociado OCL. Para simplificar la captura de las necesidades de las aplicaciones web, UWE propone una extensión que se utiliza a lo largo del proceso de autoría. Este proceso de autoría está dividido en cuatro pasos o actividades:

- **Análisis de Requisitos:** Fija los requisitos funcionales de la aplicación Web para reflejarlos en un modelo de casos de uso.
- **Diseño Conceptual:** Materializado en un modelo de dominio, considerando los requisitos reflejados en los casos de uso.
- **Diseño Navegacional:** Lo podemos subdividir en:
 - **Modelo del Espacio de Navegacional**
 - **Modelo de la Estructura de navegación:** Muestra la forma de navegar ante el espacio de navegación.
- **Diseño de Presentación:** Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML.

2.1.INTRODUCCIÓN

En este capítulo se da a conocer los fundamentos teóricos, para la realización del actual proyecto, donde especificaremos conceptos, métodos, metodologías, técnicas y herramientas convenientes en el desarrollo del sistema.

2.2.INGENIERÍA DE SOFTWARE

Como disciplina, la ingeniería del software ha progresado mucho en un corto periodo de tiempo. Actualmente, se construyen sistemas muy grandes en cuanto a tamaño y complejidad y el software está presente en casi todos los aspectos de la vida, a pesar del rápido progreso, todavía existen grandes problemas para conseguir proporcionar a los clientes productos software de alta calidad en los plazos establecidos. Hay muchos desafíos que hay que tratar para progresar hacia un campo de la ingeniería más maduro que permita obtener productos de alta calidad.

2.2.1.DEFINICIÓN DE INGENIERÍA DE SOFTWARE

La ingeniería del software es una disciplina de ingeniería preocupada por todos los aspectos de la producción de software desde las primeras etapas de especificación del sistema hasta el mantenimiento del sistema después de que éste se haya puesto en uso. Se preocupa de las teorías, métodos y herramientas para el desarrollo profesional de software. La ingeniería del software se preocupa del desarrollo de software rentable.

La ingeniería del software debería adoptar un enfoque sistemático y organizado para su trabajo y usar las herramientas y técnicas apropiadas dependiendo del problema a solucionar, las restricciones de desarrollo y los recursos disponibles.

El enfoque sistemático, disciplinado y cuantificable es con frecuencia calificado de modelo de proceso de software o de proceso de desarrollo de software. El proceso de desarrollo de software consiste en un conjunto particular de prácticas de desarrollo de software que son realizadas por el ingeniero de software en un orden predeterminado.

Cuando se habla de prácticas de desarrollo de software se hace referencia a un requisito empleado para recomendar un enfoque disciplinado y uniforme del proceso de desarrollo de software, es decir, una actividad bien definida que contribuye a la satisfacción de objetivos del proyecto. Entre las prácticas de desarrollo de software se encuentran las siguientes:

- Ingeniería de requisitos
- Análisis de sistemas
- Diseño/arquitectura a alto nivel
- Diseño a bajo nivel
- Codificación
- Integración
- Diseño y revisiones de código
- Pruebas
- Mantenimiento
- Gestión de proyectos
- Gestión de la configuración

La mayoría de las disciplinas reconocen algunas prácticas como mejores prácticas. Una mejor práctica es una práctica que, a través de la experiencia e investigación, se ha probado que lleva al resultado deseado fiablemente y se considera prudente y recomendable hacerla en una variedad de contextos.

2.2.2. CAPAS DE LA INGENIERÍA DE SOFTWARE

El enfoque de ingeniería del software cuenta con un compromiso organizacional con la calidad porque no es posible incorporar la ingeniería del software en una organización que no está centrada en conseguir calidad.

La ingeniería del software es una tecnología multicapa. Se puede ver como un conjunto de componentes estratificados, que reposan sobre ese enfoque de calidad.

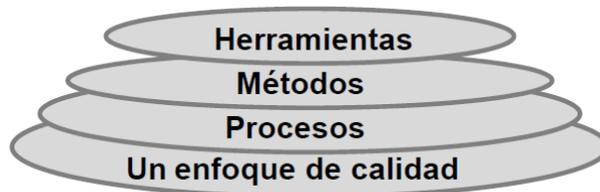


Figura 2.1. Capas de la Ingeniería de Software
Fuente: Pressman, 2005

Estos componentes que forman parte de la ingeniería del software son:

- **Procesos:** un marco de trabajo que ayuda al jefe de proyecto a controlar la gestión del proyecto y las actividades de ingeniería.
- **Métodos:** las actividades técnicas requeridas para la creación de productos de trabajo.
- **Herramientas:** la ayuda automatizada para los procesos y métodos.

2.2.3. MODELOS DEL PROCESO DE SOFTWARE

Un modelo del proceso del software es una presentación abstracta de un proceso del software. Cada modelo de proceso representa un proceso desde una perspectiva particular, y así proporciona solo información parcial sobre ese proceso.

2.2.3.1. MODELO ITERATIVO

Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien, después de cada iteración, evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente.

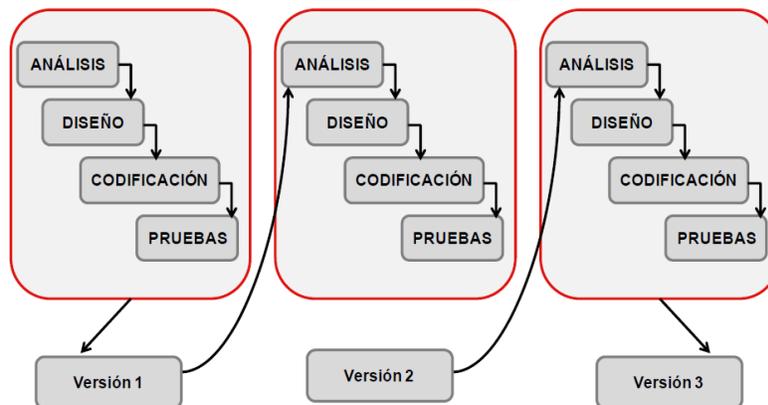


Figura 2.2. Modelo Iterativo
Fuente: Sommerville, 2005

2.2.3.2.MODELO EVOLUTIVO

La definición y especificación de requerimientos y el desarrollo de software es un proceso evolutivo que demanda la experimentación previa con algún componente (o la totalidad) el Sistema Programado (Ej. Interfaz U-S, función o subsistema) antes de desarrollar la totalidad del sistema.

Logran su objetivo por medio del desarrollo de una serie de prototipos que van evolucionando a medida que se tiene realimentación del cliente

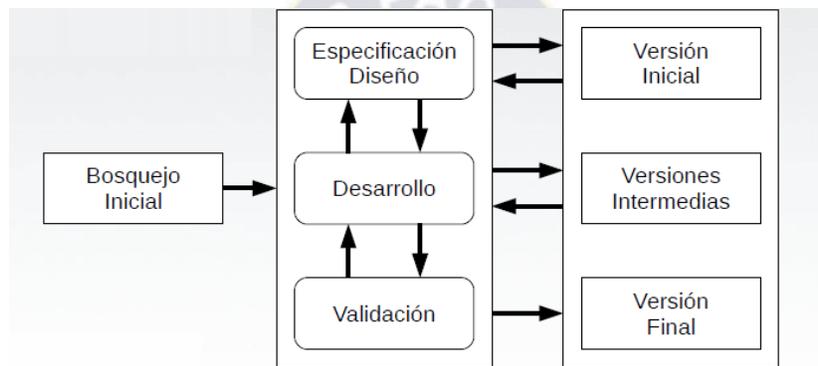


Figura 2.3. Modelo evolutivo
Fuente: Sommerville, 2005

2.2.3.3.MODELO CASCADA

El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo se va fluyendo hacia abajo (como una cascada) sobre las fases que componen el ciclo de vida.

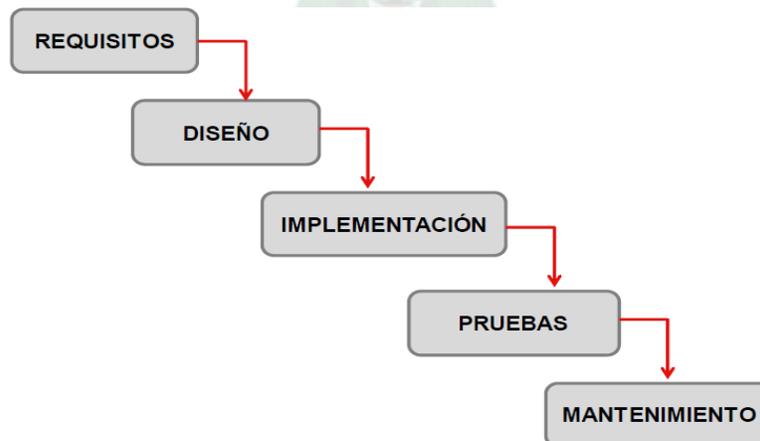


Figura 2.4. Modelo cascada
Fuente: Sommerville, 2005

2.2.3.4.MODELO EN V

El modelo en v es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Describe las actividades y resultados que han de ser producidos durante el desarrollo del producto. La parte izquierda de la v representa la descomposición de los requisitos y la creación de las especificaciones del sistema. El lado derecho de la v representa la integración de partes y su verificación. V significa “Validación y Verificación”.

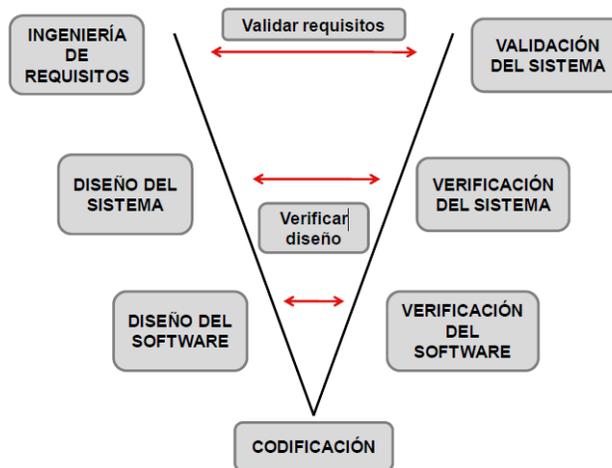


Figura 2.5. Modelo en v
Fuente: Sommerville, 2005

2.2.3.5.MODELO DE DESARROLLO INCREMENTAL

El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.

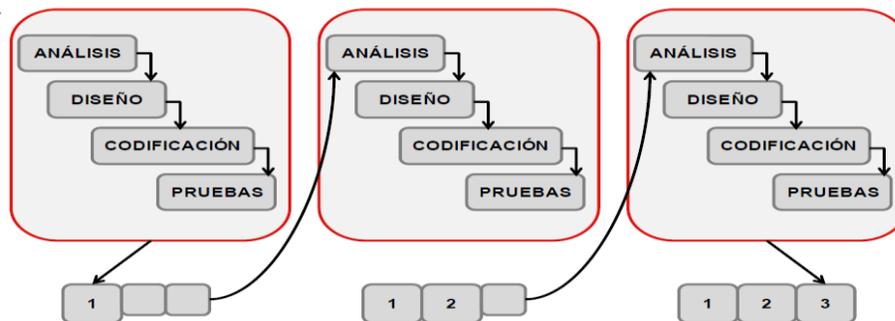


FIGURA: 2.6. Modelo en Incremental
FUENTE: Sommerville, 2005

2.2.3.6.MODELO DE PROTOTIPOS

El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar.



Figura 2.7. Modelo de prototipos
Fuente: Sommerville, 2005

Estos son algunos modelos de procesos de software que podemos utilizar en una metodología de desarrollo. Además establecer el trabajo en facetas distribuye el desarrollo de una forma ordenada lo que hace que cada uno se ocupe de su trabajo y no de aquel por el cual no le pagan.

2.3.METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software son un conjunto de instrucciones, técnicas y ayudas a la documentación para el desarrollo de productos de software. Es como un libro de recetas de cocina, en el que se van mostrando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además que personas deben participar en el desarrollo de las actividades y que papel deben desempeñar.

Se puede mencionar los métodos que están dentro de las metodologías de desarrollo de software como ser: iterativos, evolutivos y ágiles.

2.3.1. METODOLOGÍAS AGILES

Las metodologías ágiles de desarrollo de software, conocidos anteriormente como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. Las metodologías de desarrollo de software ágiles son vistas de la siguiente manera:

- El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas.
- Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación.
- Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración.
- Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos Ágiles enfatizan principalmente:

- Las comunicaciones cara a cara en vez de la documentación.
- La mayoría de los equipos Ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (bullpen en inglés).
- La oficina debe incluir revisores, diseñadores de iteración, escritores de documentación y ayuda y directores de proyecto.
- El software funcional es la primera medida del progreso.

Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.

En la siguiente tabla se puede observar la convergencia y divergencia entre las principales metodologías ágiles. (Ver Tabla 2.8)

Metodología	Acrónimo	Creación	Tipo de modelo	Característica
Adaptive Software Development	ASD	Highsmith 2000	Prácticas + ciclo de vida	Inspirado en sistemas adaptativos complejos
Agile Modeling	AM	Ambler 2002	Metodología basada en la práctica	Suministra modelado ágil a otros métodos
CrystalMethods	CM	Cockburn 1998	Familia de metodologías	Metodología ágil con énfasis en modelo de ciclos
Agile RUP	dX	Booch, Martin, Newkirk 1998	Framework/Disciplina	XP dado vuelta con artefactos RUP
Dynamic Solutions Delivery Model	DSDM	Stapleton 1997	Framework/ modelo de ciclo de vida	Creado por 16 expertos en RAD
eXtreme Programming	XP	Beck 1999	Disciplina en prácticas de ingeniería	Método ágil radical

Tabla 2.1. Convergencia y Divergencia entre las principales metodologías ágiles.
Fuente: J. Muñoz, 2000

2.3.2. PROCESO UNIFICADO ÁGIL (AUP)

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. [Ambler S, 2005]

AUP nace en 2005 en manos del desarrollador Scott W. Ambler el cual lo define de la siguiente manera: El proceso unificado ágil (AUP) es un desarrollo de programas basado en el proceso unificado racional de IBM (RUP). El ciclo vital de AUP es en serie en lo grande, iterativo en lo pequeño, entregando productos incrementales en un cierto plazo.

Es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas.

En la siguiente figura se observa las fases, disciplinas e iteraciones correspondientes. (Ver Figura 2.8).

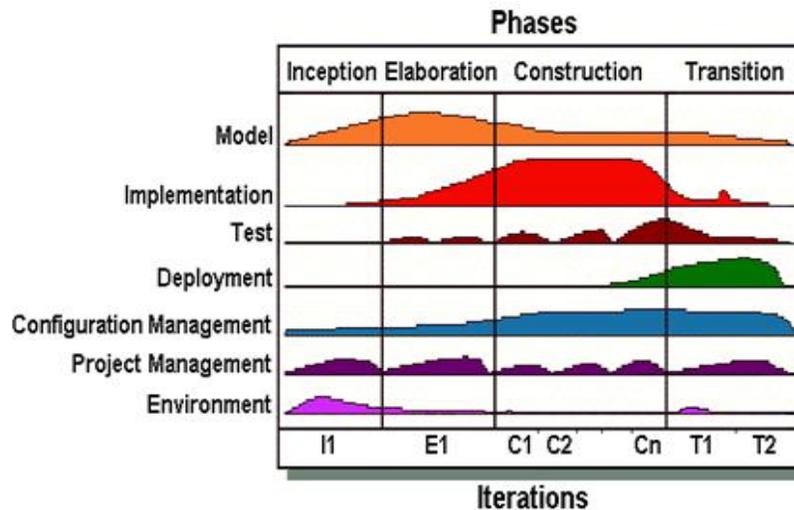


Figura 2.8. Fases y Disciplinas del AUP
Fuente: Ambler S, 2005

2.3.3. PRINCIPIOS DEL AUP

El AUP es ágil, porque está basada en los siguientes principios:

- **El personal sabe lo que está haciendo:** La gente no va a leer detallado el proceso de documentación, pero algunos quieren una orientación de alto nivel y / o formación de vez en cuando. La AUP producto proporciona enlaces a muchos de los detalles, si usted está interesado, pero no obliga a aquellos que no lo deseen.
- **Simplicidad:** Todo se describe concisamente utilizando poca cantidad de páginas, no miles de ellos.
- **Agilidad:** Se ajusta a los valores y a los principios de la Alianza Ágil.
- **Poner importancia a actividades de alto valor:** La atención se centra en las actividades que se ve que son esenciales para el de desarrollo, no todas las actividades que suceden forman parte del proyecto.
- **Independencia de la herramienta:** Usted puede usar cualquier conjunto de herramientas que usted desea con el ágil UP. Lo aconsejable es utilizar las herramientas que son las más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.

- **Adaptación de este producto para satisfacer sus propias necesidades:** La AUP producto es de fácil acomodo común a través de cualquier herramienta de edición de HTML. No se necesita comprar una herramienta especial, o tomar un curso, para adaptar la AUP.

2.3.4. CARACTERÍSTICAS DE LA AUP

Se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

- **Dirigido por Casos de Uso** Se centra en la funcionalidad que el sistema debe poseer para satisfacer las necesidades de un usuario (persona, sistema externo, dispositivo) que interactúa con él. Casos de uso como el hilo conductor que orienta las actividades de Desarrollo. (Ver Figura 2.9)

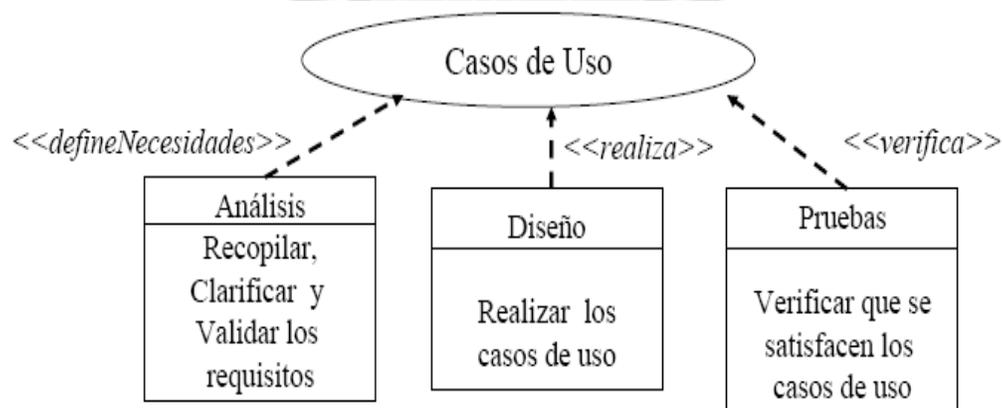


Figura: 2.9 Dirigido por Casos de Uso
Fuente: Ambler S, 2005

- **Centrado en la Arquitectura** Concepto similar a la arquitectura de un edificio, arquitectura en software, arquitectura: determina la forma del sistema, casos de uso: determinan la función del sistema.
- **Iterativo e Incremental.** Descomposición de un proyecto grande en mini-proyectos, cada mini-proyecto es una iteración, las iteraciones deben estar controladas, cada iteración trata un conjunto de casos de uso
- **Dimensión Dinámica del proceso**

Hito: punto en el tiempo donde se evalúan los objetivos logrados y se pueden tomar decisiones críticas. (Ver Figura 2.10)

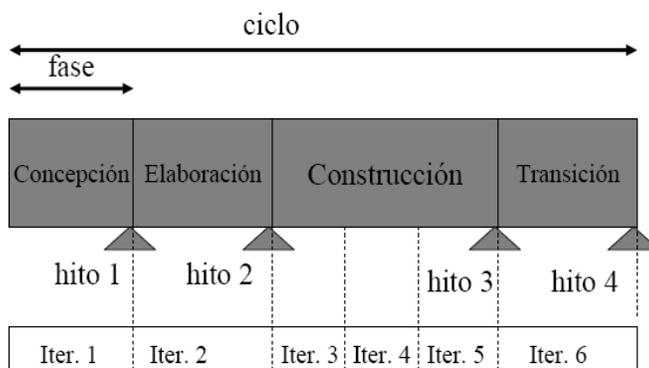


Figura 2.10, Dinámica del proceso
Fuente: Ambler S, 2005

- **Desarrollo Iterativo** se puede observar en la siguiente figura los pasos a seguir para el desarrollo. (Ver Figura 2.11)

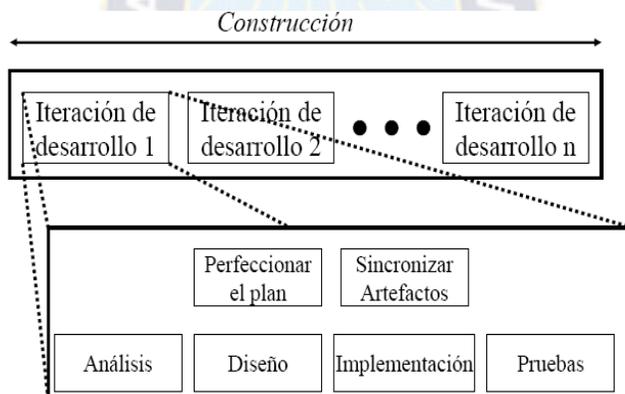


Figura 2.11 Desarrollo Iterativo
Fuente: Ambler S, 2005

2.3.5. FASES DEL AUP

AUP cuenta con las siguientes fases:

2.3.5.1. PRIMERA FASE: CONCEPCIÓN

Identificar el alcance inicial del proyecto, proveer una arquitectura potencial para el sistema, y obtener un financiamiento inicial del proyecto y la aceptación de los stakeholders.

- Visión = QUÉ + PARA QUÉ + CUÁNTO

- **Actividades**

- ✓ Especificación de los criterios de éxito del proyecto
- ✓ Definición de los requisitos
- ✓ Estimación de los recursos necesarios
- ✓ Cronograma inicial de fases.

- **Artefactos** (Pieza de información producida, modificada y utilizada en un Proceso)

- ✓ Documento de definición del proyecto

Hito: Objetivos del ciclo de vida (LCO).

2.3.5.2.SEGUNDA FASE: ELABORACIÓN

Obtener la arquitectura del sistema.

- **Actividades**

- Análisis del dominio del problema
- Definición de la arquitectura básica
- Análisis de riesgos
- Planificación del proyecto

- **Artefactos**

- Modelo del dominio
- Modelo de procesos
- Modelo funcional de alto nivel
- Arquitectura básica

Hito: Arquitectura del ciclo de vida (LCA).

2.3.5.3.TERCERA FASE: CONSTRUCCIÓN

Implementar un software sobre una base incremental la que debe estar relacionada con los objetivos de los involucrados.

Actividades

- ✓ Análisis
- ✓ Diseño
- ✓ Implementación / Codificación
- ✓ Pruebas (individuales, de integración)

Hito: Capacidad operacional inicial (IOC).

2.3.5.4.CUARTA FASE: TRANSICIÓN

Validar y entregar el sistema en un ambiente de producción. El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Actividades:

- ✓ Test del sistema
- ✓ Test de usuarios
- ✓ Trabajo del sistema
- ✓ Instalación del sistema

Hito: Lanzamiento del producto (PR).

2.3.6.DISCIPLINAS DEL AUP

Definen actividades que el equipo de desarrolladores debe realizar para construir, validar y entregar un software que satisfaga las necesidades de los stakeholders.

Las disciplinas son ejecutadas en una forma iterativa, definiendo las actividades que el equipo de desarrollo ejecuta para construir, aprobar y liberar software funcional, el cual cumple con las necesidades del usuario.

2.3.6.1.DISCIPLINA DEL MODELADO

Su objetivo principal es identificar una estrategia de arquitectura viable, entrada crítica dentro del plan del proyecto tanto como en el esfuerzo de implementación. La mejor forma de

trabajo es poner personal técnico, incluyendo algunos sino todos los desarrolladores, juntos en un lugar para desarrollar una estrategia de arquitectura que se discute en las pizarras creando diagramas de estilo libre quizás algún tipo de forma inicial de modelo de despliegue.

De los cuales se llega a observar dos casos como ser el flujo de trabajo y la fase por fase, a continuación se detalla lo mencionado.

a) Flujo de Trabajo

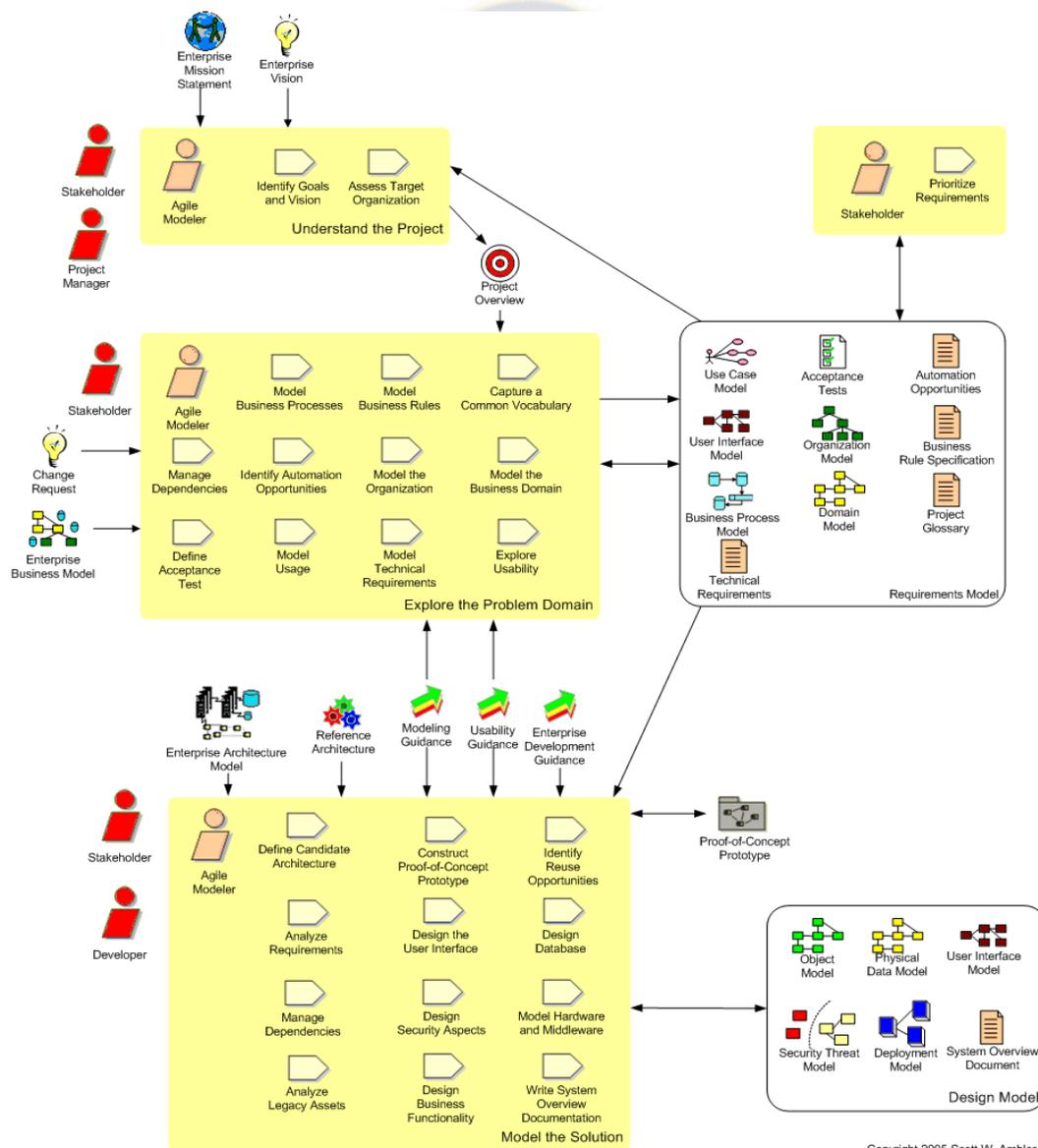


Figura 2.12 Flujo de trabajo de la Disciplina del modelado
Fuente: Ambler S, 2005

a) Fase por Fase

En la siguiente tabla se muestra la disciplina del modelado. (Ver Tabla 2.2)

Fases	Actividades
Inicio	<ul style="list-style-type: none"> • Explorar el uso de casos de uso. • Identifique los procesos del negocio para la creación de diagramas de flujo de datos (DFDs). • Identifique las entidades principales del negocio y sus relaciones trabajando con modelos de dominio livianos. • Identifique las principales reglas del negocio y requerimientos técnicos
Elaboración	<ul style="list-style-type: none"> • Identificar riesgos técnicos. • Modelado de la Arquitectura • Prototipito de interfaces de usuario.
Construcción	<p>Análisis de modelo de lluvia de ideas.</p> <ul style="list-style-type: none"> • Participación activa de interesados y modelado inclusivo que usan técnicas y herramientas simples que son críticas para su negocio. • Si lo desea, puede profundizar en los detalles de sus casos de uso. • Explore las reglas del negocio y los requerimientos técnicos en la misma forma. • Puede que necesite hacer interfaces de activos legados tales como sistemas actuales o una base de datos. • En lugar de las descripciones de casos de uso, de reglas del negocio y de requerimientos técnicos, usted puede encontrar más efectivo simplemente escribir casos de prueba de aceptación. • Mantenga su glosario del proyecto actualizado si lo tiene. • Diagrama de secuencia de UML. • Modelo de despliegue. • Documentos de resumen del sistema típicamente incluyen diagramas de estilo libre. • Modelo de amenazas de seguridad. • Modelo físico de datos. • Documento crítico decisiones de diseño.
Transición	<p>Modelado de lluvia de ideas.</p> <p>Finalice la documentación de resumen del sistema</p>

Tabla 2.2. Fases de la disciplina del modelado
Fuente: Ambler S, 2005

2.3.6.2.DISCIPLINA DE LA IMPLEMENTACIÓN

Transformar los modelos en código ejecutable y aplicar pruebas básicas en unidades particulares de prueba.

a) Flujo de Trabajo

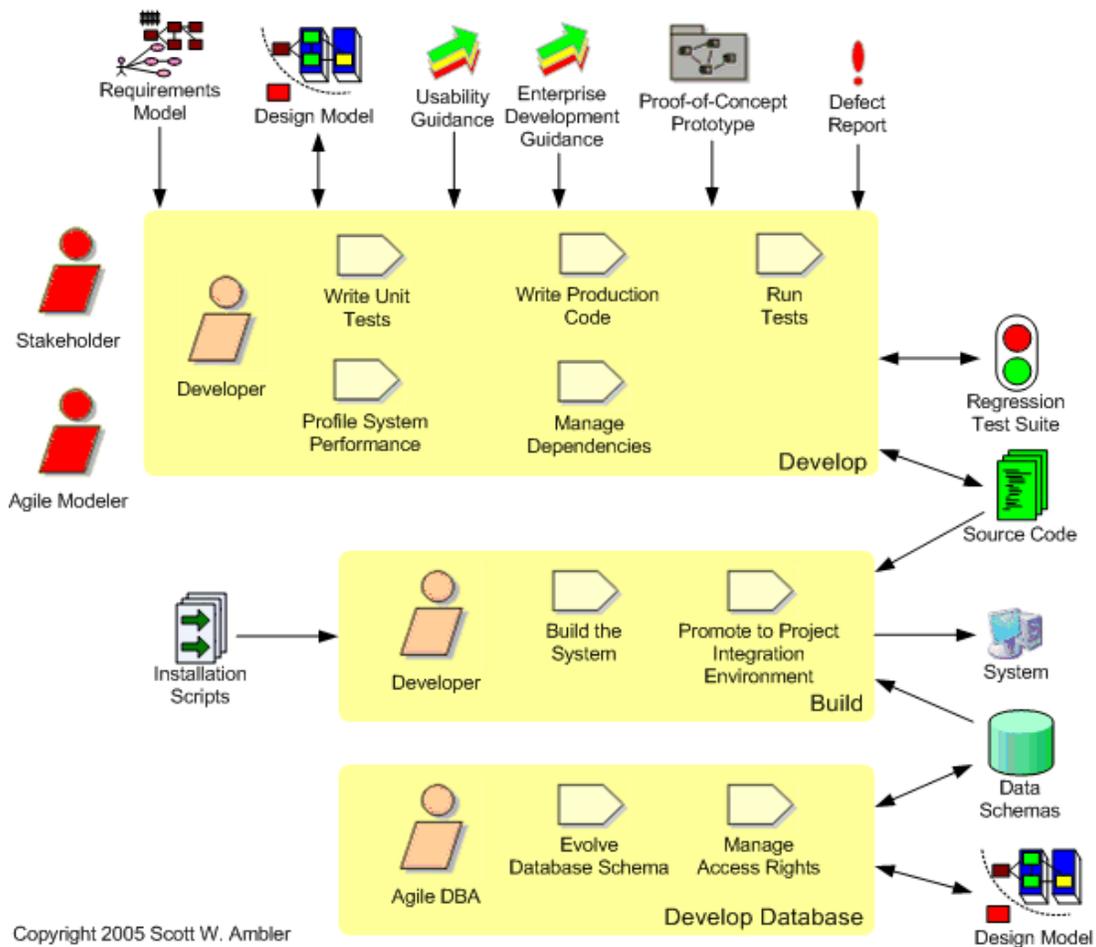


Figura 2.13. Flujo de Trabajo de la Disciplina de la implementación
Fuente: Ambler S, 2005

b) Fase por Fase

Fases	Actividades
Inicio	<p>Prototipado técnico</p> <p>Prototipado de Interfaces de Usuario. Para más usuarios las interfaces de usuario (UI) - pantallas, reportes, y manuales-- en el sistema.</p>

