

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



PROYECTO DE GRADO

**“SISTEMA DE REGISTRO, SEGUIMIENTO Y CONTROL DE TESIS
CASO: BIBLIOTECA CENTRAL”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: DANNY ALMANZA AMARU

TUTOR METODOLÓGICO: M.Sc. ALDO RAMIRO VALDEZ ALVARADO

ASESOR: Lic. MOISÉS MARTIN SILVA CHOQUE

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 ESTRICAMENTE 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

Dedico este trabajo primeramente a Dios que me dio la oportunidad de vivir y regalarme una familia maravillosa.

A mis padres Mario y Bárbara que con su amor y dedicación, me dieron la fuerza para seguir adelante.

A mis hermanos Dorian y Mary Luz que siempre han estado conmigo apoyándome en todo momento.

A los amigos que de alguna manera me apoyaron y me dieron su comprensión y sabiduría.

AGRADECIMIENTOS

A toda mi familia que me apoyó y me dio consejos para salir adelante.

A mi tutor M.Sc. Aldo Ramiro Valdez Alvarado y a mi asesor Lic. Moisés Martín Silva Choque por su orientación, comprensión y consejos, dedicando tiempo y paciencia para que pueda culminar este proyecto.

A Mg.Sc. Marilin Sánchez Rada, Jefe Biblioteca Central de la Universidad Mayor de San Andrés, por darme oportunidad de desarrollar el presente trabajo.

A los docentes que me transmitieron sus conocimientos y me formaron profesionalmente.

A mi querida Carrera de Informática que fue prácticamente mi segundo hogar y de seguro voy a extrañar.

A los Administrativos de la Biblioteca Central UMSA por su buena disposición y colaboración.

A los amigos y compañeros que me colaboraron y apoyaron en el transcurso de mi carrera, gracias a todos.

RESUMEN

La Biblioteca Central de la Universidad Mayor de San Andrés brinda el servicio de préstamo de libros, de periódicos que es la sección de hemeroteca, mapas y folletos de años anteriores y otros a la comunidad Universitaria, Docentes y Público en General. Otro servicio prestado a la comunidad universitaria es la recepción de las Tesis de los estudiantes egresados y profesionales de Postgrado. Teniendo que atender a los estudiantes para la entrega de los certificados para cumplir con los requisitos necesarios del estudiante.

El presente proyecto refleja el desarrollo del Sistema de Registro, Seguimiento y Control de Tesis para la Biblioteca Central de la UMSA, con el fin de mejorar el manejo de la información de las Tesis, de manera eficiente.

En las bases teóricas se explica detalladamente los métodos, técnicas y herramientas, que se utilizaron para la elaboración del proyecto.

En primer lugar se obtuvieron los requerimientos con las Historias de Usuarios, en donde es imprescindible conocer el problema a profundidad para poder construir la aplicación y analizar por medio de Diagramas Entidad Relación, Diagramas de Clases Navegacionales y Diagrama de Contexto Navegacional.

En segundo lugar, se planteó la estructura de solución, concentrando los esfuerzos en las necesidades del usuario gracias a las “Historias de usuario y las tareas”, esta es la fase de exploración, y seguidamente se establecieron tiempos (fechas), para cada iteración, en la fase de planificación.

Seguidamente, en la fase de iteraciones y pruebas (la más importante), se diseñó y desarrolló el programa a ser aplicado, además se realizaron pruebas de aceptación a fin de verificar que el software no presente fallas y cumpla con los requisitos del cliente.

Por último, se calcularon las métricas de calidad, análisis de costos y se establecieron algunas políticas de seguridad.

SUMMARY

The Central Library of the Universidad Mayor de San Andrés offers the service of borrowing books, newspaper which is the section of periodicals, maps and brochures and other previous years to the University community, teachers and the general public. Another service to the university community is receiving the thesis of graduates and professionals graduate students. Having to serve students for the delivery of certificates to meet the requirements of the student.

This project reflects the development of the Registration System, Monitoring and Control Thesis for the Central Library of the UMSA, in order to improve the handling of information of Thesis efficiently.

In the theoretical basis it is explained in detail the methods, techniques and tools that were used to prepare the project.

First the requirements were obtained with User Stories, where it is essential to know the problem in depth to build the application and analyze through Entity Relationship Diagrams, Class Navigational Diagrams and Diagram Navigational Context.

Second, the solution was raised structure, concentrating on the needs of the user thanks to the "user stories and tasks" This is the exploration phase, and then settled times (dates) for each iteration in the planning phase.

Then, in the phase of iterations and testing (the most important), it was designed and developed the program to be applied in addition acceptance tests were conducted to verify that the software does not present failures and meets customer requirements.

Finally, the calculated quality metrics, cost analysis and some security policies were established.

CONTENIDO

1. CAPÍTULO I MARCO REFERENCIAL.....	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES.....	2
1.3 PROBLEMA CENTRAL.....	3
1.3.1 PROBLEMAS SECUNDARIOS	3
1.4 PLANTEAMIENTO DE OBJETIVOS.....	4
1.4.1 OBJETIVO GENERAL.....	4
1.4.2 OBJETIVOS ESPECÍFICOS.....	4
1.5 JUSTIFICACIÓN	5
1.5.1 JUSTIFICACIÓN TÉCNICA.....	5
1.5.2 JUSTIFICACIÓN SOCIAL.....	5
1.5.3 JUSTIFICACIÓN ECONÓMICA.....	5
1.6 LIMITES Y ALCANCES	6
1.6.1 LIMITES.....	6
1.6.2 ALCANCES.....	6
1.7 PLANIFICACIÓN	7
1.8 METODOLOGÍA	7
1.8.1 INVESTIGACIÓN CIENTÍFICA	7
1.8.2 ÁRBOL ANALÍTICO	7
1.8.3 INGENIERÍA	7
1.8.3.1 HERRAMIENTAS.....	7
1.8.3.2 TÉCNICAS.....	8
2. CAPÍTULO II MARCO TEÓRICO	9
2.1 INSTITUCIÓN	9
2.1.1 BIBLIOTECA CENTRAL.....	9
2.1.2 MISIÓN.....	9
2.1.3 VISIÓN	9
2.2 INGENIERÍA DE SOFTWARE	10
2.3 EXTREME PROGRAMMING – XP	10
2.3.1 VALORES QUE INSPIRAN XP	10

2.3.1.1 COMUNICACIÓN.....	11
2.3.1.2 SIMPLICIDAD	11
2.3.1.3 VALENTÍA.....	11
2.3.1.4 FEEDBACK.....	12
2.3.2 PRÁCTICAS.....	12
2.3.3 ACTORES Y SUS RESPONSABILIDADES.....	13
2.3.4 CICLO DE VIDA DEL MÉTODO XP	14
2.3.4.1 PLANIFICACIÓN DEL PROYECTO	15
2.3.4.2 DISEÑO	17
2.3.4.3 CODIFICACIÓN.....	19
2.3.4.4 PRUEBAS.....	19
2.4 INGENIERÍA WEB	20
2.5 OBJECT ORIENTED HYPERMEDIA DESIGN METHOD - OOHDM.....	21
2.5.1 INTRODUCCIÓN.....	21
2.5.2 FASES DEL MODELO OOHDM.....	21
2.5.2.1 FASE CONCEPTUAL.....	22
2.5.2.2 FASE NAVEGACIONAL.....	23
2.5.2.3 FASE DE INTERFAZ ABSTRACTA	25
2.5.2.4 FASE DE IMPLEMENTACIÓN	26
2.6 HERRAMIENTAS DE DESARROLLO	27
2.6.1 DEFINICIÓN DEL LENGUAJE DE PROGRAMACIÓN.....	27
2.6.1.1 ¿QUÉ ES PHP?.....	27
2.6.1.2 CARACTERÍSTICAS DE PHP	28
2.6.1.3 CÓMO FUNCIONA PHP	28
2.6.2 HOJA DE ESTILO CSS.....	29
2.7 ARQUITECTURA DEL SISTEMA	29
3. CAPÍTULO III MARCO APlicativo.....	30
3.1 INTRODUCCIÓN.....	30
3.2 FASE DE EXPLORACIÓN	31
3.2.1 HISTORIAS DE USUARIO Y TAREAS	31
3.2.1.1 DETALLE DE HISTORIAS DE USUARIOS.....	38
3.2.2 DIAGRAMA ENTIDAD RELACIÓN	39
3.2.2.1 DIAGRAMA DE CLASES NAVEGACIONALES.....	40

3.2.2.2 DIAGRAMA DE CONTEXTOS NAVEGACIONALES	40
3.3 FASE DE PLANIFICACIÓN	42
3.3.1 ESTIMADORES DE ESFUERZO	42
3.3.2 PLANIFICACIÓN POR HISTORIAS	42
3.3.3 CRONOGRAMA DE DESARROLLO	43
3.4 FASE DE ITERACIONES Y PRUEBAS	43
3.4.1 PRIMERA ITERACIÓN	43
3.4.2 SEGUNDA ITERACIÓN	47
3.4.3 TERCERA ITERACIÓN	50
3.4.4 CUARTA ITERACIÓN	52
3.5 FASE DE PRUEBAS	55
3.5.1 PRUEBAS DE INTEGRACIÓN	55
3.5.1.1 TEST LOGIN USUARIO	56
3.5.1.2 TEST USUARIO	57
3.5.1.3 TEST FACULTAD	58
3.5.1.4 TEST CARRERA	59
3.5.1.5 TEST MODALIDAD	60
3.5.1.6 TEST REGISTRO TESIS	61
3.5.1.7 RESULTADO DE LAS PRUEBAS DE INTEGRACIÓN	62
3.5.2 PRUEBAS DE ESTRÉS	62
3.6 FASE DE MANTENIMIENTO	64
3.7 DISEÑO DE LA ARQUITECTURA DEL SISTEMA	64
3.8 FASE CIERRE DE PROYECTO	65
3.9 VALORES	65
3.9.1 COMUNICACIÓN	65
3.9.2 SIMPLICIDAD	65
3.9.3 VALENTÍA	65
3.9.4 FEEDBACK	65
4. CAPÍTULO IV CALIDAD DE SOFTWARE Y SEGURIDAD	66
4.1 INTRODUCCIÓN	66
4.2 CALIDAD DE SOFTWARE	66
4.2.1 FUNCIONALIDAD	66

4.2.2 CONFIABILIDAD	70
4.2.3 PORTABILIDAD	71
4.2.4 MANTENIBILIDAD.....	71
4.2.5 RESULTADOS	73
4.3 SEGURIDAD	73
4.3.1 SEGURIDAD DEL SISTEMA	73
4.3.2 SEGURIDAD DE INFORMACIÓN.....	74
5. CAPÍTULO V ANÁLISIS, COSTO/BENEFICIO	75
5.1 ANÁLISIS DE COSTOS	75
5.2 COSTO DEL SOFTWARE DESARROLLADO.....	75
5.3 COSTO DE IMPLEMENTACIÓN	81
5.4 COSTO DE LA ELABORACIÓN DEL PROYECTO	81
5.5 COSTO TOTAL.....	82
5.6 VALOR ACTUAL NETO (VAN).....	82
5.7 TASA INTERNA DE RETORNO (TIR).....	83
5.8 COSTO BENEFICIO	84
5.9 RESULTADO VAN, TIR, B/C.....	84
6. CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES	85
6.1 RESUMEN	85
6.2 CONCLUSIONES.....	85
6.3 RECOMENDACIONES.....	86
7. BIBLIOGRAFÍA	87
8. ANEXOS	89
ANEXO A - ÁRBOL DE PROBLEMAS	89
ANEXO B - ÁRBOL DE OBJETIVOS.....	90
ANEXO C - ÁRBOL DE ALTERNATIVAS	91
ANEXO D - MARCO LÓGICO	92
ANEXO E - CRONOGRAMA DE DESARROLLO	94
ANEXO F – CERTIFICADO DE TESIS.....	95

ÍNDICE DE FIGURAS

Figura 2.1: Valores del Método XP	11
Figura 2.2: Fases de XP	14
Figura 2.3: Fases del Modelo OOHDM	21
Figura 2.4: Diagrama entidad relación	23
Figura 2.5: Diagrama de clases navegacionales	24
Figura 2.6: Diagrama de contextos navegacionales	25
Figura 2.7: Abstract Data Views	26
Figura 3.1: Diagrama Entidad Relación	39
Figura 3.2: Diagrama de clases Navegacionales	40
Figura 3.3: Diagrama Contextual Navegacional	41
Figura 3.4. Abstract Data Views	43
Figura 3.5: Listado de Usuarios.....	44
Figura 3.6. Formulario de llenado de usuarios	44
Figura 3.7. Formulario de actualización de usuarios.....	45
Figura 3.8. Formulario de ingreso al Sistema.....	45
Figura 3.9. Formulario de registro de Tesis.....	47
Figura 3.10. Formulario de actualización de Tesis.....	48
Figura 3.11. Reporte en PDF para la impresión	48
Figura 3.12. Formulario de Búsqueda	50
Figura 3.13. Formulario de Búsqueda por Código	50
Figura 3.14. Formulario de Búsqueda por Título	51
Figura 3.15. Formulario de Búsqueda por Autor.....	51
Figura 3.16. Formulario de Reportes.....	53
Figura 3.17. Reporte Excel por Facultad	53
Figura 3.18. Reporte Excel por Año	54
Figura 3.19. Reporte Excel por mes	54
Figura 3.20: Test Login Usuario	56

Figura 3.21: Test Usuario	57
Figura 3.22: Test Facultad	58
Figura 3.23: Test Carrera.....	59
Figura 3.24: Test Modalidad	60
Figura 3.25: Test Registro Tesis.....	61
Figura 3.26: Resultado de las Pruebas de Integración.....	62
Figura 3.27: Reporte de resumen.....	62
Figura 3.28: Gráfico del resultado.....	63
Figura 3.29. Arquitectura del Sistema.....	64
Figura 8.1: Cronograma de Desarrollo	94
Figura 8.2: Certificado de Tesis	95

ÍNDICE DE TABLAS

Tabla 2.1: Tarjeta de Historia de Usuario.....	15
Tabla 2.2: Esquema Conceptual	22
Tabla 3.1: Historia de Usuario Administración de usuarios.....	31
Tabla 3.2: Tarea Diseñar la estructura de datos para administrar usuarios	31
Tabla 3.3: Tarea Crear interfaz Usuarios.....	32
Tabla 3.4: Tarea Crear interfaz para registrar nuevos usuarios	32
Tabla 3.5: Tarea Crear interfaz para modificar usuarios	33
Tabla 3.6: Tarea Crear interfaz de acceso al sistema	33
Tabla 3.7: Historia de Usuario Registro de Tesis	34
Tabla 3.8: Tarea Diseñar la estructura de datos para registrar la Tesis	34
Tabla 3.9: Tarea Crear interfaz para registrar la nueva Tesis.....	35
Tabla 3.10: Tarea Crear interfaz para editar los registros de las Tesis.....	35
Tabla 3.11: Tarea Crear interfaz para visualizar la Tesis para imprimir	36
Tabla 3.12: Historia de Usuario Búsqueda de Tesis.....	36
Tabla 3.13: Tarea Crear interfaces para Filtrar las Tesis.....	37
Tabla 3.14: Historia de Usuario Reportes de Tesis	37
Tabla 3.15: Tarea Crear interfaces para los reportes de las Tesis	38
Tabla 3.16: Estimaciones de Esfuerzo.....	42
Tabla 3.17: Planificaciones por historia	43
Tabla 4.1 Cálculo de Punto Función	67
Tabla 4.2 Ajuste de Complejidad	69
Tabla 4.3: Tabla de resultados de calidad.....	73
Tabla 5.1 Conversión de puntos función	76
Tabla 5.2 Conductores de coste	78
Tabla 5.3 Coeficientes	80
Tabla 5.4 Detalle de Elaboración del proyecto.....	82
Tabla 5.5 Detalle del Costo Total	82
Tabla 5.6: Resultado de costos	84

CAPÍTULO I MARCO REFERENCIAL

1.1 INTRODUCCIÓN

La necesidad del manejo de la información cada vez se hace más frecuente en la sociedad. En la actualidad todas las organizaciones tienden a depender de herramientas más rápidas para el procesamiento de datos y toma de decisiones. El más requerido, usado en la mayoría de las empresas es la computadora, sistemas informáticos, con el fin de centralizar toda su información para el logro de sus objetivos.