Elaboración	<p>Probar la arquitectura. Las actividades críticas dentro de la fase de Elaboración es identificar la arquitectura potencial y luego probar que esta arquitectura funcione a través del desarrollo de la arquitectura del prototipo extremo a extremo para su sistema, y a la vez mitigando gran parte de los riesgos técnicos en su proyecto.</p>
Construcción	<p>Primeras pruebas. Obtenga un acercamiento de la base del Desarrollo Dirigido por Pruebas (TDD) para todos los aspectos de la aplicación.</p> <p>Construya constantemente. Creaciones diarias son un buen comienzo, pero idealmente usted quiera construir su sistema cada vez que el código fuente cambie.</p> <p>Evolución de la lógica de dominio. Implemente su lógica del negocio in sus clases de negocio/dominio.</p> <p>Evolucionar el esquema de datos. Su esquema de datos debe ir evolucionado en conjunto con su dominio y código de interfaces de usuario.</p> <p>Desarrollo de interfaces de activos existentes. Necesitará acceder frecuentemente a la funcionalidad dentro de los sistemas existente.</p>
Transición	<p>Corregir defectos. Concéntrese en la corrección de defectos encontrados como resultado de las pruebas.</p>

Tabla 2.3. Fases de la disciplina de implementación
Fuente: Ambler S, 2005

2.3.6.3.DISCIPLINA DE PRUEBA

Realizar una evaluación objetiva para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema funcione como fue diseñado, y verificar que los requerimientos estén abordados por las funcionalidades.

a) Flujo de Trabajo

En la siguiente figura se describe todo el flujo de trabajo de la disciplina de despliegue. (Ver Figura 2.14)

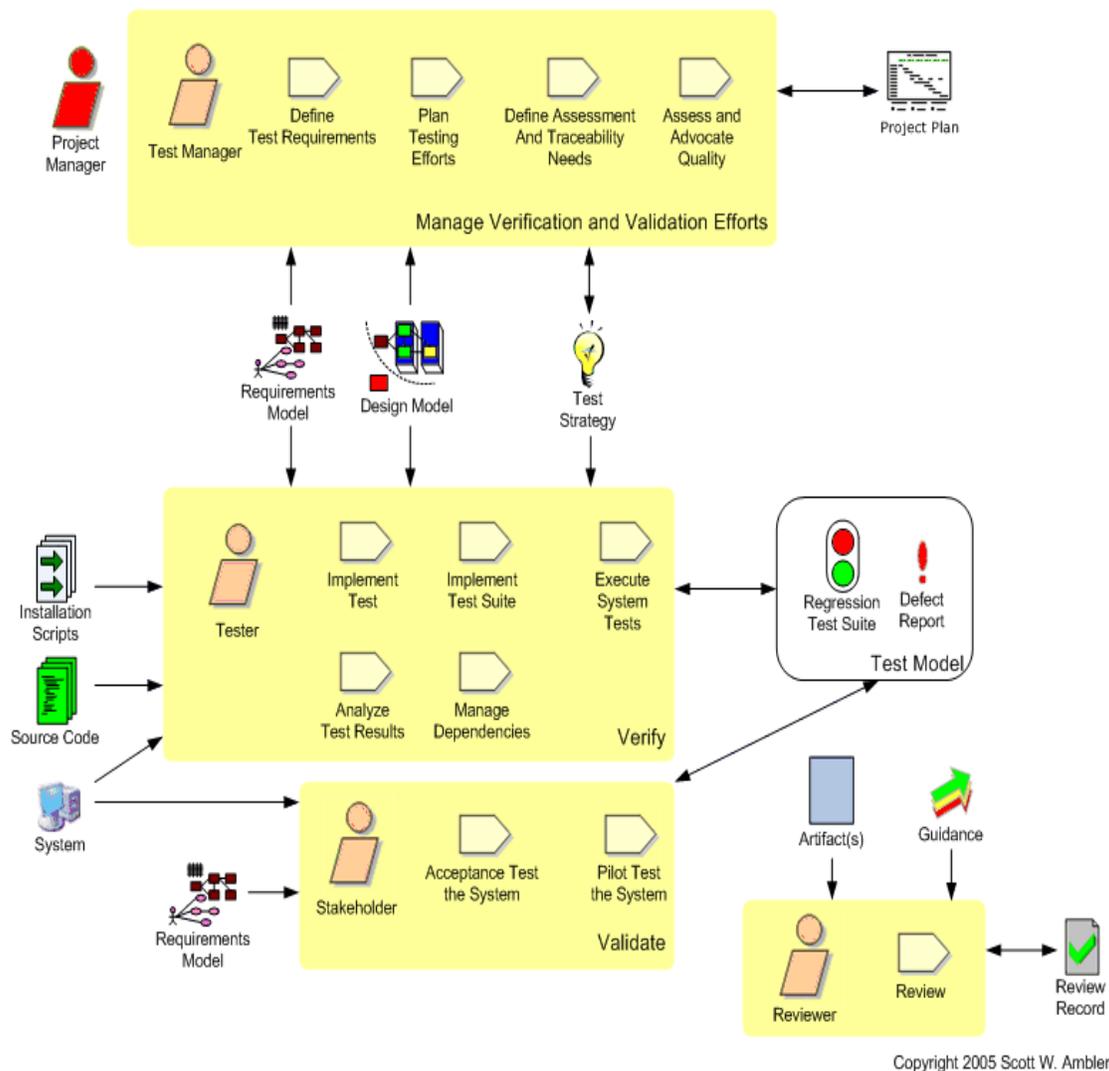


Figura 2.14 Flujo de Trabajo de la disciplina de Prueba
Fuente: Ambler S, 2005

b) Fase por Fase

Fases	Actividades
Inicio	Planificación inicial de pruebas. Deben ser a muy alto nivel al principio.
	Exanimación inicial de los productos de trabajo del proyecto
	Exanimación inicial de modelos. A un alto nivel

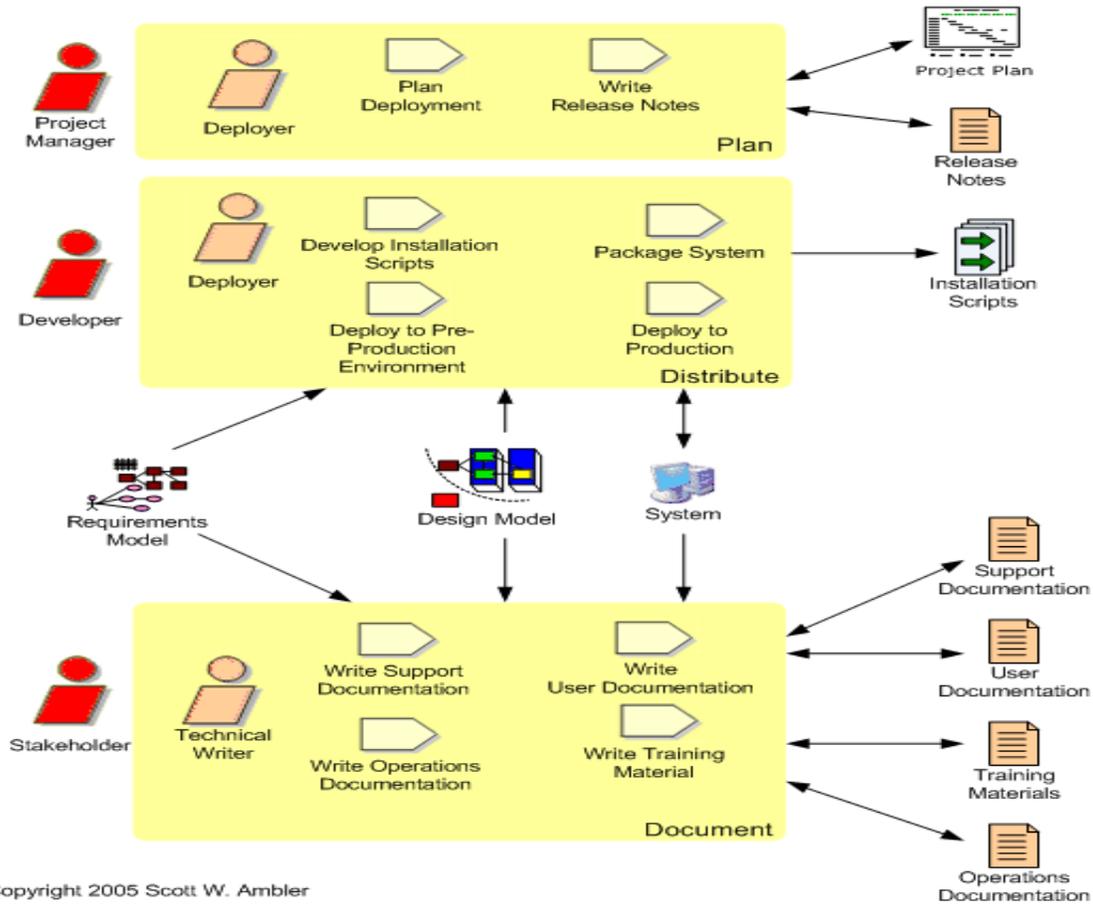
Elaboración	<p>Validación de la Arquitectura. Usted debe tomar un enfoque de desarrollo controlado por pruebas (TDD) para construir su prototipo técnico el cual compruebe la arquitectura de su sistema.</p> <p>Evoluciones su modelo de pruebas. Su equipo deberá desarrollar un paquete de pruebas de regresión.</p>
Construcción	<p>Pruebas de software. Además de las unidades de prueba de los desarrolladores deberá hacer pruebas de instalación del script de despliegue o liberación, sistema de pruebas de esfuerzos tales como la carga / pruebas de tensión y las pruebas de función, y sus pruebas de aceptación de usuario. Debido a que su sistema evoluciona a través de sus proyectos, su paquete de pruebas también lo hará. Lo más común es que promueva su código en un ambiente de pruebas de pre-producción, lo mejor de la fase de Transición serán las actividades de prueba.</p> <p>Evolucione su modelo de pruebas. Ver arriba.</p>
Transición	<p>Validación del sistema. Usted se concentrará en las "grandes pruebas" de actividades tales como las del sistema, las de integración y las de aceptación, y las pruebas piloto/beta. Su objetivo es probar completamente el sistema dentro del ambiente de pruebas de pre-producción.</p> <p>Validación de la documentación. Su documentación de sistema (vista general del sistema, usuarios, soporte, y documentación de operaciones), y sus materiales de capacitación necesitará validarlos. Todo esto puede ser hecho por medio de las revisiones o mejor aún como parte de sus pruebas piloto/betas.</p> <p>Analice su modelo de pruebas. Va a tener que seguir ejecutando de paquete de pruebas de regresión y actualizarlo tanto como necesite, hasta que su sistema esté listo para ser desplegado en producción.</p>

Tabla 2.4. Fases de la disciplina de prueba
Fuente: Ambler S, 2005

2.3.6.4. DISCIPLINA DE DESPLIEGUE

Planificar la entrega del sistema y ejecutar el plan para que el sistema esté disponible para los usuarios.

a) Flujo de trabajo



Copyright 2005 Scott W. Ambler

Figura 2.15. Flujo de trabajo de la disciplina de Despliegue
Fuente: Ambler S, 200

b) Fase por Fase

Fases	Actividades
Inicio	<p>Identificar el rango liberación potencial. Por ejemplo, se le solicita entregar su sistema antes de que finalice el año.</p> <p>Comience con un plan de entregables de alto nivel. Este esfuerzo se debe enfocar en planificar los entregables de su sistema, identificando el rango de liberación potencial.</p>
Elaboración	<p>Actualizar el plan de implementación. Una parte importante de definir la arquitectura es definir la configuración de los entregables del sistema.</p>

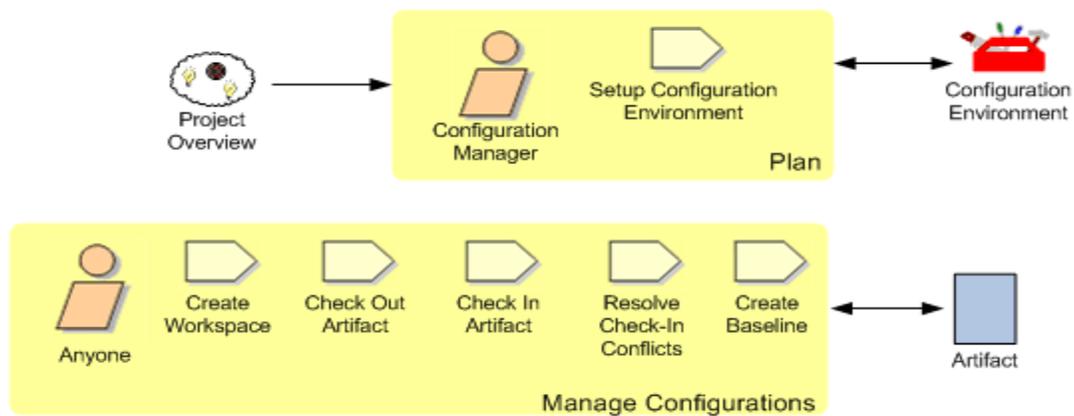
Construcción	<p>Desarrollar el script de instalación. Como desarrolla el sistema debe también escribir y probar el script de instalación necesario para entregarlo en el pre producción del ambiente de pruebas.</p> <p>Desarrollar notas del entregable. Sus notas de entregables deben resumir los avances que posee el entregable actual del sistema que actualmente está construyendo.</p> <p>Desarrollar documentación inicial. Además de entregar software funcional, también se debe entregar la documentación del sistema (operaciones, soporte, visión general, y la documentación al usuario), así como su material de formación.</p> <p>Actualice su plan. De manera cómo progrese el desarrollo del sistema, debe avanzar su plan de implementación.</p> <p>Implementar el sistema en entornos de pre-producción. Debe entregar regularmente el sistema en un ambiente de pre-producción para efectuar pruebas y llevar a cabo el control de calidad necesario, así como realizar demostraciones a los involucrados.</p>
Transición	<p>Concluir el proceso de implementación. Para concluir este proceso debe definir una línea base de entrega como referencia, las actividades de la administración de la configuración, y realizar una "última" revisión de software, así como la implementación del flujo de trabajo.</p> <p>Finalizar la documentación. La mayor parte de la documentación del sistema (operaciones, soporte, visión general, y la documentación al usuario) es generalmente realizada durante esta etapa, debido a que la funcionalidad del sistema se estabiliza en este momento.</p> <p>Anunciar la implementación. Se debe anunciar el calendario de implementación de manera anticipada e incluyendo las fechas estimadas de capacitación e instalación. Debe instruir también su equipo de operación, soporte y la comunidad de usuarios según proceda en esta fase.</p> <p>Capacitar el personal. Capacitar los clientes o usuarios de su proyecto</p> <p>Puesta en producción. En este punto se debe realizar cualquier conversión o migración de datos, y puede ser todo de una vez, un trabajo por lotes o una conversión gradual de los datos, conforme lo requieran los usuarios.</p>

Tabla 2.5. Fases de la disciplina de despliegue
Fuente: Ambler S, 2005

2.3.6.5. DISCIPLINA DE ADMINISTRACIÓN DE LA CONFIGURACIÓN

Administrar el acceso a los artefactos del proyecto. Esto no solo incluye los seguimientos de las versiones de los artefactos, sino también controlar y administrar los cambios sobre ellos.

a) Flujo de Trabajo



Copyright 2005 Scott W. Ambler

Figura 2.16. Flujo de trabajo de la disciplina de Administración de la Configuración

Fuente: Ambler S, 2005

b) Fase por Fase

Fases	Actividades
Inicio	<p>Establecer la configuración del ambiente. Usted tiene que hacer varias cosas:</p> <ul style="list-style-type: none"> • La estructura de directorios apropiada, la cual debe seguir los lineamientos corporativos, necesita ser creada para el equipo del proyecto. • Los miembros del equipo del proyecto necesitan tener acceso al folder o directorios del proyecto y cualquier otro software cliente instalados • Los miembros del equipo del proyecto también necesitan ser entrenados con los conceptos básicos de CM y las herramientas necesarios. • Su repositorio de CM necesitará ser instalado si este aún no se ha instalado. <p>Ponga todos los productos del trabajo bajo el control de CM. Cada uno debe poner su trabajo bajo el control de CM en una base regular, verificar las entradas y salidas en cada caso, resolver conflictos de actualización cuando se requiera, y presupuestar los productos del proyecto cuando las versiones más actualizadas estén aprobadas.</p>
Elaboración	<p>Poner todos productos del proyecto sobre el control de Configuración del Manager.</p>

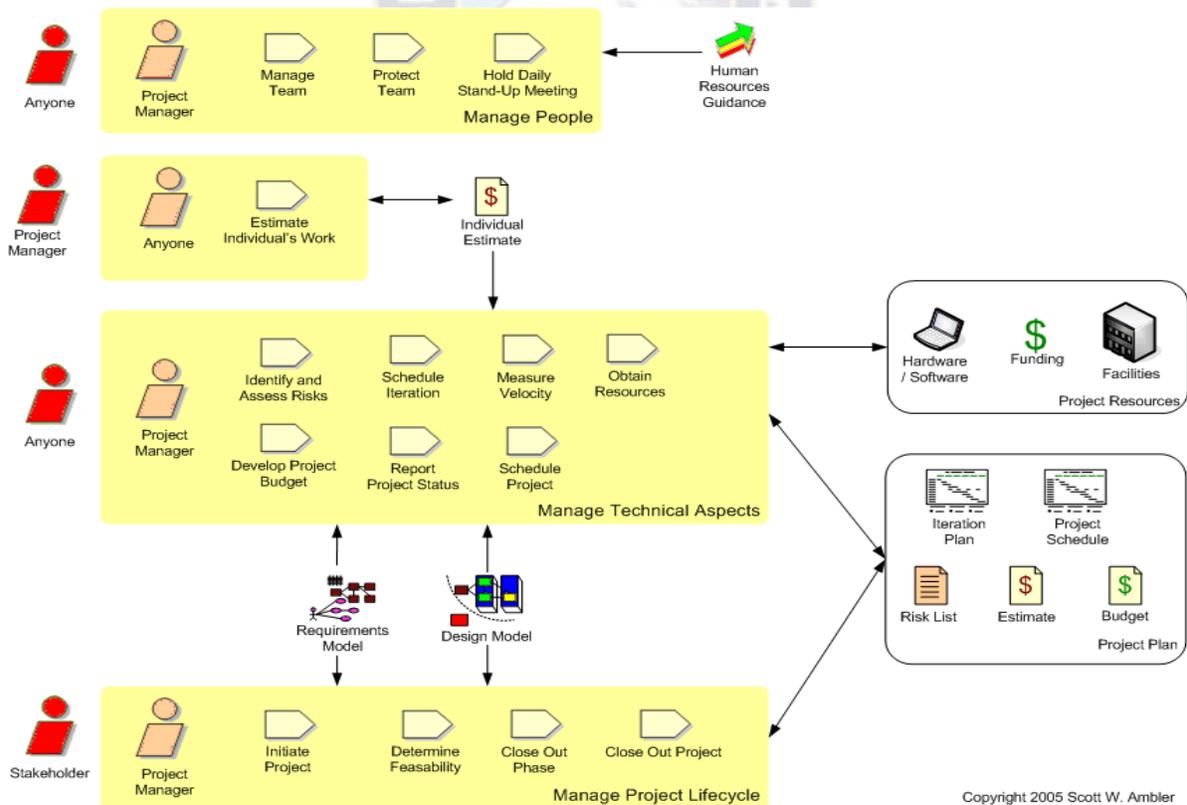
Construcción	Poner todos productos del proyecto sobre el control de CM.
Transición	Poner todos productos del proyecto sobre el control de CM.

Tabla 2.6. Fases de la disciplina de Administración de la Configuración
Fuente: Ambler S, 2005

2.3.6.6. DISCIPLINA DE LA ADMINISTRACIÓN DEL PROYECTO

Dirigir las actividades que forman parte del proyecto. Esto incluye administración de riesgos, dirigir personas y coordinar personas con sistemas que están fuera del alcance del proyecto.

a) Flujo de Trabajo



Copyright 2005 Scott W. Ambler

Figura 2.17. Flujo de trabajo de la disciplina de Administración del Proyecto
Fuente: Ambler S, 2005

b) Fase por Fase

Fases	Actividades
Inicio	<p>Inicie conformando el equipo. En este punto necesitará alguien con habilidades de modelado</p> <p>Crear relaciones con sus involucrados del proyecto. El soporte a los usuarios y la participación es crítica para su éxito.</p> <p>Determinar. El proyecto debe ser financiera, técnica, operacional, y políticamente viable.</p> <p>Desarrollar un cronograma de alto nivel para todo el proyecto. El cronograma del proyecto debe mostrar su proyecto organizado en iteraciones.</p> <p>Desarrollar un plan detallado de iteración para la siguiente iteración. La planificación detallada se realiza basada en el principio justo a tiempo.</p> <p>Manejo del riesgo. Siempre hay riesgos en un proyecto de desarrollo de software: técnicos y organizacionales.</p> <p>Obtener financiamiento y apoyo de los involucrados. Tendrá que demostrar que usted entiende el alcance, que el proyecto es viable.</p> <p>Cerrar esta fase. Debe ejecutar la Revisión de los entregables de los objetivos del ciclo de vida.</p>
Elaboración	<p>Construya el equipo. Conforme su proyecto tome forma y crezca, necesitará agregar miembros al equipo.</p> <p>Proteger el equipo. Las políticas de empresa son una realidad y un buen administrador de proyectos protegen a sus equipos lo mejor posible.</p> <p>Obtener recursos. Su equipo necesita financiación, instalaciones hardware, software, y así sucesivamente para hacer su trabajo.</p> <p>Manejo del riesgo. Continúe los esfuerzos de administración del riesgo.</p> <p>Actualice el plan del proyecto. Continúe las actividades planeadas.</p> <p>Cerrar esta fase. Tendrá que ejecutar la revisión del ciclo de vida de la arquitectura.</p>
Construcción	<p>Administre el equipo. Continúe desarrollando el equipo, manténgase protegiéndolos y proveyéndoles los recursos que necesitan.</p> <p>Manejo del riesgo. Continúe los esfuerzos de administración del riesgo.</p> <p>Actualizar su plan de proyecto. Durante la fase de construcción necesitará asegurar que tiene identificadas las principales dependencias involucradas en el desarrollo exitoso de su sistema.</p> <p>Cerrar esta fase. Tendrá que ejecutar la revisión de la capacidad operativa inicial.</p>
Transición	<p>Administrar el equipo. Debe incluir en el equipo de desarrolladores, encargados de pruebas e implementadores.</p> <p>Cerrar esta fase. Tendrá que ejecutar la revisión de los productos entregables</p> <p>Iniciar el próximo ciclo del proyecto. Los sistemas se desarrollan y se ponen en producción de manera incremental.</p>

Tabla 2.7. Fases de la disciplina de Administración del Proyecto
Fuente: Ambler S, 2005

2.3.6.7.DISCIPLINA DEL AMBIENTE

En la disciplina del ambiente, el objetivo principal es el de facilitar todo el entorno que permita el normal desarrollo del proyecto, para lo cual se observa el flujo de trabajo y la fase por fase que se tiene en cada punto.

a) Flujo de Trabajo

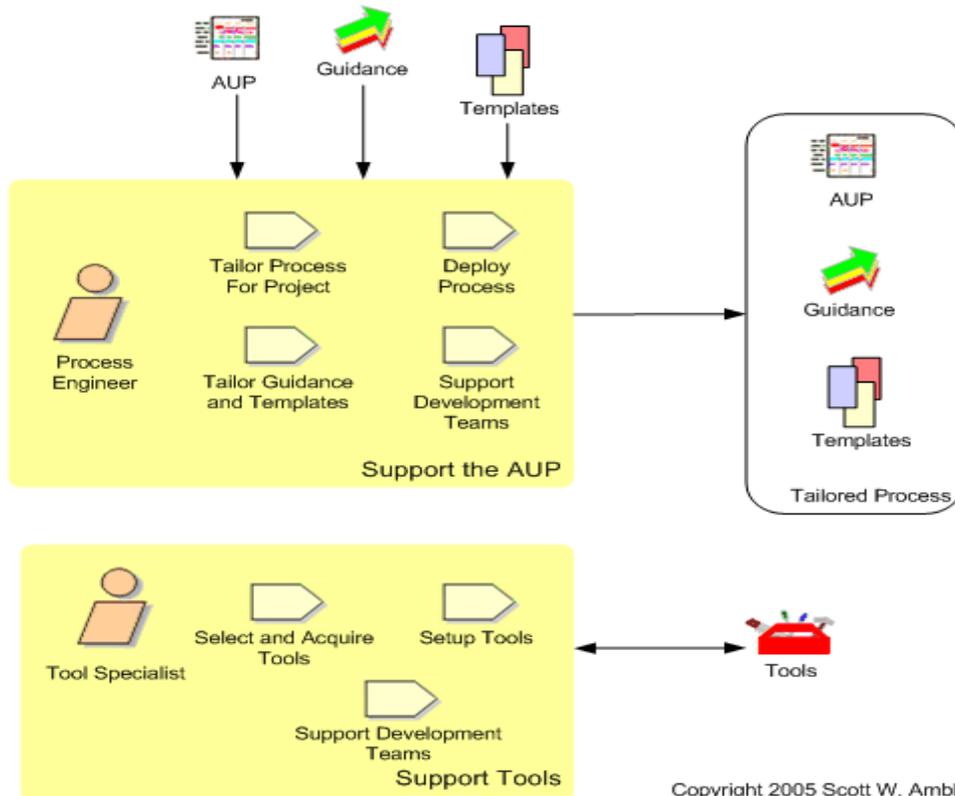


Figura 2.18. Flujo de trabajo de la disciplina del Ambiente
Fuente: Ambler S, 2005

b) Fase por Fase

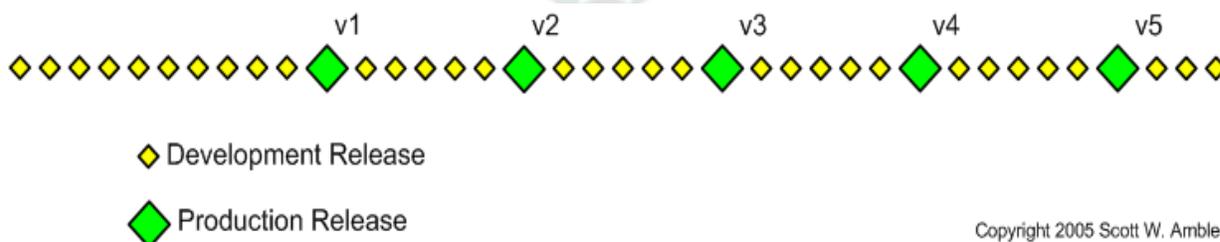
Fases	Actividades
Inicio	<p>Configure el entorno de trabajo. Esta será una tarea permanente ya que hay gente que se añade al equipo en el tiempo.</p> <p>Identifique la categoría del proyecto. Muchas organizaciones desarrollan varias versiones de sus procesos de software, por ejemplo uno para equipos pequeños, uno para reemplazar sistemas legales, otro para sistemas de plataforma comercial, etc.</p>

Elaboración	Evolucionar el entorno de trabajo. Su proyecto progresa a medida que se entiende la evolución de los requisitos, la estrategia de la arquitectura, y su enfoque general. Ajuste de los procesos de materiales. Se debe ajustar AUP para cumplir con las necesidades del equipo.
Construcción	Apoyar al equipo. Miembros del equipo del proyecto necesita ayuda para utilizar y / o la configuración de diversas herramientas para satisfacer sus necesidades. Evolucionar el entorno de trabajo. Ver elaboración. Establecer el ambiente de capacitaciones. A medida que progrese en el plan de despliegue o liberación se debe descubrir que se necesita entrenar al usuario, personal de soporte y el personal de operación.
Transición	Configuración de las operaciones y soporte de los entornos. Personal de soporte, y algunas veces personal de operación, frecuentemente se necesita una versión del sistema configurada que se use para simular reportes de defectos en una forma segura. Recobrar licencias de software. A medida que su proyecto llega a la conclusión puede ser necesario desinstalar las licencias de software los equipos que ya no necesitan el software para que las licencias puedan estar disponibles a los demás dentro de su organización.

Tabla 2.8. Fases de la disciplina del ambiente
Fuente: Ambler S, 2005

2.3.6.8. ITERACIONES E INCREMENTOS DE TIEMPO

Los equipos del AUP entregan típicamente lanzamientos del desarrollo en el final de cada iteración. Como se puede observar en la siguiente figura el primer reléase puede tomar más tiempo que el segundo, eso se debe en que en primera instancia hay más imprevistos a resolver que en las instancias futuras. Como se puede observar en la siguiente figura lo que propone AUP es que a medida que el proyecto avanza las iteraciones se vuelvan más cortas. (Ver Figura 2.19)



Copyright 2005 Scott W. Ambler

Figura 2.19. Iteraciones Production Release
Fuente: Ambler S, 2005

2.3.6.9.ROLES DEL EQUIPO

Los roles del equipo sobre los que se basan AUP son:

- ✓ Las roles pueden llevarse a cabo por varias personas.
- ✓ Una persona puede adquirir roles múltiples.
- ✓ Un rol no es una posición.

En la siguiente tabla se muestra la descripción de los roles del equipo. (Ver Tabla 2.9)

Rol	Descripción	Disciplina(s)
DBA Ágil	Un administrador de bases de datos (DBA) que trabaja de manera colaborativa con los integrantes del equipo del proyecto para diseñar, probar y brindar soporte a los diferentes esquemas de datos.	Implementación
Modelador Ágil	Alguien que cree y desarrolle modelos, ya sean dibujos, tarjetas, o archivos complejos realizados con herramientas CASE, de manera colaborativa y evolutiva	Modelado Implementación
Cualquiera	Cualquier otra persona en otro rol distinto.	Administración de la Configuración Administración del Proyecto
Administrador de la configuración	Un administrador de configuración se encarga de proporcionar la infraestructura y crear el medio ambiente para el equipo de desarrollo.	Administración de la Configuración
Implementador	Un implementador es responsable de poner en disponer el sistema en los ambientes de pre-producción y producción.	Desarrollo

Desarrollador	Es quien escribe código, realiza pruebas y construye el software.	Modelado Implementación Desarrollo
Especialista del proceso	Desarrolla, adapta y apoya el material de los procesos de la organización (descripción de procesos, plantillas, guías, ejemplos, entre otros).	Entorno
Administrador del proyecto	Administra los miembros de los equipos de trabajo, crea relaciones con los involucrados, coordina las interacciones con los involucrados, planea, administra y dispone recursos, enmarca prioridades y mantiene el equipo enfocado.	Modelado Pruebas Desarrollo Administración del Proyecto
Examinador	Evalúa los productos del proyecto, inclusive "el trabajo en progreso", suministrando retroalimentación al equipo de trabajo.	Pruebas
Involucrado	Un involucrado es cualquiera que sea usuario directo, usuario indirecto, administrador de usuarios, administrador, miembro de equipo de operación o soporte, desarrolladores que trabajan en otros sistemas que se integran o interactúan con el sistema implementado, en fin todo aquel que se vea afectado de una u otra forma con el proyecto.	Modelado Implementación Pruebas Desarrollo Administración del Proyecto
Documentador técnico	Es responsable de producir documentación para los involucrados, tal como: materiales de capacitación, documentación de operaciones, documentación de mantenimiento, y documentación de usuario.	Desarrollo
Administrador de pruebas	El administrador de pruebas es el responsable del éxito de las pruebas, incluye planificar la administración, y promover las pruebas y las actividades de calidad.	Pruebas

Equipo de pruebas	El equipo de pruebas es responsables de ejecutar las pruebas y documentar los resultados que proyecten.	Pruebas
Especialista en herramientas	Es responsable de seleccionar, adquirir, configurar y brindar mantenimiento al equipo requerido.	Ambiente

Tabla 2.9. Descripción de roles del equipo
Fuente: Ambler S, 2005

2.3.6.10. ENTREGABLES

AUP clasifica los entregables en tres tipos:

Productos a entregar mínimos: es la documentación mínima requerida para realizar el proyecto.

Otros productos del trabajo de proyecto: es la documentación no esencial según AUP.

Productos del trabajo de la empresa: son los entregables que la empresa debe realizar para el proyecto.

1. Mantenga sus productos de trabajos tan simples y concisos como sea posible.
2. Usted necesita menos documentación de la que cree.
3. Trabaje cerca de las personas con las cuales usted está creando el producto de trabajo para que haga sólo lo que ellos necesitan.
4. Documentos Ágiles son los suficientemente buenos para la tarea en cuestión.
5. Producir un documento es la peor forma de comunicar información, muchas personas paradas frente a una pizarra hablando es la mejor manera.
6. Use herramientas simples como una pizarra (antigua), hojas de papel, y wikis para modelar y capturar documentación.
7. Considere adoptar las plantillas de código abierto como base de partida para crear su propio trabajo. Estas tienen, probablemente más de lo que usted necesita, pero son un buen comienzo.

Los productos mínimos a entregar se lo prioriza en la siguiente tabla. (Ver Tabla 2.10)

Entregable	Descripción	Sugerencias
Sistema	El software de trabajo, el hardware y la documentación para ser liberada a producción.	Hay más en su sistema que sólo el software que se escribe.
Código fuente	El código de programa para su sistemas	Siga las directrices de codificación común.
Suite de Pruebas de Regresión	Una colección de casos de prueba, y el código para correrlas en un orden adecuado.	Automatice sus pruebas. Ejecútelas tan frecuentemente como sea posible, sobre todo cuando ocurre algún cambio.
Scripts de Instalación	Código para instalar su sistema su ambiente de pre-producción.	Necesitará un script, o scripts, para instalarlo en un ambiente de pre-producción tan exacto como el de producción. Probablemente deba desinstalar los scripts si su instalación falla.
Documentación del Sistema	La documentación liberada como una parte del sistema para ayudar al usuario al trabajar con él, y a los desarrolladores para mantenerlo actualizado.	Mantenga su documentación tan liviana como sea posible.
Notas	Sus notas deben resumir "las buenas cosas a saber" acerca de las versiones actuales que se están construyendo.	Una lista puntual es a menudo suficiente.

<p style="text-align: center;">Modelado de requerimientos</p>	<p>Describe los requisitos que su sistema debe cumplir.</p>	<p>Su objetivo es entender y luego construir lo que sus usuarios quieren, no escribir montículos de documentación.</p> <p>No necesita mantener todos los aspectos para su modelo de requerimientos, sólo la porción que resuma el alcance de su sistema. Considere mantener:</p> <ul style="list-style-type: none"> • Diagramas de Procesos del Negocio(s). • El glosario del proyecto. • Cualquier otro requisito de productos. • Pruebas de aceptación para requerimientos implementados.
<p style="text-align: center;">Modelo de Diseño</p>	<p>Describe el diseño de su sistema. Consta de una variedad de productos de trabajo, incluye potencialmente un modelo de despliegue, un modelo de objetos, un modelo de datos físico (PDM), un modelo de seguridad de amenazas, un documento de resumen del sistema, y un modelo de interface de usuario.</p>	<p>Mantenga su modelo tan simple como sea posible, y descarte en cuanto le sea posible una vez que haya extraído el valor de ellos.</p> <p>El mejor lugar para documentar es en su unidad de pruebas y código fuente.</p> <p>Mantenga el documento de resumen del sistema y el modelo físico de datos para documentación permanente. Debe también mantener unos pocos diagramas de diseño detallados, tales como diagramas de secuencia o diagramas de las máquinas de estado.</p>

Tabla 2.10: Productos mínimos a entrega
Fuente: Ambler S, 2005

2.4.INGENIERÍA WEB

La ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.

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

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

2.4.1.APLICACIONES WEB

Las aplicaciones Web son un sistema de información donde una gran cantidad de datos, altamente estructurados, son consultados, procesados y actualizados mediante navegadores.

Categorías de las aplicaciones Web

Las aplicaciones Web se categorizan en:

- **Informativa.** Contenido solo de lectura con navegación y enlaces simples.
- **Descarga.** Se descarga información desde un servidor apropiado.
- **Personalizable.** Se puede cambiar el contenido a necesidades específicas.
- **Interacción.** La comunicación entre una comunidad de usuarios ocurre mediante un espacio chat (charla), anuncios o mensajería.
- **Entrada del usuario.** Entrada basada en formularios.
- **Orientada a transacciones.** Realización de una solicitud que es completada por la aplicación Web.

- **Orientado a servicios.** Proporción de un servicio al usuario.
- **Portal.** La aplicación canaliza al usuario llevándolo a otros contenidos o servicios Web fuera del dominio de la aplicación del portal.
- **Acceso a bases de datos.** El usuario consulta en una base de datos grande y extrae información.
- **Almacenes de datos.** El usuario consulta en una colección de base de datos grande y extrae información.

Tecnologías Web

El diseño y la implementación de sistemas basados en Web incorporan tres tecnologías, el desarrollo basado en componentes, la seguridad y los estándares de Internet.

- **Desarrollo basado en componentes.** Proporcionan una infraestructura que permite a los que diseñan emplear y personalizar componentes de terceras partes permitiéndoles así comunicarse unos con otros y con servicios en el ámbito de sistemas, estándares como JavaBeans.
- **Seguridad.** Mediante la infraestructura de red se proporciona una variedad de medidas de seguridad, tales como encriptación, cortafuegos y otras.
- **Estándares de Internet.** Estándares para la creación del contenido y la estructura de la aplicación Web como ser, HTML y XML.

2.4.2. MODELO DE PROCESO DE INGENIERÍA WEB

Las Aplicaciones Web necesitan de un marco de trabajo de Ingeniería Web que acompañe a un modelo de proceso.

El proceso de Ingeniería Web comienza con la Formulación, actividad que identifica las metas y los objetivos de las Aplicaciones Web. La Planificación evalúa los riesgos asociados con el esfuerzo del desarrollo. El Análisis establece los requisitos técnicos e identifica los elementos del contenido que se van a incorporar. La actividad de ingeniería incorpora dos tareas paralelas, el Diseño del Contenido y la Producción, son tareas llevadas a cabo por personas no técnicas. El objetivo de estas tareas es diseñar, producir y/o adquirir todo el

contenido de texto, gráfico y vídeo que se vayan a integrar en la Aplicación Web. Al mismo tiempo se lleva a cabo un conjunto de tareas de diseño. Cada incremento producido como parte del proceso de Ingeniería Web se revisa durante la actividad de evaluación del cliente.

Existen diversas metodologías para el desarrollo de sitios Web, como ser:

- ✓ **WSDM:** Web Site Design Method [TROYER & LEUNE, 1997]. Es una propuesta para el desarrollo de sitios Web, en la que el sistema se define en base a los grupos de usuarios. Su proceso de desarrollo se divide en cuatro fases: modelo de usuario, diseño conceptual, diseño de la implementación e implementación.
- ✓ **SOHDM:** Scenario-based Object-Oriented Hypermedia Design Methodology [LEE, LEE & YOO, 1998]. Presenta la necesidad de disponer de un proceso que permita capturar las necesidades del sistema. Para ello, propone el uso de escenarios, cada escenario describe el proceso de interacción entre el usuario y el sistema.
- ✓ **HFPM:** Hypermedia Flexible Process Modeling [OLSINA, 1998]. Describe un proceso detallado que cubre todo el ciclo de vida de un proyecto software, propone un total de trece fases para las cuales se especifican a su vez una serie de tareas.
- ✓ **UWE:** UML-Based Web Engineering (HENNICKER & KOCH, 2000), [KOCH, 2001]. Es una propuesta metodológica basada en UML para el desarrollo de aplicaciones Web. UWE cubre todo el ciclo de vida de este tipo de aplicaciones.
- ✓ **W2000** (BARESI, GARZOTTO & PAOLINI, 2001). Supone una propuesta que amplía la notación de UML con conceptos para modelar elementos de multimedia. El proceso de desarrollo de se divide en tres etapas: análisis de requisitos, diseño de hipermedia y diseño funcional.
- ✓ **UWA:** Ubiquitous Web Applications. Ha nacido de la colaboración entre diferentes grupos de trabajo, por lo que resulta realmente una agrupación de propuestas y técnicas. En concreto, la propuesta de W2000 se encuentra incluida en UWA.
- ✓ **NDT:** Navigational Development Techniques [ESCALONA, TORRES & MEJÍAS, 2002]. Es una técnica para especificar, analizar y diseñar el aspecto de la navegación en aplicaciones Web.

A continuación se presenta una comparativa de requisitos y de orientación:

	Req. datos	Req. interfaz	Req. Navegacion	Req. Personaliz.	Req. Transacion.	Req. no funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFPM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
UWA	✓	✓	✓	✓	✓	
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

Tabla 2.11: Requisitos de cada propuesta
Fuente: ESC, 2002

	Orientación al proceso	Orientación a la técnica	Orientación al resultado
WSDM	o	-	-
SOHDM	-	+	-
RNA	+	-	-
HFPM	+	o	+
OOHDM	o	+	-
UWE	+	o	o
W2000	o	o	-
UWA	+	o	-
NDT	o	+	+
DDDP	+	o	-

Tabla 2.12: Orientación de cada propuesta
Fuente: ESC, 2002

Donde:

Orientación al proceso: describe claramente los pasos a seguir (+), describe el proceso sin detallarlo (o), no describe ningún proceso (-).

Orientación a la técnica: describe claramente las técnicas y la forma de aplicarlas (+), enumera las técnicas a aplicar (o), no propone ninguna técnica concreta o referencia a técnicas generales (-).

Orientación al producto: describe claramente la estructura del producto a obtener (+), describe el contenido del producto sin entrar en detalle de su estructura (o), no comenta nada sobre el producto resultante (-).

2.5.METODOLOGÍA DE MODELADO WEB UWE

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

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

Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo navegacional, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad.

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

Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son los que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web.

De esta manera, se obtiene una notación UML adecuada aun dominio en específico a la cual se le conoce como Perfil UML.

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

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

2.5.1. ACTIVIDADES DE MODELADO DE UWE

Las actividades base de modelado de UWE son:

- Análisis de requerimientos.
- Modelo conceptual
- Modelo navegacional.
- Modelo de presentación.

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

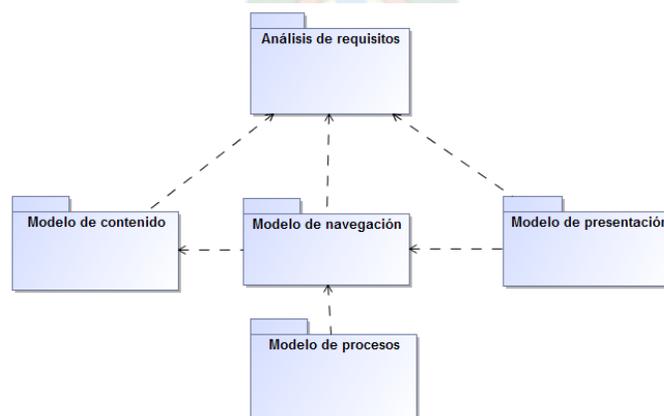


Figura 2.20. Actividades del modelado UWE

Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

A. Especificando los requisitos

Una de las primeras actividades en la construcción de aplicaciones Web es la identificación de los requisitos, y en UWE se especifican mediante el modelo de requerimientos, que involucra el modelado de casos de uso con UML.

El diagrama de casos de uso está conformado por los elementos actor y caso de uso. Los actores se utilizan para modelar los usuarios de la aplicación Web que para este caso de estudio son los diferentes tipos de usuarios (anónimo, consultor, tutor, alumno) que pueden interactuar con el mismo.

Los casos de uso se utilizan para visualizar las diferentes funcionalidades que la aplicación tiene que proporcionar, como son: crear a un nuevo usuario, identificar al usuario, realizar una búsqueda, realizar la composición de un nuevo objeto y guardar el objeto compuesto. En la figura 2.21 se ilustra el diagrama de casos de usos para la aplicación web. Es de mencionar que para cada etapa del modelado, UWE provee diferentes estereotipos. La lista de todos los estereotipos que pueden utilizarse en esta etapa se encuentra en el Perfil UWE (Profile UWE) del sitio oficial de UWE.

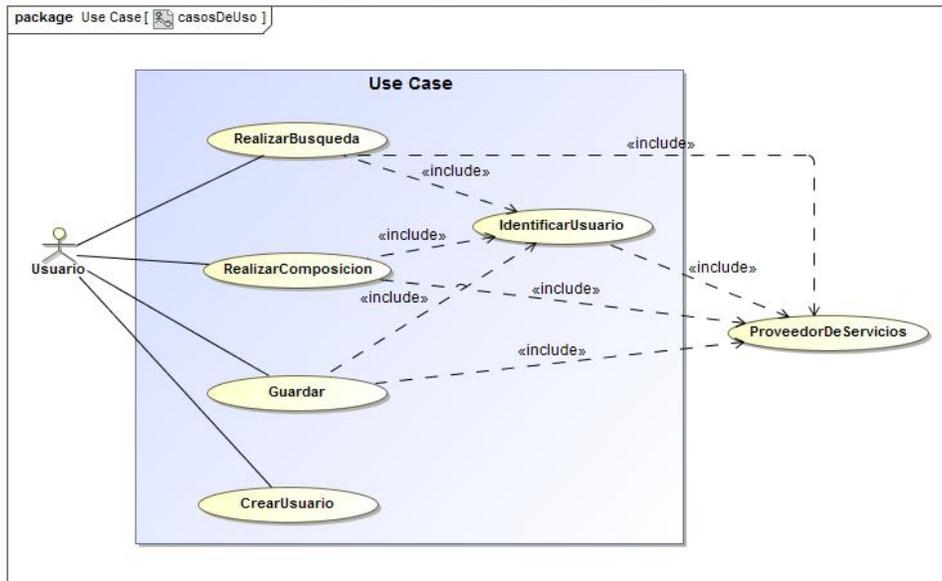


Figura 2.21. Casos de uso, requisitos en UWE
Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

El caso de uso "RealizarBusqueda" es del estereotipo explorar («browsing»). Modela la búsqueda de los objetos de aprendizaje por medio de las características de los objetos y de los usuarios para que el sistema pueda proporcionar una recomendación personalizada.

El caso de uso "RealizarComposicion" es del estereotipo procesar («processing»). Según la lista final seleccionada por el usuario, ejecuta el proceso de composición para conformar un nuevo objeto de mayor nivel de instrucción añadiendo cambios a los metadatos si el usuario así lo decide.

El caso de uso "IdentificarUsuario" es del estereotipo explorar («browsing»). Ejecuta el proceso de inicio de sesión el cual verifica si el usuario proporcionado existe en el sistema.

El caso de uso "Guardar" es del estereotipo procesar («processing»). Ejecuta la conversión del objeto compuesto al estándar IEEE-LOM y almacena el objeto compuesto en la computadora o en el repositorio para su posterior uso.

El caso de uso "CrearUsuario" es el estereotipo procesar («processing»). Registra los datos de un nuevo usuario que se agrega al sistema, lo que facilita información de su perfil y mejora la personalización de los resultados.

El nivel de detalle y la formalidad de la especificación de requerimientos dependen de los riesgos del proyecto y de la complejidad de la aplicación Web a construir. A menudo una especificación basada solamente en casos de uso no es suficiente.

Siguiendo el principio de usar UML para la especificación hasta donde sea posible, es factible emplear diagramas de actividades en esta fase. Para cada caso de uso descrito para actividades no triviales se puede construir al menos un diagrama de actividad por cada flujo principal de tareas realizadas en orden.

B. Definiendo el contenido

El objetivo del modelo de contenido es proporcionar una especificación visual de la información en el dominio relevante para la aplicación Web.

Este es un diagrama UML normal de clases, por ello se debe pensar en las clases que son necesarias para el caso de estudio presentado.

En la figura 2.22 se presenta el diagrama de clases para el modelo de contenido. En particular, la información de los usuarios es modelada por la clase "PerfilUsuario" donde se almacenan las propiedades que describen a los diferentes tipos de usuarios.

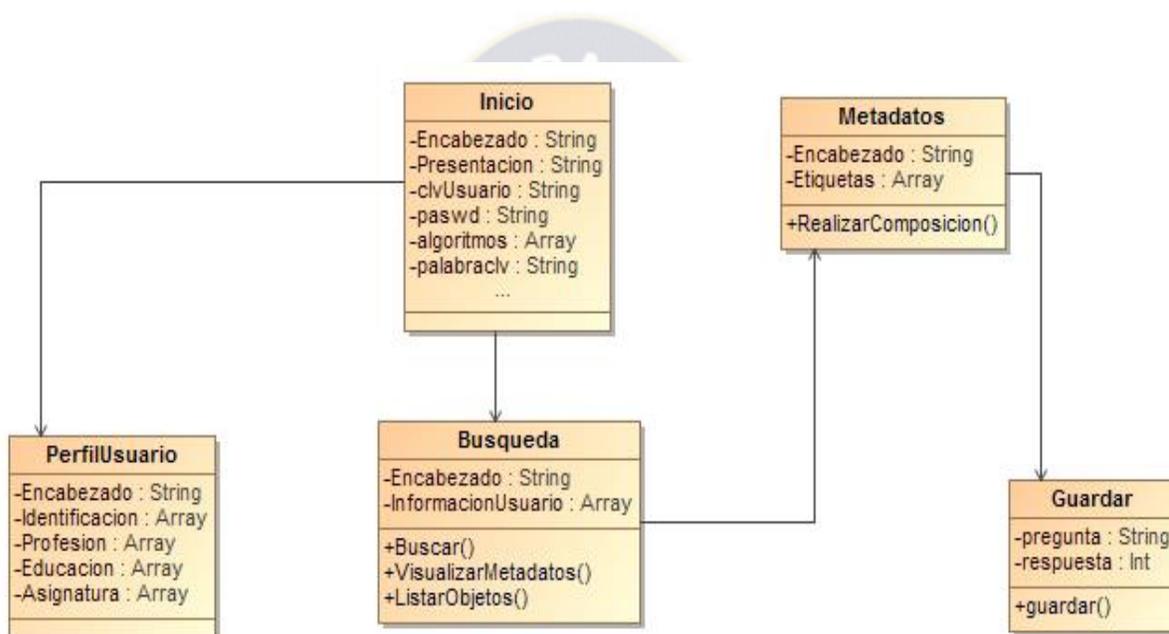


Figura 2.22. Modelo de Contenido
Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

En la clase "Inicio" se modela el inicio de la aplicación web, se almacenan las credenciales y propiedades que sirven para identificar al usuario que quiere iniciar sesión. La clase "Búsqueda" modela la información que el usuario proporciona para realizar una consulta y los métodos que se ejecutan para generar la lista de recomendación, la selección de los objetos y la recuperación de los mismos con sus metadatos. La clase "metadatos" modela las características devueltas por los objetos de aprendizaje que el usuario ha seleccionado y el método de realizar la composición con la selección y los metadatos proporcionados. La clase "guardar" modela las características de almacenamiento del nuevo objeto compuesto.

C. Estructura de Navegación

En una aplicación para la Web es útil saber cómo están enlazadas las páginas. Ello significa que se requiere un diagrama de navegación con nodos y enlaces. Este diagrama se modela con base en el análisis de los requisitos y el modelo de contenido.

UWE provee diferentes estereotipos para el modelado de navegación, en la figura 2.24 se presentan los usados en este caso de estudio y seguidamente se da una descripción de cada uno de ellos.

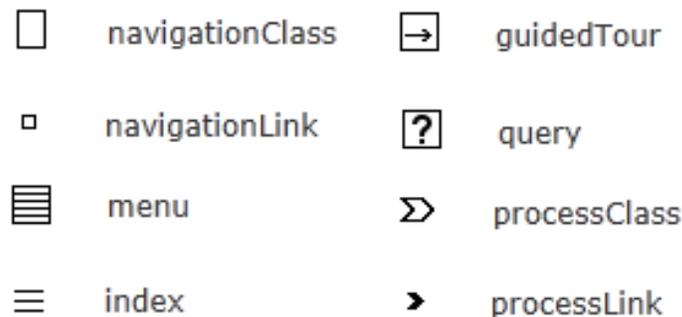


Figura 2.23. Estereotipos de estructura de navegación
Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

Las clases de navegación («navigationClass») representan nodos navegables de la estructura de hipertexto; los enlaces de navegación («navigationLink») muestran vínculos directos entre las clases de navegación; las rutas alternativas de navegación son manejadas por menú («menu»). Los accesos se utilizan para llegar a múltiples instancias de una clase de navegación («index» o «guidedTour») o para seleccionar los elementos («query»). Las clases de procesos («processClass») forman los puntos de entrada y salida de los procesos de negocio en este modelado y la vinculación entre sí y a las clases de navegación se modela por enlaces de procesos («processLink»).

En la figura 2.24, las clases de navegación "Inicio y PerfilUsuario" representan nodos navegables de la estructura de hipertexto y se consideran relevantes para la navegación. Los enlaces de navegación "navigationLink" y "processLink" muestran vínculos directos entre

las clases de navegación y representan posibles pasos a seguir por el usuario y, por lo tanto, estos vínculos tienen que ser dirigidos.

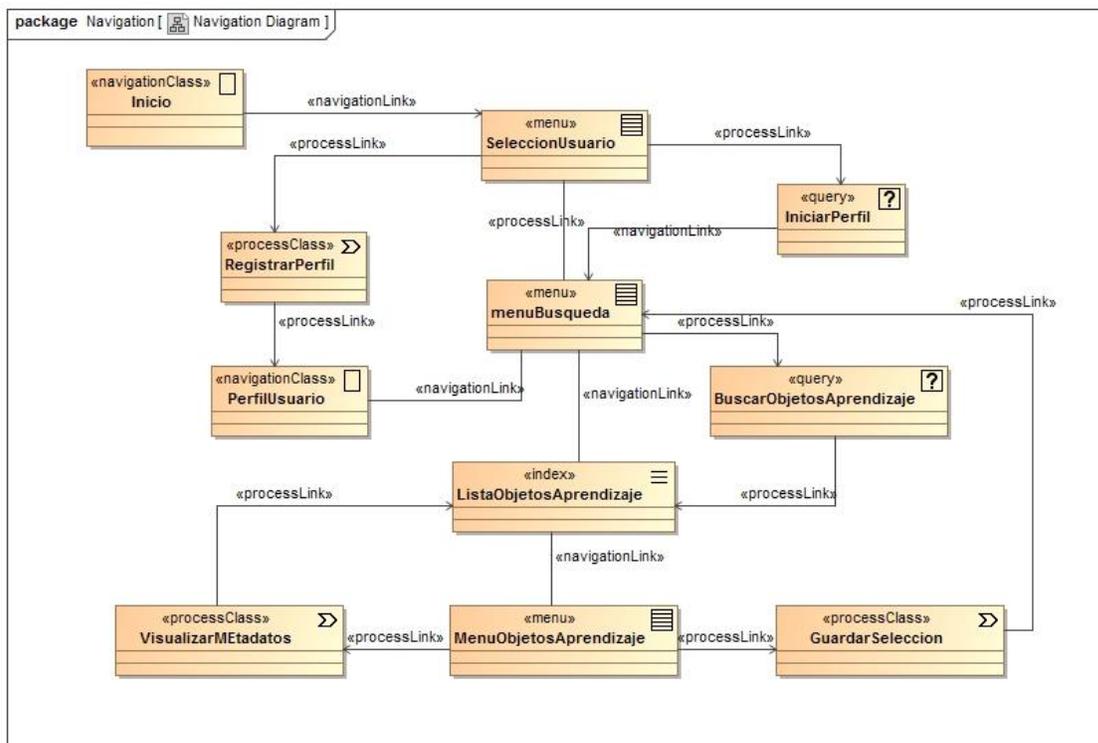


Figura 2.24. Clases de navegación.
Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

La navegación por diferentes alternativas es representada por las clases «menu» ("SeleccionUsuario, MenuBusqueda y MenuObjetosAprendizaje") que se añaden a cada clase de navegación que tiene más de una asociación saliente.

Las primitivas de acceso «index» como es "ListaObjetosAprendizaje" se utilizan para llegar a múltiples instancias de una clase de navegación o para seleccionar los elementos con los tipos «query» como "IniciarPerfil y BuscarObjetosAprendizaje", este tipo de clase se debe agregar entre dos clases de navegación cada vez que la multiplicidad de la meta final de su asociación de enlace sea mayor que 1. Las entradas y salidas de las clases "RegistrarPerfil, VisualizarMetadatos y GuardarSeleccion" son modeladas por las clases «process».

Es así que desde la página de Inicio un usuario puede, por medio de "SeleccionUsuario", tener una representación personalizada según sea su tipo de usuario con el que accede al sistema. Puede optar por usar "IniciarPerfil" para consultar si existe su clave de usuario proporcionada, o por "registrarPerfil" que inicia el proceso de registro del nuevo usuario. El usuario que ingresa a la aplicación proporciona palabras clave para "BuscarObjetosAprendizaje" que arroja una "ListaObjetosAprendizaje" para la selección por parte del usuario. De los objetos que son seleccionados en un "MenuObjetosAprendizaje", el usuario puede "VisualizarMetadatos" de los objetos que son candidatos a conformar un nuevo Objeto de Aprendizaje de nivel superior de complejidad para "GuardarSeleccion".

D. Modelo de presentación

El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de una aplicación Web. Se basa en el modelo de navegación y en los aspectos concretos de la interfaz de usuario (IU). Describe la estructura básica de la IU, es decir, ¿qué elementos de interfaz de usuario (por ejemplo, texto, imágenes, enlaces, formularios) se utilizan para presentar los nodos de navegación. Su ventaja es que es independiente de las técnicas actuales que se utilizan para implementar un sitio Web, lo que permite a las partes interesadas discutir la conveniencia de la presentación antes de que realmente se aplique.

Una clase de presentación está compuesta de elementos de IU como texto («text»), enlaces («anchor»), botones («button»), imágenes («image»), formularios («form») y colecciones de enlaces («anchored collection»). La figura 4 muestra un ejemplo de la clase de presentación para la clase de navegación Inicio.

En la figura 6 se modela la página de presentación "PaginaInicio". Existe una representación de texto para el encabezado y un mensaje de presentación. Modela también un formulario de entrada para que el usuario introduzca clave y contraseña, así como los botones de "iniciarperfil" y "registrarPERfil".

Usualmente la información de varios nodos de aplicación es presentada en una página Web, la cual es modelada por páginas en UWE, por ejemplo, en la figura 2.25 se tiene una («presentationPage»). Las páginas de presentación también pueden contener grupos de («presentationGroup»), grupos de presentación iterativos («iteratedPresentationGroup»), y presentaciones alternativas («presentationAlternative»), por ejemplo ajustar la interfaz al dispositivo utilizado para ejecutar la aplicación. Un grupo de presentación puede contener a si mismo grupos de presentación y clases de presentación.

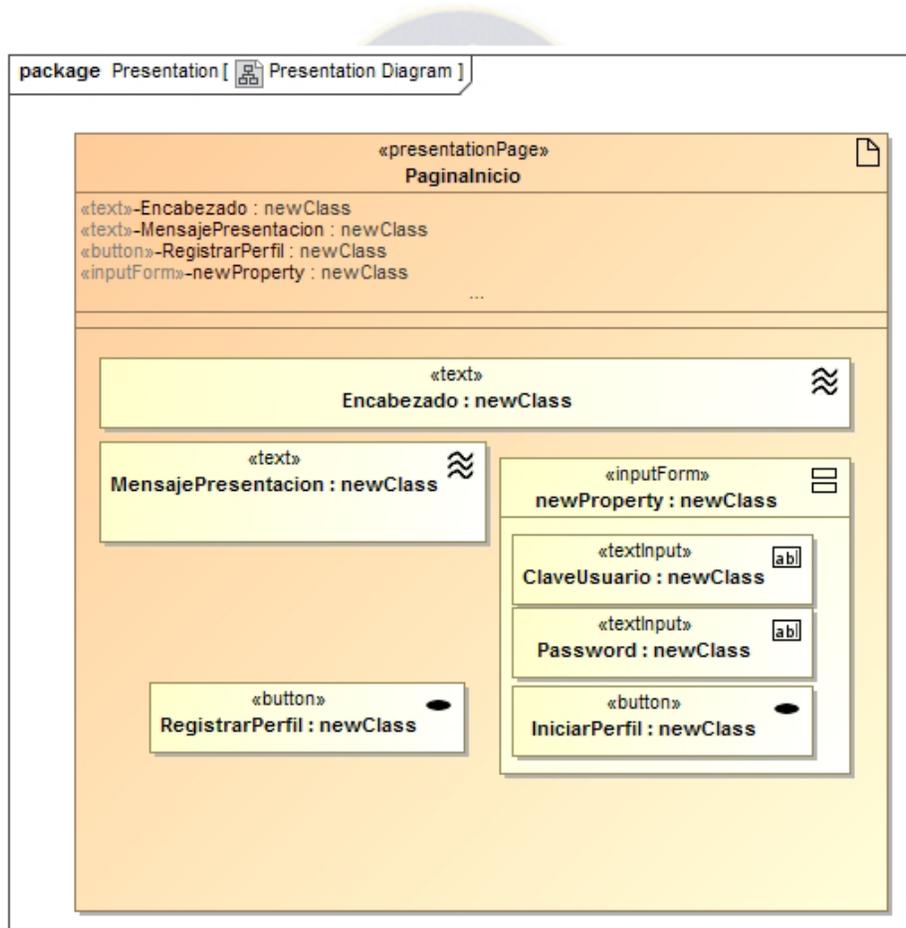


Figura 2.25. Página de presentación: Inicio.
Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

En la figura 2.26 se modela la página de presentación "paginaBusqueda" donde se representa como texto un encabezado y el nombre del usuario. Existe un formulario donde se puede introducir las palabras clave de búsqueda así como seleccionar los algoritmos que se pueden

aplicar. Esta página de presentación contiene un grupo de presentación para modelar las listas de objetos candidatos a la composición y los botones de buscar y ver metadatos.

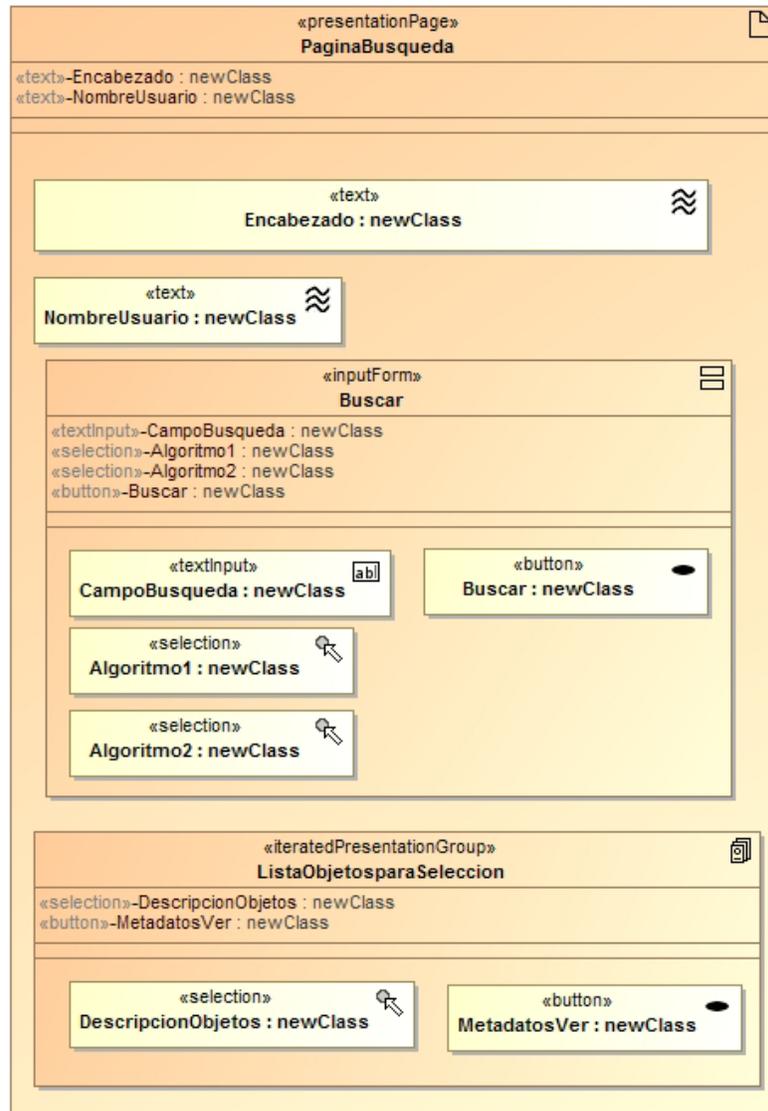


Figura 2.25. Página de presentación: Búsqueda.
Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

E. Modelo de proceso

La estructura de navegación puede ser extendida mediante clases de procesos que representan la entrada y la salida de procesos de negocio. El modelo del proceso representa el aspecto

que tienen las acciones de las clases de proceso. En este modelo se tienen dos tipos de modelos:

- Modelo de estructura del proceso, que describe las relaciones entre las diferentes clases de proceso, y
- Modelo de flujo del proceso, que especifica las actividades conectadas con cada « processClass».

A continuación se describen cada uno de ellos:

- **Modelo de estructura del proceso.** Es representado por un diagrama de clases donde se describen las relaciones entre las diferentes clases de proceso. La figura 2.26 presentala aplicación del modelo para el caso de estudio analizado.
- **Modelo del flujo del proceso.** Siguiendo el principio de la utilización de UML se han refinado los requisitos con los diagramas de actividad UML. Los diagramas de actividades incluyen actividades, actores responsables de estas actividades (opcional) y elementos de flujo de control. Ellos pueden ser enriquecidos con flujos de objetos que muestran objetos relevantes para la entrada o salida de esas actividades.

Estos diagramas representan el flujo del proceso, describiendo el comportamiento de una clase de proceso. En la figura 2.27 se ilustra el diagrama de actividad para el proceso "Inicio". El diagrama muestra que al generar la página de inicio el usuario puede optar por dos opciones:

- proporcionar su clave de usuario y contraseña si es un usuario registrado,
- activar el botón para registrarse como nuevo usuario.

En el caso de la primera opción, el sistema debe validar al usuario proporcionando el acceso a la búsqueda de objetos a aquellos usuarios que sean confirmados como válidos o mostrando un mensaje de error para el caso contrario. En la segunda opción, se debe activar el proceso de registro para capturar el perfil del nuevo usuario.

En la figura 2.28 se ilustra el diagrama de actividad para el proceso "Buscar". El diagrama muestra que se activa con el botón buscar y el usuario proporciona las palabras clave para iniciar la búsqueda. La aplicación regresa una lista de objetos de aprendizaje candidatos a ser seleccionados por el usuario. Si existe la información se recupera la misma desde el repositorio, en caso contrario se regresa a la página de búsqueda. Si la información listada es de interés para el usuario, este selecciona la misma, en caso contrario cambia sus parámetros de búsqueda.