La información debe estar disponible en cualquier momento de tal manera que sea oportuna, verdadera, confiable y efectiva. Es cierto que no es posible obtener información con estas características, si no se contara con la tecnología informática que está caracterizada por las computadoras y los programas que las hacen funcionar.

La Biblioteca Central de la Universidad Mayor de San Andrés brinda el servicio de préstamo de libros, de periódicos que es la sección de hemeroteca, mapas y folletos de años anteriores y otros a la comunidad Universitaria, Docentes y Público en General. Otro servicio prestado a la comunidad universitaria es la recepción de las Tesis de los estudiantes egresados de Pregrado y profesionales de Postgrado. Teniendo que atender a los estudiantes para la entrega de los certificados para cumplir con los requisitos necesarios del estudiante y continuar con los trámites para la colación.

Con el paso de los años son más estudiantes que salen egresados de las diferentes carreras de Pregrado y de Postgrado que cuenta la Universidad Mayor de San Andrés, y guardar la información se hace más complicado acorde al crecimiento de los datos.

La propuesta a realizar es presentar un sistema para automatizar los registros del formulario de los certificados emitidos a los interesados, facilitando el uso de la información de manera más eficiente y confiable, optimizando el tiempo invertido en el proceso del llenado de los certificados de entrega de las Tesis.

1.2 ANTECEDENTES

Por lo general un buen servicio que se puede brindar es la generación de información confiable para evitar errores en el momento de emitir los formularios de certificados de confirmación a los interesados que se presentan en la Biblioteca Central de la UMSA.

Los procesos para el control y manejo de la información de las Tesis son realizados de forma manual, situación que la información proporcionada para el formulario de los certificados de Tesis sea la correcta, porque puede presentar errores en el llenado de los certificados.

Los proyectos de grado relacionados con el control de inventarios, entre los cuales podemos mencionar los siguientes:

Inventario para los suministros médicos para el Seguro Social Universitario SSU, realizado con la metodología RUP¹ y UML² para la elaboración del Sistema Web local. (Mamani, 2013)

Inventario para pedidos de muebles para la empresa Antiguo Arte Europeo S.A., utiliza las herramientas de Visual Basic 6.0 para elaborar el sistema, la Base de Datos utilizada es Microsoft Access, y utilizando Ingeniería de Software. (Villa, 2007)

Los proyectos de grado relacionados con el registro de las tesis se lo pueden mencionar:

Inventario de Libros y Tesis que cuenta la Biblioteca de la Carrera de Informática UMSA, realizado con la metodología UWE³ para la elaboración web, utilizando los metadatos estándar DUBLIN-CORE y una base de datos MySQL. (Álvarez, 2005)

¹Rational Unified Process, es una metodología estándar para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

²Unified Modeling Language, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

³UML – Based Web Engineering, es una herramienta para modelar aplicaciones web.

Biblioteca Virtual en Web Semántica para el postgrado en Informática, este trabajo se introduce a conceptos y notaciones de la Web Semántica, utiliza la metodología de desarrollo es OOHDM. (Apaza, 2008)

El proyecto relacionado con la metodología a ser utilizado se lo puede mencionar:

Sistema de Gestión de Correspondencia. Caso: Carrera de Administración de Empresas, utiliza las herramientas de PHP, con la metodología XP para el seguimiento de la correspondencia para ser consultado por los interesados. (Monasterio, 2012)

Sistema de Control y Gestión Hotelera, proyecto realizado con la metodología XP y diagramas UML para el desarrollo e implementación, también utilizaron las herramientas de PHP, Javascript y JQuery. (Carrasco, 2009)

1.3 PROBLEMA CENTRAL

La Biblioteca Central tiene fechas para la recepción de las Tesis y por lo tanto los encargados responsables tienen la función de llenar los formularios de los certificados y entregar a los interesados.

¿Cómo se puede disponer de manera inmediata las tesis, y el pronto acceso de los datos fiables y ordenados, para el personal de la Biblioteca Central de la Universidad Mayor de San Andrés?

1.3.1 PROBLEMAS SECUNDARIOS

En cuanto a los problemas que se encuentra se tiene que:

- Llenado manual de los certificados de Tesis a los interesados, lo que produce información incorrecta de las tesis.
- Entrega de certificados con datos erróneos de la Tesis a los interesados, ocasionando pérdidas del material de certificados.

- Solicitud de copias de certificado por parte del interesado. Lo que produce la búsqueda de registro en copias de certificados almacenados.
- Registro manual y repetitivo a Excel de las tesis recibidas al finalizar la semana de recepción, que produce demora en sus trabajos habituales diarias del personal.
- Elaboración de informes mensuales y anuales del funcionario manualmente para presentar los trabajos realizados. Lo que ocasiona pérdida de tiempo de los funcionarios, no realizando sus labores cotidianas después de la recepción de las Tesis.
- Trabajo arduo del personal.

(Ver ANEXO A - ÁRBOL DE PROBLEMAS)

1.4 PLANTEAMIENTO DE OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar e Implementar un sistema web para el registro, seguimiento y control de las Tesis ingresadas a la Biblioteca Central de la Universidad Mayor de San Andrés, facilitando al funcionario en el llenado, entrega de los formularios de certificados y la búsqueda de los certificados en tiempo real.

1.4.2 OBJETIVOS ESPECÍFICOS

- Facilitar la impresión de los certificados de las Tesis recibidas con información correcta.
- Permitir el llenado de los datos de las Tesis de los certificados de manera automatizada.
- Facilitar la copia de los certificados utilizando la Base de Datos para la entrega a los interesados.
- Agilizar la búsqueda de las Tesis almacenados en la Base de Datos.
- Generar los reportes automáticamente de las Tesis ingresadas a diferentes formatos.

- Elaborar informes de forma automática de las Tesis ingresadas por facultad y año.
- Ahorrar el tiempo de los funcionarios para la elaboración de reportes e informes.
- Facilitar el trabajo al funcionario para la entrega de certificados.

(Ver ANEXO B - ÁRBOL DE OBJETIVOS, y ANEXO C - ÁRBOL DE ALTERNATIVAS)

1.5 JUSTIFICACIÓN

1.5.1 JUSTIFICACIÓN TÉCNICA

Se justifica técnicamente, en un aporte para la Biblioteca Central UMSA, donde se ejecuta el servicio de entrega de certificados de Tesis a los interesados de forma manual. La Biblioteca Central cuenta con equipos de computación para la elaboración del sistema y la optimización de la entrega de los certificados.

Este sistema permitirá operar adecuadamente los trabajos de la Institución mecanizando el trabajo de los funcionarios responsables del llenado del formulario de los certificados de las Tesis. Se espera que sea una herramienta que facilitara las actividades en la Biblioteca Central de la Universidad Mayor de San Andrés (UMSA), actualmente realizados de manera manual, mejorando la gestión de los sistemas informáticos.

1.5.2 JUSTIFICACIÓN SOCIAL

Se justifica socialmente, porque beneficia a los funcionarios de la Biblioteca Central de realizar el llenado del formulario de los certificados de las Tesis, Proyectos de Grado, Monografías, Trabajos Dirigidos y otros, dado que el sistema se constituirá como una ayuda para realizar las entregas de los certificados a los interesados.

1.5.3 JUSTIFICACIÓN ECONÓMICA

La elaboración del proyecto se justifica económicamente debido a que la Biblioteca Central, la Universidad Mayor de San Andrés UMSA tiene una política de inversión

orientada a reducir costos. Realizando la reducción de costos de trabajo y costos en tiempo más aún si lo realizan de manera eficiente.

Disminuir la cantidad de documentación física que se manejan para verificación e información de las Tesis ingresadas cada mes.

1.6 LIMITES Y ALCANCES

1.6.1 LIMITES

El sistema se enfocara en el área de manejo de los registros de las tesis, tanto en los reportes que tienen que realizar y de la entrega de los certificados a los interesados.

- No tomara en cuenta el registro de los libros u otros datos usados por la Biblioteca Central UMSA.
- El sistema estará conformado por la implementación del proceso de Registro, Búsqueda de certificados ingresados, Reportes e informes a diferentes formatos.
- El proyecto se limitara en la Biblioteca Central – UMSA.

1.6.2 ALCANCES

El sistema pretende analizar las operaciones dentro de la Biblioteca Central de la UMSA, de ahí que se identifican los siguientes alcances:

- Registrar los certificados entregados de las Tesis que fueron recibidas en la Biblioteca Central de la Universidad Mayor de San Andrés.
- Realizar un listado de las Tesis de acuerdo a la fecha, título, autor.
- Facilitar el acceso a la información sobre las Tesis, minimizando el tiempo de búsqueda por el encargado.
- Reporte para la generación de información de cada Tesis ingresada por carrera, por facultad y por año.

1.7 PLANIFICACIÓN

El Sistema de Registro, Seguimiento y Control de Tesis será un aporte para la Biblioteca Central UMSA, debido a que permitirá cubrir las necesidades de los formularios de las Tesis que ingresan cada mes de las diferentes Facultades de la Universidad Mayor de San Andrés.

El sistema desarrollada una interfaz amigable para los funcionarios para utilizarlo de manera eficiente.

(Ver ANEXO D - MARCO LÓGICO)

1.8 METODOLOGÍA

1.8.1 INVESTIGACIÓN CIENTÍFICA

La investigación es un conjunto de procesos sistemáticos, críticos y empíricos que se aplican al estudio de un fenómeno. Para este estudio utilizamos el enfoque cuantitativo porque plantea un estudio delimitado y concreto para recolectar los datos y resolver el problema. (Hernández, Fernández, & Baptista, 2010)

1.8.2 ÁRBOL ANALÍTICO

Árbol de problemas

Es una técnica participativa que ayuda a desarrollar ideas creativas para identificar el problema y organizar la información recolectada. (Martínez & Fernández, 2006)

Árbol de objetivos

Es la versión positiva del Árbol de Problemas. Permite determinar las áreas de intervención que plantea el proyecto. (Martínez & Fernández, 2006)

1.8.3 INGENIERÍA

1.8.3.1 HERRAMIENTAS

Las herramientas a utilizar en el desarrollo e implementación son las siguientes: tecnologías web de Procesador de HiperTexto PHP 5.9, para el diseño web utilizaremos las

Hojas de Estilo CSS, para el comportamiento del sistema web y validación de los formularios utilizaremos JavaScript, Jquery, Jquery-UI, Jquery Validate, Bootstrap 3, para el almacenamiento de los datos del formulario de los certificados utilizaremos MySQL para la Base de Datos y el Framework Laravel 5.

1.8.3.2 TÉCNICAS

El modelado del sistema se elabora con algunos diagramas UML como los casos de uso, Diagramas de Clases, Diagramas de Secuencia, Diagrama de Actividades.



CAPÍTULO II MARCO TEÓRICO

2.1 INSTITUCIÓN

2.1.1 BIBLIOTECA CENTRAL

La Biblioteca Central UMSA fue fundado por el presidente de la República el Dr. Hernando Siles, mediante Decreto Supremo de 22 de marzo del año 1930. Desde agosto de 1949 cuenta con su local y ambientes propios ubicados en el edificio del MONOBLOCK de la Universidad Mayor de San Andrés. (Verástegui, 1980)

La Biblioteca Central UMSA cuenta con una sección de TESITECA⁴, donde se guardan las Tesis de Grado, Tesis de Maestría, Tesis de Doctorado, Proyectos de Grado, Tesinas de Grado, Monografías, Trabajos Dirigidos, Trabajos de Aplicación y varios, que se reciben cada mes de los estudiantes que realizarán su colación de Grado de las diferentes carreras que cuenta la Universidad Mayor de San Andrés.

2.1.2 MISIÓN

Es misión de la Biblioteca, recopilar, procesar y gestionar información, científico-técnica y cultural, en apoyo a planes y programas de estudios de Facultades, Carreras e Institutos de investigación, que forman parte de la comunidad universitaria de San Andrés, para apoyar a la actualización permanente de docentes, investigadores y alumnos, construyendo a la generación de nuevos conocimientos.

2.1.3 VISIÓN

Consolidarse en el nodo central del Sistema de Unidades de Información (SUI – UMSA) y nacional de las bibliotecas universitarias de nuestro país. (DTIC, 2005)

⁴TESITECA – Almacén de las tesis guardadas de las diferentes Carreras que cuenta la Universidad Mayor de San Andrés.

2.2 INGENIERÍA DE SOFTWARE

La ingeniería de software es el desarrollo, operación, mantenimiento del software de forma sistemática, disciplinada, cuantificable, y el estudio de los métodos que se aplican para generarlos. Por ende la Ingeniería de Software se podría definir *como el establecimiento y aplicación de principios de la Ingeniería para obtener software*, teniendo en cuenta factores como el coste económico, la fiabilidad del sistema y el funcionamiento eficiente para satisfacer las necesidades del usuario. (Del Valle, 2009)

2.3 EXTREME PROGRAMMING – XP

La programación extrema (XP) es probablemente la metodología ágil más conocida; está centrada en la colaboración, la creación temprana y rápida de software, y una serie de prácticas útiles en el desarrollo de software. (Fernández, 2002)

Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar (Beck, 2002). Una encuesta que realizó la IBM muestra la aprobación de la gente hacia este método.