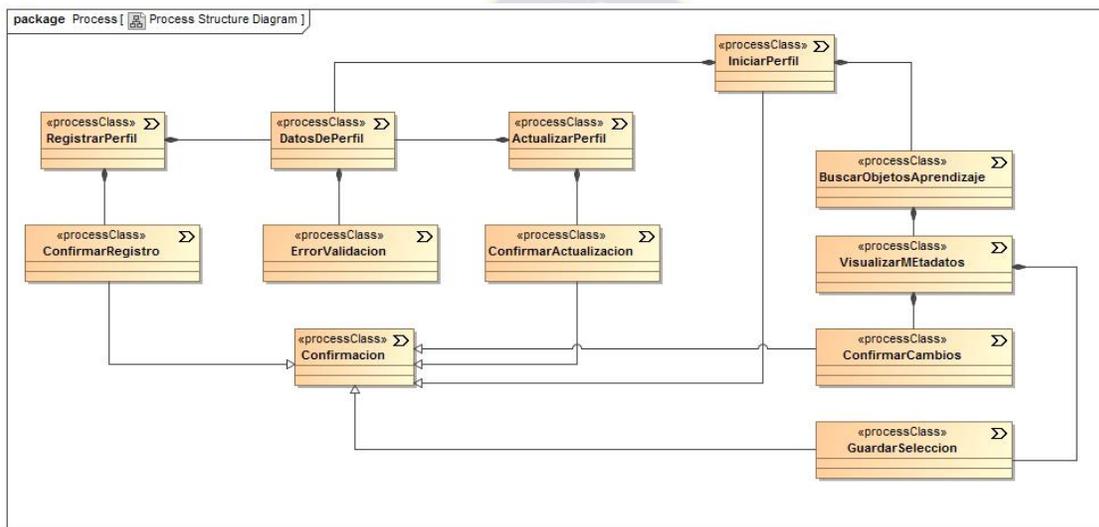


Figura 2.26. Estructura del proceso.

Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

Se ilustra el diagrama de actividad para el proceso "Inicio" (ver figura 2.27)

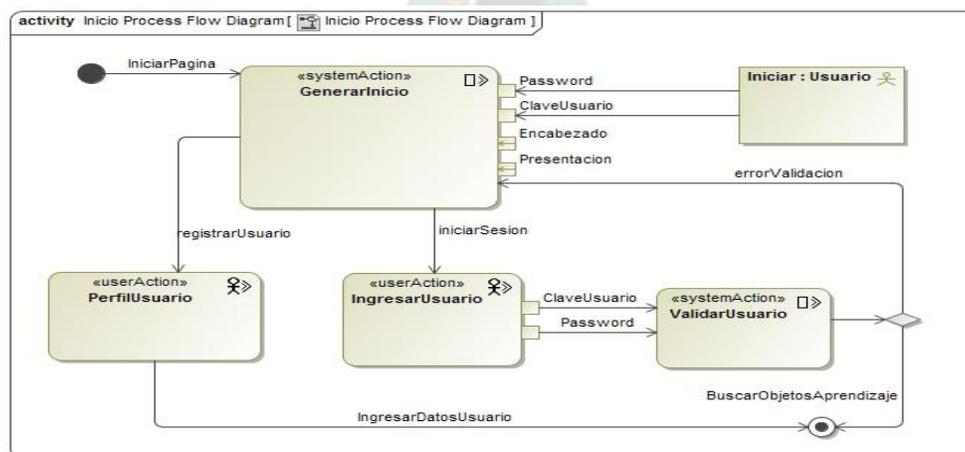


Figura 2.27. Flujo del proceso: Inicio.

Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

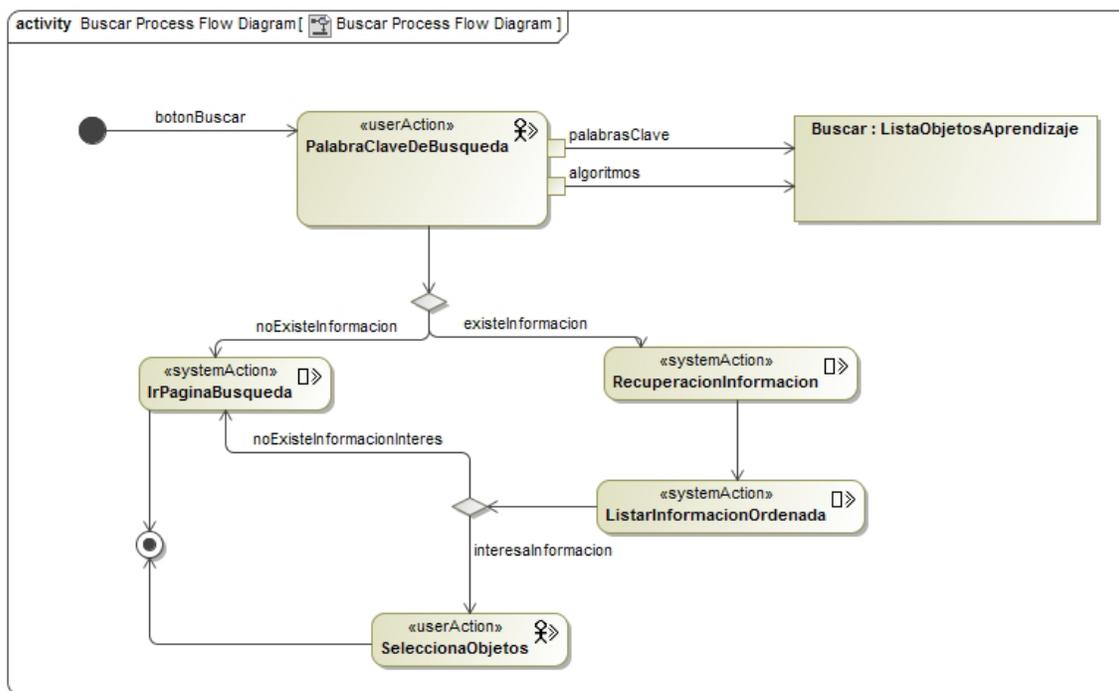


Figura 2.28. Flujo del proceso: Buscar.
Fuente: Revista Latinoamericana de Ingeniería de Software, 2014

2.5.2. ETAPAS O SUB-MODELOS DE UWE

- Modelo de Casos de Uso
- Modelo de contenido
- Modelo de estructura
- Modelo Abstracto
- Modelo de Adaptación
- Modelo de flujo de presentación
- Modelo de ciclo de vida del objeto
- Modelo Lógico-Conceptual

UWE apunta a construir un modelo conceptual de una aplicación Web, procurando hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web.

El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web.

2.5.3. FASES Y ETAPAS DE LA UWE

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

Fases:

1) Captura, análisis y especificación de requisitos:

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

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

2) Diseño del sistema:

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

3) Codificación del software:

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

4) Pruebas:

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

5) La Instalación o Fase de Implementación:

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

6) El Mantenimiento:

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



Figura 2.29. Etapas del modelado UWE
Fuente: Revista Latinoamericana de Ingeniería de Software

2.6.MODELO VISTA CONTROLADOR

Un problema muy acostumbrado para los programadores es la reutilización o mejora de código existente. Para los casos que hay que resolver un problema parecido a algo que ya tenemos hecho, mejorar el aspecto de un programa, mejorar su algoritmo, etc. Esta tarea se facilita mucho si a la hora de programar tenemos la precaución de separar el código en varias partes que sean susceptibles de ser reutilizadas sin modificaciones.

En casi cualquier programa que hagamos podemos encontrar tres partes bien diferenciadas:

- Por un lado tenemos el problema que tratamos de resolver. El problema suele ser independiente de como queramos que nuestro programa recoja los resultados o como queremos que los presente. El código construirá el modelo.
- Otra parte clara es la presentación visual que queramos hacer de la aplicación. El código es la vista. Le llamare interfaz gráfica por ser lo más común, pero podría ser texto, de comunicaciones con otro programa externo, con la impresora, etc.
- La tercera parte de código, es aquel que toma decisiones, algoritmos, etc. Es código que no tiene que ver con las ventanas visuales ni con las reglas de nuestro modelo.

Tras este tipo de ordenación, si queremos reaprovechar cosas en futuros programas, está claro que el modelo debe ser independiente. Las clases del modelo no deben ver a ninguna clase de los otros grupos. De esta forma podremos compilar el modelo en una librería independiente que podremos utilizar en cualquier programa que hagamos. Siguiendo con el orden de posibilidad de reutilización, el controlador podría ver clases del modelo, pero no de la vista. De esta forma, con los siguientes riesgos de estropearlo además del trabajo del retoque.

La vista es lo más cambiante, así se podrá hacer que vea clases del modelo y del controlador. Si cambiamos algo del controlador o del modelo, es bastante seguro que tendremos como mínimo que recompilar la interfaz gráfica.



3.1. INTRODUCCIÓN

La finalidad de este capítulo, es establecer el análisis y diseño del “SISTEMA WEB ACADÉMICO INTEGRAL PARA EL INSTITUTO TECNOLÓGICO AYACUCHO”, que se desarrollara utilizando para este fin, la metodología de desarrollo ágil AUP y siguiendo con el lenguaje de modelo web UWE, como se muestra en la figura 3.1.

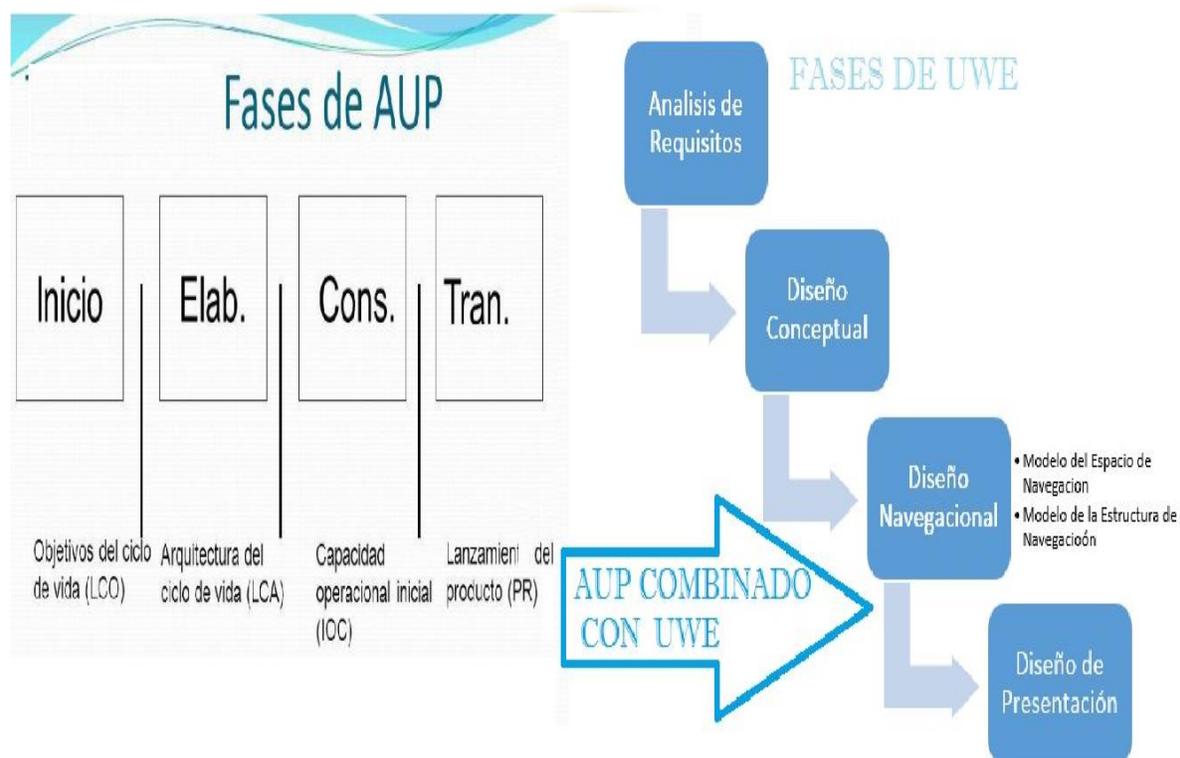


FIGURA 3.1. Modelos AUP y UWE
FUENTE: Elaboración Propia

Se planea una estructura gráfica, organizativa de herramientas y metodologías que intervienen directamente en el presente proyecto.

Esta estructura grafica nos guiara durante todo el desarrollo del capítulo III, mediante una serie de fases en las cuales podremos analizar, diseñar e implementar en una aplicación a los métodos descritos, como se puede observar en la figura 3.2.

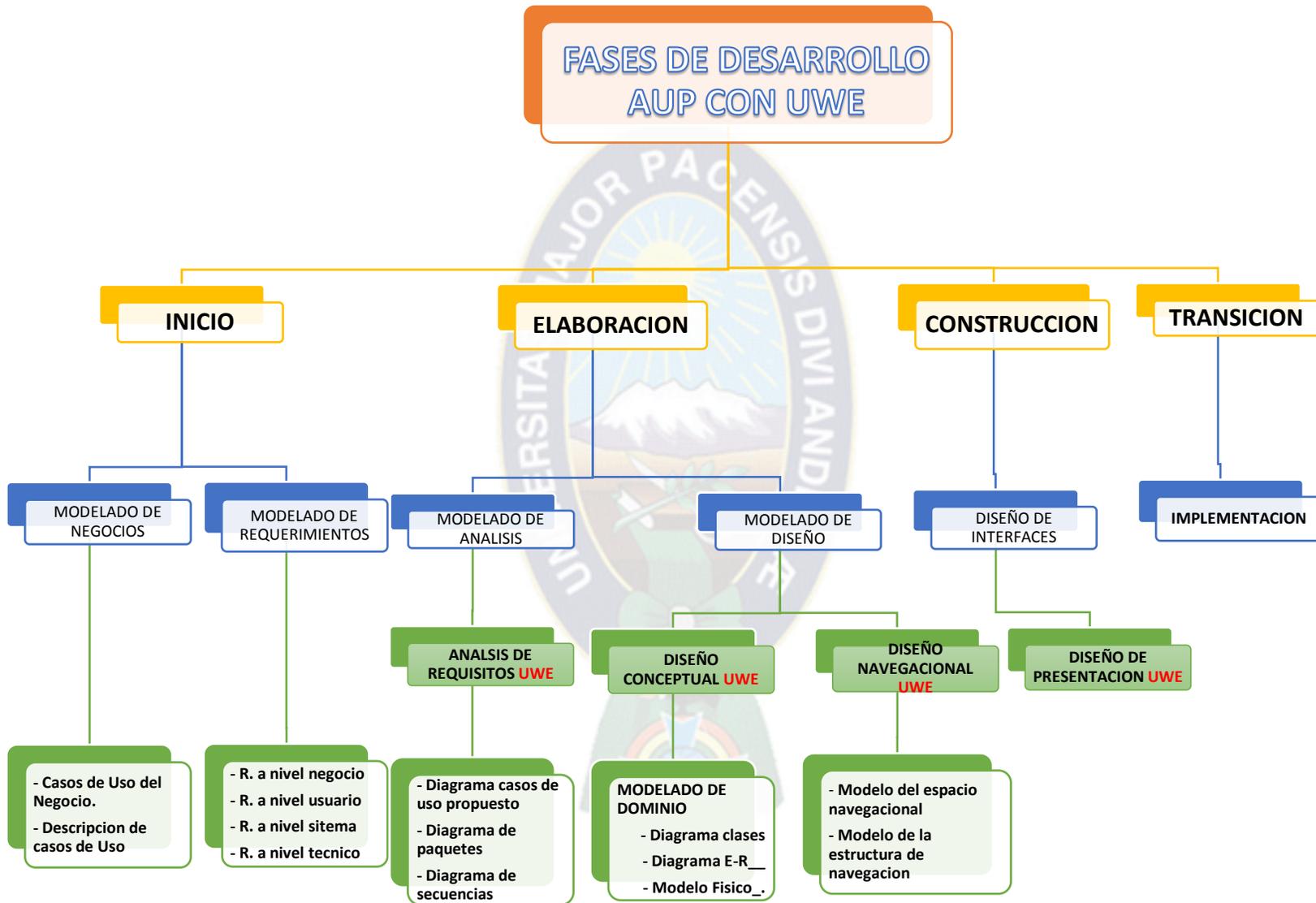


FIGURA 3.2. Combinación Modelos AUP y UWE
FUENTE: Elaboración Propia

3.2 FASE DE INICIO

El objetivo principal de la fase de inicio es archivar el consenso de los interesados del proyecto en relación a los objetivos del proyecto para obtener el financiamiento requerido.

Las principales actividades en esta fase son:

- a) **Definir el alcance del proyecto.** Esto incluye la definición, a un alto nivel, de qué es lo que hará el sistema.
- b) **Estimación de costos y calendario.** En un nivel alto, el calendario y el costo del proyecto son estimados.
- c) **Definición de Riesgos.** La lista de riesgos es una compilación en vivo que cambiará en el tiempo cuando los riesgos serán identificados, mitigados, evitados y / o materializados o exterminados
- d) **Determinar la factibilidad del proyecto.** Su proyecto debe tener sentido desde la perspectiva técnica, operacional y del negocio
- e) **Preparar el entorno del proyecto.** Esto incluye la reserva de áreas de trabajo para el equipo.

3.2.1 MODELADO DEL NEGOCIO

Con el modelado del negocio, se logrará entender mejor el funcionamiento de los respectivos procesos que el personal del INSTITUTO TECNOLÓGICO AYACUCHO.

3.2.1.1 MODELADO DE CASOS DE USO DEL NEGOCIO

En el diagrama de caso de uso del negocio (Ver Figura 3.3) se muestra la operación que se realizan en la inscripción de un alumno, cuando se presenta la convocatoria para el ingreso al instituto, el secretario (a) es la persona que se encargara de recibir los requisitos que se piden, posteriormente se le facilita al alumno una boleta de inscripción el cual se encuentra los datos personales del alumno, la carrera que se encuentra y la matrícula de la gestión actual, entre otros documentos.

El responsable de kardex se encargara posteriormente de revisar los documentos y datos recibidos, el director se encargaran de dar el visto bueno a la postulación del nuevo alumno, como también se muestra la apertura del registro, la responsable de kardex estará encomendada a realizar el registro de los datos y documentos que se recabaron en la postulación, como también tendrá el deber de ejecutar bienales, registrar mensualidades, proporcionar reporte de mensualidades, entre otros.

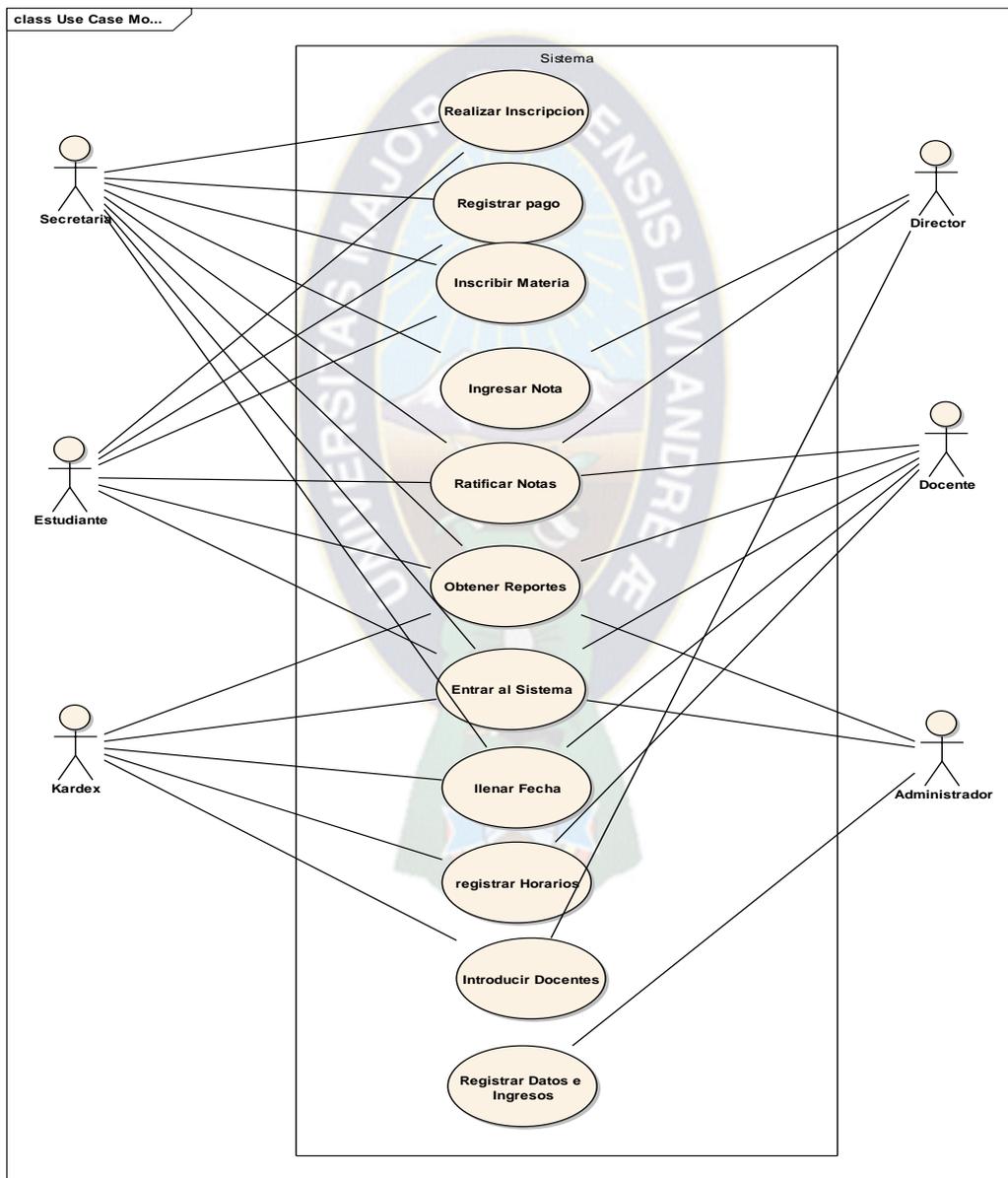


Figura 3.3: Diagrama de casos de uso del negocio
Fuente: Elaboración Propia

3.2.1.2 DESCRIPCIÓN DE ACTORES DEL CASO DE USO DE NEGOCIO

Descripción de los actores de los diagramas de caso de uso del negocio que son respectivamente descritos en la siguiente tabla. (Ver Tabla 3.1)

ACTOR	DESCRIPCIÓN
ADMINISTRADOR	Es el encargado de dar Soporte al sistema, incluyendo crear y eliminar usuarios. También es el encargado de registrar ingresos ,obtener respectivos reportes y ver información de docentes
DIRECTOR	Encargado de planificar y supervisar los aspectos relacionados a las inscripciones, obtener reportes académicos, reportes de ingresos y gastos del Instituto
SECRETARIA	Podrá registrar estudiantes nuevos, ratificar notas por resolución, registrar notas, registrar pago de pensiones y matriculas, asignar fechas de inscripción, generar reportes y realizar inscripción de los estudiantes.
KARDEX	La persona encargada de registrar Docentes, Obtener reportes de records académicos, certificado de Notas, historial académico y registrar Horarios.
DOCENTE	Encargado de guiar al estudiante en toda su carrera, el cual valora al estudiante mediante reportes.
ESTUDIANTE	Solicita su inscripción y realiza su seguimiento académico.

Tabla 3.1. Descripción de actores de casos de uso del negocio
Fuente: Elaboración Propia

3.2.2 MODELADO DE REQUERIMIENTOS

El modelado de requerimientos, es un proceso de descubrimiento, refinamiento, modelado y especificación que nos permite conocer los elementos necesarios para definir un proyecto de software.

Es la fase más crítica del desarrollo de un proyecto de software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el sistema.

Consta de una variedad de productos de trabajo, como ser: requerimientos técnicos, requerimientos de usuario, requerimientos a nivel de negocio. Los requerimientos son la parte final de la fase de inicio.

3.2.2.1 DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL DE NEGOCIOS

Desarrollar un Sistema Web Académico para el Instituto Tecnológico Ayacucho.

3.2.2.2 DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL DE USUARIO

- Registrar los datos personales de un usuario.
- Registrar información personal y documentación entregada del usuario.
- Administrar los datos del usuario, donde se pueda adicionar eliminar, buscar y modificar una determinada información.
- Mantener actualizada la documentación entregada.
- Registrar usuario de administración para mantener la información segura
- Registrar usuarios para tener acceso a todos los datos que se refiere a este.
- Generar historial para tener claro que datos y documentos se necesita actualizar o entregar del usuario.
- Emitir reportes de los datos y documentación recabada.

3.2.2.3 DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL DE SISTEMA

- El sistema debe poseer una interfaz para el registro de los datos personales del usuario.

- El sistema debe tener una interfaz que nos permita elegir entre adicionar, eliminar, buscar y modificar la información del usuario.
- El sistema debe poseer una interfaz para el registro de toda la información personal del usuario.
- El sistema debe de contener una interfaz para registrar el historial de cada usuario.
- El sistema tendrá la opción de imprimir los reportes generados del usuario.

3.2.2.4 DESCRIPCIÓN DE REQUERIMIENTOS A NIVEL TÉCNICOS

Se tiene los requerimientos a nivel hardware y software.

- La codificación será efectuada sobre la plataforma de Microsoft, utilizando Php 5.5.12, MySql 5.6.17.
- Para realizar los reportes se utilizará plugins de php.
- Para el servidor local se utiliza Apache Server 2.5

3.3. FASE DE ELABORACIÓN

El principal objetivo de la fase de elaboración es probar la arquitectura del sistema a desarrollar.

En esta fase se determinan las soluciones técnicas del proyecto durante la cual se elabora los modelos de análisis, diseño, navegación e implementación, de los cuales se desarrollaran los respectivos diagramas.

3.3.1 MODELADO DE ANÁLISIS

3.3.1.1. ANÁLISIS DE REQUISITOS

Su objetivo es encontrar los requisitos funcionales del Sistema Web académico del Instituto Tecnológico Ayacucho para representarlos como casos de uso. Da lugar a un diagrama de casos de uso.

A. MODELO DE CASOS DE USO PROPUESTO

Los casos de uso propuesto, representan una descripción específica de acciones e interacciones entre los actores y el sistema que proporciona valor a un actor en particular, que define un conjunto de instancias de casos de uso.

DIAGRAMA DE CASOS DE USO DE ALTO NIVEL

Los diagramas de casos de uso de alto nivel es donde se trata de realizar una descripción breve de las acciones del caso de uso. (Ver Figura 3.4)

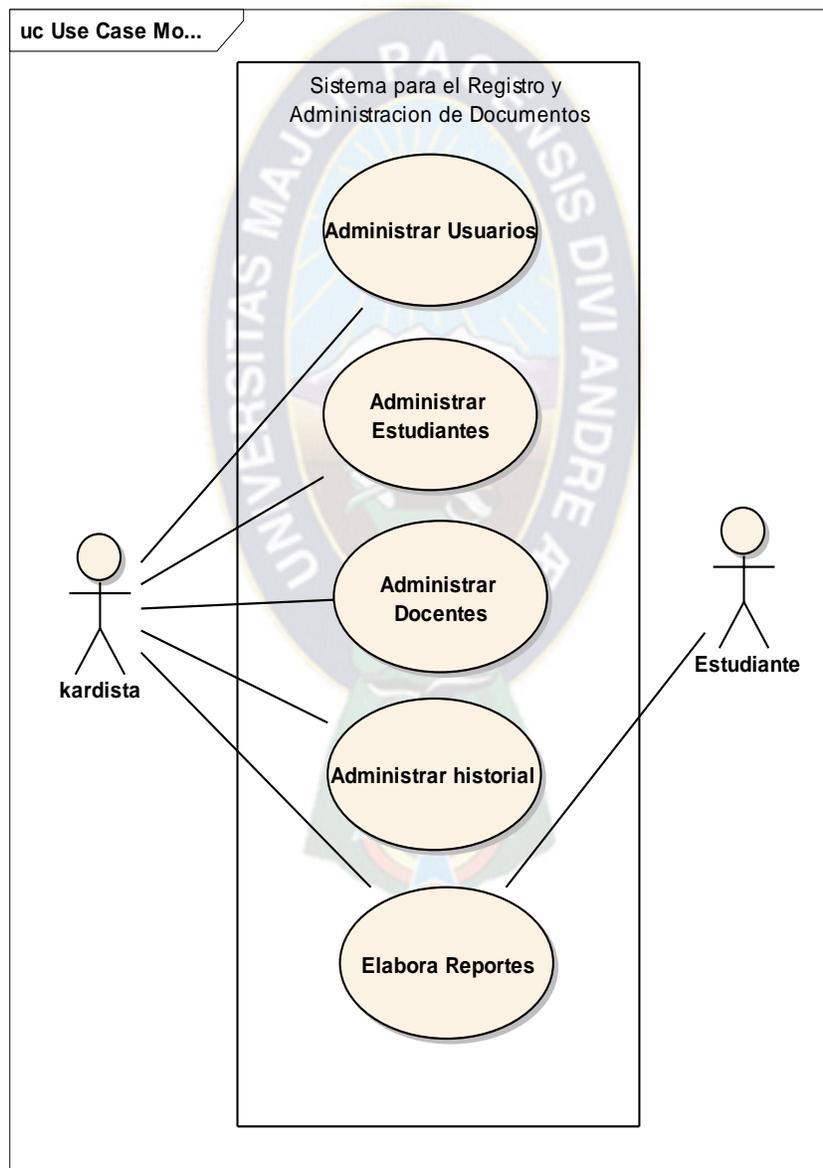


Figura 3.4. Diagrama de casos de uso del sistema de registro y administración
Fuente: Elaboración Propia

B. DESCRIPCIÓN DE CASOS DE USO

A continuación, se describen los diagramas de casos de uso de la figura 3.4, estos diagramas proporcionan una guía para los siguientes flujos de trabajo y su descripción, desde el diseño hasta las pruebas.

i. CASO DE USO: ADMINISTRAR USUARIOS

Administrar usuarios, consiste en la asignación de usuarios al personal que tiene un rol en el sistema, la modificación de usuario en caso de haber cometido algún error al habilitar a algún empleado y finalmente la eliminación de algún usuario que se lo realiza cuando la persona deja de pertenecer al centro. La persona a cargo de estos procesos es el director administrativo. En la siguiente figura se muestra todo lo mencionado. (Ver Figura 3.5)

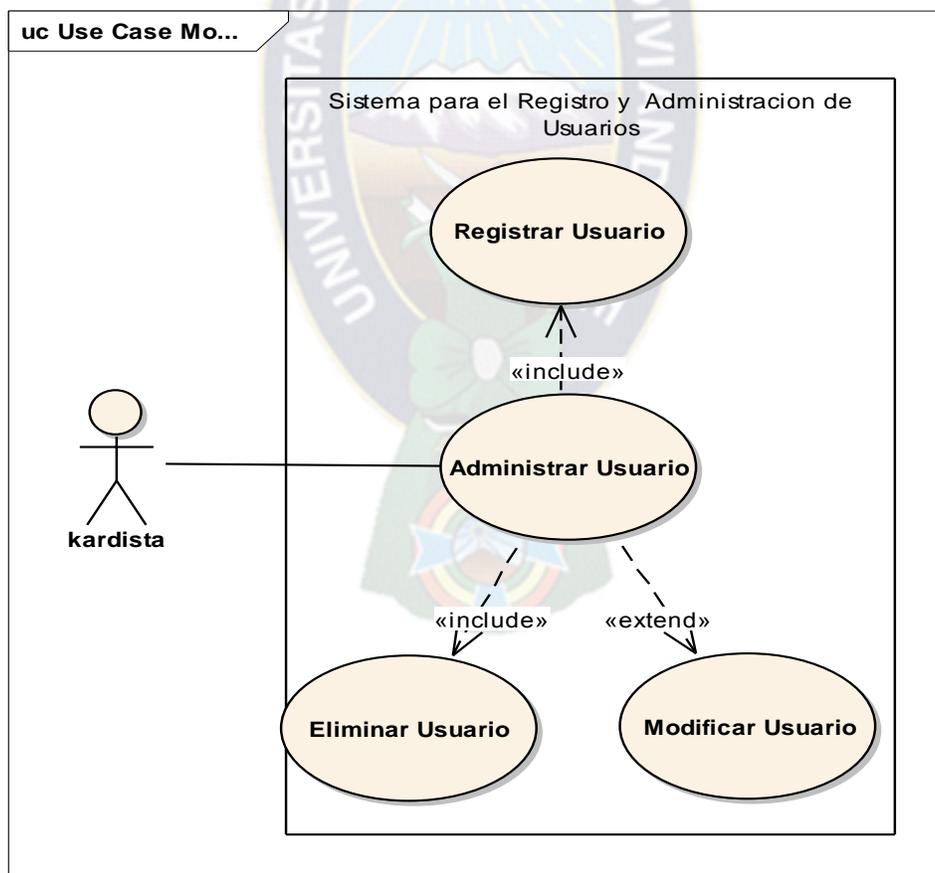


Figura 3.5. Casos de uso de administrar usuario
Fuente: Elaboración Propia

Descripción del diagrama de Casos de Uso: Administrar Usuario	
Nombre:	Administrar Usuario
Autor:	Eddy Richard Agne Quispe
Descripción: Permitir al director administrativo registrar, modificar y eliminar usuario.	
Actores: Kardista	
Precondiciones: El responsable debe ser usuario del sistema y tener el rol de administrador.	
Flujo Normal:	
Actor <ul style="list-style-type: none"> ➤ Autenticación del actor en el sistema. ➤ Elegir la opción administración de usuario. ➤ Elegir la opción de adicionar, eliminar y modificar usuario ➤ Salir del sistema. 	Sistema <ul style="list-style-type: none"> ➤ Verifica datos introducidos en la base de datos. ➤ Registra los datos introducidos en la base de datos.
Flujo Alternativo: En caso de cometer algún error en la administración del usuario, el sistema informara dicho error.	
Post condicione: El sistema guardara el registro en el inicio de postulación.	

Tabla 3.2. Descripción de caos de uso administrar usuario
Fuente: Elaboración Propia

ii. CASO DE USO: ADMINISTRAR ESTUDIANTES

Administrar estudiantes, consiste en registrar los datos principales, que no es más que introducir todos los datos personales y aspectos esenciales, como ser nombres, apellidos, dirección, teléfonos, procedencia de Colegios, entre otros, posteriormente se realiza la

revisión de estos requisitos dando el visto bueno o emitiéndolos según algunos requerimientos no cumplidos. A continuación se detalla lo mencionado en la siguiente figura.