Los objetivos perseguidos son:

- ✓ El objetivo principal que persigue XP es la satisfacción del cliente. Esta metodología fue diseñada para proporcionar el software que el cliente cuando lo necesite.
- ✓ El segundo objetivo es potenciar al máximo el trabajo en equipo. Tanto los jefes de Proyecto, como los clientes y desarrolladores, son parte del equipo.

2.3.1 VALORES QUE INSPIRAN XP

La Programación Extrema es una metodología ligera de desarrollo de software, se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito. (Peréz, 2010)

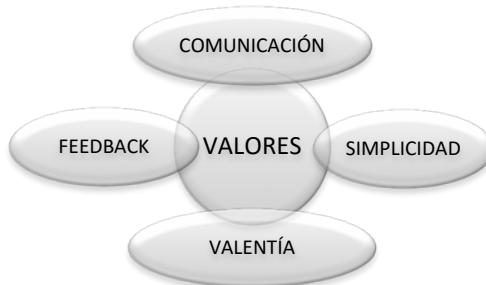


Figura 2.1: Valores del Método XP

Fuente: (Monasterio, 2012)

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

2.3.1.1 COMUNICACIÓN

XP pone en comunicación directa y continua a clientes y desarrolladores. El cliente se integra en el equipo para establecer prioridades y resolver dudas. De esta forma ve el avance día a día, y es posible ajustar la agenda y las funcionalidades de forma consecuente

2.3.1.2 SIMPLICIDAD

La simplicidad consiste en desarrollar sólo el sistema que realmente se necesita. Implica resolver en cada momento sólo las necesidades actuales.

Con este principio de simplicidad, junto con la comunicación y el feedback resulta más fácil conocer las necesidades reales.

2.3.1.3 VALENTÍA

Los desarrolladores involucrados deben de tener la actitud necesaria para lograr un cambio y alcanzar resultados de alta calidad, sin importar cuánto tiempo se ha invertido previamente en el mismo.

2.3.1.4 FEEDBACK

El cliente brinda retroalimentación a las funciones desarrolladas e incluyen sus comentarios en la próxima iteración para una mejor comprensión de sus necesidades, la retroalimentación constante y el involucramiento por parte del cliente garantizan un alto grado de satisfacción en el cliente.

Ésta metodología está basada en el desarrollo incremental de pequeñas partes, con entregas y pruebas frecuentes y continuas, proporciona un flujo de retro-información valioso para detectar los problemas o desviaciones.

- De esta forma fallos se localizan muy pronto.
- La planificación no puede evitar algunos errores, que sólo se evidencian al desarrollar el sistema.
- La retro-información es la herramienta que permite reajustar la agenda y los planes.

2.3.2 PRÁCTICAS

Los Pasos fundamentales inmersos en las fases del método son:

- **Desarrollo iterativo e incremental:** Pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas:** Son frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- **Programación en parejas:** Se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- **Frecuente integración del equipo de programación con el cliente o usuario:** Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores antes de añadir nueva funcionalidad.** Hacer entregas frecuentes.

- **Refactorización del código:** Es decir, reescribir ciertas partes del código para aumentar su legibilidad y Mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartido:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad del código:** es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

2.3.3 ACTORES Y SUS RESPONSABILIDADES

Hay diferentes roles en XP para diferentes tareas y propósitos durante el proceso y sus prácticas. (Fernández, 2002)

- **Programador.** El programador escribe las pruebas y mantiene el código del programa tan simple y definido como sea posible.
- **Cliente.** Escribe las historias (especificaciones) y pruebas funcionales, decide cuándo es logrado cada requisito y determina la prioridad de la implementación de cada uno de ellos.

- **Verificadores (Testers).** Ayudan al cliente a escribir pruebas funcionales, las corren regularmente, comunican los resultados de las mismas y mantienen las herramientas de prueba.
- **Seguidor de rastros (Tracker).** El seguidor de rastros retroalimenta en XP. Sigue las estimaciones hechas por el equipo, es decir, la estimación del esfuerzo; y da retroalimentación en cómo están desacertados para mejorar las futuras estimaciones.
- **Facilitador.** Es la persona responsable del proceso como un todo. Un buen entendimiento de XP es importante en este rol para habilitar al entrenador para guiar a los otros miembros del equipo en el siguiente proceso.
- **Consultor técnico.** El consultor técnico es un miembro externo que procesa el conocimiento técnico específico necesario y guía al equipo en resolver sus problemas específicos.
- **Administrador.** El administrador toma las decisiones. Para lograr esto, él se comunica con el equipo del proyecto para determinar la situación actual y distinguir cualquier dificultad o deficiencia en el proceso.

Tratándose de un proyecto de grado individual el proyectista asumirá todos estos papeles excepto el del cliente.

2.3.4 CICLO DE VIDA DEL MÉTODO XP

Si bien el ciclo de vida de un proyecto XP es muy dinámico, se puede separar en las siguientes fases:

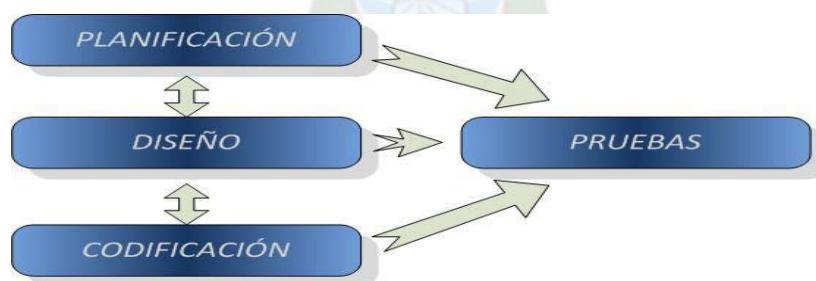


Figura 2.2: Fases de XP
Fuente: (Peréz, 2010)

2.3.4.1 PLANIFICACIÓN DEL PROYECTO

- **Historias de usuario:**

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

HISTORIA DE USUARIO	
Numero:	Usuario:
Nombre Historia:	
Prioridad en negocio:	Riesgo en desarrollo:
Puntos estimados:	Iteración asignada:
Descripción:	
Observaciones:	

Tabla 2.1: Tarjeta de Historia de Usuario

Fuente: (Peréz, 2010)

- **Release Planning:**

Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, en inglés "Release plan", donde se indiquen las historias de usuario que se

crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Un "Release plan" es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa. Después de un "Release plan" tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado. (*Release plan: Planificación de publicaciones).

- **Iteraciones:**

Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

- **La Velocidad del Proyecto:**

Es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 o 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Plan".

- **Programación en Parejas:**

La metodología XP aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado.

El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad.

- **Reuniones Diarias:**

Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

2.3.4.2 DISEÑO

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo. Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto.

Los aspectos que se tratarán a continuación son: simplicidad en el diseño, metáfora del sistema, tarjetas CRC, spike solution, no solucionar antes de tiempo y refactoring.

- **Diseños Simples:**

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implemémentable que a la larga costará menos tiempo y esfuerzo desarrollar.

- **Tarjetas de clase, responsabilidad, colaboración (CRC cards)**

La principal funcionalidad que tienen estas, es ayudar a dejar el pensamiento procedimental para incorporarse al enfoque orientado a objetos. Cada tarjeta representa una clase con su nombre en la parte superior, en la sección inferior izquierda están descritas las responsabilidades y a la derecha las clases que le sirven de soporte.

En el proceso de diseñar el sistema por medio de las tarjetas CRC como máximo dos personas se ponen de pie adicionando o modificando las tarjetas, prestando atención a los mensajes que éstas se transmiten mientras los demás miembros del grupo que permanecen sentados, participan en la discusión obteniendo así lo que puede considerarse un diagrama de clases preliminar.

- **Glosarios de Términos:**

Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

- **Riesgos:**

Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.

- **Funcionabilidad extra:**

Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

- **Refactorizar:**

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos.

2.3.4.3 CODIFICACIÓN

Como ya se dijo en la introducción, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

2.3.4.4 PRUEBAS

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. El uso de los test en XP es el siguiente:

- Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.
- Hay que someter a tests las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.

- Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican.
- Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.
- Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

2.4 INGENIERÍA WEB

La Ingeniería Web (IWeb) aplica “*sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad*”.

Actualmente existen millones de sitios web que cubren diferentes necesidades de la vida, seguramente la comunicación es la más importante teniendo en cuenta que las distancias pasaron a un segundo plano.

El avance de Internet y las nuevas tecnologías de comunicación, marcan una nueva tendencia en el mercado del software. La exigencia de los usuarios, cada vez más numerosos, hizo que Internet creciera en forma acelerada y descontrolada, produciendo un impacto social, económico y político impensado.

El nacimiento de las aplicaciones Web (WebApp) no se hizo esperar, aquellos sitios que en un comienzo fueron sólo páginas informativas, se vieron obligadas a brindar a sus visitantes algún tipo de servicio que conjugue páginas planas con datos almacenados. Todo este proceso, caótico, dio lugar al nacimiento de sitios no planificados en los que no se empleó una metodología apropiada. (Del Valle, 2009)

2.5 OBJECT ORIENTED HYPERMEDIA DESIGN METHOD - OOHDM

2.5.1 INTRODUCCIÓN

En la actualidad son pocas las metodologías existentes que permiten a los desarrolladores conseguir productos de software hipermedia reusables y fáciles de mantener. A pesar de ello, ha nacido una tendencia a considerar el desarrollo hipermedial, por lo que ya se han propuesto algunas metodologías para este fin.

Una de ellas es el modelo OOHDM o Método de Diseño de Hipermedia Orientado a Objetos, para diseño de aplicaciones hipermedia y para la Web, fue diseñado por D. Schwabe, G. Rossi, y S. D. J. Barbosa, este método se inspira en el modelo HDM, extendiéndose a la orientación de objetos, convirtiéndose en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web. (Escalona, 2004)

2.5.2 FASES DEL MODELO OOHDM

OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de cuatro fases:



Figura 2.3: Fases del Modelo OOHDM

Fuente: (Apaza, 2008)

2.5.2.1 FASE CONCEPTUAL

Durante esta actividad se construye un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos.

El esquema conceptual está construido por clases, relaciones y subsistemas.

	Esquema Conceptual
Productos:	Diagrama de Clases, División en subsistemas y relaciones
Herramientas:	Técnicas de modelado O. O, patrones de diseño
Objetivo de diseño:	Modelo semántico de la aplicación

Tabla 2.2: Esquema Conceptual

Fuente: (Monasterio, 2012)

Las *clases* son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real.

Diagrama Entidad – Relación. Se pueden representar gráficamente mediante el diagrama entidad relación (DER). El DER fue propuesto originalmente por Peter Chen para el diseño de sistemas de bases de datos relacionales y ha sido ampliado por otros. Se identifica un conjunto de componentes primarios para el DER: objetos de datos, atributos, relaciones y varios indicadores tipo. El propósito primario del DER es representar objetos de datos y sus relaciones. (Pressman, 2010)

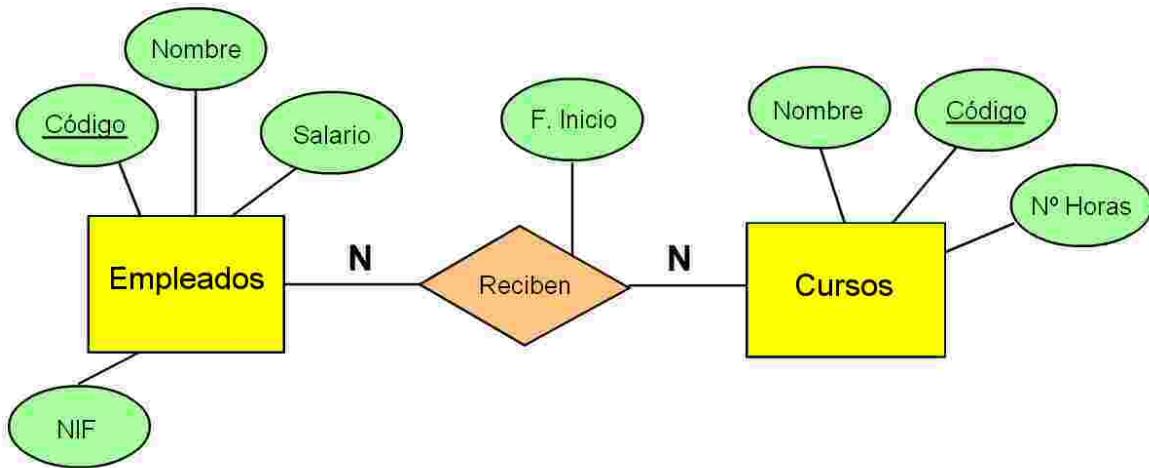


Figura 2.4: Diagrama entidad relación

Fuente: (Álvarez, 2005)

2.5.2.2 FASE NAVEGACIONAL

El modelo navegacional es construido como una vista sobre un diseño conceptual, aceptando la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual. El diseño de navegación es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales.

En OOHDM existe un conjunto de tipos predefinidos de clases navegacionales: nodos, enlaces y estructuras de acceso. La semántica de los nodos y los enlaces son las tradicionales de las aplicaciones hipermedia, y las estructuras de acceso, tales como índices o recorridos guiados, representan los posibles caminos de acceso a los nodos.

Diagrama de Clases Navegacionales

El diagrama de clases navegacionales es una vista del modelo estructural. *Cada conjunto de usuarios tiene una en particular*, y sus elementos son clases y asociaciones. Las clases reflejan la estructura de un documento de hipermedia, y son definidas como una vista de las clases conceptuales.