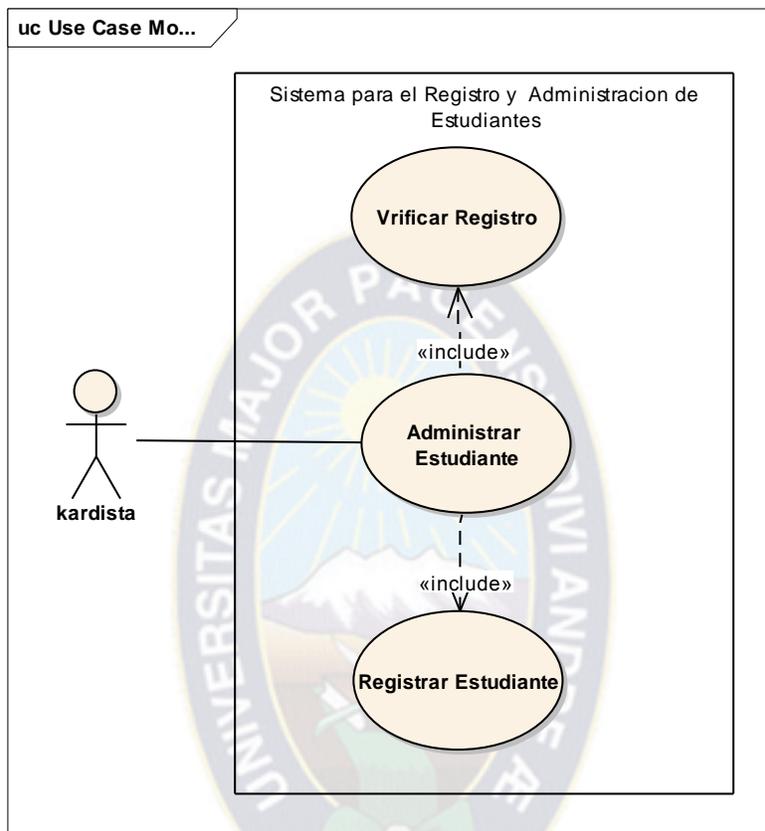


Figura 3.6. Diagrama de casos de uso de administrar estudiantes
Fuente: Elaboración Propia

En la siguiente tabla se describe los diagramas de caso de uso de la figura 3.4 administrar estudiantes

Descripción del diagrama de Casos de Uso: Administrar Estudiantes	
Nombre:	Administrar Estudiante
Autor:	Eddy Richard Agne Quispe
Descripción:	Permitir al Kardista registrar datos del Estudiante.

Actores: Kardista	
Precondiciones: El responsable debe ser usuario del sistema y tener el rol correspondiente.	
Flujo Normal	
Actor ➤ Elegir la opción de registrar.	Sistema ➤ Almacena la información introducida en la base de datos. ➤ Salir del sistema.
Flujo Alternativo: En caso de cometer algún error en el proceso de registro el sistema informara dicho error.	
Post condicione: El sistema guardara el registro de estudiante.	

Tabla 3.3. Descripción de casos de uso administrar Estudiantes
Fuente: Elaboración Propia

iii. CASO DE USO: ADMINISTRAR DOCENTE

La administración del docente, consiste en el registro del docente, para esto se debe introducir todos los datos personales de este que sea importantes para la institución.

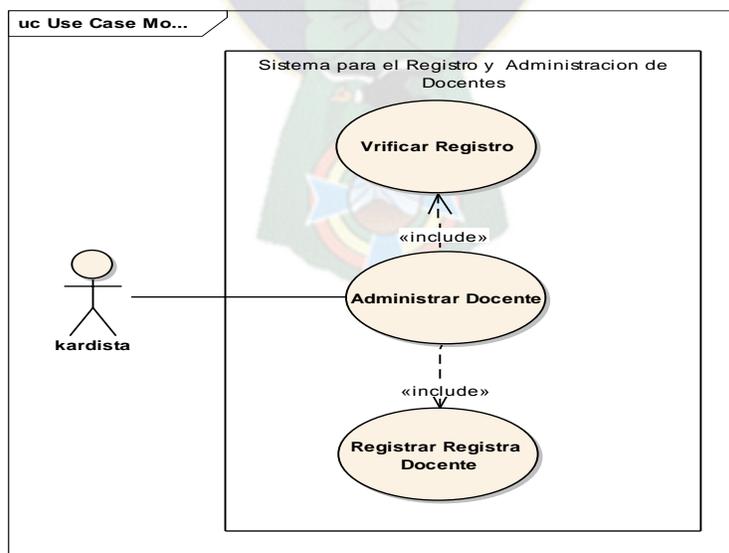


Figura 3.7. Diagrama de casos de uso administrar Docente
Fuente: Elaboración Propia

Descripción del diagrama de Casos de Uso: Administrar Docente	
Nombre:	Administrar Docente
Autor:	Eddy Richard Agne Quispe
Descripción: Permitir al kardista, buscar, modificar y eliminar Docente.	
Actores: Kardista	
Precondiciones: El responsable debe ser usuario del sistema y tener el rol correspondiente.	
Flujo Normal:	
Actor	Sistema
<ul style="list-style-type: none"> ➤ Autenticación del actor en el sistema. ➤ Elegir la opción de adicionar, buscar, modificar y cancelar. ➤ Salir del sistema 	<ul style="list-style-type: none"> ➤ Se guardara todos los datos introducidos en la base de datos si es el caso de registrar. ➤ Se rebuscara en la base de datos y mostrara toda la información registrada, para modificar o cancelar.
Flujo Alternativo: En caso de cometer algún error en el proceso de registro el sistema informara dicho error.	
Post condiciones: El sistema guardara el registro en administrar datos y documentos.	

Tabla 3.4. Descripción de casos de uso administrar Docente
Fuente: Elaboración Propia

iv. CASO DE USO: ADMINISTRAR HISTORIAL

Administrar historial, consiste en el registro del nuevo historial que se le asignara al estudiante, en donde se jalaran los datos personales, que se le asignó al momento de registrarse como usuario, que se encuentran guardados de la base de datos, El encargado de estos procesos es el kardista que su labor será el de registrar mensualidades, notas respectivas. En la siguiente figura, se puede observar todo lo detallado anteriormente. (Ver Figura 3.8)

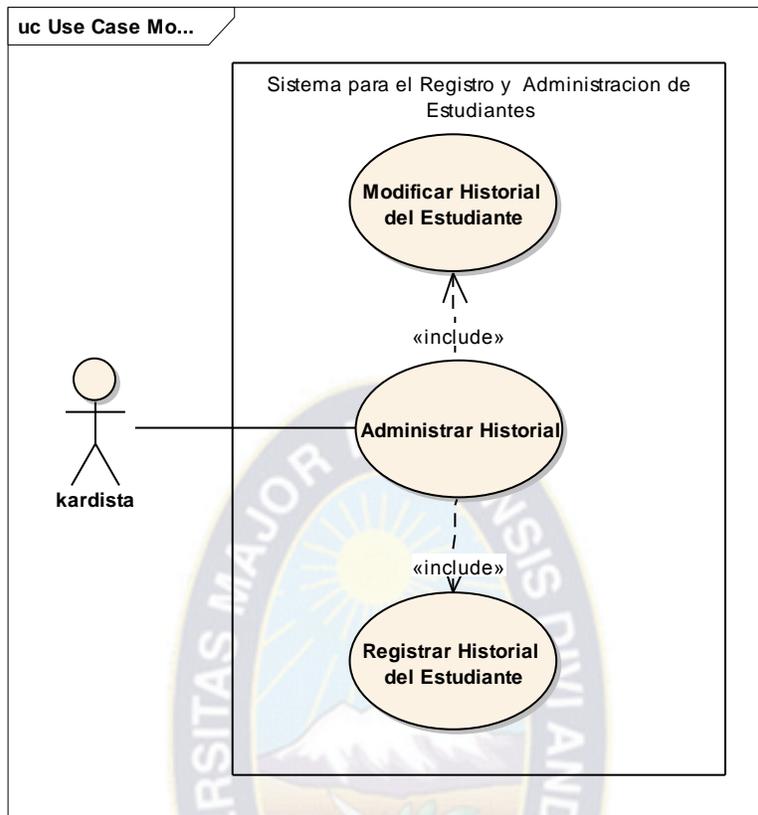


Figura 3.8. Diagrama de casos de uso administrar Historial.
Fuente: Elaboración Propia

En la siguiente tabla se describe los casos de uso de la figura 3.8 administrar historial.

Descripción del diagrama de Casos de Uso: Administrar Historial	
Nombre:	Administrar Historial del Estudiante
Autor:	Eddy Richard Agne Quispe
Descripción:	Permitir al Kardista registrar y modificar el historial de un Estudiante.
Actores:	Kardista
Precondiciones:	El responsable debe ser usuario del sistema y tener el rol correspondiente.
Flujo Normal:	

<ul style="list-style-type: none"> ➤ Autenticación del actor en el sistema. ➤ Elegir la opción de registrar, modificar historial. ➤ Salir del sistema. 	<ul style="list-style-type: none"> ➤ Se almacenara los datos introducidos en a base de datos. ➤ Buscará el código y desplazara la información registrada.
<p>Flujo Alternativo: En caso de cometer algún error en el proceso de registro el sistema informara dicho error.</p>	
<p>Post condicione: El sistema guardara el registro en operar mensualidades y asistencias.</p>	

Tabla 3.5. Descripción de casos de uso registrar historial

Fuente: Elaboración Propia

v. CASO DE USO: ELABORAR REPORTES

Elaborar reportes, consiste en imprimir los datos del Estudiante, Docente, Administrativo, información del historial del Estudiante, donde también se encuentra la documentación entregada de este, se realiza una búsqueda, introduciendo el código referencial, el sistema buscara estos datos y desplazara la información correspondiente, además de un reporte de estadísticas sobre todos los estudiantes. De estos procesos se encarga el Kardista, también el Estudiante accediendo al sistema podrá verificar estos datos. A continuación se puede observar la siguiente figura donde se describen los procesos explicados anteriormente.

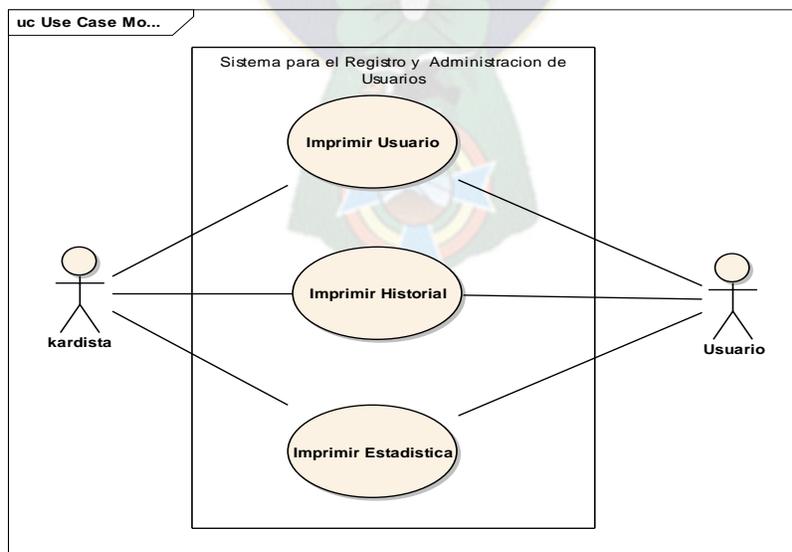


Figura 3.9. Casos de uso de reportes

Fuente: Elaboración Propia

Descripción del diagrama de Casos de Uso: Elaborar Reportes	
Nombre:	Elaborar Reportes
Autor:	Eddy Richard Agne Quispe
Descripción: Permitir al Kardista imprimir los datos del Usuario, historial como también estadísticas.	
Actores: Kardista, Usuario	
Precondiciones: El kardista y usuarios, deben ser usuarios del sistema y tener el rol correspondiente.	
Flujo Normal:	
Actor	Sistema
<ul style="list-style-type: none"> ➤ Autenticación del actor en el sistema. ➤ Elegir la opción de imprimir usuario, historial del usuario y estadísticas. ➤ Salir del sistema. 	<ul style="list-style-type: none"> ➤ Verificara datos introducidos para acceder al sistema. ➤ Se buscara el código desplazando la información y para imprimir
Flujo Alternativo: En caso de cometer algún error en el proceso de registro el sistema informara dicho error.	
Post condicione: El sistema guardara el registro en el inicio de postulación e historial.	

Tabla 3.6 Descripción de caos de uso de reportes
Fuente: Elaboración Propia

C. DIAGRAMA DE PAQUETES

El diagrama de paquetes muestra la forma en la que el Sistema web para el registro y administración, está dividido en agrupaciones lógicas mostrando la dependencia entre ellas,

los diagramas de paquetes muestran la descomposición jerárquica lógica de un sistema, es decir se muestra un esquema de los módulos que comprende nuestro sistema. A continuación se muestra el diagrama de paquetes en la figura para entender la organización del sistema. (Ver Figura 3.10)

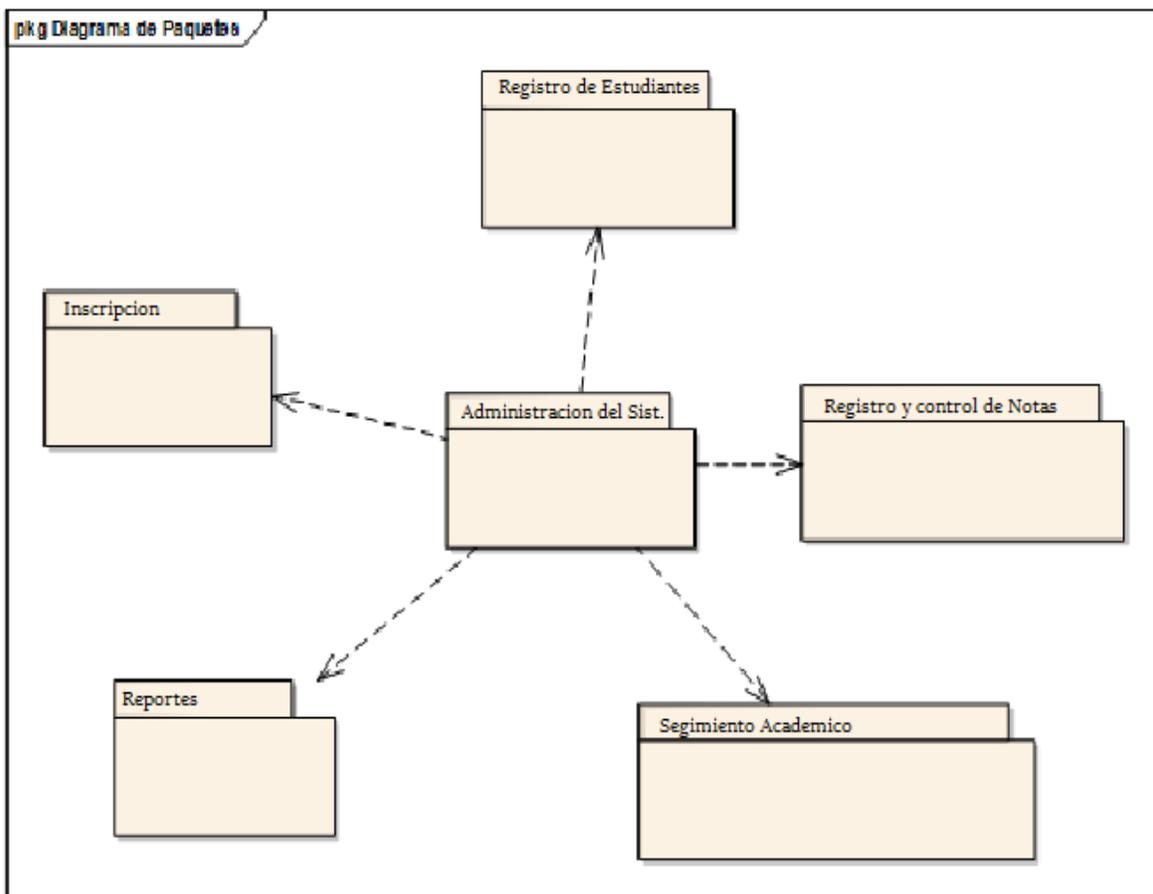
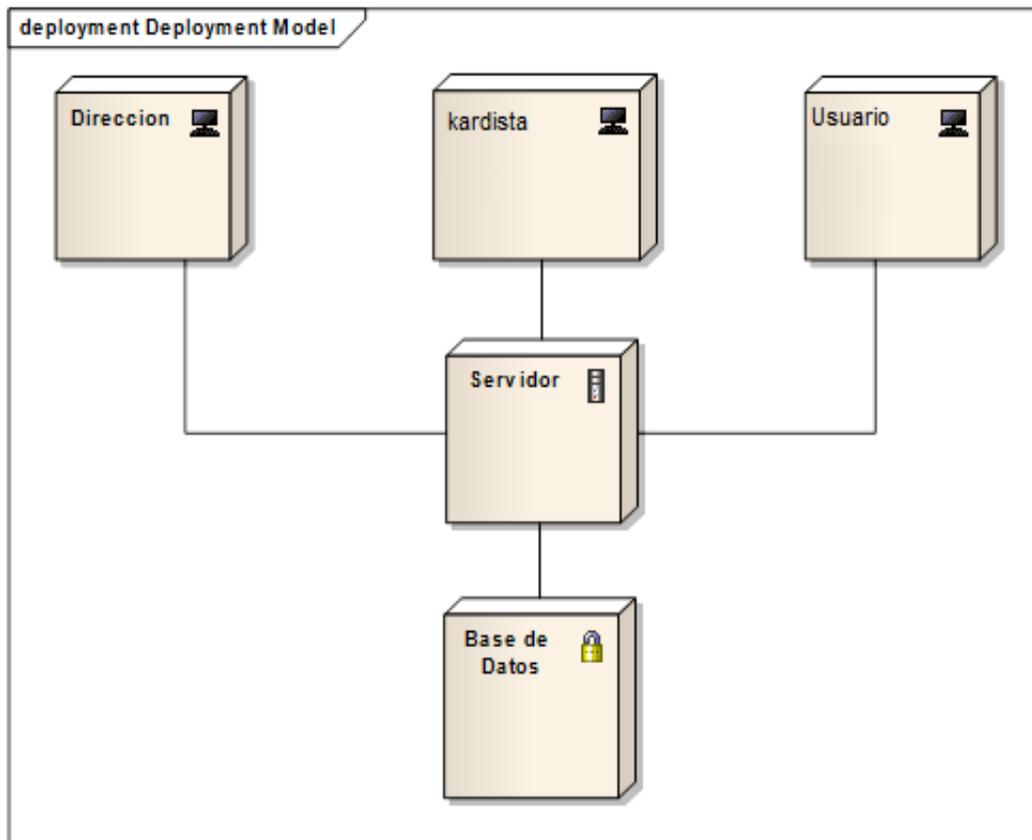


Figura 3.10 Diagrama de paquetes del sistema de registro y administración de documentos
Fuente: Elaboración Propia

E. DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue representa, como y donde se instalara el sistema para el registro y administración de documentos, que está realizada en forma física, para este caso se presenta el servidor, la base de datos, el kardista, la dirección y el usuario, los cuales tendrán el acceso al sistema para el registro y administración de documentos de los usuarios, la figura 3.11. nos ilustra una configuración básica.



Fuente 3.11. Diagrama de Despliegue
Fuente: Elaboración Propia

F. DIAGRAMA DE SECUENCIA

El diagrama de secuencia del sistema muestra de manera gráfica los eventos que propician los actores directos con el sistema. Para su elaboración deberá haberse elaborado, previamente los casos de uso específicos, de los cuales se obtiene el curso normal de eventos, teniendo en cuenta los cursos opcionales más interesantes. [LARM, 1999].

Autenticación.- Empieza cuando un usuario ingresa a la página de autenticación del sistema. El sistema muestra una pantalla donde pide introducir usuario y contraseña, una vez digitados los datos por el usuario, el sistema verifica estos y dependiendo del usuario muestra las pantallas de interfaz y el usuario puede acceder al sistema. En la siguiente figura se muestra este proceso. (Ver Figura 3.12)

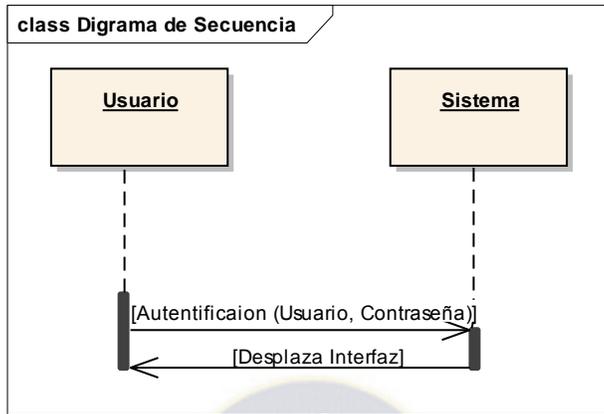


Figura 3.12. Diagrama de Secuencia: Autenticación
Fuente: Elaboración Propia

Administración de Usuarios.- Cuando el kardista ya está accediendo al sistema, puede administrar usuarios, registrando modificando o eliminando, en la siguiente figura se observa este procedimiento. (Ver Figura 3.13)

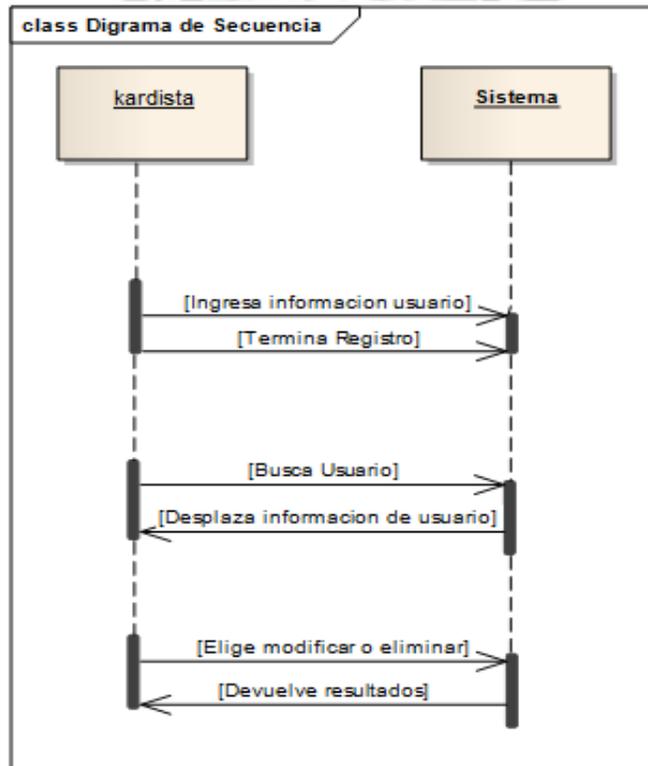


Figura 3.13. Diagrama de Secuencia: Administración Usuarios
Fuente: Elaboración Propia

Administración Estudiante.- El kardista una vez autenticado, elije registrar al estudiante, llenando todos los campos requeridos, una vez terminado presionara registrar, el kardista elegirá modificar o eliminar, según lo que se quedó en la verificación de requisitos. En la siguiente figura se observa este proceso. (Ver Figura 3.14)

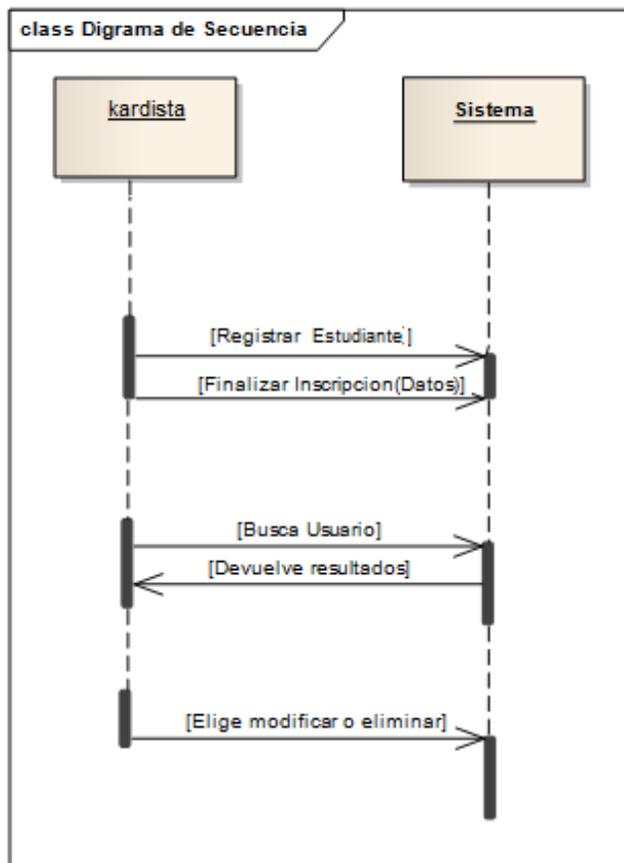


Figura 3.14. Diagrama de Secuencia: Administrar Estudiante
Fuente: Elaboración Propia

Administración Historial.- El kardista es el que se encarga de esta parte una vez pasado el nivel de registro del usuario, el podrá registrar sus documentos en el historial, este elegirá la opción de registrar historial, el sistema desplegara una tabla donde pedirá la cedula del usuario, el sistema devolverá los datos completos del Usuario, igualmente el sistema devolverá toda la información registrada en la base de datos, después de todo esto, el kardista empezara a registrar los documentos que son requeridos, y finalizara con la opción de registrar.

También el kardista podrá elegir la opción de editar el historial por x motivos, por el que tendrá la opción de buscar por código, el sistema desplegara la información registrada en el historial, donde logrará modificar datos por otros a su vez actualizarlos. En la siguiente figura se observa lo mencionado. (Ver Figura 3.15)

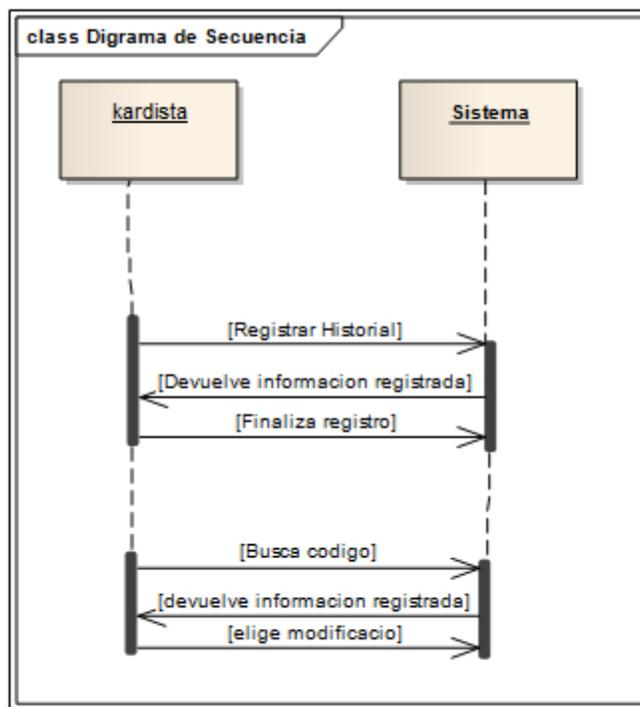


Figura 3.15. Diagrama de Secuencia: Administración Historial
Fuente: Elaboración Propia

Administración Reporte.- En este módulo podrán tener acceso tanto como el kardista, usuarios y administradores, previamente su autenticación, llegaran a tener acceso a esta parte, el kardista podrá realizar un reporte tanto como del Estudiante, como del historial del Docente, este elegirá la opción de búsqueda por código, e sistema desplegara todos los datos registrados en la base de datos, el kardista elegirá la opción de imprimir, el sistema devolverá la información en formato pdf y Microsoft office Excel. Si el Usuario es el que tiene acceso al sistema automáticamente el sistema desplegara sus datos más la documentación entregada, elegirá la opción de imprimir, el sistema devolverá los datos registrados en formato pdf o Microsoft office Excel. De los cuales tanto como Karditsa o usuario podrán imprimir esta información.

Además se cuenta con la opción de estadísticas, para tener un panorama más claro de los registros de cada año. (Ver Figura 3.16)

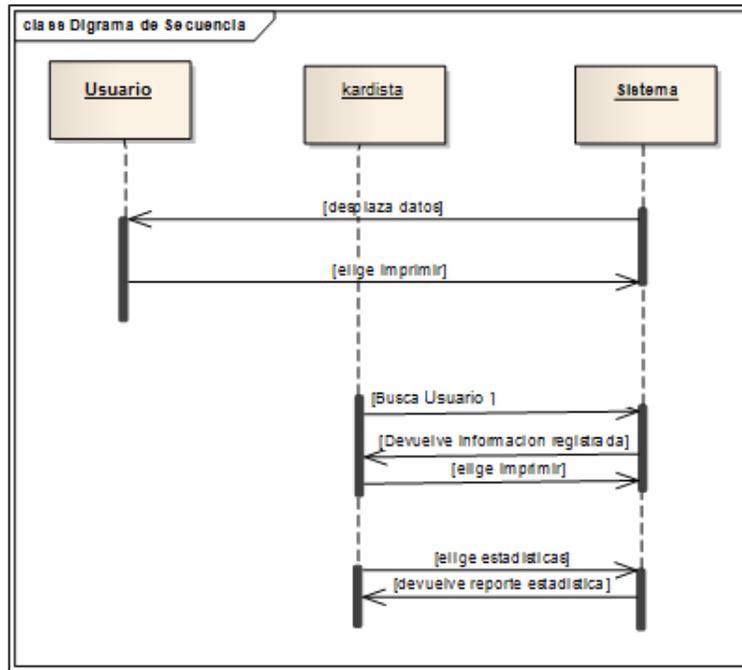


Figura 3.16. Diagrama de Secuencia: Elaboración de Reportes
Fuente: Elaboración Propia

3.3.2. MODELO DE DISEÑO

El modelo de diseño refleja decisiones en cuanto a asignación de responsabilidades entre las operaciones, muestra cómo se relacionan componentes de software para resolver el problema planteado es el paso previo a la implementación.

3.3.2.1. DISEÑO CONCEPTUAL

- **Diagrama de clases**

Representan los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación.

La clase define el ámbito de definición de un conjunto de objetos, cada objeto pertenece a una clase, los objetos se crean por instanciación de las clases. Un diagrama de clases está

compuesto por los siguientes elementos: nombre de la clase, atributos, operaciones, sus relaciones son de herencia, composición, agregación, asociación y uso.

Este es el diagrama principal para el análisis y diseño, en la figura se representan las relaciones entre las clases, atributos y sus operaciones para representar la información del sistema. Después de haber realizado el análisis para la base de datos e identificar todas las entidades que intervienen en el sistema, se elabora el diagrama de clases como se muestra a continuación.