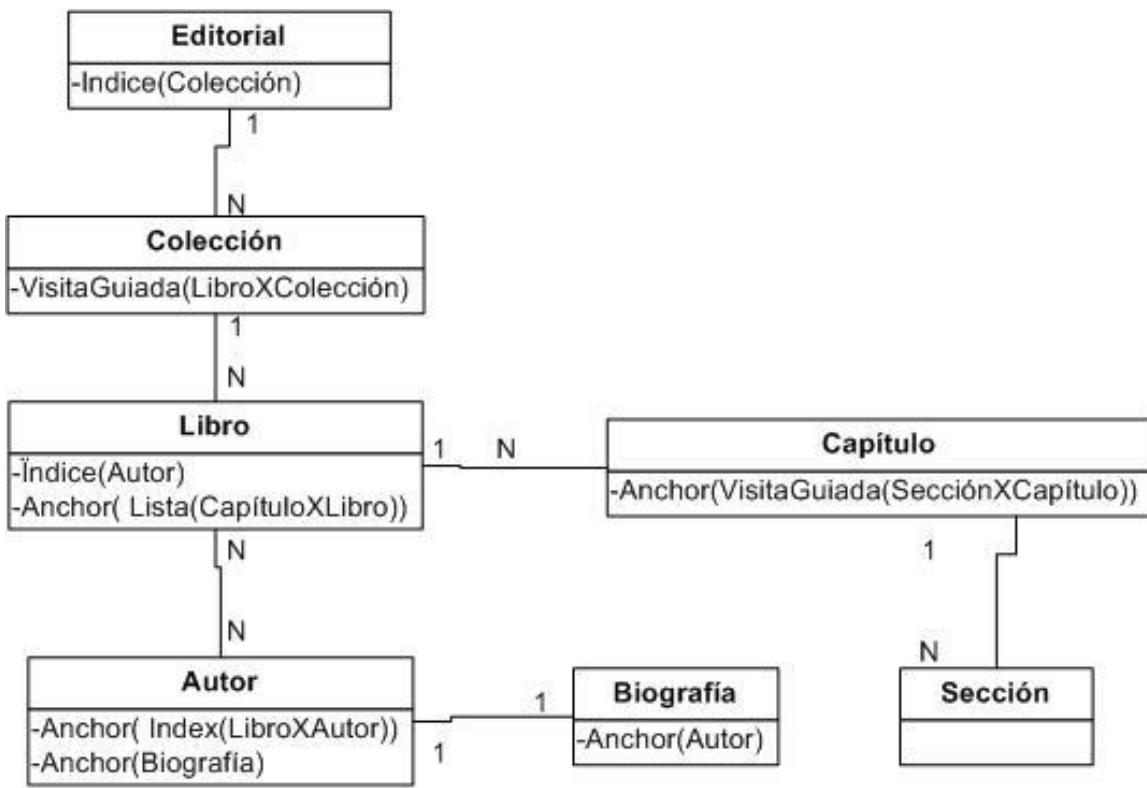


Figura 2.5: Diagrama de clases navegacionales

Fuente: (Solís, 2008)

- **Diagrama de Contextos Navegacionales**

Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio naveacional en conjuntos convenientes que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada. Los nodos son enriquecidos con un conjunto de clases especiales que permiten de un nodo observar y presentar atributos (incluidos las anclas), así como métodos (comportamiento) cuando se navega en un particular contexto

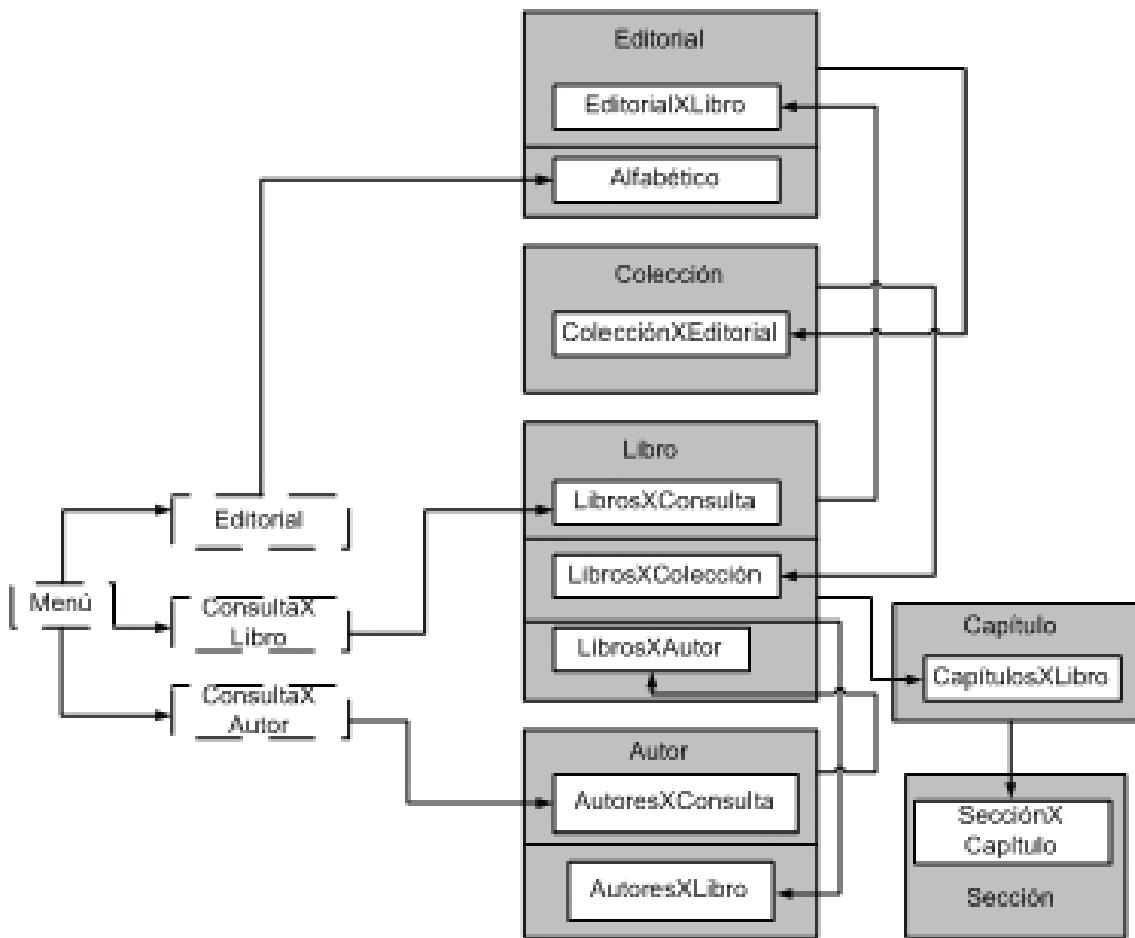


Figura 2.6: Diagrama de contextos navegacionales

Fuente: (Solís, 2008)

2.5.2.3 FASE DE INTERFAZ ABSTRACTA

Una vez definidas las estructuras navegacionales, se deben especificar los aspectos de interfaz. Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, de cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación, qué transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas.

Una clara separación entre diseño navegacional y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional, dejando un alto grado de independencia de la tecnología de interfaz de usuario.

El aspecto de la interfaz de usuario de aplicaciones interactivas (en particular las aplicaciones Web) es un punto crítico en el desarrollo que las modernas metodologías tienden a descuidar.

En **OOHDM** se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia. El modelo de interfaz **ADV's** (Vista de Datos Abstracta) especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las **ADV's** en la pantalla real son hechas en la fase de implementación.

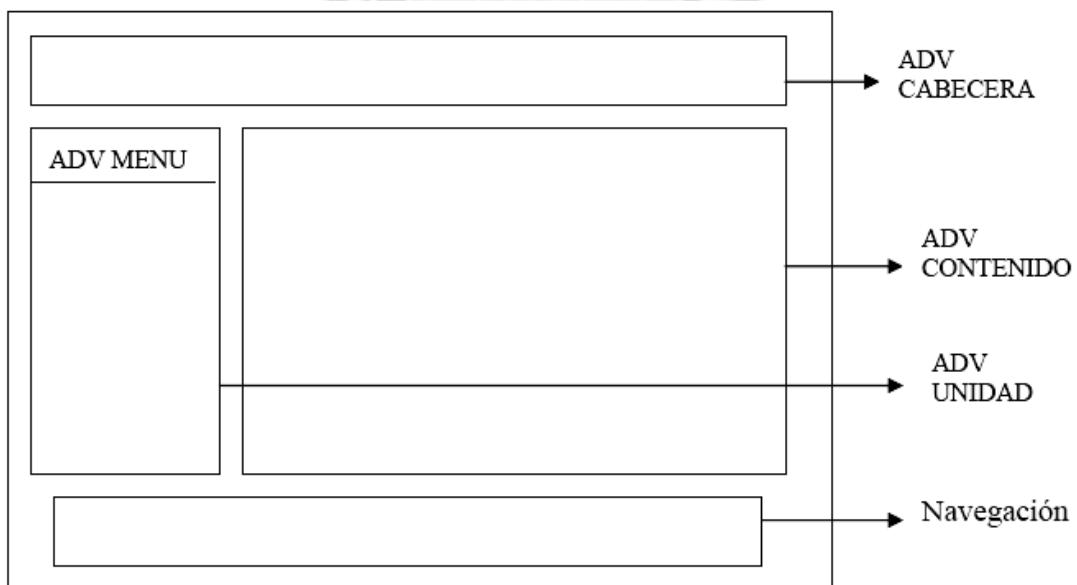


Figura 2.7: Abstract Data Views

Fuente: (Monasterio, 2012)

2.5.2.4 FASE DE IMPLEMENTACIÓN

Una vez obtenido el modelo conceptual, el modelo de navegación y el modelo de interfaz abstracta; en esta fase se toma en cuenta el entorno particular en el cual se va a correr la

aplicación. Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno Web, el diseñador debe decidir además qué información debe ser almacenada.

En los diagramas de clases navegacionales corresponden a vistas del esquema conceptual y los esquemas de contexto modelan el espacio de navegación incluyendo estructuras de acceso y contextos (que corresponde a un conjunto de instancias de una clase navegacional). Se podrían crear vistas parciales por usuario agrupando los contextos a partir de los tipos de usuarios que tienen acceso a los mismos. Las vistas por módulos o subsistemas no las modela de manera explícita, pero en los esquemas de contextos pueden modelarse fácilmente sub.-módulos.

Construir la interfaz de una aplicación Web también es una tarea compleja; no sólo se necesita especificar cuáles son los objetos de la interfaz que deberían ser implementados, sino también la manera en la cual estos objetos interactuarán con el resto de la aplicación. Esta metodología propone dedicar un tiempo importante en las fases previas a la implementación.

2.6 HERRAMIENTAS DE DESARROLLO

2.6.1 DEFINICIÓN DEL LENGUAJE DE PROGRAMACIÓN

2.6.1.1 ¿QUÉ ES PHP?

PHP (Hypertext Pre-Processor) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite, entre otras. La mayoría del código de PHP ha sido rescatado de C, Java y Perl, con unas cuantas características específicas propias. La meta de este lenguaje es permitirte escribir páginas generadas dinámicamente de forma rápida.

2.6.1.2 CARACTERÍSTICAS DE PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Biblioteca nativa de funciones sumamente amplia e incluida y permite las técnicas de programación Orientada a Objetos.
- No requiere definición de tipos de variables (Esta característica también podría considerarse una desventaja del lenguaje).
- Tiene manejo de excepciones (desde PHP5).

2.6.1.3 CÓMO FUNCIONA PHP

- Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos).
- El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.

2.6.2 HOJA DE ESTILO CSS

Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML, la idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación, esta información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML.

2.7 ARQUITECTURA DEL SISTEMA

La arquitectura del sistema está comprendido tanto en la descripción del diseño y contenido, incluye información sobre la organización fundamental del sistema, que incluye sus componentes, las relaciones entre si y el ambiente, y los principios que gobiernan su diseño y evolución.



CAPÍTULO III MARCO APLICATIVO

3.1 INTRODUCCIÓN

En este capítulo se empleará el método ágil de desarrollo XP, que por sus características se adapta extraordinariamente a esta propuesta, dicho método exige una evaluación de resultados y una retroalimentación de manera constante, lo que permite determinar si se está logrando los objetivos propuestos y se está cumpliendo con las expectativas finales del cliente.

- **FASE DE EXPLORACIÓN.** Se combinará las fases de los métodos XP y OOHDM, la fase de exploración del Método XP, se define las necesidades del cliente de forma rápida y sencilla en las “Historias de Usuario y sus Tareas”. En OOHDM se realizará la fase conceptual para la elaboración de Diagrama Entidad Relación, y la fase navegacional para los diagramas de clases navegacionales y contextos navegacionales.
- **FASE DE PLANIFICACIÓN.** Se utilizará la fase de planificación de XP para revisar el orden de las historias de usuario, el tiempo aproximado para cada historia de usuario., y subdividir las historias de usuario si el tiempo estimado es mayor a tres semanas.
- **FASE DE ITERACIONES Y PRUEBAS.** En OOHDM se utilizan vistas abstractas de datos (abstract data views, ADV) que representa la estructura estática de la interfaz, la composición de objetos y los eventos a los que responden. En XP se implementa las historias de usuario para hacer entregas de lo avanzado al cliente.
- **FASE DE MANTENIMIENTO.** Se realizará la metodología XP para las modificaciones o actualización de los requerimientos, dado que durante esta fase si se encuentra algún nuevo requerimiento que implementar o algo que modificar se comienza todo el ciclo nuevamente.

3.2 FASE DE EXPLORACIÓN

3.2.1 HISTORIAS DE USUARIO Y TAREAS

A continuación se detalla la Historia de Usuario: Registro de Usuarios, donde el administrador será capaz de crear, eliminar y modificar. (Ver Tabla 3.1).

HISTORIA DE USUARIO	
Número: 1	Usuario: ADMINISTRADOR
Nombre de Historia: Registro de usuarios	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Tiempo estimado: 17 días	Iteración asignada: 1
Descripción: El administrador será capaz de ver los usuarios registrados, además de adicionar nuevos usuarios, editar y eliminar, para que estos puedan acceder al sistema.	
Observaciones: Las interfaces deberán ser fáciles de comprender.	

Tabla 3.1: Historia de Usuario Administración de usuarios

Fuente: (Elaboración propia)

En esta tabla se realiza la siguiente Tarea: Diseñar la estructura de datos para administrar usuarios. (Ver Tabla 3.2.)

TAREA	
Número: 1.1	Número de Historia: 1
Nombre de Tarea: Diseñar la estructura de datos para administrar usuarios	
Tipo de Tarea: Diseño	Tiempo estimado: 3 días
Fecha Inicio: 9-feb-2015	Fecha Final: 11-feb-2015
Responsable: Danny Almanza Amaru	
Descripción: Creación de la estructura de la Base de datos para dar altas, bajas y modificaciones a usuarios	

Tabla 3.2: Tarea Diseñar la estructura de datos para administrar usuarios

Fuente: (Elaboración propia)

Realizar la tarea de crear interfaz de usuario listar usuarios registrados, mostrando detalles del usuario y la categoría asignada. (Ver Tabla 3.3)

TAREA	
Número: 1.2	Número de Historia: 1
Nombre de Tarea: Crear interfaz de Usuarios	
Tipo de Tarea: Desarrollo	Tiempo estimado: 3 días
Fecha Inicio: 12-feb-2015	Fecha Final: 14-feb-2015
Responsable: Danny Almanza Amaru	
Descripción: Mediante la interfaz se podrán listar los usuarios, además estarán incluidos dos botones para cada usuario, botón de “eliminar” el cual borrará el registro del usuario y el botón de “editar” el cual mandará a otro interface donde se podrá editar al usuario.	