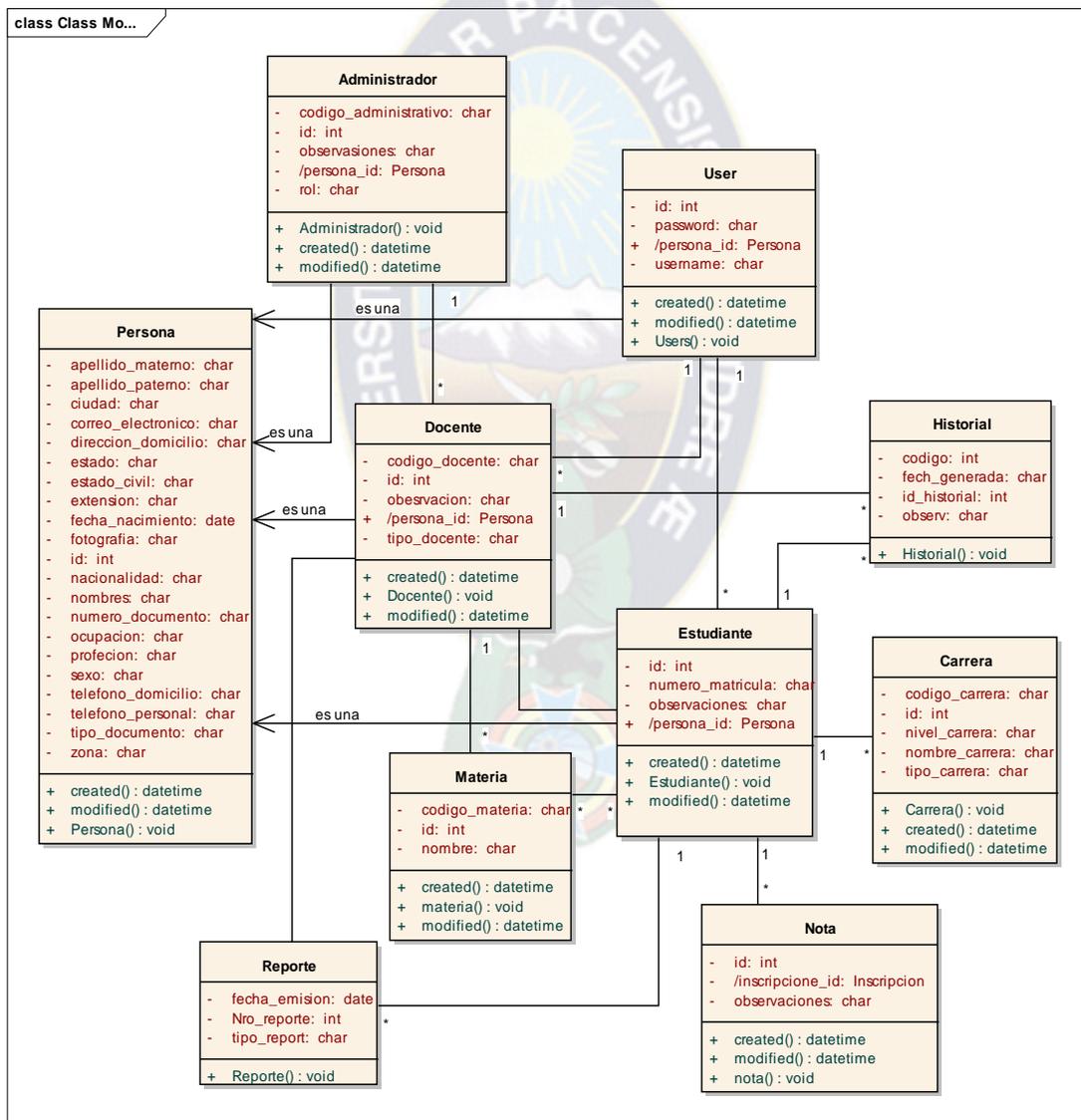


Figura 3.17. Diagrama de clases del sistema
Fuente: Elaboración Propia

▪ **DIAGRAMA ENTIDAD RELACIÓN**

El diagrama Entidad/Relación se fundamenta en entidades y en las interrelaciones que se constituyen entre estas entidades y sitúa énfasis en el tipo de interrelaciones y la cardinalidad que existe entre los elementos. (Ver Figura 3.18)

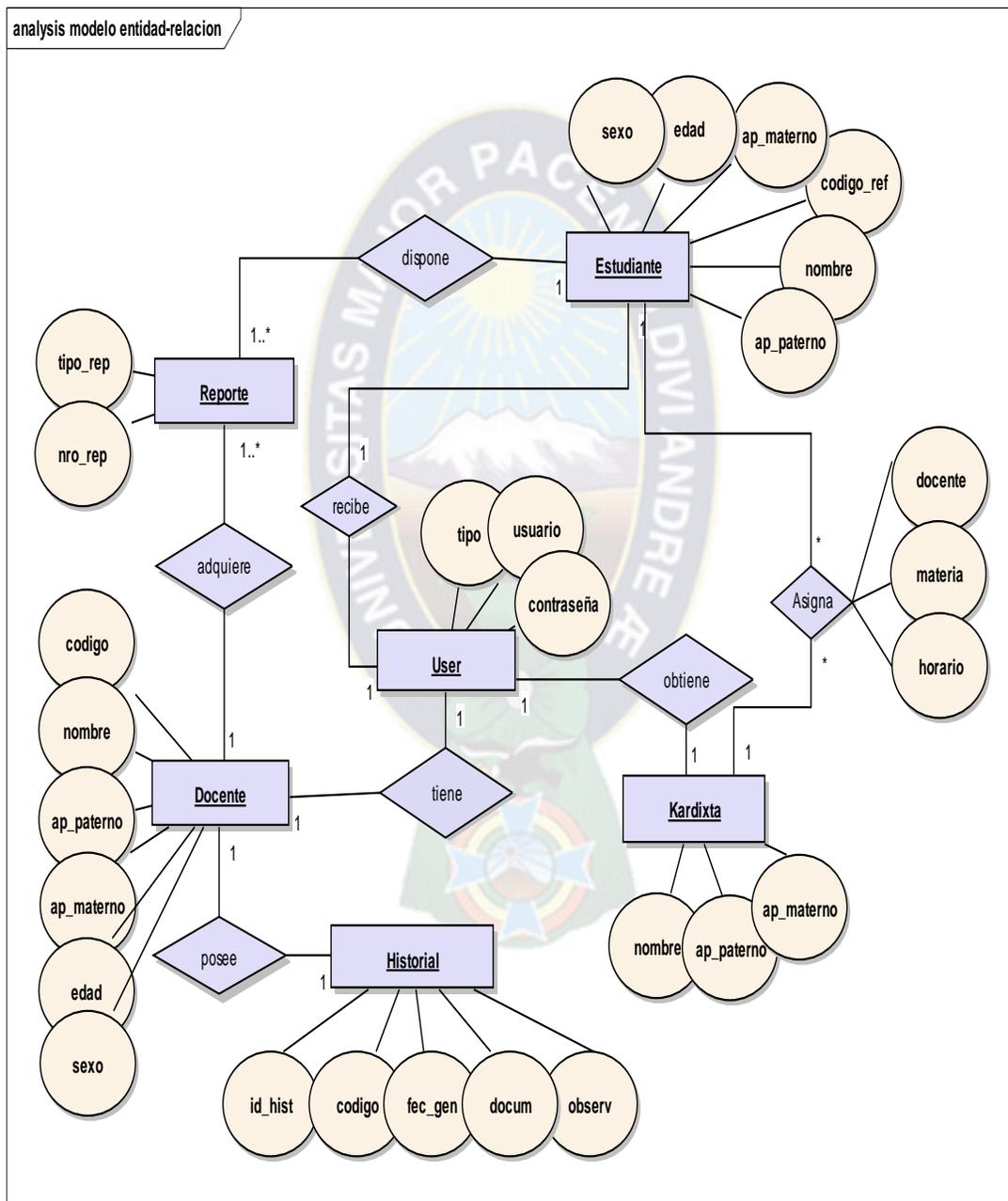


Figura 3.18. Modelo Entidad Relación
Fuente: Elaboración Propia

▪ MODELO FÍSICO

En esta parte el objetivo es transformar el esquema conceptual obtenido en la fase previa, adaptándola al modelo de datos (relacional) que se va a utilizar y de acuerdo a un sistema de gestión de base de datos particular.

La meta de esta fase es producir un esquema físico donde muestre las llaves primarias y los atributos respectivos, que sea eficiente para las operaciones de consulta y actualización en el sistema. En la siguiente figura se observa todo lo mencionado. (Ver Figura 3.19)

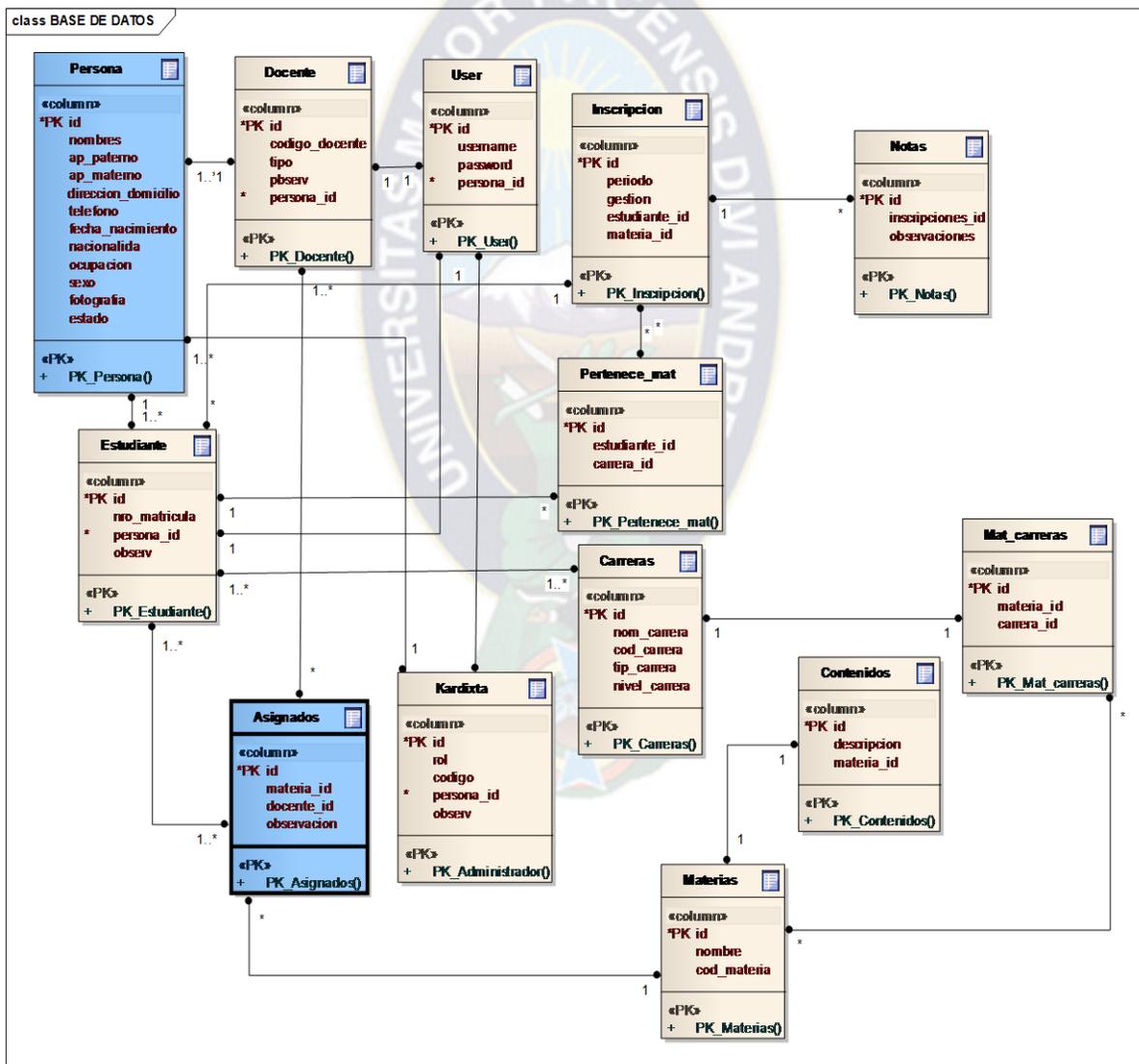


Figura 3.19. Modelo físico de base de datos
Fuente: Elaboración Propia

3.3.2.2.DISEÑO NAVEGACIONAL

A. MODELO DEL ESPACIO NAVEGACIONAL

En la figura se muestra el área de navegación que detalla los objetos que logran ser visitados mediante la navegación de los usuarios, especificados por el rol que poseen.

En el siguiente diagrama de navegación se observa dos usuarios: administrador, y el usuario y en cada uno se percibe su respectiva navegación. (Ver Figura 3.20)

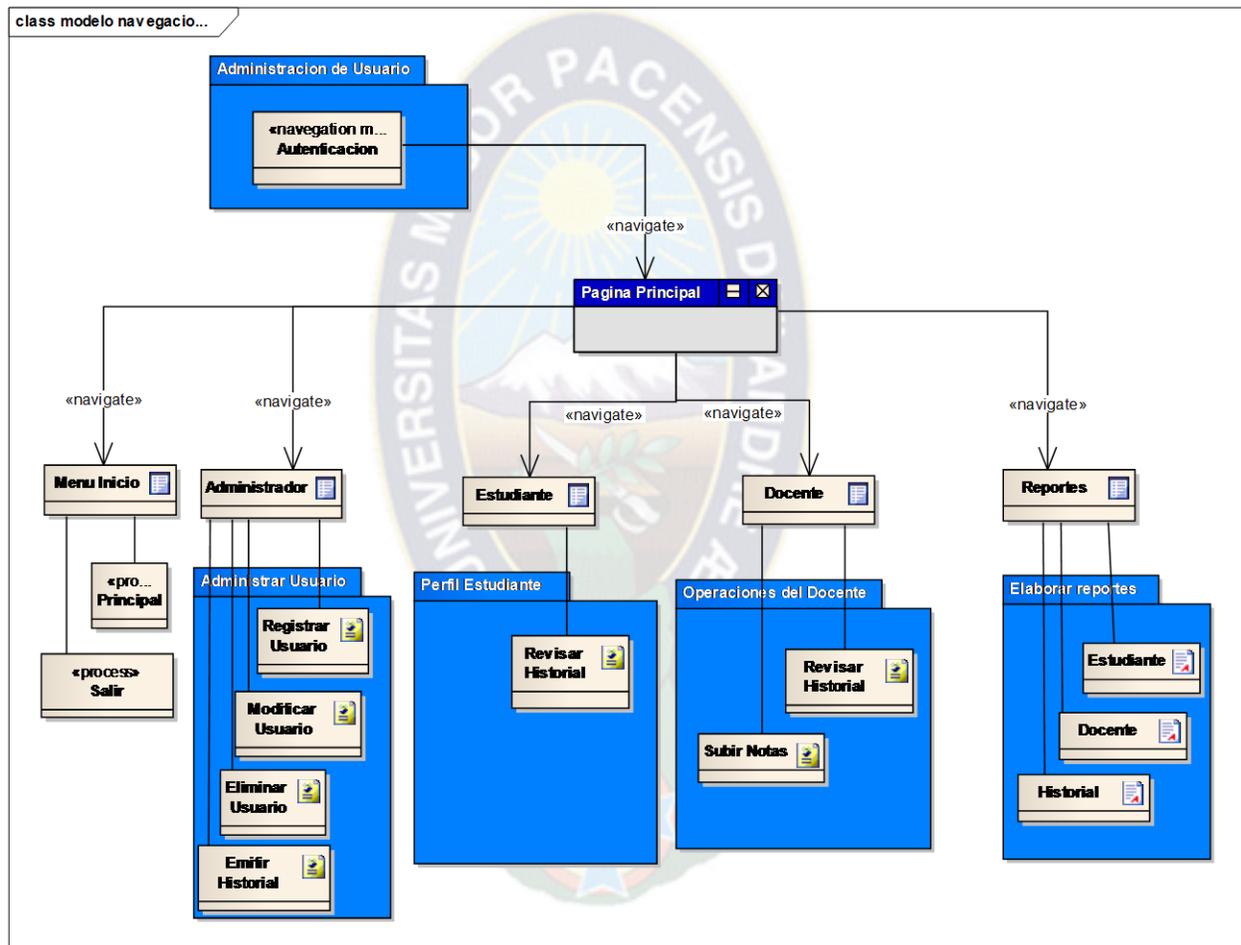


Figura 3.20. Modelo de espacio de navegación
Fuente: Elaboración Propia

B. MODELO DE LA ESTRUCTURA DE NAVEGACIÓN

La estructura de navegación es un diseño de cómo se exhibe nuestro sistema, obviamente dependiendo del usuario que acceda, es decir dependiendo el rol que presente el usuario.

En la figura 3.21 se llega a observar los accesos del menú que posee el respectivo usuario, cada usuario que requiera ingresar al sistema, deberá previamente autenticarse antes de ingresar.

Si el usuario ingresado es administrador, tiene la opción de administrar usuarios, como el de registro, modificación y eliminación, administrar, como el de registrar, actualizar y buscar, además de administrar historial de registro, edición y búsqueda, el administrador también estará encargado de imprimir reportes.

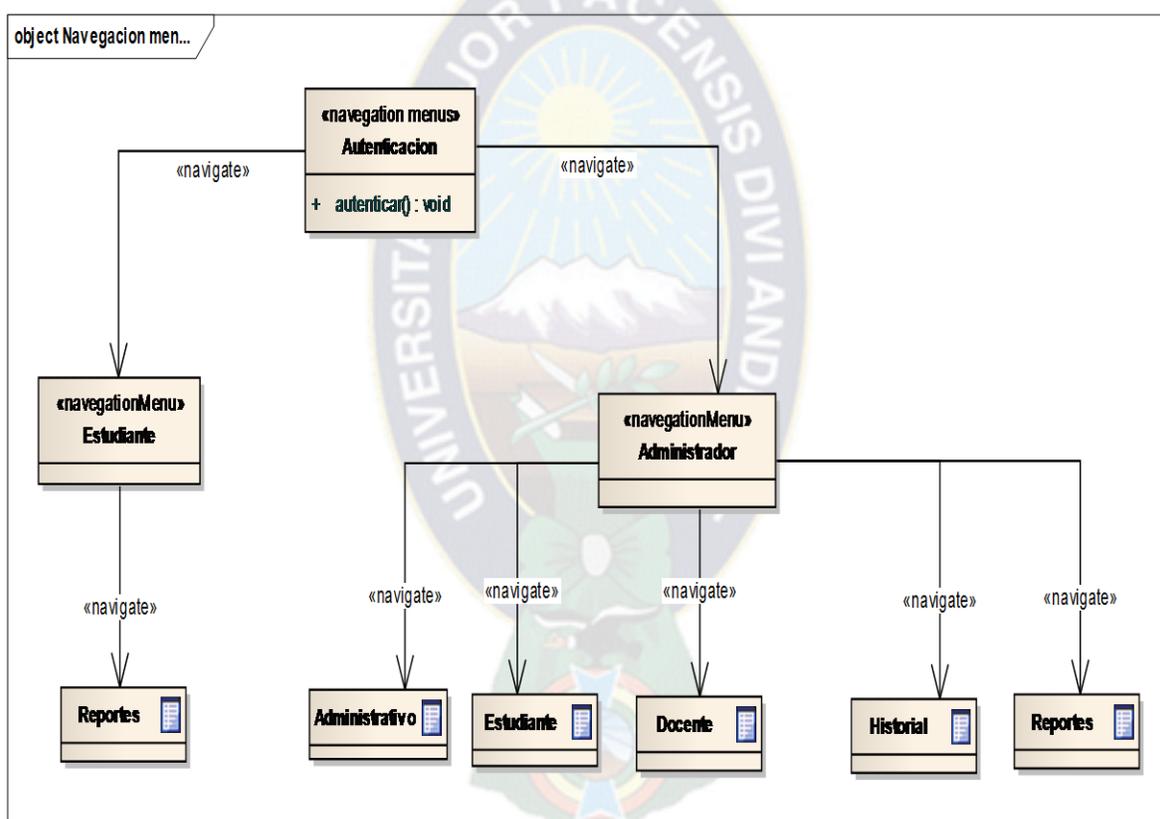


Figura 3.21 Modelo de navegación acceso de usuarios
Fuente: Elaboración Propia

3.4. FASE DE CONSTRUCCIÓN

El objetivo de esta fase consiste en desplegar el sistema hasta el punto en que esté listo para producción de pruebas, abordaremos el diseño de interfaces abstractas que es parte de la metodología UWE.

3.4.1. DISEÑO DE INTERFACES

Una vez finalizado el diseño navegacional, será necesario especificar las diferentes interfaces de la aplicación. El diseño de interfaz se desarrolló siguiendo el modelado de requerimientos y el modelado de diseño.

3.4.2 DISEÑO DE PRESENTACIÓN

El modelo de presentación nos permite mostrar en forma abstracta la interfaz de usuario del sistema, expresando el contenido y estructura de nodos simples. Primeramente se muestra el modelo de presentación de la página principal del sistema. (Ver figura 3.22)

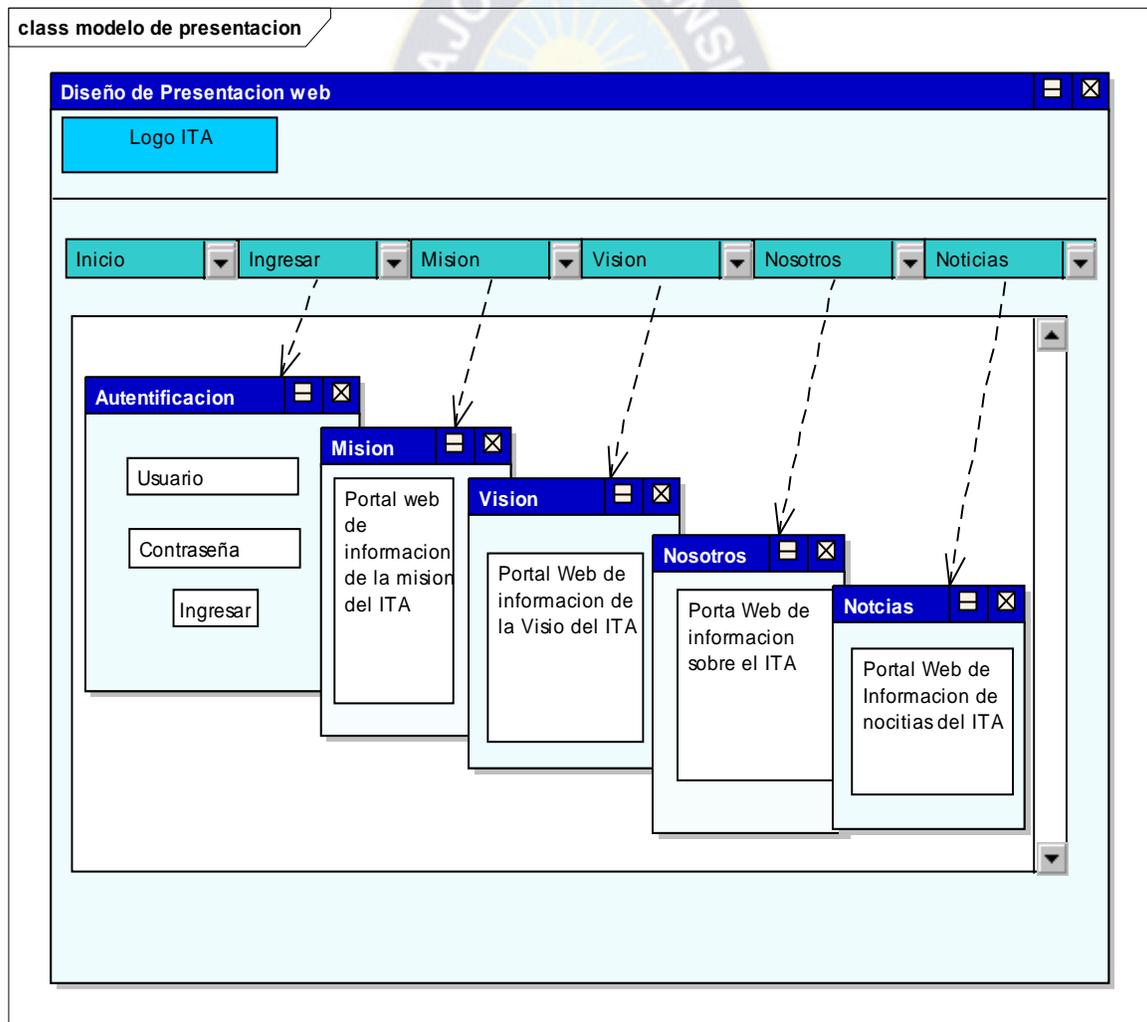


Figura 3.22 Diseño de presentación
Fuente: Elaboración Propia

A. AUTENTIFICACIÓN

El usuario que desee acceder al sistema debe de tener previamente, su cuenta de usuario y contraseña, que son asignados por el administrador del sistema. (Ver Figura 3.23)

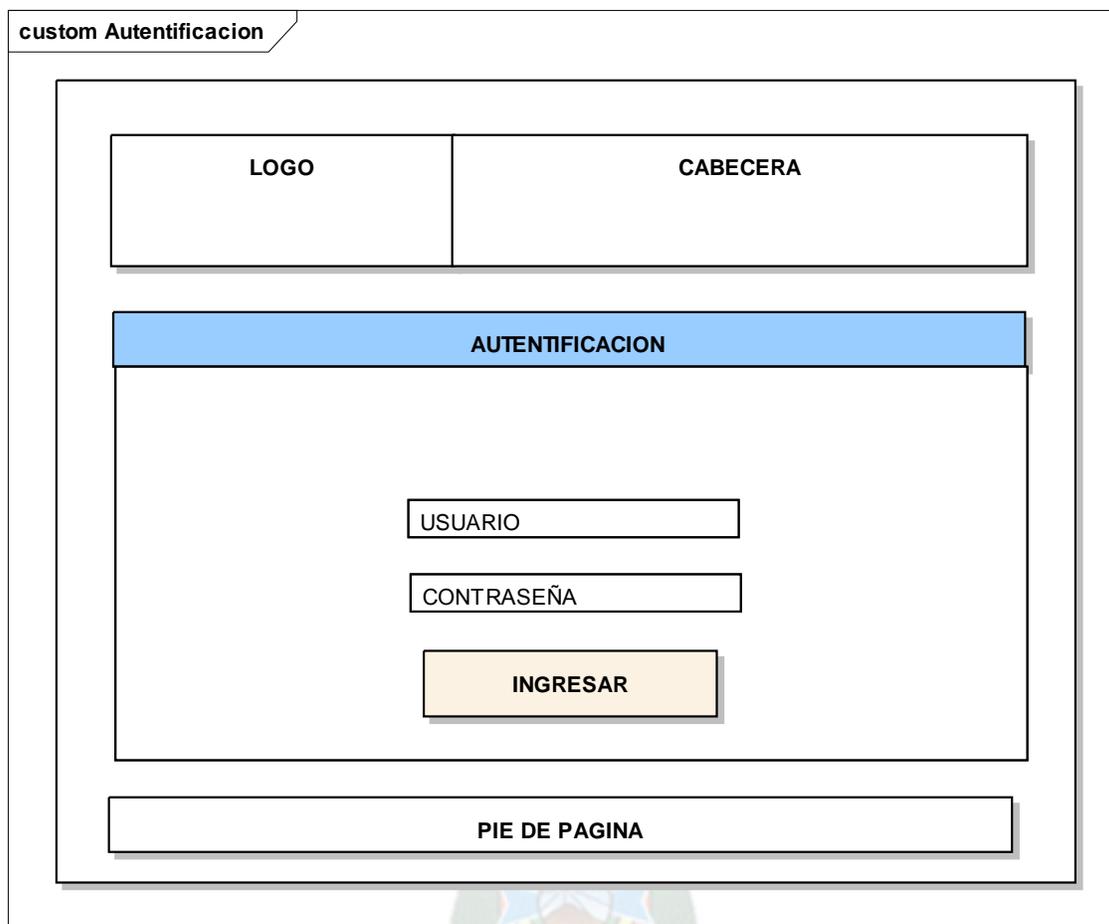


Figura 3.23. Autenticación del sistema
Fuente: Elaboración Propia

B. INICIO

Una vez que el usuario ha ingresado al sistema con su usuario y contraseñas respectivas, el sistema le permite el acceso a la página de Inicio, donde dependerá del usuario ingresado, el acceso a diferentes opciones del menú del sistema que se muestra a continuación. (Ver Figura 3.24)

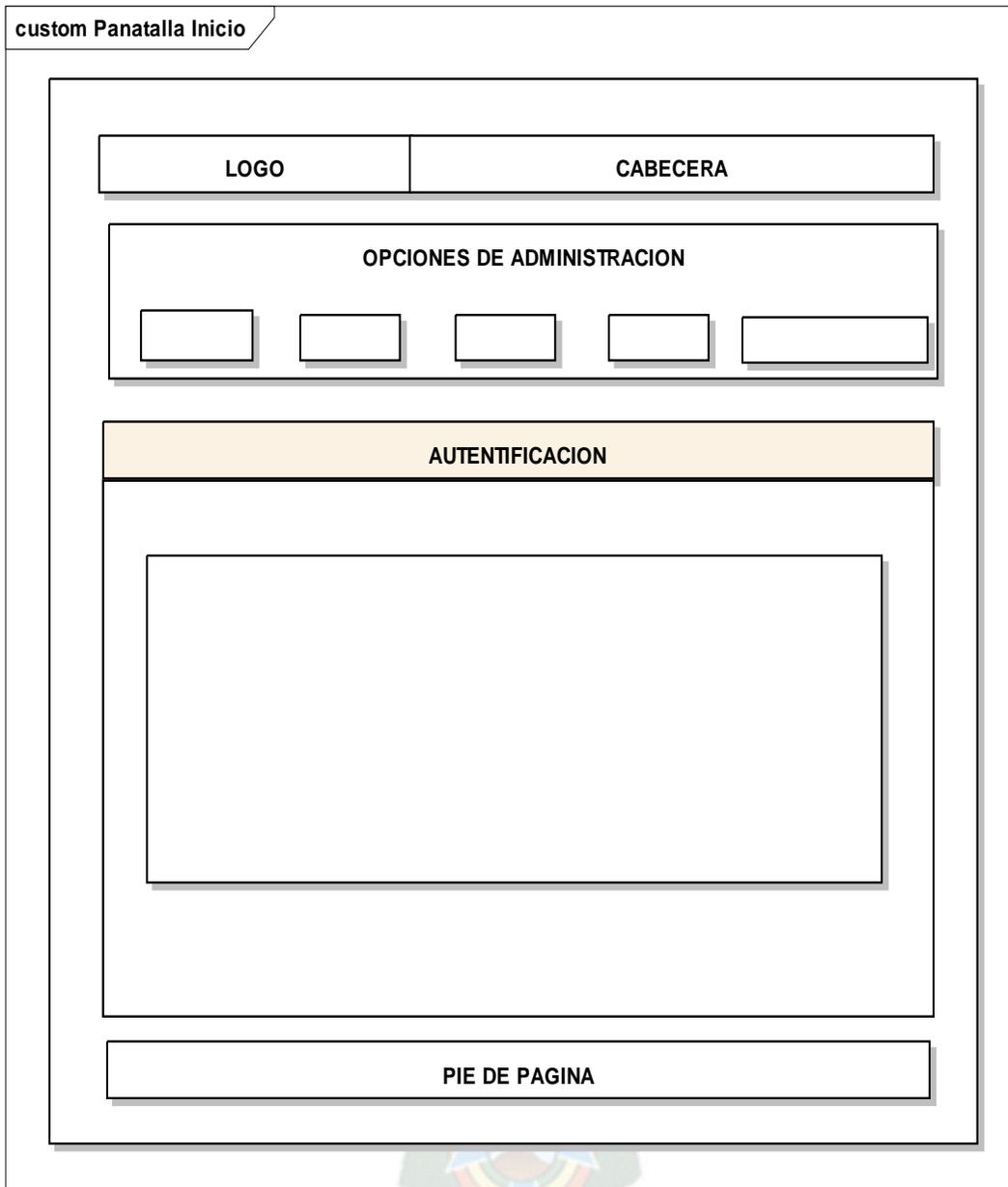


Figura 3.24. Pantalla de Inicio del sistema
Fuente: Elaboración Propia

C. ADMINISTRACIÓN DE USUARIO

La administración de usuario se realiza al desplegar el menú de usuario, donde se muestra tres opciones de elección principales, que son el de registro de usuario, modificación de usuario y eliminación de este, la siguiente figura muestra estos detalles. (Ver figura 3.25)

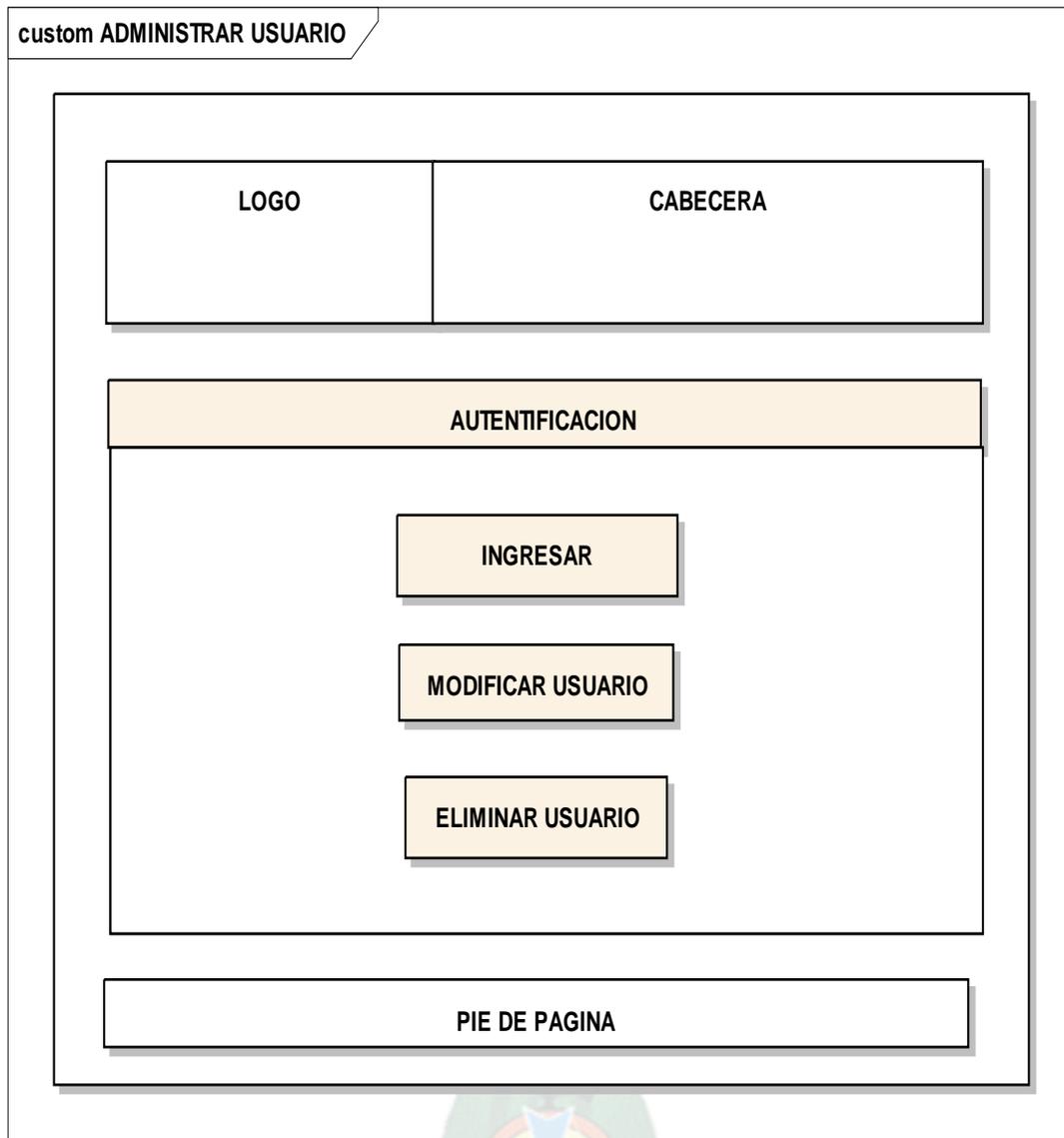


Figura 3.25. Pantalla Administración de Usuario
Fuente: Elaboración Propia

D. REGISTRAR USUARIO

Al desplegar la opción usuario, se elige la opción de registro de usuario en donde se llenan los datos respectivos del usuario, como ser nombre, usuario, contraseña, tipo de usuario, y se elige la opción de registrar y limpiar, en la siguiente figura se observa el detalle. (Ver Figura 3.26)

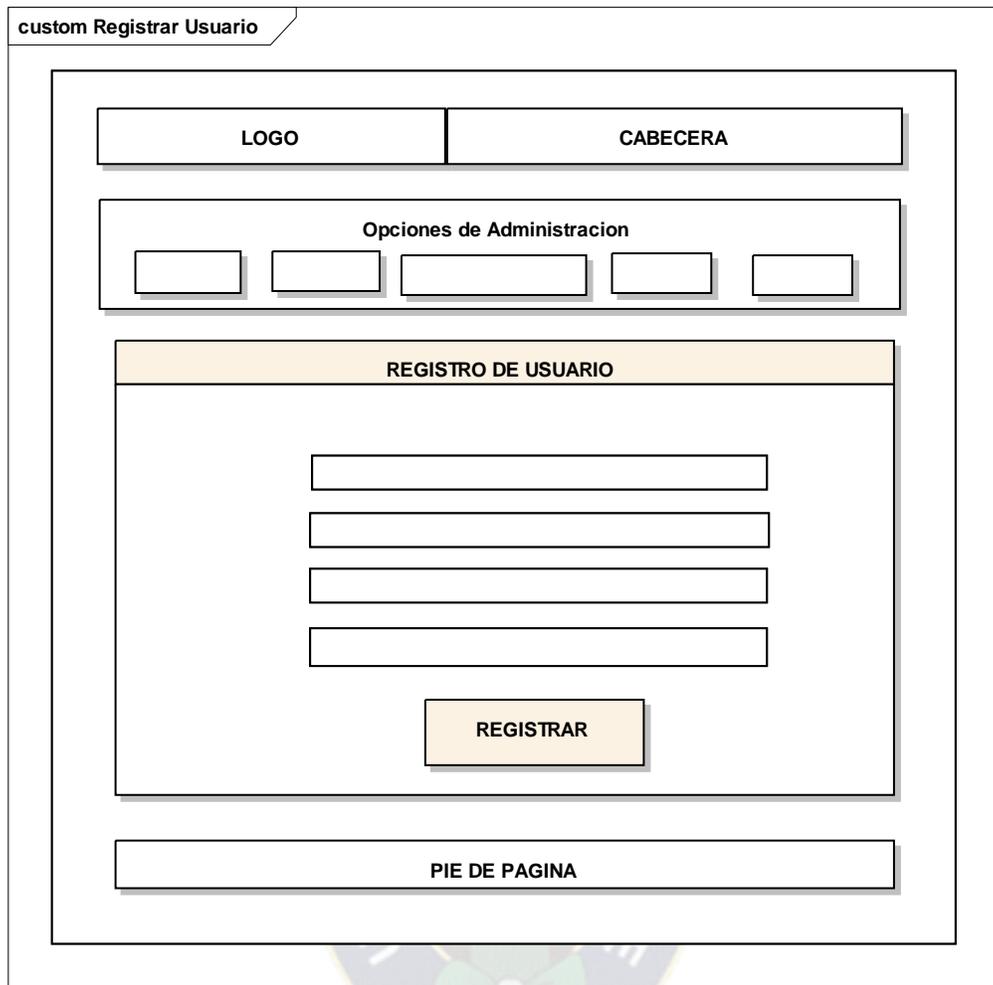


Figura 3.26. Pantalla Registro de usuario
Fuente: Elaboración Propia

Después de registrar los datos respectivos en el sistema, nos muestra en un recuadro amarillo que es el respectivo mensaje de que el usuario fue ingresado correctamente y no existe ningún error, esto muestra que el procedimiento se realizó satisfactoriamente y se llegó a registrar en la base de datos todo lo introducido.