Tabla 3.3: Tarea Crear interfaz Usuarios

Fuente: (Elaboración propia)

Realizar la tarea para crear el interfaz de nuevos usuarios, el usuario administrador podrá crear nuevos usuarios. (Ver Tabla 3.4)

TAREA	
Número: 1.3	Número de Historia: 1
Nombre de Tarea: Crear interfaz para ingresar nuevos usuarios	
Tipo de Tarea: Desarrollo	Tiempo estimado: 5 días
Fecha Inicio: 15-feb-2015	Fecha Final: 19-feb-2015
Responsable: Danny Almanza Amaru	
Descripción: Los usuarios registrados podrán acceder al sistema tan solo con ingresar el Login y el Password	

Tabla 3.4: Tarea Crear interfaz para registrar nuevos usuarios

Fuente: (Elaboración propia)

Crear la Tarea para crear la interfaz para modificar usuarios, el usuario administrador podrá realizar modificaciones de un usuario. (Ver tabla 3.5)

TAREA	
Número: 1.4	Número de Historia: 1
Nombre de Tarea: Crear interfaz para modificar usuarios	
Tipo de Tarea: Desarrollo	Tiempo estimado: 3 días
Fecha Inicio: 20-feb-2015	Fecha Final: 22-feb-2015
Responsable: Danny Almanza Amaru	
Descripción: Los usuarios registrados podrán acceder al sistema tan solo con ingresar el Login y el Password	

Tabla 3.5: Tarea Crear interfaz para modificar usuarios

Fuente: (Elaboración propia)

Realizar la tarea de crear interfaz de acceso al sistema con los campos de login y password, y poder realizar actividades dentro de ellas. Si los datos son incorrectos enviar un mensaje de error. (Ver Tabla 3.6)

TAREA	
Número: 1.5	Número de Historia: 1
Nombre de Tarea: Crear interfaz de acceso al sistema	
Tipo de Tarea: Diseño	Tiempo estimado: 3 días
Fecha Inicio: 23-feb-2015	Fecha Final: 25-feb-2015
Responsable: Danny Almanza Amaru	
Descripción: Los usuarios registrados podrán acceder tan solo con ingresar el Login y el Password, para ello se deberán manejar sesiones	

Tabla 3.6: Tarea Crear interfaz de acceso al sistema

Fuente: (Elaboración propia)

Realizar la Historia de Usuario: Registro de Tesis, podrán realizar esta actividad los usuarios registrados. (Ver Tabla 3.7)

HISTORIA DE USUARIO	
Número: 2	Usuario: ADMINISTRADOR, PRIVADO
Nombre de Historia: Registro de Tesis	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Tiempo estimado: 21 días	Iteración asignada: 1
<p>Descripción:</p> <p>Los usuarios registrados podrán ser capaces de registrar las Tesis que son recibidas por los alumnos de las diferentes carreras que cuenta la Universidad Mayor de San Andrés.</p>	
Observaciones: Solamente tendrán acceso al sistema los usuarios registrados.	

Tabla 3.7: Historia de Usuario Registro de Tesis

Fuente: (Elaboración propia)

Realizar la estructura de los datos para realizar el registro de las tesis, estos atributos se obtendrán al ingresar como usuario registrado. (Ver Figura 3.8)

TAREA	
Número: 2.1	Número de Historia: 2
Nombre de Tarea: Diseñar la estructura de datos para registrar la Tesis	
Tipo de Tarea: Diseño	Tiempo estimado: 3 días
Fecha Inicio: 28-feb-2015	Fecha Final: 2-mar-2015
Responsable: Danny Almanza Amaru	
<p>Descripción:</p> <p>Los atributos de la estructura se obtendrán al ingresar como usuario registrado</p>	

Tabla 3.8: Tarea Diseñar la estructura de datos para registrar la Tesis

Fuente: (Elaboración propia)

Realizar la tarea, crear interfaz para registrar la nueva Tesis, ingresando el llenado con la información requerida. (Ver Figura 3.9)

TAREA	
Número: 2.2	Número de Historia: 2
Nombre de Tarea: Crear interfaz para registrar la nueva Tesis	
Tipo de Tarea: Desarrollo	Tiempo estimado: 9 días
Fecha Inicio: 3-mar-2015	Fecha Final: 11-mar-2015
Responsable: Danny Almanza Amaru	
Descripción: La interfaz para llenar la Tesis para los usuarios registrados	

Tabla 3.9: Tarea Crear interfaz para registrar la nueva Tesis

Fuente: (Elaboración propia)

Realizar la tarea, crear interfaz para editar los registros de las Tesis, para modificar la información necesaria. (Ver Figura 3.10)

TAREA	
Número: 2.3	Número de Historia: 2
Nombre de Tarea: Crear interfaz para editar los registros de las Tesis	
Tipo de Tarea: Desarrollo	Tiempo estimado: 4 días
Fecha Inicio: 12-mar-2015	Fecha Final: 15-mar-2015
Responsable: Danny Almanza Amaru	
Descripción: La interfaz para editar la Tesis para los usuarios registrados	

Tabla 3.10: Tarea Crear interfaz para editar los registros de las Tesis

Fuente: (Elaboración propia)

Realizar la tarea, crear interfaz para visualizar la Tesis para imprimir, donde se mostrarán los datos al interesado. (Ver Figura 3.11)

TAREA	
Número: 2.4	Número de Historia: 2
Nombre de Tarea: Crear interfaz para visualizar la Tesis para imprimir	
Tipo de Tarea: Desarrollo	Tiempo estimado: 3 días
Fecha Inicio: 16-mar-2015	Fecha Final: 18-mar-2015
Responsable: Danny Almanza Amaru	
Descripción:	
Esta interfaz muestra los datos de las Tesis para imprimir	

Tabla 3.11: Tarea Crear interfaz para visualizar la Tesis para imprimir

Fuente: (Elaboración propia)

Realizar la Historia de Usuario: Búsqueda de Tesis, donde se podrá buscar las Tesis a determinadas características. (Ver Figura 3.12)

HISTORIA DE USUARIO	
Número: 3	Usuario: PRIVADO, PÚBLICO
Nombre de Historia: Búsqueda de Tesis	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Tiempo estimado: 10 días	Iteración asignada: 1
Descripción:	
Los clientes podrán buscar las tesis de acuerdo a determinadas características, la búsqueda será inmediata	
Observaciones:	
Solamente tendrán acceso al sistema los usuarios registrados.	

Tabla 3.12: Historia de Usuario Búsqueda de Tesis

Fuente: (Elaboración propia)

Realizar la tarea, crear interfaces para filtrar las Tesis, donde se podrá buscar las tesis por características definidas al usuario. (Ver Figura 3.13)

TAREA	
Número: 3.1	Número de Historia: 3
Nombre de Tarea: Crear interfaces para Filtrar las Tesis	
Tipo de Tarea: Desarrollo	Tiempo estimado: 10 días
Fecha Inicio: 18-mar-2015	Fecha Final: 27-mar-2015
Responsable: Danny Almanza Amaru	
Descripción: La filtración de tesis según su atributo específico, por código, por autor, por título de la tesis	

Tabla 3.13: Tarea Crear interfaces para Filtrar las Tesis

Fuente: (Elaboración propia)

Realizar la Historia de Usuario: Reportes de Tesis, aquí se podrá realizar el exportado de las Tesis a archivo Excel. (Ver Figura 3.14)

HISTORIA DE USUARIO	
Número: 4	Usuario: PRIVADO
Nombre de Historia: Reportes de Tesis	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Tiempo estimado: 21 días	Iteración asignada: 1
Descripción: Los reportes son guardados en archivos Excel para guardar físicamente las Tesis registradas y guardarlas por año.	
Observaciones: Solicitudada por el cliente	

Tabla 3.14: Historia de Usuario Reportes de Tesis

Fuente: (Elaboración propia)

Realizar la tarea, crear interfaces para reportes de las Tesis, donde se harán por Facultad, Año, y por Mes. (Ver Tabla 3.15)

TAREA	
Número: 4.1	Número de Historia: 4
Nombre de Tarea: Crear interfaces para los reportes de las Tesis	
Tipo de Tarea: Desarrollo	Tiempo estimado: 21 días
Fecha Inicio: 27-mar-2015	Fecha Final: 16-abr-2015
Responsable: Danny Almanza Amaru	
Descripción: Los reportes emitidos por el sistema por Facultad, Año, Mes.	

Tabla 3.15: Tarea Crear interfaces para los reportes de las Tesis

Fuente: (Elaboración propia)

3.2.1.1 DETALLE DE HISTORIAS DE USUARIOS

A continuación se describen de manera no detallada cada historia de usuario con sus tareas respectivas.

Módulo de Administración

- Historia de usuario 1:Administración de Usuarios
 - Tarea 1.1: Diseñar la estructura de datos para administrar usuarios
 - Tarea 1.2: Crear interfaz de usuarios
 - Tarea 1.3: Crear interfaz para registrar nuevos usuarios
 - Tarea 1.4: Crear interfaz para modificar usuarios
 - Tarea 1.5: Crear interfaz de acceso al sistema

Módulo de Tesis

- Historia de usuario 2:Registro de Tesis
 - Tarea 2.1: Diseñar la estructura de datos para registrar la Tesis

- Tarea 2.2: Crear interfaz para registrar la nueva Tesis
- Tarea 2.3: Crear interfaz para editar los registros de las Tesis
- Tarea 2.4: Crear interfaz para visualizar la Tesis para imprimir

Módulo de Búsqueda

- Historia de usuario 3: Búsqueda de Tesis
 - Tarea 3.1: Crear interfaces para Filtrar las Tesis

Módulo de Reportes

- Historia de usuario 4: Reportes de Tesis
 - Tarea 4.1: Crear interfaces para los reportes de las Tesis

3.2.2 DIAGRAMA ENTIDAD RELACIÓN

Se realiza el diagrama de Entidad Relación teniendo en cuenta las Historias de Usuarios hechas anteriormente. (Ver Figura 3.1)

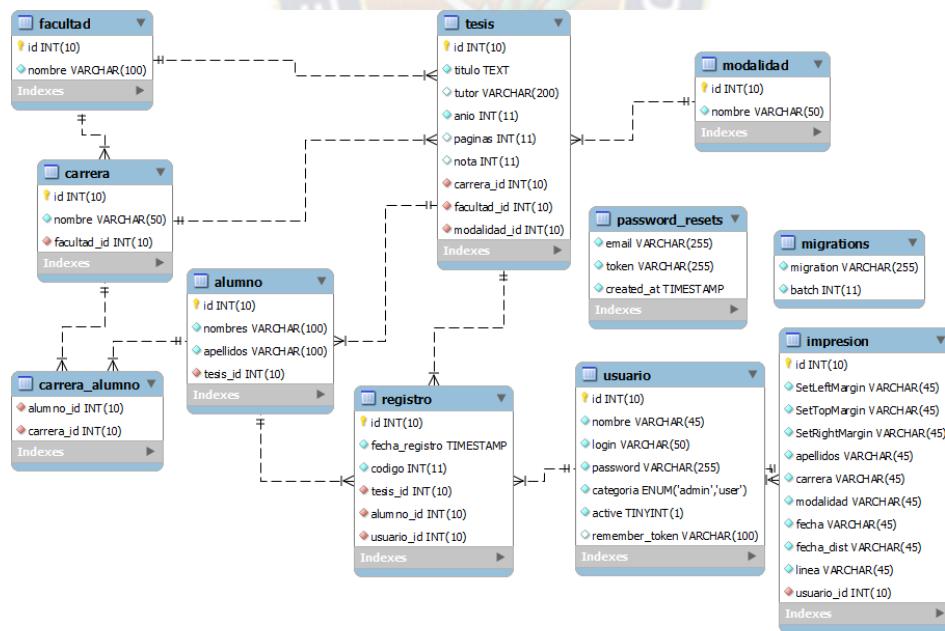


Figura 3.1: Diagrama Entidad Relación
Fuente: (Elaboración propia)

3.2.2.1 DIAGRAMA DE CLASES NAVEGACIONALES

En esta fase se realiza el modelado de la estructura del sistema, se captura los conceptos involucrados en el dominio de la aplicación y se los describe en detalle haciendo uso de diagramas que permitan expresar con claridad el comportamiento, la estructura de los atributos y las relaciones entre conceptos.