E. MODIFICACIÓN DE USUARIO

En la modificación de usuario el administrador podrá realizar los cambios que son requeridos como equivocaciones para el respectivo usuario, como ser nombre, usuario, contraseña, o simplemente cambiar los datos, esto para una mayor seguridad en el sistema, la

administración y para el mismo usuario que utiliza el sistema de registro y administración de documentos. (Ver Figura 3.27)

custom Modificar Usua...

LOGO CABECERA

Opciones de Administracion

REGISTRO DE USUARIO

BUSCAR

Usuario: Text BUSCAR

MODIFICAR

PIE DE PAGINA

Figura 3.27. Modificación de Usuario
Fuente: Elaboración Propia

F. ELIMINACIÓN DE USUARIO

Para la eliminación de usuarios se debe buscar por nombre de usuario o por usuario, así se accede a la página de eliminación verificando datos se procede con lo decidido.

(Ver Figura 3.28)

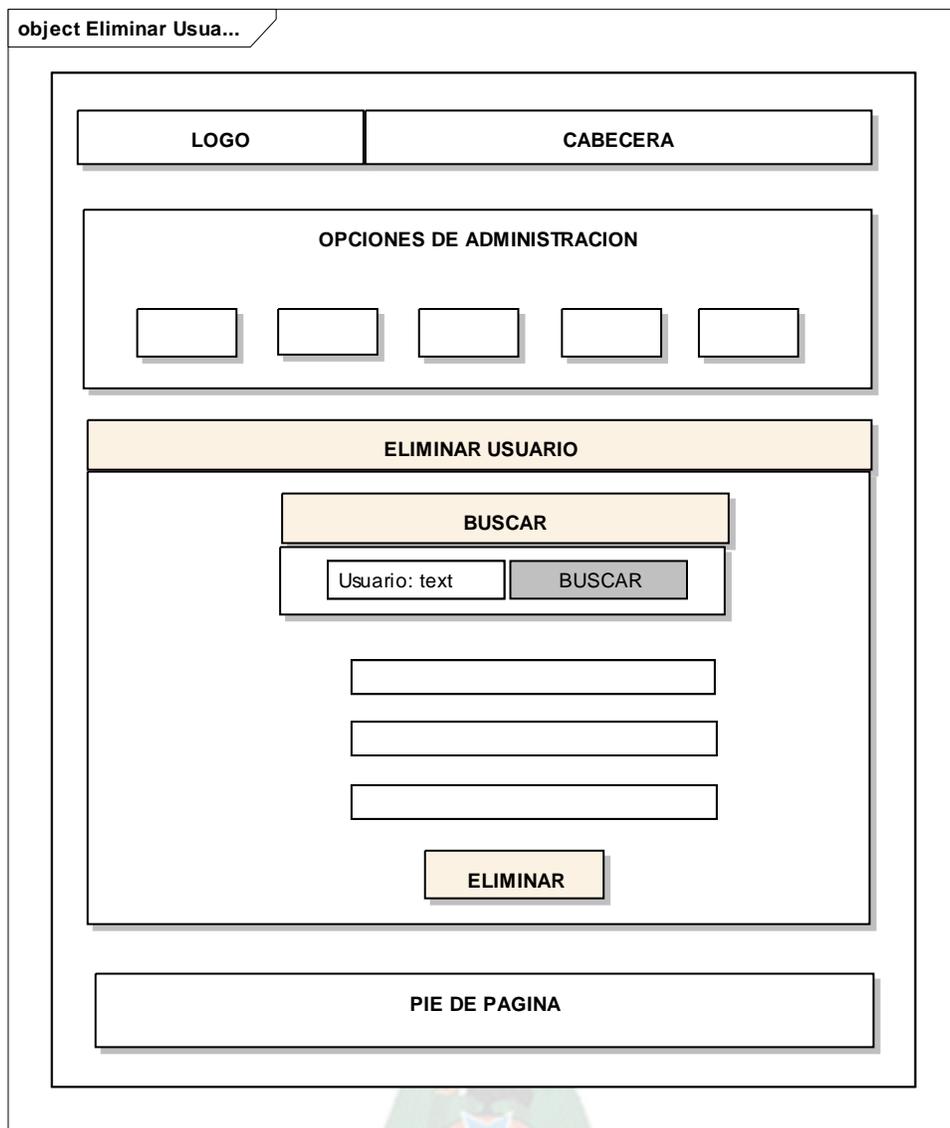


Figura 3.28. Eliminación de Usuario
Fuente: Elaboración Propia

G. BÚSQUEDA Y EDICIÓN DE HISTORIAL

En la opción buscar historial, se tiene que ingresar el código de usuario, para generar los datos correspondientes, una vez ingresado, podrá realizar los cambios adecuados de la información extraída del patrocinado, una vez realizado los cambios este mandara un mensaje que dirá que los datos fueron actualizados correctamente. En la figura 3.29 se representa el proceso anterior. La edición de historial se encarga de modificar y actualizar los datos

registrados en el sistema para una mayor administración además de notificar al usuario que es lo que le está faltando entregar para su respectivo registro, en la siguiente figura se muestra el proceso de edición de historial del sistema.

custom Edicion Historial

LOGO CABECERA

Opciones de Administracion

REGISTRO DE USUARIO

ADV BUSCAR

usuario: Text BUSCAR

MODIFICAR

PIE DE PAGINA

Detailed description: The diagram illustrates a web interface for editing system history. It is titled 'custom Edicion Historial'. The layout is organized into several sections. At the top, there are two boxes labeled 'LOGO' and 'CABECERA'. Below these is a section for 'Opciones de Administracion' containing five empty rectangular boxes. The main content area is titled 'REGISTRO DE USUARIO'. Inside this area, there is a search section with a header 'ADV BUSCAR', a text input field labeled 'usuario: Text', and a 'BUSCAR' button. Below the search section are four empty horizontal input fields. At the bottom of the main content area is a 'MODIFICAR' button. Finally, there is a 'PIE DE PAGINA' section at the very bottom.

Figura 3.29. Búsqueda y modificación de historial
Fuente: Elaboración Propia

H. REPORTE HISTORIAL

Es opción detallara el historial completo del usuario, como ser sus datos personales, además de la documentación entregada, que es una manera de ver si a este le falta entregar documentación que debe ser completada. El reporte, permite informar a la administración sobre los datos que se recabaron sobre el usuario, así mismo poder tener documentación legible a la vista... La figura 3.30 muestra este detalle

custom Reporte Historial del usuario

LOGO CABECERA

Opciones de Administracion

REGISTRO DE USUARIO

ADV BUSCAR

usuario: Text BUSCAR

IMPRIMIR

PIE DE PAGINA

Figura 3.30. Reporte Historial
Fuente: Elaboración Propia

3.5. FASE DE TRANSICIÓN

La fase de transición se dirige en liberar el sistema en producción para clausurar esta fase se cumple, con la aprobación y visto bueno del diseño e implementación del sistema que será por parte de los usuarios, los cuales harán las pruebas necesarias.

3.5.1. IMPLEMENTACIÓN

Para esto se aplica pruebas extensivas a lo largo de esta fase, una buena afinación del proyecto tiene lugar aquí, incluyendo modificaciones a los defectos significativos, se destaca las pruebas de caja blanca y caja negra, con el objeto de descubrir defectos que tiene el sistema.

”La prueba no puede asegurar la ausencia de defectos, solo puede demostrar que existe defectos en el software” [PRESSMAN, 2005].

3.5.1.1. PRUEBAS DEL SISTEMA

El motivo de la elección de este método, es la utilidad que tiene para probar software. La prueba de caja negra intenta descubrir errores en las siguientes categorías:

- 1) Funciones incorrectas o ausentes.
- 2) Errores de interfaz.
- 3) Errores en la estructura de datos o acceso a base de datos externas.
- 4) Errores de rendimiento.
- 5) Errores en la inicialización y terminación.

El mismo consiste en catalogar las entradas y salidas del sistema, primeramente en forma global y después en forma consecutiva en cada uno de sus componentes que se encuentran elaborados.

Esto se lo realiza en forma independiente y la manera en que estas son transformadas, para tener un mejor y buen entendimiento del objetivo de cada uno de los distintos componentes. (Ver Figura 3.31)



Figura 3.31. Prueba de caja negra
Fuente: Elaboración Propia

- ✓ Se realizó pruebas unitarias, es decir cada módulo en el momento de la construcción previamente puesta en producción.
- ✓ La prueba de integración se realiza cuando todos los módulos, están diseñados y desarrollados, para luego integrarlos y posteriormente realizar pruebas de todo el sistema.
- ✓ Las pruebas del software se realiza una vez integrado los módulos, se procede a las pruebas con datos reales.
- ✓ Las pruebas de implantación, se realiza en forma incremental, es decir que ya parte del sistema está en plena producción acorde a la metodología AUP.
- ✓ Las pruebas de aceptación, después del periodo de pruebas, se espera que el sistema sea aceptado por los usuarios finales.

3.5.1.2. PRUEBAS DE ESTRÉS

Conocidas como “stress testing”, muchas veces engloban de forma errónea al resto de pruebas mencionadas anteriormente. El objetivo de estas pruebas es obtener datos, sobre la carga del sistema, que ayuden a realizar el dimensionamiento del sistema o “capacity planning”. Esta prueba genera carga en el sistema hasta hacerlo inutilizable. Una vez que la aplicación ha dejado de funcionar, nuestros consultores se centran en distintos objetivos, como por ejemplo: verificar la calidad de los mensajes de error del sistema o establecer alertas para poder anticipar un fallo total del sistema. Las pruebas de estrés son uno de los últimos tipos de pruebas que se deben ejecutar, ya que, por su carácter poco realista, podría darse el caso de que la situación de carga simulada nunca se diera en la vida real.

Las pruebas de rendimiento son ejecutadas por medio de scripts automatizados, éstos se encargan de emular las acciones que realizaría un usuario final sobre la aplicación bajo pruebas. Los scripts se ejecutan en paralelo, cada uno de ellos emulando un “usuario virtual”, de esta forma, anticipando la carga esperada cuando el sistema pase a producción. Durante la ejecución de las pruebas, nuestros consultores se encargan de vigilar el sistema, que recibe la carga por medio de indicadores de rendimiento, esta acción es comúnmente llamada monitorización del sistema.

Para el caso se utiliza la herramienta de Apache-jmeter, con el cual se pudo sacar los siguientes resultados, como se observa en la siguiente figura. (Ver Figura 3.32)

Etiqueta	# Muestras	Media	Mediana	Línea de ...	Mín	Máx	% Error	Rendimi.	Kb/sec
204 /proyecto/theme/images/hbg.gif	30	4	3	8	2	10	0.00%	2,2/min	,0
201 /proyecto/theme/images/portada.jpg	30	9	5	15	4	81	0.00%	2,2/min	2,2
202 /proyecto/theme/images/cn-bg.gif	30	6	3	13	3	26	0.00%	2,2/min	,2
205 /proyecto/theme/images/bot-bg.gif	30	5	4	10	2	14	0.00%	2,2/min	,0
200 /proyecto/theme/images/logoss.png	30	8	3	14	2	77	0.00%	2,2/min	,1
203 /chart	20	7618	8032	11849	2195	13262	0.00%	6,1/min	1,3
199 /proyecto/theme/slide/slide/bannern01.jpg	30	10	7	12	5	78	0.00%	2,2/min	3,5
198 /chart	20	8894	9024	14006	1934	25009	0.00%	6,0/min	1,3
197 /proyecto/theme/js/superfish.js	30	9	5	16	3	76	0.00%	2,2/min	,1
207 /proyecto/theme/images/shadow.png	30	6	3	9	2	49	0.00%	2,2/min	,1
206 /proyecto/theme/images/arrows-ffffff.png	30	3	3	6	2	10	0.00%	2,2/min	,0
208 /proyecto/includes/ejemplo.php	30	22	15	37	14	82	0.00%	2,2/min	,2
209 /chart	20	4695	3884	7867	430	15874	0.00%	6,0/min	1,3
210 /proyecto/includes/ejemplo.php	30	20	16	29	13	92	0.00%	2,2/min	,2
211 /chart	20	5032	2794	6240	873	33786	0.00%	5,2/min	1,1
212 /proyecto/mod_reportes/rep_his.php	30	17	12	34	10	80	0.00%	2,2/min	,2
213 /proyecto/mod_reportes/rep_his.php	30	35	24	50	23	99	0.00%	2,2/min	,5
214 /proyecto/mod_registros/	30	23	20	31	16	77	0.00%	2,2/min	,1
215 /proyecto/mod_impresion/imp_hist.php	30	120	117	148	85	187	0.00%	2,2/min	22,6
216 /proyecto/mod_reportes/rep_his.php	30	29	27	40	23	46	0.00%	2,2/min	,5
217 /proyecto/theme/css/superfish.css	30	6	3	7	2	54	0.00%	2,2/min	,1
221 /proyecto/theme/js/funciones.js	30	9	3	30	3	69	0.00%	2,2/min	,1
219 /proyecto/theme/slide/slide.js	30	10	5	35	4	49	0.00%	2,2/min	,1
218 /proyecto/theme/css/style.css	30	6	4	11	2	28	0.00%	2,2/min	,2
220 /proyecto/theme/js/Query.js	30	11	6	16	4	67	0.00%	2,2/min	2,8
223 /proyecto/theme/js/superfish.js	30	9	3	20	2	50	0.00%	2,2/min	,1
222 /proyecto/theme/js/hoverIntent.js	30	5	3	6	2	36	0.00%	2,2/min	,1

Figura 3.32. Apache JMeter: Informe Agregado
Fuente: Elaboración Propia

Se tiene una tabla con **una fila para cada petición realizada**. Se observa que las peticiones todas los campos se han repetido 30 veces (columna Muestras): esto es porque existían 3 muestras de peticiones a esas URL's en diferentes momentos de la Prueba. Aquí salen agrupadas.

Los valores están calculados tomando el tiempo total que ha tardado la prueba (la ejecución de mis 10 usuarios), por lo tanto, cuantas más muestras haya más tiempo total de ejecución; influye en cada línea de la tabla. Para cada línea (petición) tenemos:

- el máximo de tiempo invertido por una petición (columna Max).
- el mínimo de tiempo invertido por una petición (columna Min).
- la media de tiempo invertido por una petición (columna Media).
- la mediana de tiempo invertido por una petición: significa que el 50% de las muestras tardaron menos del valor reflejado.
- El tanto por ciento de respuestas con error.
- El rendimiento (*throughput*): número de peticiones procesadas en una unidad de tiempo, que puede ser segundos, minutos y horas.
- El rendimiento en Kb/segundo: igual que la anterior pero con cantidad de datos en lugar de peticiones

Por lo que se puede concluir, que el servidor es capaz de responder a 2.2 peticiones por minuto a la página de login (primera muestra), o que la petición a “imprimir reportes de estadísticas”, es la que más tarda en procesarse con respecto a la media. En la siguiente figura se muestra el gráfico en barras (Ver Figura 3.33)

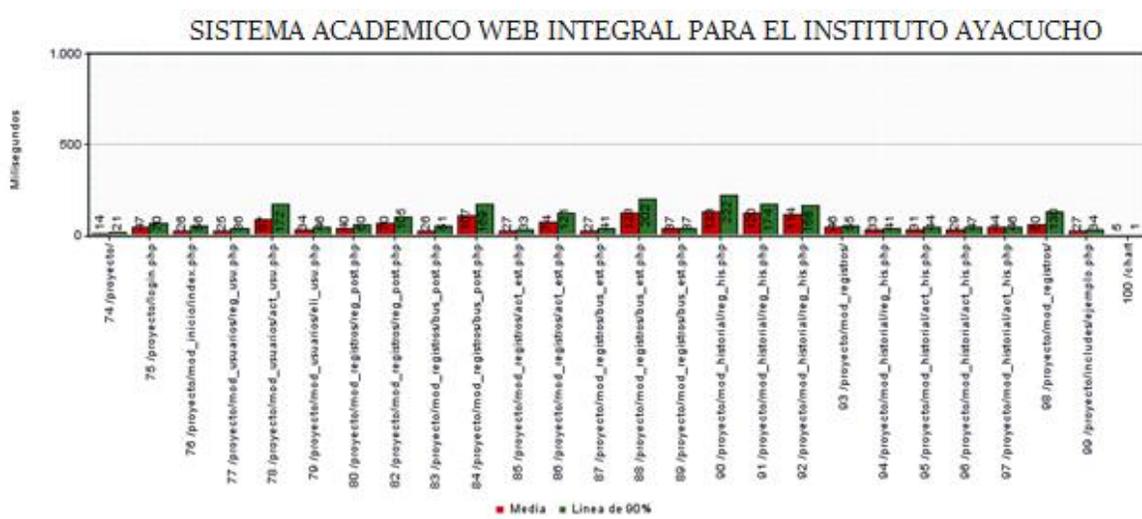


Figura 3.33. Apache JMeter: Grafico de barras
Fuente: Elaboración Propia

3.5.1.3. PRUEBAS DE SEGURIDAD

La seguridad es responsabilidad de todos aquellos que están en contacto con el sistema. La cual lo enfocamos desde tres aspectos interrelacionados entre sí: físico, lógico y conductual

A. SEGURIDAD FÍSICA

- El servidor debe ser manejado por el personal autorizado, en este caso sería la responsable de patrocinio, para una buena administración de base de datos. En caso de mantenimiento técnico, este deberá consultar al administrador del sistema.
- A la oficina del administrador, se debe restringir, el acceso físico de otras personas, que nos son propios de esta oficina, para una mayor seguridad.
- El personal de limpieza debe estar consciente de las restricciones de seguridad.

B. SEGURIDAD LÓGICA

Se especifica, al acceso de los usuarios, a los recursos y a la información que dispone en un ordenador, como también al uso correcto de los mismos. Cuando se utilizan, permiten al usuario entrar al sistema o a una parte particular de una base de datos con una contraseña correcta, entre las cuales se destacan:

- ✓ Seguridad del motor Sql Manger of MySQL, contraseña de acceso a base de datos.
- ✓ Seguridad del sistema, mediante el manejo de usuarios del sistema y contraseñas.
- ✓ Se puede utilizar Back Up de la base de datos, solo por usuarios con este rol de manejo de sistemas.
- ✓ Sistema de encriptación de contraseñas, para que el usuario confíe en el sistema.
- ✓ Se puede realizar restauración del sistema en caso de eventualidades, es decir recuperar una copia anterior.

C. SEGURIDAD CONDUCTUAL

- Crear conciencia en el manejo de información de los usuarios a través de capacitaciones planificaciones, para que los mismos reconozcan el valor que tiene la información.

- Identificación de personal que eventualmente tendrán acceso a la computadora, datos e información para asegurar que sus intereses son consistentes con los intereses del Instituto y que entienden por completo la importancia de llevar a cabo los procedimientos de seguridad.
- Se crea un manual de funciones, que describan las políticas con respecto a la seguridad, para que los usuarios estén totalmente conscientes de las expectativas y responsabilidades que se deben asumir.

D. RESTRICCIONES DEL SISTEMA

- ✓ El sistema funciona dentro del entorno de Windows XP adaptable a sus siguientes versiones.
- ✓ El sistema funciona en un entorno de red intranet, pero se tiene previsto, para que pueda ser utilizado desde la web, una vez se adquiera su respectivo dominio, de acuerdo a las políticas que el centro adopte.

E. CAPACITACIÓN

La capacitación del personal del Instituto Tecnológico Ayacucho, que maneja el sistema para el registro y administración de documentos, se realiza de la siguiente manera:

- Instalación del sistema.
- Back Up y restauración de la base de datos del sistema.
- Altas de usuarios, estudiantes e historiales.
- Ubicación física de equipos.
- Políticas de manejo de usuarios.
- Ingreso al sistema.
- Registro, modificación y eliminación de usuarios.
- Registro, actualización y búsqueda de Usuarios.
- Registro, edición y búsqueda de historial.
- Elaboración de reportes estadísticos de Usuarios.

4.1. INTRODUCCIÓN

La calidad del software es la concordancia con los requerimientos funcionales y de rendimiento directamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

Un producto de alta calidad requiere menos mantenimiento y facilita tanto el desarrollo como el mantenimiento de la productividad. Con la medición de la calidad se pueden lograr estos objetivos. En lo que se refiere al mantenimiento, la medición de la calidad del software ayuda a identificar problemas de confiabilidad y a mejorar las técnicas para identificar las necesidades de mantenimiento. A continuación se detallan los factores de calidad con el objeto de evaluar la calidad del software.

Al término del Sistema Web Académico Integral, se debe medir la calidad del producto, existe dos tipos de medición directa e indirecta. En el presente proyecto se aplicara las medidas indirectas planteada por la norma ISO 9126

4.2. CARACTERÍSTICAS PROPUESTAS POR ISO-9126

La necesidad de comparar productos motiva el trabajo para la definición de un modelo estándar.

ISO 9126 entrega la definición de las características y los procesos de evaluación de calidad asociados para usar cuando se especifican los requisitos y la evaluación de los productos de software a lo largo de su vida útil.

Los factores relacionados con la norma ISO 9126, están representados por seis los cuales son: funcionalidad, confiabilidad, eficiencia, usabilidad, mantenibilidad y portabilidad.

A continuación se muestra en la siguiente figura 4.1 las preguntas usuales que se hacen en cada factor.



Figura 4.1: Norma ISO-9126
Fuente: Elaboración Propia

Funcionalidad: conjunto de atributos que soporta la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son tales que satisfacen las necesidades implícitas o establecidas.

Confiabilidad: El conjunto de atributos que soporta la capacidad del software para mantener su nivel de rendimiento bajo condiciones establecidas por un periodo de tiempo establecido.

Usabilidad: El conjunto de atributos que soporta el esfuerzo necesario para el uso y la evaluación individual de tal uso mediante un conjunto de usuarios establecidos e implícitos.

Eficiencia: el conjunto de atributos que soporta las relaciones entre el nivel de rendimiento del software y el monto de recursos empleados, bajo condiciones establecidas.

Mantenibilidad: El conjunto de atributos que soporta el esfuerzo necesario para realizar modificaciones especificadas

Portabilidad: El conjunto de atributos que soporta la habilidad del software para transferirlo de un entorno a otro.

4.2.1. CONFIABILIDAD

La confiabilidad de un sistema es una unidad muy importante en su calidad general. Para comprobar la confiabilidad se toma en cuenta las fallas que se producen en el sistema en un tiempo determinado, también es el grado en que el sistema responde bajo las condiciones definidas durante un intervalo de tiempo dado. En primer lugar se considera la confiabilidad de cada módulo independientemente. Para esto se requiere el modelo del sistema, que se observa en la siguiente figura 4.2.

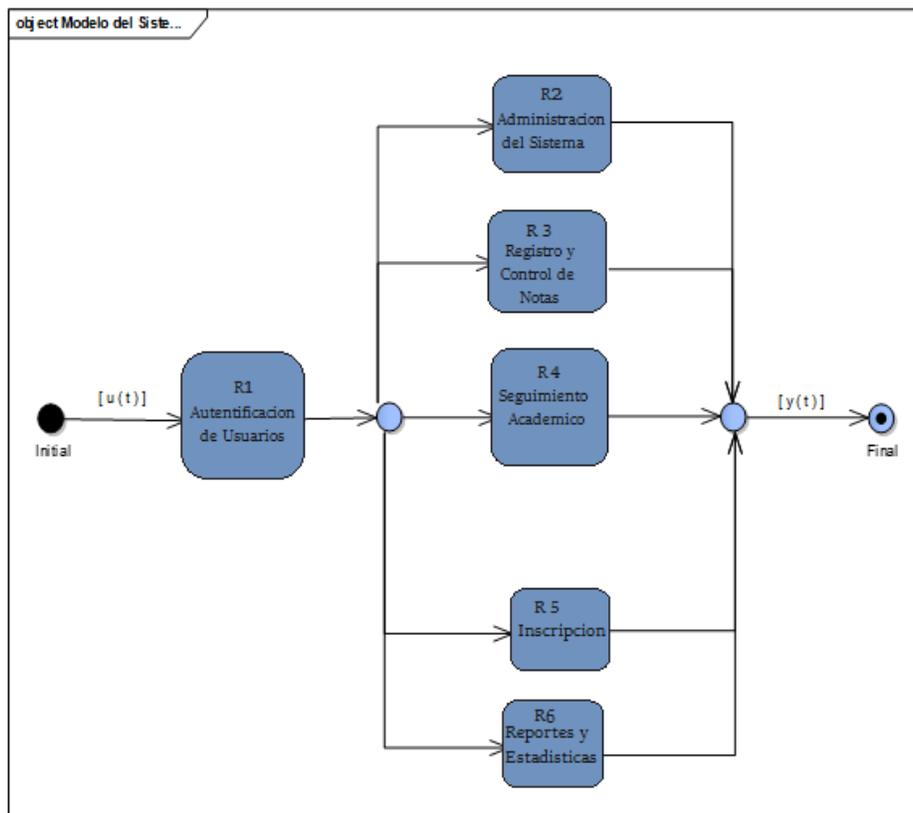


Figura 4.2: Modelo del Sistema Web Académico Integral
Fuente: Elaboración Propia

Se toma en cuenta la siguiente relación:

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

Donde:

$R(t)$ = Confiabilidad de un componente o subsistema t.

$e^{-\lambda t}$ = Probabilidad de falla de un componente o subsistema en tiempo t.

T = Tiempo de trabajo sin falla.

λ = Tasa constante de fallos.

t = Periodo de operacion de tiempo.

Posteriormente se realiza los cálculos para cada módulo utilizando la ecuación (1).

	MÓDULOS	λ	T	R(t)
1	Autenticación de Usuario	0.02	2 Hrs.	0.96
2	Administración del Sistema	0.03	3 Hrs.	0.91
3	Registro y control de Notas	0.04	3 Hrs.	0.89
4	Seguimiento Académico	0.03	3 Hrs.	0.91
5	Inscripción	0.04	3 Hrs.	0.89
6	Reportes y Estadísticas	0.03	3 Hrs.	0.92

Tabla 4.1: Calculo de Confiabilidad
Fuente: Elaboración Propia

En la figura 4.1 se muestra claramente que el modelo del sistema tiene una conexión inicial en serie y posteriormente una conexión en paralelo, por lo que realizando los cálculos se obtiene:

$$R = R_i * R_s \dots\dots\dots(2)$$

Donde $R_i = R_1 = 0.96$

$$R_s(t) = 1 - \{(1 - R_2(t)) * (1 - R_3(t)) * \dots * (1 - R_{n-1}(t)) * (1 - R_n(t))\} \dots\dots (3)$$

Remplazando los datos en la ecuación 3 se obtiene:

$$R_s(t) = 1 - \{(0.09) * (0.11) * (0.09) * (0.11) * (0.08)\}$$

$$R_s(t) = 0.9999922$$

Reemplazando en la ecuación 2 se tiene:

$$R = 0.96 * 0.9999922$$

$$R = 96\%$$

Interpretación

“Con el resultado obtenido anteriormente, se puede decir que el “Sistema Web Académico Integral”, demuestra que del 100% un 96% es confiable y un 4% de que cualquiera de los componentes falle.”

4.2.2. FUNCIONALIDAD

Es preciso evaluar un conjunto de características y capacidades del sistema, ya que la funcionalidad no se mide directamente. El sistema debe ser apto, para proveer las funciones que cumplan con las necesidades explícitas e implícitas, cuando es utilizado en la condiciones especificadas por el cliente.

Para el procesamiento de datos de la funcionalidad utilizaremos la métrica de punto función, para esto se debe determinar cinco características de dominios de información, donde se proporciona las cuentas en la posición apropiada a la tabla. Los valores de dominio de información se definen de la siguiente forma:

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

Para el procesamiento de datos de punto función se utiliza la siguiente ecuación:

$$PF = \text{cuenta total} * (0.65 + 0.01 * \sum F_i) \dots \dots \dots (1)$$

Dónde:

PF = Medida de Funcionalidad.

Cuenta Total = Es la suma de todas las entradas obtenidas en Nro de entradas, Nro de salidas, Nro de peticiones, Nro de archivos, Nro de interfaces externas.

$\sum F_i$ = Son los valores de ajuste de complejidad según las respuestas a preguntas destacadas en la siguiente tabla.

Con todos estos pasos tomados en cuenta se prosigue con el cálculo del punto función.

4.2.2.1. NÚMERO DE ENTRADAS DE USUARIO

Es todos los datos que vienen desde el exterior, que tiene una sola dirección, que es del exterior al interior. En la siguiente tabla 4.2 se observa las entradas de usuario que tiene el sistema.

Nº	ENTRADAS DE USUARIO
1	Pantalla de Ingreso al Sistema.
2	Registro de Usuario.
3	Registro de Historial.

Tabla 4.2. Entrada de Usuarios
Fuente: Elaboración Propia

4.2.2.2. NÚMERO DE SALIDAS DE USUARIO

Es la información elaborada por el sistema que son transmitidas al usuario, también actualizan algunos archivos, tiene una sola dirección del interior al exterior. En la siguiente tabla 4.3 se puede observar las salidas del usuario.

Nº	SALIDA DE USUARIO
1	Detalle del Usuario
2	Reporte del Historial del Usuario
3	Reporte de estadísticas de registros de Usuarios

Tabla 4.3. Salidas de usuario
Fuente: Elaboración Propia

4.2.2.3. NÚMERO DE PETICIONES DE USUARIO

Es una entrada interactiva que produce la generación de algunas respuestas del sistema inmediata en forma interactiva. Se puede observar las peticiones de usuario en la siguiente tabla 4.4.

Nº	PETICIONES DE USUARIO
1	Autenticación de Usuario
2	Modificación de Usuario
5	Modificación de Historial
6	Listado de Estudiantes
7	Listado de Docentes
8	Listado de Administrativos

Tabla 4.4. Petición de Usuario
Fuente: Elaboración Propia

4.2.2.4. NUMERO DE ARCHIVOS

Es un grupo lógico de datos, que puede llegar a ser una parte de una gran base de datos o un archivo independiente. En la siguiente tabla 4.5 se puede observar el número de archivos.