Para ello se utiliza el “Diagrama de Clases” para describir la estructura de la Base de datos.
(Ver Figura 3.2)

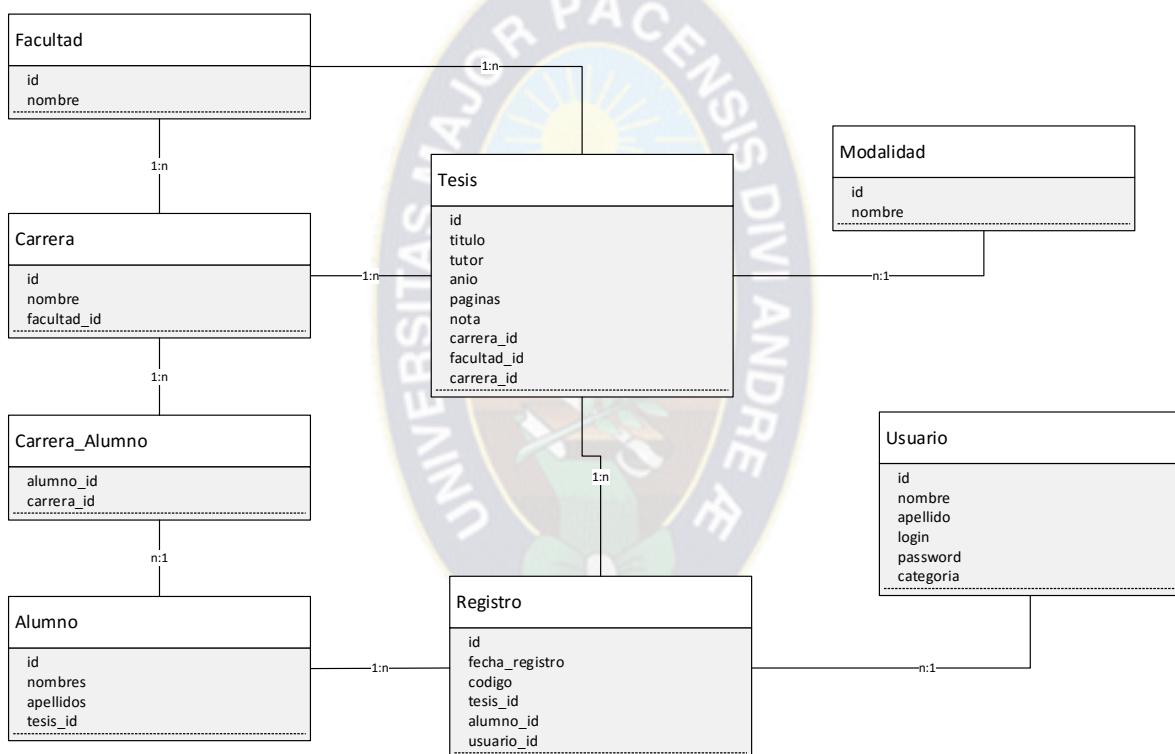


Figura 3.2: Diagrama de clases Navegacionales
Fuente: (Elaboración propia)

3.2.2.2 DIAGRAMA DE CONTEXTOS NAVEGACIONALES

En esta fase se realiza el diagrama de contexto navegacional, Una de las mayores innovaciones de OOHDM es que el usuario no navega directamente dentro de las entidades conceptuales, sino en los Nodos (objetos navegacionales) que son construidos desde uno o

varias entidades conceptuales. Entonces los atributos de los nodos se obtienen a partir de los atributos del modelo conceptual. (Ver Figura 3.3)

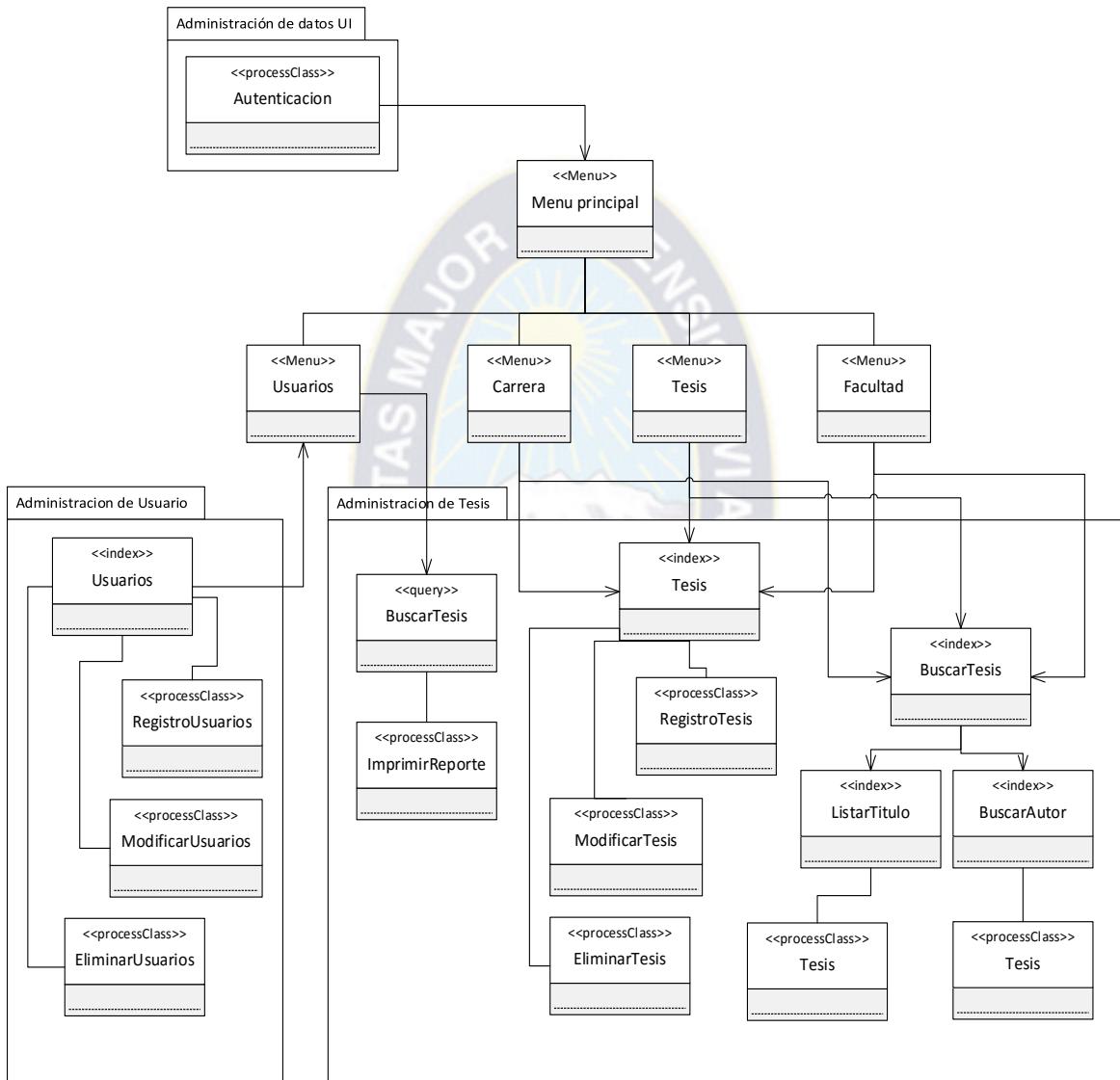


Figura 3.3: Diagrama Contextual Navegacional

Fuente: (Elaboración propia)

3.3 FASE DE PLANIFICACIÓN

La cantidad y el orden de las iteraciones de las historias de usuario así como las estimaciones de esfuerzo y la planificación por historias, se detallan a continuación, de las cuales algunas de ellas han ido modificándose en el transcurso del desarrollo del proyecto.

3.3.1 ESTIMADORES DE ESFUERZO

En la Tabla 3.16 se detallará los estimadores de esfuerzo de las Historias de Usuario realizado por días.

Historias de Usuario	Duración (días)
Administración de Usuarios	17
Registro de Tesis	21
Búsqueda de Tesis	10
Reportes de Tesis	21

Tabla 3.16: Estimaciones de Esfuerzo
Fuente: (Elaboración propia)

3.3.2 PLANIFICACIÓN POR HISTORIAS

En la Tabla 3.17 se detalla la elaboración de las Historias de Usuario con la fecha de inicio y la fecha de culminación de la misma.

Iteraciones	HISTORIA	Nº	Inicio	Fin	Prioridad
1	Administración de Usuarios	1	9-feb-2015	25-feb-2015	Media
2	Registro de Tesis	2	28-feb-2015	18-mar-2015	Alta
3	Búsqueda de Tesis	3	18-mar-2015	27-mar-2015	Media

4	Reportes de Tesis	4	27-mar- 2015	16-abr-2015	Alta
5	Reportes Estadísticos	5	17-abr-2015	10-may- 2015	Media

Tabla 3.17: Planificaciones por historia

Fuente: (Elaboración propia)

3.3.3 CRONOGRAMA DE DESARROLLO

Para ver el cronograma de desarrollo véase ANEXO E - CRONOGRAMA DE DESARROLLO

3.4 FASE DE ITERACIONES Y PRUEBAS

3.4.1 PRIMERA ITERACIÓN

Se realizará el Diseño de la Vista Abstracta ADV, para la navegación del sistema y cumplir con las historias de usuario realizadas en la anterior fase. (Ver Figura 3.4)

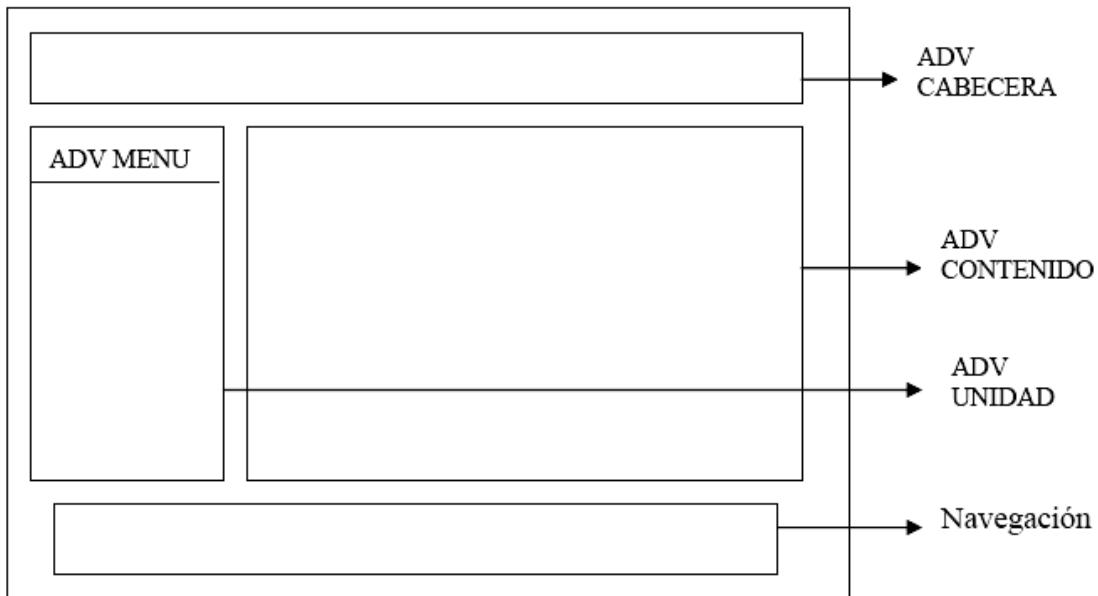


Figura 3.4. Abstract Data Views

Fuente: (Elaboración propia)

HISTORIAS DE USUARIO 1: ADMINISTRACIÓN DE USUARIOS

Tarea 1.1 Crear interfaz Perfil de Usuarios. En la figura se detalla la lista de usuarios registrados para la modificación o eliminación. (Ver Figura 3.5)

USUARIOS

NOMBRE	APELLIDOS	USUARIO	CATEGORIA	
Danny	Danny	Danny	admin	 
Delia	Delia	Delia	private	 
patty	patty	patty	private	 
Rocio	Rocio	Rocio	private	 

[Añadir](#)

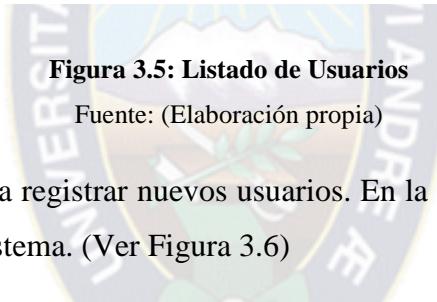


Figura 3.5: Listado de Usuarios

Fuente: (Elaboración propia)

Tarea 1.2 Crear interfaz para registrar nuevos usuarios. En la figura se realiza la vista crear usuario para el ingreso al sistema. (Ver Figura 3.6)

Crear Usuario

Nombre	<input type="text" value="Escriba el nombre"/>
Apellido	<input type="text" value="Escriba el apellido"/>
Usuario	<input type="text" value="Escriba el usuario"/>
Categoría	<input type="text" value="Privado"/> ▼

[Crear Usuario](#)

Figura 3.6. Formulario de llenado de usuarios

Fuente: (Elaboración propia)

Tarea 1.3 Crear interfaz para modificar usuarios. En la figura se realizara la vista para modificar el usuario, mostrando los datos anteriormente. (Ver Figura 3.7).

Actualizar Usuario

Nombre	Delia
Apellido	Cazas
Usuario	Delia
Categoría	Privado

Actualizar Usuario

Figura 3.7. Formulario de actualización de usuarios

Fuente: (Elaboración propia)

Tarea 1.4 Crear interfaz de ingreso al sistema. En la figura se muestra el diseño para el ingreso de usuario y password para el ingreso del sistema. (Ver Figura 3.8).

Usuario
Introduzca usuario
Password
Password
Ingresar

Figura 3.8. Formulario de ingreso al Sistema

Fuente: (Elaboración propia)

PRUEBAS DE ACEPTACIÓN DE LAS HISTORIAS DE USUARIO 1

a) Identificar todos los posibles resultados observables de la historia.

- Interfaz de Perfil de usuarios con los botones “editar”, “eliminar”, “crear usuario”.

- Formulario de Registro de Nuevos Usuarios.
- Formulario para modificar Usuarios.
- Interfaz de Ingreso al Sistema.

b) Identificar los resultados que terminan la historia y los que permiten continuar dentro de la historia.

La historia de usuario concluye:

- Al salir de la interfaz de Usuarios, o al oprimir eliminar, editar o crear a un usuario.
- Al llenar el formulario de Registro de Nuevos usuarios.
- Al editar el formulario de Registro de Usuario.
- Al introducir nombre de usuario y contraseña para acceder a Sistema.

c) Identificar los caminos de ejecución posible.

- Cuando el Administrador decide ingresar un nuevo usuario, el sistema valida los datos y los envía a la base de datos.
- Cuando el Administrador decide modifica un usuario, el sistema valida los datos y los envía a la base de datos.
- Cuando el Administrador decide eliminar usuario, el sistema valida los datos y los envía a la base de datos.

d) Asignar un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- El conjunto de valores válidos está dado por el conjunto de datos requeridos por el formulario en caso de ingresar o modificar usuarios.

e) Eliminación de caminos redundantes.

- No existe caminos redundantes.

3.4.2 SEGUNDA ITERACIÓN

HISTORIA DE USUARIO 2: REGISTRO DE TESIS

Tarea 2.2: Crear interfaz para registrar la nueva Tesis. En la figura se realiza el registro de las tesis con los campos requeridos. (Ver Figura 3.9).

Crear Registro

Código	
Apellidos	
Nombre(s)	
Título	
Tutor	
Facultad	Seleccione opcion...
Carrera	Seleccione opcion...
Año	
Páginas	
Valoración	
Modalidad	Seleccione opcion...

Crear Registro

Figura 3.9. Formulario de registro de Tesis

Fuente: (Elaboración propia)

Tarea 2.3: Crear interfaz para editar los registros de las Tesis. En la figura se detalla los campos de actualización, mostrando todos los datos guardados anteriormente para modificar o agregar. (Ver Figura 3.10).