Nº	ARCHIVOS
1	Usuario
2	Estudiantes
3	Docentes
4	Administrativos
5	Reporte
6	Historial

Tabla 4.5. Archivos
Fuente: Elaboración Propia

4.2.2.5. NUMERO DE INTERFACES EXTERNAS

Son todas las interfaces legibles por la computadora que se utilizan para transmitir información a otro sistema. Una vez que se ha seleccionado los datos anteriores, a la cuenta se asocia un valor de complejidad. En la siguiente tabla 4.6 se puede observar las interfaces externas.

Nº INTERFACES EXTERNAS	
1	Internet
2	Intranet

Tabla 4.6. Interfaces Externas
Fuente: Elaboración Propia

4.2.2.6. PONDERACIÓN

En seguida se reúne todos los datos encontrados, para su respectiva ponderación. Para realizar puntos de función es necesario elegir un criterio de ponderación, que en el caso se utiliza el factor medio. En la siguiente tabla 4.7 detallamos lo explicado.

Parámetro de Medición	Cuenta	*	FACTORES DE PONDERACIÓN			=	Total
			Simple	Medio	Complejo		
Nº de entradas de Usuario	5	*	3	4	6	=	20
Nº de salidas de Usuario	3	*	4	5	7	=	15
Nº de peticiones de Usuario	7	*	3	4	6	=	28
Nº de archivos	6	*	7	10	15	=	60
Nº de interfaces externas	2	*	5	7	10	=	14
Cuenta total							137

Tabla 4.7. Cálculos de puntos de Función no ajustados
Fuente: Elaboración Propia

Para el ajuste del modelo en función de la complejidad del proceso es necesario analizar las características propias del proceso, respondiendo 14 preguntas utilizando la siguiente tabla 4.8

GRADO	DESCRIPCIÓN
0	No está presente o no influye
1	Influencia Mínima
2	Influencia Moderada
3	Influencia Promedio
4	Influencia Significativa
5	Influencia Fuerte

Tabla 4.8. Ponderaciones
Fuente: Elaboración Propia

Las 14 preguntas a responder se muestran en la tabla 4.9.

Nº	CARACTERÍSTICAS	FACTOR F _i
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?	5
2	¿Se requiere comunicación de datos?	5
3	¿Existen funciones de procesamiento distribuido?	2
4	¿Es crítico el rendimiento?	2
5	¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	4
6	¿Requiere el sistema entrada de datos interactiva?	3
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	4
8	¿Se actualizan los archivos maestros de forma interactiva?	2
9	Son complejas las entradas, las salidas, los archivos o las peticiones?	3
10	¿Es complejo el procesamiento interno?	4
11	¿Se ha diseñado el código para ser reutilizable?	3
12	¿Están incluidas en el diseño la conversión y la instalación?	5

13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
14	14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	4
Factor de Ajuste de Complejidad		51

Tabla 4.9. Valores de Ajuste de Complejidad
Fuente: Elaboración Propia

Remplazando los datos anteriores en la fórmula de punto función de la ecuación 1, se tiene:

$$PF = 137 * (0.65 + 0.01 * 51)$$

El punto función del Sistema Web Académico Integral, es:

$$PF = 180$$

Luego comparando los valores de funcionalidad del sistema con el punto función, máximo que se puede alcanzar es:

$$PF = 137 * (0.65 + 0.01 * 70)$$

$$PF = 190$$

Por lo que la funcionalidad del Sistema Web Académico, será:

$$\text{Funcionalidad} = \left(\frac{180}{190}\right) * 100$$

$$\text{Funcionalidad} = 94.73\%$$

Interpretación:

“De un 100% un 94.73% del sistema se concluye que es funcional, pero un 6% puede llegar a no ser funcional.”

4.2.3. MANTENIBILIDAD

Para el procesamiento de datos de la mantenibilidad del sistema, utilizaremos las medidas directas proporcionadas por la IEEE 982.1 – 1998, el cual propone un índice de madurez del sistema, que consisten en los cambios que producen, en cada versión del producto, para lo dicho se tiene la siguiente ecuación.

$$IMS = \frac{[M_T - (F_a + F_b + F_c)]}{M_T} \dots\dots\dots (1)$$

Donde:

IMS = Índice de madurez del sistema

M_T = Número de módulos en la versión actual.

F_a = Número de módulos en la versión actual que se han cambiado.

F_b = Número de módulos en la versión actual que se han añadido.

F_c = Número de módulos en la versión anterior que se han borrado en la versión actual.

Si el valor del IMS se aproxima a 1, el sistema empieza a estabilizarse.

Por lo tanto los valores encontrados son:

$$M_T = 6$$

$$F_a = 0$$

$$F_b = 0$$

$$F_c = 0$$

Reemplazando estos valores en la ecuación 1 se obtiene:

$$IMS = \frac{[6 - (0 + 0 + 0)]}{6}$$

$$IMS = 1 * 100$$

$$IMS = 100\%$$

Interpretación:

“Por lo tanto se deduce que el índice de madurez del sistema en un 100% tiende a estabilizarse, lo que indica es el sistema es estable.”

4.2.4. PORTABILIDAD

Es la capacidad del software para ser transferido de un ambiente de operaciones a otro. La portabilidad del software se enfoca en tres aspectos: a nivel de aplicaciones, a nivel de sistema operativo y adaptación al cambio. Dado por la siguiente fórmula:

$$GP = 1 - \left(\frac{CT}{CRD} \right) \dots\dots\dots (1)$$

Dónde:

GP = Grado de portabilidad.

CT = Costo de Transportar.

CRD = Costo de Re-Desarrollo.

Si GP

- **Es > 0, la portabilidad es mas rentable que el re-desarrollo.**
- **Es = 1, la portabilidad es perfecta.**
- **Es < 0, el re-desarrollo es mas rentable que la portabilidad.**

Reemplazando en la ecuación 1, tenemos:

$$GP = 1 - \left(\frac{75}{2000} \right)$$

GP = 0.96258

Interpretación:

“Por lo tanto del 100% de grado de portabilidad del sistema puede movilizarse de un entorno a otro en un 96%, que es más rentable que el re-desarrollo y un 4% puede llegar a no tener portabilidad.”

4.2.4.1. NIVEL DE APLICACIONES

El sistema es desarrollado en el lenguaje de PHP, utilizando Aptana Estudio como su editor de código. Con su gestor de base de datos Mysql, utilizan el servidor Apache Server, en cuanto al sistema desarrollado es portable, ya que el sistema puede ser distribuido en CD's, memory flash.

4.2.4.2. NIVEL DEL SISTEMA OPERATIVO

El sistema llega a ser portable para los sistemas operativos de Windows como ser: Windows XP, Windows Vista, Windows Seven y Windows 8.

4.2.4.3. NIVEL DE HARDWARE

En el nivel de hardware, se puede ver que el sistema es portable, es decir que se lo puede implantar fácilmente en cualquier momento y aptas para todas las computadoras mayores o iguales a la tecnología Pentium IV.

4.2.5. USABILIDAD

La usabilidad, es el grado en el que el software es fácil de usar, y viene manifestado por la: facilidad de comprensión, facilidad de aprendizaje y operatividad. Para calcular la usabilidad del sistema se utiliza las siguientes tres métricas:

4.2.5.1. LA COMPLEJIDAD DE LA DESCRIPCIÓN

Que se encuentra dada por la fórmula:

$$X = \frac{A}{B} \dots \dots \dots (1)$$

Dónde:

A= Número de funciones (casos de uso) o tipos de funciones explicadas en la descripción del producto.

B= Número total de funciones (casos de uso).

$$X = \frac{11}{12}$$

$$X = 0.92$$

Interpretación:

“Por lo tanto el 100% del sistema presenta un 92% de entendimiento por parte de los usuarios finales respecto a la capacidad del producto y un 8% les falta un poco más de entendimiento acerca del sistema.”

4.2.5.2. CONSISTENCIA OPERACIONAL

Dada por la fórmula:

$$X = 1 - \frac{A}{B} \dots \dots \dots (2)$$

Dónde:

A= Número de instancias con operaciones con comportamiento inconsistente.

B= Número total de operaciones.

Reemplazando en la ecuación 2 tenemos:

$$X = 1 - \frac{1}{12}$$

$$X = 0.92$$

Interpretación:

“Por lo que el sistema muestra de un 100% un 92% de no instancias de operaciones con comportamiento inconsistente y en un 8% presentan instancias inconsistentes.”

4.2.5.3. CONSISTENCIA OPERACIONAL EN EL USO

El cual está dado por la fórmula:

$$X = 1 - \frac{A}{B} \dots \dots \dots (1)$$

Dónde:

A= Número de funciones que el usuario encontró inaceptables, inconsistentes según sus expectativas.

B= Número de funciones usadas por el usuario durante el periodo de prueba.

Reemplazando en la ecuación 2 tenemos:

$$X = 1 - \frac{1}{18}$$

$$X = 0.96$$

Interpretación:

“Por lo tanto de un 100% del sistema, el usuario encuentra un 4% del sistema inaceptable, en el periodo de prueba, de acuerdo a los resultados obtenidos anteriormente, el usuario se encuentra satisfecho con la consistencia operacional del uso del sistema.”

4.2.5.4. RESULTADO FINAL

En la siguiente tabla 4.10 se despliega el resultado de los factores de la norma ISO 2196.

FACTORES	RESULTADO
Confiabilidad	96
Funcionalidad	94.73
Mantenibilidad	100
Portabilidad	96
Usabilidad	93
EVALUACIÓN DE CALIDAD TOTAL	95.95

Tabla 4.10. Resultado total
Fuente: Elaboración Propia

De un usuario que acceda al sistema en su totalidad tendrá un grado de satisfacción del 96% al utilizarla.

5.1. INTRODUCCIÓN

Las pruebas en el desarrollo del software son realizados con la intención de descubrir errores, y se deben realizar para todos los artefactos generados durante la construcción de un producto, se aplican diferentes tipos de técnicas de prueba a cada tipo de software. Para disponer de información continua, confiable y en un tiempo oportuno, se deben de considerar varios factores de seguridad. La seguridad informática debe garantizar: la disponibilidad de los sistemas informáticos y la confidencialidad.

5.2. PRUEBAS DEL DISEÑO ARQUITECTÓNICO

Se concentran en las características de la arquitectura del programa, con especial énfasis en la estructura arquitectónica y en la eficacia de los módulos y es una métrica de caja negra por que no requiere de ningún conocimiento de trabajo interno del módulo.

a) **Complejidad estructural**, se define de la siguiente manera:

$$S(i) = f^2_{out(i)}$$

Donde $f^2_{out(i)}$ es la expansión del módulo i .

Para la aplicación se tomó en cuenta:

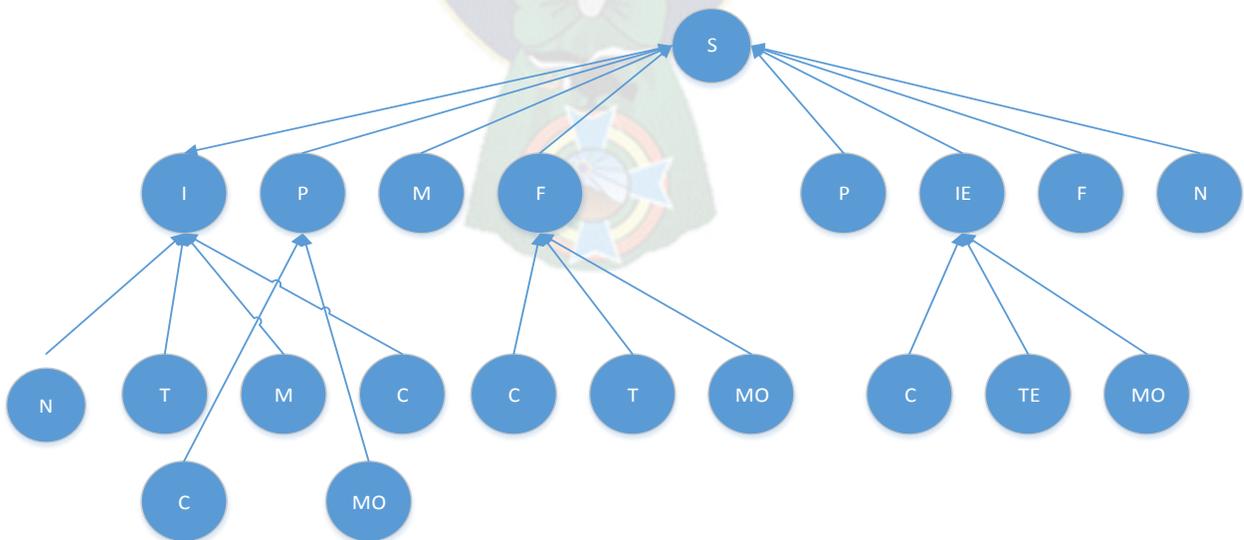


Figura 5.1 Grafo del sistema
Fuente: Elaboración propia

Donde $S(i) = 20$ por ser la cantidad de módulos, si el análisis comienza desde el modulo i que es el mismo sistema.

b) **Complejidad de datos**, que proporciona una indicación de complejidad interna de un módulo i y se define:

$$D(i) = v(i)/[f_{out(i)} + 1]$$

Considerando a las variables de entrada y salida como una sola por modulo se tiene:

$$D(i) = \frac{29}{20 + 1} = 1.38$$

El resultado nos dice que tiende a complicarse.

c) **Complejidad del sistema**, se define como la suma de complejidades estructurales y de datos y se define como:

$$C(i) = S(i) + D(i)$$

Remplazando tenemos:

$$C(i) = 20 + 1.38 = 21.38$$

A medida que crecen los valores de complejidad, la complejidad arquitectónica o global del sistema también aumenta. Esto lleva a una mayor probabilidad de que aumente el esfuerzo necesario para la integración y las pruebas.

5.3 SEGURIDAD

5.3.1 SEGURIDAD FÍSICA

La seguridad física consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial. Se refiere así como los controles y mecanismos de seguridad dentro y alrededor del centro de cómputo así como los medios de acceso remoto y desde el mismo, implementados para proteger el hardware y medios de almacenamiento de datos.

Las principales amenazas que se prevén en la seguridad física son:

- Desastres naturales, incendios, accidentales, tormentas e inundaciones.
- Amenazas ocasionales por el hombre.
- Disturbios, sabotajes internos y externos deliberados.

Se consideraron los siguientes mecanismos de seguridad física.

Protección de hardware, el hardware es frecuentemente el elemento más caro de todo sistema informático. Por tanto, las medidas encaminadas a asegurar su integridad son una parte importante de la seguridad física de la institución, son muchas las amenazas al hardware de una instalación informática, se presenta algunas de las posibilidades soluciones, si no para evitar los problemas si al menos minimizar sus efectos.

- a) Acceso físico, el nivel de seguridad física depende completamente del entorno donde se ubiquen los puntos a proteger, se recomienda a los funcionarios de la institución alojar el servidor donde será instalado la aplicación en un ambiente con acceso restringido a los usuarios o personas particulares.
- b) Desastres naturales, para los casos de terremotos o sismos se recomienda ubicar los servidores alejados de ventanas , ubicados en superficie muy elevadas, utilizar instrumentos que aseguren su estabilidad en cuanto al posicionamiento del servidor, no situar objetos pesados encima del servidor para prevenir posibles caídas al mismo, en cuanto a las tormentas eléctricas se recomienda apagar los servidores y desconectarlas ante una tormenta, también que los medios magnetos como los dispositivos de almacenamientos sean alejados lo más alejado posible de la estructura metálica de los edificios. Para prevenir desastres causadas por inundaciones o humedad se debe considerar situar a los equipos a una altura no muy elevada del suelo
- c) Desastres del entorno, para el caso de la electricidad se tiene una solución es utilizar tomas de corriente a tierra, lo cual desvía el exceso de corriente al suelo, para la corriente estática se propone un spray antiestático. Para los casos de incendios y humos se tiene la compra de extintores de dióxido de carbono, para el humo, un

potente abrasivo que ataca especialmente los discos magnéticos y ópticos, se sugiere prohibir dentro la sala donde se ubican las computadoras.

5.3.2 SEGURIDAD DE BASE DE DATOS

Para recuperar o almacenar información se necesita conectarse a la base de datos, enviar una consulta valida, recoger el resultado y cerrar la conexión, para realizar la consulta se utiliza el lenguaje de consultas usando comúnmente, Lenguaje de Consultas Estructurado – SQL.

- a) Asignar un súper usuario quien pueda realizar altas, bajas y modificaciones a la base de datos, y para que otros usuarios puedan usarla, deben otorgarse privilegios que restringen el uso de la misma, para ello se crea diferentes usuarios para cada aspecto de la aplicación con derechos muy limitados sobre las tablas de la base de datos.
- b) Inyecciones SQL, las consultas SQL pueden burlar los controles de acceso, y de este modo evitar los chequeos estándares de autenticación y autorización, la inyección directa de comandos SQL es una técnica en la cual un atacante crea o altera comandos SQL existentes para exponer datos escondidos, o sobrescribir datos críticos, o incluso ejecutar comandos del sistema peligrosos en el servidor donde se encuentra la base de datos. Esto se consigue cuando la aplicación toma información de entrada del usuario y la combina con parámetros estáticos para construir una consulta SQL. Una solución a este tipo de problemas es revisar las entradas, PHP posee un amplio rango de funciones de validación de datos desde los más simples encontrados en funciones sobre variables y en funciones de tipo de carácter.
- c) Realiza copias de seguridad, el gestor de Base de datos MySQL incluye varias herramientas para la realización de copias de seguridad de la base de datos. Mediante ellas podrá poner a salvo los datos, para que en el eventual caso de que se pierdan, se pueda recuperar.

Existe varias opciones para realizar la copia de seguridad de una base de datos de MySQL, a partir de la versión 3.23.25 y posteriores, existe la posibilidad de realizar una copia de seguridad a través de la sentencia SQL “backup table”. Otra opción, es

posible realizar copias de seguridad a través de las herramientas que nos proporciona el propio gestor de base de datos, como pueden ser **mysqldump** ó **mysqldhotcopy**.

5.3.3 SEGURIDAD EN LA WEB

- a) Otorga acceso a las aplicaciones Web solo a personal autorizado, con privilegios mínimos que le permiten cumplir con sus funciones.
- b) Implementar políticas de contraseñas fuertes y de control de acceso a las aplicaciones Web.
- c) Se debe validar todo parámetro de entrada, incluyendo los campos de formularios teniendo en cuenta:
 - a. Tipo de datos (cadena, entero, real, etc.).
 - b. Conjunto de caracteres permitidos.
 - c. Longitud mínima y máxima.
 - d. Si el valor nulo es permitido.
 - e. Si el parámetro es requerido o no.
 - f. Si los duplicados son permitidos.
- d) El manejo inadecuado de errores puede introducir diversos problemas de seguridad en un sitio Web. Por ejemplo, la exhibición de información detallada de mensajes de error, como el contenido de variables, nombres de directorios e información sobre la base de datos, puede revelar detalles de la implementación que no deben ser expuestos bajo ninguna circunstancia a usuarios no autorizados.
- e) Almacenamiento seguro, toda información sensible por ejemplo, contraseñas en un solo sentido, es decir no podrá ser visto ni por el administrador de sistema.
- f) Se implementó controles de acceso en las aplicaciones Web, para evitar la ejecución de funciones por parte de usuarios no autorizados.
- g) La aplicación Web establece sesiones para mantener el rastro del flujo, de acciones de cada usuario. Para el establecimiento y mantenimiento de dichas sesiones se implementa mecanismos que garanticen la protección de las credenciales en tránsito y del identificador de sesión.

6.1. INTRODUCCIÓN

Unas de las tareas más importantes en la planificación de los proyectos de software es la estimación, el cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costos, tiempo y esfuerzo necesarios para el desarrollo de los mismo . En este presente capitulo examinaremos estos términos costos – beneficios, mediante COCOMO II, modelo de estimación de costos, mediante el obtendremos el esfuerzo, tiempo y personal necesarios para el desarrollo del software, también utilizaremos los modelos del VAN, TIR y C/B para obtener los beneficios a los que incurriría la empresa.

En las secciones siguientes examinaremos diversos aspectos de los cálculos de costo/beneficio:

- Análisis de costos.
- Análisis de beneficios.
- Como expresar los ahorros.
- Análisis de riesgos.

6.2. ANÁLISIS DE COSTOS

Se debe calcular todos los costos pronosticados asociados al sistema. Para establecer el costo del software desarrollado, se utiliza el modelo constructivo COCOMO II, que están orientados a los puntos de función. En la siguiente tabla 5.1 se observa la estimación de puntos de función:

Parámetro de Medición	Cuenta	Factor de Ponderación	Total
Nº de entradas de usuario	5	4	20
Nº de salidas de usuario	3	5	10
Nº de peticiones de usuario	7	4	28
Nº de archivos	6	10	60
Nº de interfaces externas	2	7	14
Cuenta Total			135

Tabla 5.1. Calculo de punto función
Fuente: Elaboración Propia

Para el cálculo del factor de complejidad técnica TCF, se toma en cuenta la tabla 5.2, para considerar la siguiente formula

$$TCF = (0.65 + 0.01 * 51)$$

$$TCF = 1.16$$

Nº	CARACTERÍSTICAS	FACTOR F _i
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?	5
2	¿Se requiere comunicación de datos?	5
3	¿Existen funciones de procesamiento distribuido?	2
4	¿Es crítico el rendimiento?	2
5	¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	4
6	¿Requiere el sistema entrada de datos interactiva?	3
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	4
8	¿Se actualizan los archivos maestros de forma interactiva?	2
9	Son complejas las entradas, las salidas, los archivos o las peticiones?	3
10	¿Es complejo el procesamiento interno?	4
11	¿Se ha diseñado el código para ser reutilizable?	3
12	¿Están incluidas en el diseño la conversión y la instalación?	5
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
14	14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	4
Factor de complejidad Técnica		51

Tabla 5.2. Calculo de punto de función ajustada
Fuente: Elaboración Propia

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

$$PF = \text{Cuenta Total} * TCF$$

$$PF = 135 * 1.16$$

$$PF = 156.6$$

Conversión de los puntos de función a KLDC

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a miles de líneas de código fuente considerando el lenguaje de implementación que se muestra en la tabla 5.3.

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
Ansi Cobol	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

Tabla 5.3. Conversión de puntos de función.
Fuente: Elaboración Propia

$$LDC = PF * \text{Factor LDC/PF}$$

$$LDC = 156.6 * 29$$

$$LDC = 4591.98$$

Las líneas de código en su totalidad son 4440.48, entonces el número estimado de líneas de código distribuidas en miles es:

$$KLCD = LCD / 1000$$

$$KLDC = 4591.98 / 1000 = 4.6$$

Por tanto existen 4.6 líneas de código distribuidas para el proyecto.

Ahora se aplican las formulas básicas de esfuerzo, tiempo calendario y personal requerido.

Las ecuaciones del COCOMO básico tienen la siguiente forma:

$$E = a_b(KLDC)^{b_b} \dots \dots \dots (1)$$

$$D = c_b D^{d_b} \dots \dots \dots (2)$$

Donde:

E: Esfuerzo aplicado en personas por mes.

D: Tiempo de desarrollo en meses cronológicos.

KLDC: Número estimado de líneas de código distribuidas (en miles).

Proyecto de software	a _b	b _b	c _b	d _b
Orgánico	2.4	1.05	2.5	0.38
Semi – acoplado	3	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Tabla 5.4. Relación de valores del modelo COCOMO
Fuente: Elaboración Propia

En la tabla 5.4 se muestran los tipos de proyectos de software, como este es un proyecto intermedio, en tamaño y complejidad, se elige semi-acoplado.

Reemplazando los datos en la ecuación 1 se tiene:

$$E = 3(4.6)^{1.12}$$

$$E = 16.57 \text{ [programadores/mes]}$$

$$D = 2.5(16.57)^{0.35}$$

$$D = 6.68 \text{ [meses]}$$

El personal requerido, en este caso el número de programadores (N° Prog) se obtiene con la siguiente fórmula:

$$N^{\circ} \text{ Prog} = E/D$$

$$N^{\circ} \text{ Prog} = 16.57 / 6.68$$

$$N^{\circ} \text{ Prog} = 2.48$$

$$N^{\circ} \text{ Prog} = 2 \text{ [programadores]}$$

El salario de un programador puede oscilar entre los 2166 \$us, cifra que es tomada en cuenta para la estimación siguiente:

Costo del software desarrollado por persona = Numero de programadores * salario de un programador

$$\text{Costo del software desarrollado por persona} = 2 * 2166 \text{ \$us}$$

$$\text{Costo del software desarrollado por persona} = 4332 \text{ \$us}$$

$$\text{Costo total del software desarrollado} = 4332 * 2$$

$$\text{Costo total del software desarrollado} = 8664 \text{ \$us}$$

De donde concluimos que para el desarrollo del “Sistema Web Académico para el Instituto Tecnológico Ayacucho”, es necesario contar con **2 programadores** durante **7 meses** para su respectivo desarrollo, el tendrá un **costo de 8664\$us** en bolivianos sería 60648 Bs.

6.3. ANÁLISIS DE BENEFICIOS

Para evaluar los beneficios que se obtendrá al implementar el proyecto se calcula con el método VAN y el TIR.

El VAN (Valor Actual Neto) es un indicador financiero que mide los flujos de los futuros ingresos y egresos que tendrá el proyecto, para determinar si luego de descontar la inversión inicial quedara alguna ganancia. Si el resultado es positivo, el proyecto es viable.

Para hallar el VAN del proyecto de inversión se requiere tres valores de acuerdo a la siguiente fórmula:

$$VAN = \sum \frac{ganacias}{(1 + 0.10)^n} - \sum \frac{costos}{(1 + 0.10)^n}$$

$$VAN = \sum \frac{15000}{(1 + 0.10)^5} - \sum \frac{8664}{(1 + 0.10)^5}$$

$$VAN = 12550 \text{ \$us}$$

La regla del VAN, que indica qué decisión tomar, es:

- Si el VAN es mayor que cero, se debe aceptar.
- Si el VAN es igual a cero, se debe ser indiferente.
- Si el VAN es menor que cero, se debe rechazar.

Entonces con VAN es mayor a cero, podemos decir que el proyecto sea aceptado.

Luego:

$$\text{CostoBeneficio} = \frac{\sum \text{ganancias}}{\sum \text{costos}}$$

$$\text{CostoBeneficio} = \frac{\sum 15000}{\sum 8664}$$

$$\text{CostoBeneficio} = 1.73 \text{ \$us}$$

De donde se concluye que por cada 1 \$us invertido ganamos 0.73 \$us, lo cual significa que el proyecto es una decisión adecuada.

El TIR (Tasa Interna de Retorno) es la tasa de descuento TD de un proyecto de inversión que permite que el BNA sea igual a la inversión (VAN igual a 0). La TIR es la máxima TD que puede tener un proyecto para que sea rentable.

La regla de la TIR, que indica qué decisión tomar, es:

- Si la TIR es mayor que la tasa de descuento, se debe aceptar.
- Si la TIR es igual a la tasa de descuento, se debe ser indiferente.
- Si la TIR es menor que la tasa de descuento, se debe rechazar.

$$\text{TIR} = \frac{\sum \text{ganancias} - \text{costos}}{(1 + 0.10)^n}$$

$$\text{TIR} = \frac{\sum 15000 - 8664}{(1 + 0.10)^5}$$

$$\text{TIR} = 3934.17$$

Por lo tanto la rentabilidad que el presente proyecto está proporcionando es de 3934.17 \$us, como es mayor a la tasa de descuento, se acepta el proyecto.

7.1 CONCLUSIONES

Con el desarrollo del presente proyecto la institución se beneficia por contar con una base de datos que contiene información trascendental para el desempeño de sus actividades en el área académica.

Se logró implementar mecanismos de control para las inscripciones considerando los prerequisites de las diferentes materias y el semestre al que le corresponde cursar.

Mucha de la información es almacenada en mecanismos de almacenamiento persistentes, lo beneficia en cuanto al ahorro de material de escritorio para contener información.

La búsqueda de información se realiza de modo automático logrando generar información oportuna.

Se facilitó la generación de copias de seguridad que pueden ser realizados de forma periódica según las necesidades de la institución.

Habiendo logrado resolver el principal problema que tenía la institución, la demora en la obtención de información a causa de la extensa información que se almacenaban en medios tangibles provocando que las funciones de los administrativos sea tediosa.

Se concluye que el **“SISTEMA WEB ACADÉMICO INTEGRAL PARA EL INSTITUTO TECNOLÓGICO AYACUCHO” - S.A.I.I.T.A.**, es un aporte tecnológico considerable, tomando en cuenta que antes de su implementación no se contaba con ningún tipo de tecnologías especializadas que ayuden a minorizar las responsabilidades de los administrativos siendo el mismo una herramienta útil de trabajo.

7.2 RECOMENDACIONES

- Utilizando de metodologías de desarrollo nos lleva a realizar software de calidad en cada fase de desarrollo a través de un buen control sobre los procesos, teniendo también una mayor transparencia del mismo. No importando las dimensiones del proyecto es siempre viable utilizar una metodología de desarrollo para producir lo que se espera en el tiempo definido y con el presupuesto calculado.

- Si el proyecto en curso contara con políticas establecidas antes de su implementación en cuanto a la elaboración del pensum académico de las diferentes carreras que ofreció la academia con anterioridad, se tendría información más completa de todos los estudiantes que pasaron por esta casa de estudios, siendo esta observación una recomendación a futuro para establecer reglas para la elaboración y documentación de nuevos pensums académicos.



REFERENCIAS

LIBROS

- CAMPDERRICH Flagueras Benet, 2003: *Ingeniería de Software*, Eureka Media, UOC, Barcelona.
- UWE, en *Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio* Citlali G. Nieves-Guerrero, Juan P. Ucán-Pech Víctor H. Menéndez-Domínguez Universidad Autónoma de Yucatán Mérida, Yucatán, México 2014
- FOWLER Martín, 1999: *UML gota a gota*, Addison Wesley Longman, México S.A., México, D. F.
- HERNÁNDEZ Roberto y FERNÁNDEZ Carlos., 1998: *Metodología de la Investigación*, McGraw-Hill Interamericana, México, D.F.
- LARMAN G., 1999: *Análisis y Diseño Orientado a Objetos UML y patrones*, 1a. ed. Prentice-Hall, México.
- PRESSMAN Roger, 2002: *Ingeniería de Software*, 5a. ed., McGraw-Hill.
- RAMOS Salavert Isidro, LOZANO Pérez Dolores, 2000: *Ingeniería del Software y Base de datos, Tendencias Actuales*, 1ra. ed., Universidad de Castilla – La Mancha, España.
- BOOCH & JACOBSON, “Proceso Unificado de Desarrollo de Software”, 1999.

- PRESSMAN, R. S. “INGENIERÍA DE SOFTWARE” Un Enfoque Práctico, 2006, sexta edición, McGraw-Hill, Madrid.

PROYECTOS DE GRADO

- Perez Elizabeth y Laura Wendy, 2001: *Sistema de Seguimiento Académico Instituto Normal Superior Simón Bolívar*, T358, Proyecto de Grado, La Paz-Bolivia.
- Chacolla Quispe, Víctor, 2003: *Sistema de Gestión Académica Vía Intranet: Inscripciones y Consultas para el Instituto Normal Superior Simón Bolívar*, T169, Proyecto de Grado, La Paz- Bolivia.
- Tarqui Carpio, Nelson Egberto, 2003: *Sistema de Seguimiento Académico Universidad Tecnológica Boliviana (U T B)*,T169, Proyecto de Grado, La Paz-Bolivia
- Flores, Jesús Lalo, 2005: *Portal de Gestión Académica Estudiantil vía Web*, T1129 Proyecto de Grado, La Paz-Bolivia.
- Sirpa Caceres, Rocio de la Azucena 2005: *Sistema de Gestión Académica Universitaria Caso: Facultad de Ingeniería*, T1241, Proyecto de Grado, La Paz-Bolivia
- Flores Condori, Freddy 2003: *Control y Seguimiento Académico Web para la Empresa Educomser SRL*, T1879, Proyecto de Grado, La Paz-Bolivia
- Diego Omar Chambi Lima 2007: “*Sistema de Gestión Académica para el Instituto Normal Superior Simón Bolívar*” T1979 Proyecto de Grado, La Paz-Bolivia

SITIOS DE INTERNET

- <http://proyectogradoingenieriasistemas.blogspot.com/2015/03/metodologia-uwe-uml-uml-based-web.html>, [consulta: 13 de septiembre 2015].
- ASSI , 2007, Asociación de investigación en software inteligente (ASSI): Ingeniería Web [en línea], < <http://www.ASSI.html> >, [consulta: 14 de septiembre 2015].
- AMBLER, 2000, Metodologías de proceso ágil, [en línea], < <http://www.Agilmodeling.com> >, [consulta: 24 de septiembre 2015].
- IEEE, Norma IEEE estándar, [en línea], < <http://www.Iee.org> >, [consulta: 10 de Octubre 2015].
- SCOTT, 2005, El proceso unificado ágil, [en línea] <<http://www.cc.una.ac.cr/AUP/index.html> >, [consulta: 15 de Octubre 2015].
- PINTO, 1627, Modelo Oohdm: el nuevo concepto dela cultura en la imagen, [en línea], <<http://www.hipertexto.info/documentos/oohdm.htm>>, [consulta: 18 de octubre 2015].
- J. MUÑOZ, 2000, Metodologías Agiles, [en línea], <<http://es.slideshare.net/jofese/metodologas-giles-y-metodologias-web-apra-el-desarrollo-de-software>>, [consulta: 4 de octubre 2015].
- ESC, 2002, Metodologías web, [en línea] <<http://es.slideshare.net/vabl/qu-es-la-ingeniera-web-7367015>>, [consulta: 10 de octubre 2015].