Actualizar Registro

Código	4889
Apellidos	Landaeta Tudela
Nombre(s)	José Antonio Gonzalo
Título	Eestudio de factibilidad para la implementación de radiodifusión digital FM con estándar DRM (digital radio mondiale) en Bolivia
Tutor	Machicao, Juan Carlos
Facultad	Ingeniería
Carrera	Ingeniería Electrónica
Año	2014
Páginas	152
Valoración	95
Modalidad	Proyecto de Grado

Actualizar Registro

Figura 3.10. Formulario de actualización de Tesis

Fuente: (Elaboración propia)

Tarea 2.4: Crear interfaz para visualizar la Tesis para imprimir. En la figura siguiente se ve el reporte en PDF antes de imprimir. (Ver Figura 3.11)

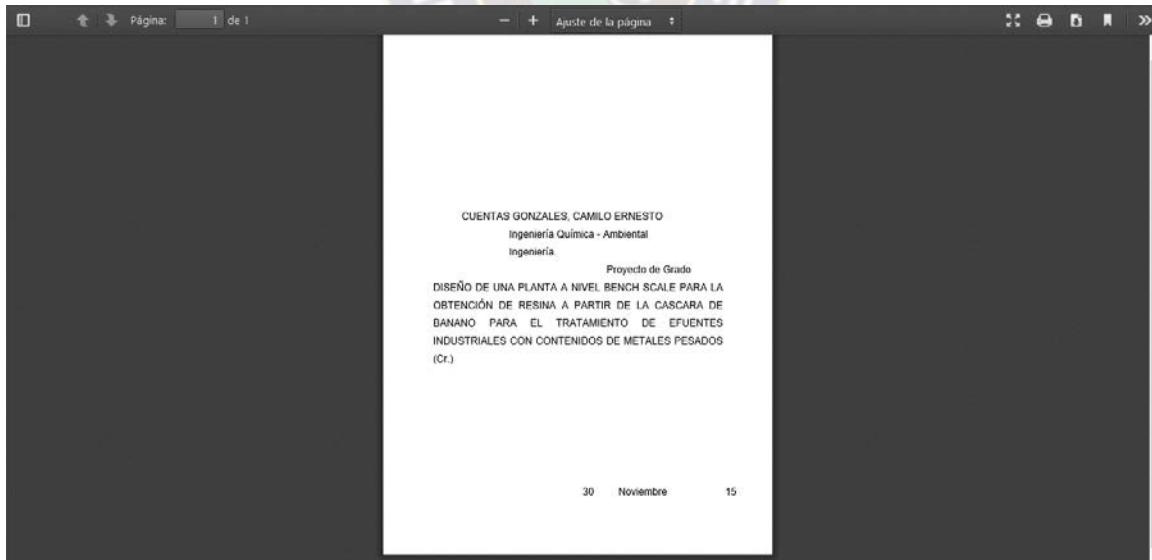


Figura 3.11. Reporte en PDF para la impresión

[Elaboración propia]

Este reporte esta realizado para imprimir en los certificados que cuenta la Biblioteca Central para la entrega a los interesados. Ver ANEXO F – CERTIFICADO DE TESIS

PRUEBAS DE ACEPTACIÓN DE LAS HISTORIAS DE USUARIO 2

a) Identificar todos los posibles resultados observables de la historia.

- Formulario de Registro de Nuevas Tesis
- Formulario para modificar Tesis
- Interfaz de visualizar las tesis registradas en el sistema al Sistema.

b) Identificar los resultados que terminan la historia y los que permiten continuar dentro de la historia.

La historia de usuario concluye:

- Al llenar el formulario de Registro de Tesis.
- Al editar el formulario de Registro de Usuario.

c) Identificar los caminos de ejecución posible.

- Cuando el usuario decide registrar el nuevo tesis, el sistema valida los datos y los envía a la base de datos.
- Cuando el usuario decide modifica una tesis, el sistema valida los datos y los envía a la base de datos.
- Cuando el usuario decide visualizar una tesis, el sistema valida los datos y los envía a la base de datos.

d) Asignar un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

- a. El conjunto de valores válidos está dado por el conjunto de datos requeridos por el formulario en caso de registrar y visualizar las tesis.

e) Eliminación de caminos redundantes.

- a. No existe caminos redundantes.

3.4.3 TERCERA ITERACIÓN

HISTORIA DE USUARIO 3: BÚSQUEDA DE TESIS

Tarea 3.1: Crear interfaces para Filtrar las Tesis. En la figura siguiente se detalla la vista de búsqueda de Tesis. (Ver Figura 3.12)

The screenshot shows a search interface with a blue header bar containing three tabs: 'CÓDIGO' (highlighted in orange), 'TÍTULO' (in light blue), and 'AUTOR' (in white). Below the header is a section labeled 'Código' with a text input field containing the placeholder 'Introducir código'. At the bottom is a green button labeled 'Búsqueda'.

Figura 3.12. Formulario de Búsqueda

Fuente: (Elaboración propia)

En la Figura 3.13 se detalla el resultado de la búsqueda por código, el código es el número del certificado entregado a los interesados.

The screenshot shows a search results table. The top part is identical to Figure 3.12, with the 'CÓDIGO' tab selected. The results table has columns: CODIGO, AUTOR, TÍTULO, CARRERA, and AÑO. One row is shown, corresponding to the search term '1111'. The row details are: CODIGO 1111, AUTOR Rivero Marquez, Rolan Franklin Celso, TÍTULO El Crecimiento Urbano De Una Ciudad Y Sus Consecuencias Para Su Aeropuerto: Caso Del Aeropuerto Jorge Wilsterman De Cochabamba, CARRERA Ingeniería Civil, and AÑO 2013. To the right of the row are three icons: a blue square with a white minus sign, an orange square with a white pencil, and a green square with a white plus sign.

CODIGO	AUTOR	TÍTULO	CARRERA	AÑO	
1111	Rivero Marquez, Rolan Franklin Celso	El Crecimiento Urbano De Una Ciudad Y Sus Consecuencias Para Su Aeropuerto: Caso Del Aeropuerto Jorge Wilsterman De Cochabamba	Ingeniería Civil	2013	

Figura 3.13. Formulario de Búsqueda por Código

Fuente: (Elaboración propia)

En la Figura 3.14 se devuelven los resultados de las tesis buscadas por título, se puede buscar por parte del título para su filtrado.

CÓDIGO	AUTOR	TÍTULO	CARRERA	AÑO	
410	Uzeda Orellana, Geila Volga	¿De obreros a empresarios o de empresarios a obreros?	Sociología	2013	

Figura 3.14. Formulario de Búsqueda por Título

Fuente: (Elaboración propia)

En la Figura 3.15 se realiza la búsqueda de las tesis por el campo autor, se puede buscar por apellidos o por los nombres.

CÓDIGO	AUTOR	TÍTULO	CARRERA	AÑO	
4296	Miranda Almanza, Raquel Síntique	Efectos Psicosociales De Las Personas Que Viven Con Cáncer En El Hospital Militar Central De La Ciudad De La Paz En El Primer Semestre De La Gestión 2013	Postgrado	2014	
87	Miranda Almanza, Soledad Raquel	Museo paleontológico	Arquitectura	2004	

Figura 3.15. Formulario de Búsqueda por Autor

Fuente: (Elaboración propia)

PRUEBAS DE ACEPTACIÓN DE LAS HISTORIAS DE USUARIO 3

- a) Identificar todos los posibles resultados observables de la historia.**
 - Interfaz del menú de búsqueda.
 - Interfaz de resultados (por código)
 - Interfaz de resultados (por título de tesis)
 - Interfaz de resultados (por autor de Tesis)
- b) Identificar los resultados que terminan la historia y los que permiten continuar dentro de la historia.**

La historia de usuario concluye:

 - Al salir de la Interfaz de menú de búsqueda.
- c) Identificar los caminos de ejecución posible.**
 - Cuando el usuario busca las tesis por alguno de los filtros el sistema despliega los resultados, es decir, lista las tesis encontradas.
- d) Asignar un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.**
 - a. El conjunto de valores válidos está dado por el conjunto de datos se las realiza de acuerdo al tipo de búsqueda que elige el usuario.
- e) Eliminación de caminos redundantes.**
 - a. No existe caminos redundantes.

3.4.4 CUARTA ITERACIÓN

HISTORIA DE USUARIO 4: REPORTES DE TESIS

Tarea 4.1: Crear interfaces para los reportes de las Tesis. En la figura se detalla la vista de reporte para exportar a Excel según los capos seleccionados, Facultad, Año, Mes. (Ver Figura 3.16)

FACULTAD	AÑO	MES
Facultad	Agronomía	
Gráficos	Año	
	Barras	
		Reportes
		Gráficos

Figura 3.16. Formulario de Reportes

Fuente: (Elaboración propia)

En la Figura 3.17 se realiza el reporte a Excel por Facultad de todas las tesis ingresadas en el sistema.

TESIS BIBLIOTECA CENTRAL INFORMATICA						
NRO	AUTOR	TITULO	TUTOR	FACULTAD	CARRERA	AÑO DEFENSA
4	Achá Huanca, María Eugenia	Modelo De Identificación Del Patrimonio Cultural Y Turístico M	Dolz De Moreno, Fátima Consuelo	Cs. Puras y Naturales	Informática	2013
5	Aduvin Yuja, Ruben	Registro Y Control De Inscripciones Para El Área De Patrocin	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013
6	Aguilar Anílles, Juan Carlos	Semillas de encriptación para la seguridad y confiabilidad de	Rodríguez Ramírez, Grover A.	Cs. Puras y Naturales	Informática	2013
7	Aguilar Quisbert, María Teresa	Modelo de Sistema Distribuido para su Aplicación en Domótic	Lic. Jorge Humberto Terán Pomier	Cs. Puras y Naturales	Informática	2013
8	Aguilar Quispe, José Antonio	Web servicios en comprobante de pago electrónico para peq	Cuevas Quiroz, Franz	Cs. Puras y Naturales	Informática	2013
9	Aguilera Ignacio, Héctor Vladimir	Un método expandido de ingeniería WEB para el desarrollo d	Reyes Pacheco, Javier	Cs. Puras y Naturales	Informática	2013
10	Aguirre Heredia, Claudia Patricia	Análisis de condiciones de la red eléctrica para el envío de da	Toledo Paz, Miguel	Cs. Puras y Naturales	Informática	2013
11	Alanoca Bautista, José Bernardo	Sistema Inteligente Para Determinar El Estado Nutricional En	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013
12	Albaracín Vallejos, Rodrigo Mario	Sistema de información georeferencial de líneas de trasporte	Flores Morales, Rosa	Cs. Puras y Naturales	Informática	2013
13	Alberto Mamani, Freddy Waldo	Predicción Del Consumo De Gas Natural Aplicando Redes Ne	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013
14	Alvarezo Alfaro, Mireya Russena	Tutor Inteligente Para El Aprendizaje De La Lectura De Partit	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013
15	Alvarado Mallo, Silvia Laura	Modelo de simulación de la erosión del suelo basado en fract	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013
16	Aparicio Manzaneda, Nelson	Minería de datos para determinar los niveles de seguridad y p	Rodríguez Ramírez, Grover	Cs. Puras y Naturales	Informática	2013
17	Aramburu Dávalos, Laura Daniela	Sistema Experto Para El Diagnóstico De La Enfermedad Cordotá	M. Sc. Fátima Consuelo Dolz De Moreno	Cs. Puras y Naturales	Informática	2013
18	Arana Paz, Deybis Willam	Sistema experto para el diagnóstico de fallas en motores a in	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013
19	Arequipa Coloma, Wilmer	Sistema tutor inteligente para la enseñanza e información de	Cuevas Quiroz, Franz	Cs. Puras y Naturales	Informática	2013
20	Athauachi Magne, Verónica Karla	Cuantificación Y Deteccción De Alteraciones Morfológicas De	Cotafá Mier, Miguel	Cs. Puras y Naturales	Informática	2013
21	Bautista Cabrera, Jaime Abraham	Tutor inteligente para el aprendizaje de la matemática en prim	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013
22	Bernal Escobar, Yola	Asistente Virtual Interactivo En Entornos Virtuales De Enseñ	Lic. Grover Alex Rodríguez	Cs. Puras y Naturales	Informática	2013
23	Blanco Apaza, Elena Rut	Cambio De Tecnología ATM a ETHERNET en Servicios ADSL	M. Sc. Fátima Consuelo Dolz De Moreno	Cs. Puras y Naturales	Informática	2013
24	Brañez Serrano, Nilton Abel	Realidad Aumentada En Aplicaciones Móviles Para La Seguri	Flores Morales, Rosa	Cs. Puras y Naturales	Informática	2013
25	Brida Correa, Juan Víctor	Tesis de grado titulada de reconstrucción y evaluación de amonita	Valdez Alvarado, Aldo Ramiro	Cs. Puras y Naturales	Informática	2013

Figura 3.17. Reporte Excel por Facultad

Fuente: (Elaboración propia)

En la Figura 3.18 se realiza el reporte a Excel por Año, mostrando la cantidad de tesis ingresadas por mes de las diferentes carreras.

	FAC/CAR	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	NOV	DIC	TOTAL
1	AGRONOMIA													
4	INGENIERIA AGRONOMICA				13	3	3	16	7	3	18		1	63
5	POSTGRADO													
6	TECNICA SUPERIOR AGROPECUARIA DE VIACHA										1			1
7	ARQUITECTURA, ARTES, DISEÑO Y URBANISMO													
8	ARQUITECTURA				21	1	2	12	11	5	7	1		59
9	ARTES									1				2
10	POSTGRADO													
11	CS. ECONOMICAS Y FINANCIERAS													
12	ADMINISTRACION DE EMPRESAS				20	1	10	9	7	20	17	2		84
13	CONTADURIA PUBLICA				15		3	11	13	8	8	3		58
14	ECONOMIA				6	1	3	3	16	11	4	1		44
15	POSTGRADO								1					1
16	CS. FARMACEUTICAS Y BIOQUIMICAS													
17	BIOQUIMICA				3			1			2			6
18	POSTGRADO						1	1	2	1	1			7
19	QUIMICA FARMACEUTICA													
20	CS. GEOLGICAS													
21	INGENIERIA GEOGRAFICA				6	1	1	1		1		2		10
22	INGENIERIA GEOLGICA						1	1	2	2	1			7
23	POSTGRADO													
24	CS. PURAS Y NATURALES													

Figura 3.18. Reporte Excel por Año

Fuente: (Elaboración propia)

En la Figura 3.19 se realiza el reporte a Excel por mes de las tesis ingresadas, para el guardado en archivo físico.

Nro	CODIGO	AUTOR	TITULO	CARRERA	FACULTAD	ANO	FECHA
1	93	Adrian Rojas, Anelin Nirvana	optimización de los métodos de limpieza CIP	Química Industrial	Tecnología	2013	2013-01-30
4	2	264 Alegria Meneses, Zuely Betty	Metodologías de control para la Supervisión	Ingierencia Civil	Ingierencia	2013	2013-01-31
5	3	263 Alvarado Villareal, Christian R	Terminological database for improving the trilingualism e idiomas	Humanidades y Cs de la Edu	Humanidades y Cs de la Edu	2013	2013-01-31
6	4	740 Apaza Vargas, Iver	Modificación al art 51 del reglamento de prod Derecho	Derecho y Cs. Políticas	Derecho y Cs. Políticas	2013	2013-01-27
7	5	258 Aranda Asturizaga, Juana Milen	La satisfacción laboral y la motivación de logr Psicología	Humanidades y Cs de la Edu	Humanidades y Cs de la Edu	2013	2013-01-31
8	6	209 Arenas Vásquez, Bárbara Guisela	La limitada educación de emisión de facturas	Contaduría Pública	Cs. Económicas y Financieras	2013	2013-01-30
9	7	80 Arguedas Ortuño, Claudio Patricio	Recuperación de plata a partir del reciclado d	Ingierencia Química - Alimen	Ingierencia	2013	2013-01-30
10	8	743 Arroyo Jiménez, Javier Marcelo	Indicadores d ejecución de los recursos del	Administración de Empresas	Cs. Económicas y Financieras	2013	2013-01-27
11	9	212 Astorga Rada, Leidy Susana	Demandas potenciales para el turismo social de	Turismo	Humanidades y Cs de la Edu	2013	2013-01-31
12	10	79 Avendaño Rodríguez, Jesus Chir	Diseño e implementación (original) de manu	Contaduría Pública	Cs. Económicas y Financieras	2013	2013-01-30
13	11	259 Avilés Hilari, Hernan Gregorio	Diseño de instalaciones electromecánicas en	Ingierencia Electromecánica	Ingierencia	2013	2013-01-31
14	12	81 Ayala Ticona, Carlos	Diversificación de la línea de fibra de carbono	Ingierencia Industrial	Ingierencia	2013	2013-01-30
15	13	735 Baquedano Antequera, Ruddy Renato	Diseño normalizado de un que estacionamiento	Ingierencia Mecánica	Ingierencia	2013	2013-01-27
16	14	240 Barrios Pachón, Fabiola Moreno	el de un sistema de estacionamiento automático	Derecho y Cs. Políticas	Derecho y Cs. Políticas	2013	2013-01-31
17	15	273 Borda Tapia, María Carolina	Propuesta institucional para la creación d	Derecho y Cs. Políticas	Derecho y Cs. Políticas	2013	2013-01-31
18	16	320 Castillo Callaquejus, Horacio	Plan de negocios implementación de inversi	Administración de Empresas	Cs. Económicas y Financieras	2013	2013-01-29
19	17	185 Celis Yola, Mayta Humerez	La intervención del trabajo social en program	Trabajo Social	Cs. Sociales	2013	2013-01-30
20	18	82 Chamí Canqui, Romer Wilfredo	Optimización del tiempo de reajuste d	Ingierencia Civil	Ingierencia	2013	2013-01-30
21	19	316 Chambi Larico, Juan Jesus	Lineamientos estratégicos d desarrollo turis	Ingierencia Civil	Humanidades y Cs de la Edu	2013	2013-01-29
22	20	745 Chirino Flores, Fabiola Andrea	Propuesta de planificación d uso de suel	Ingierencia Geográfica	Cs. Geológicas	2013	2013-01-27
23	21	181 Chiquipa Vásquez, Omar Cristhian	Correlación entre el coeficiente d saturaci	Ingierencia Civil	Ingierencia	2013	2013-01-30
24	22	275 Choque Mamani, Cesar	Creación del archivo central en la dirección	Bibliotecología y Cs. Inform	Humanidades y Cs de la Edu	2013	2013-01-31
25	23	325 Choque Vásquez, German Felix	Influencia d la edad d comité d plástico	Ingierencia Agronómica	Agronomía	2013	2013-01-29
26	24	208 Churata Apaza, Tedilio	Medios audiovisuales como recurso didácti	Cs. de la Educación	Humanidades y Cs de la Edu	2013	2013-01-30
27	25	182 Churata Apaza, Amilia	Interpretación d cuentos infantiles, para m	Cs. de la Educación	Humanidades y Cs de la Edu	2013	2013-01-30
28	26	746 Coqueirua Campuzano, Irving Yan	Propuesta d planificación d uso d suel	Ingierencia Electromecánica	Ingierencia	2013	2013-01-28
29	27	180 Crespo Bezires, Mirella Nancy	Analisis térmico d sistema d concentraci	Ingierencia Civil	Cs. Sociales	2013	2013-01-30

Figura 3.19. Reporte Excel por mes

Fuente: (Elaboración propia)

PRUEBAS DE ACEPTACIÓN DE LAS HISTORIAS DE USUARIO 4

- a) Identificar todos los posibles resultados observables de la historia.**
 - Interfaz del menú de reportes.
 - Se identificaron dos interfaces para los reportes de Tesis: Por Facultad y por Año.
- b) Identificar los resultados que terminan la historia y los que permiten continuar dentro de la historia.**

La historia de usuario concluye:

 - Al salir de la Interfaz de menú de reportes.
 - Las 2 interfaces de reportes de Tesis: Por Facultad y Año son exportadas a Excel.
- c) Identificar los caminos de ejecución posible.**

A continuación se detalla los caminos de cada interfaz desarrollada.

- Las dos interfaces de reportes de Tesis, se lo exporta a Excel para guardarlos y/o imprimirlos.
- d) Asignar un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.**
 - Esta historia no requiere de asignar un conjunto de valores.
- e) Eliminación de caminos redundantes.**
 - No existe caminos redundantes.

3.5 FASE DE PRUEBAS

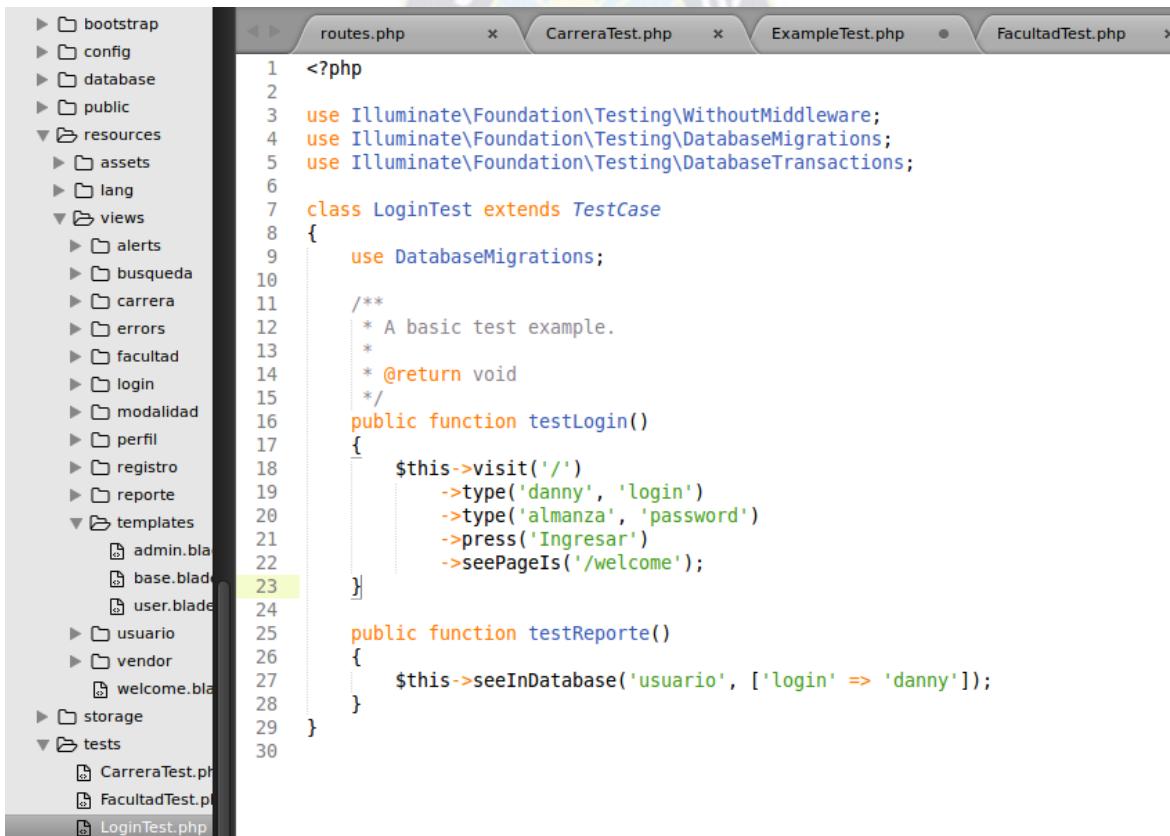
3.5.1 PRUEBAS DE INTEGRACIÓN

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados en unidad y estructurar un programa que esté de acuerdo con el que dicta el diseño.

Laravel se construye con las pruebas en mente. De hecho, el apoyo a las pruebas con PHPUnit se incluye fuera de la caja, y un archivo `phpunit.xml` ya está configurado para su aplicación. El marco también se incluye con métodos auxiliares convenientes que le permite probar expresivamente sus aplicaciones.

3.5.1.1 TEST LOGIN USUARIO

En la Figura 3.20 se puede observar la función realizada para verificar que el usuario que ingrese al sistema debe estar autenticado, caso contrario no podrá ingresar al sistema.

A screenshot of a code editor showing a file structure on the left and several tabs at the top. The tabs are labeled 'routes.php', 'CarreraTest.php', 'ExampleTest.php', and 'FacultadTest.php'. The 'routes.php' tab is active. The code editor displays a PHP file named 'LoginTest.php' with the following content:

```
1 <?php
2
3 use Illuminate\Foundation\Testing\WithoutMiddleware;
4 use Illuminate\Foundation\Testing\DatabaseMigrations;
5 use Illuminate\Foundation\Testing\DatabaseTransactions;
6
7 class LoginTest extends TestCase
8 {
9     use DatabaseMigrations;
10
11     /**
12      * A basic test example.
13      *
14      * @return void
15      */
16     public function testLogin()
17     {
18         $this->visit('/')
19             ->type('danny', 'login')
20             ->type('almanza', 'password')
21             ->press('Ingresar')
22             ->seePageIs('/welcome');
23     }
24
25     public function testReporte()
26     {
27         $this->seeInDatabase('usuario', ['login' => 'danny']);
28     }
29 }
30
```

The code uses the Laravel testing framework to perform a login test. It includes imports for middleware, database migrations, and transactions. It defines a `LoginTest` class that extends `TestCase` and uses the `DatabaseMigrations` trait. The `testLogin` method performs a visit to the root URL, types 'danny' into the login field, types 'almanza' into the password field, presses the 'Ingresar' button, and checks if the page is '/welcome'. The `testReporte` method checks if the 'usuario' table has a record where 'login' is 'danny'.

Figura 3.20: Test Login Usuario
Fuente: (Elaboración propia)

3.5.1.2 TEST USUARIO

En la Figura 3.21 se realiza las pruebas integraciones del módulo de usuario para su correcto funcionamiento del código, de la inserción de nuevos usuarios.



The figure shows a code editor interface with a sidebar containing a file tree and the main area displaying a PHP test script. The file tree on the left lists several directories and files, including bootstrap, config, database, public, resources (with assets, lang, and views subfolders), storage, tests (with CarreraTest.php, FacultadTest.php, LoginTest.php, ModalidadTest.php, RegistroTest.php, TestCase.php, and UsuarioTest.php), vendor, .env, .env.example, .gitattributes, .gitignore, artisan, and composer.json. The main area contains the following code:

```
<?php
use Illuminate\Foundation\Testing\WithoutMiddleware;
use Illuminate\Foundation\Testing\DatabaseMigrations;
use Illuminate\Foundation\Testing\DatabaseTransactions;

class UsuarioTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */

    public function testEnlace()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user);
        $response = $this->call('GET', '/usuario');
        $this->assertEquals(200, $response->status());
    }

    public function testIndex()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/usuario')
            ->see('USUARIOS');
    }

    public function testCreate()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/usuario/create')
            ->type('Danny Almanza', 'nombre')
            ->select('admin', 'categoria')
            ->press('Crear Usuario')
            ->call('POST', '/usuario');
    }
}
```

Figura 3.21: Test Usuario

Fuente: (Elaboración propia)

3.5.1.3 TEST FACULTAD

En la Figura 3.22 se realiza el test de facultad para su correcto funcionamiento en crear facultad, existencia del enlace, y la página principal funcionando.

The screenshot shows a code editor with a sidebar containing a file tree and three tabs at the top: routes.php, CarreraTest.php, and ExampleTest.php. The routes.php tab is active, showing a portion of a PHP file with syntax highlighting. The CarreraTest.php tab is visible but inactive. The ExampleTest.php tab is also visible but inactive. The file tree on the left lists several directories and files, including bootstrap, config, database, public, resources (with assets, lang, and views sub-directories), tests (with CarreraTest.php, FacultadTest.php, LoginTest.php, ModalidadTest.php, RegistroTest.php, TestCase.php, and UsuarioTest.php), vendor, .env, .env.example, .gitattributes, and .gitignore.

```
<?php
use Illuminate\Foundation\Testing\WithoutMiddleware;
use Illuminate\Foundation\Testing\DatabaseMigrations;
use Illuminate\Foundation\Testing\DatabaseTransactions;

class FacultadTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testEnlace()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user);
        $response = $this->call('GET', '/facultad');
        $this->assertEquals(200, $response->status());
    }

    public function testIndex()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/facultad')
            ->see('FACULTADES');
    }

    public function testCreate()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/facultad/create')
            ->type('Cs. Puras y Naturales', 'nombre')
            ->press('Crear Facultad')
            ->call('POST', '/facultad');
    }
}
```

Figura 3.22: Test Facultad

Fuente: (Elaboración propio)

3.5.1.4 TEST CARRERA

En la Figura 3.23 se realiza el test de la carrera para su correcto funcionamiento en crear carrera, existencia del enlace, y la página principal funcionando.

The screenshot shows a code editor with a sidebar displaying a project's directory structure. The main area contains a PHP test script named `CarreraTest.php`.

```
<?php
use Illuminate\Foundation\Testing\WithoutMiddleware;
use Illuminate\Foundation\Testing\DatabaseMigrations;
use Illuminate\Foundation\Testing\DatabaseTransactions;

class CarreraTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */

    public function testEnlace()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user);
        $response = $this->call('GET', '/carrera');
        $this->assertEquals(200, $response->status());
    }

    public function testIndex()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/carrera')
            ->see('Carreras');
    }

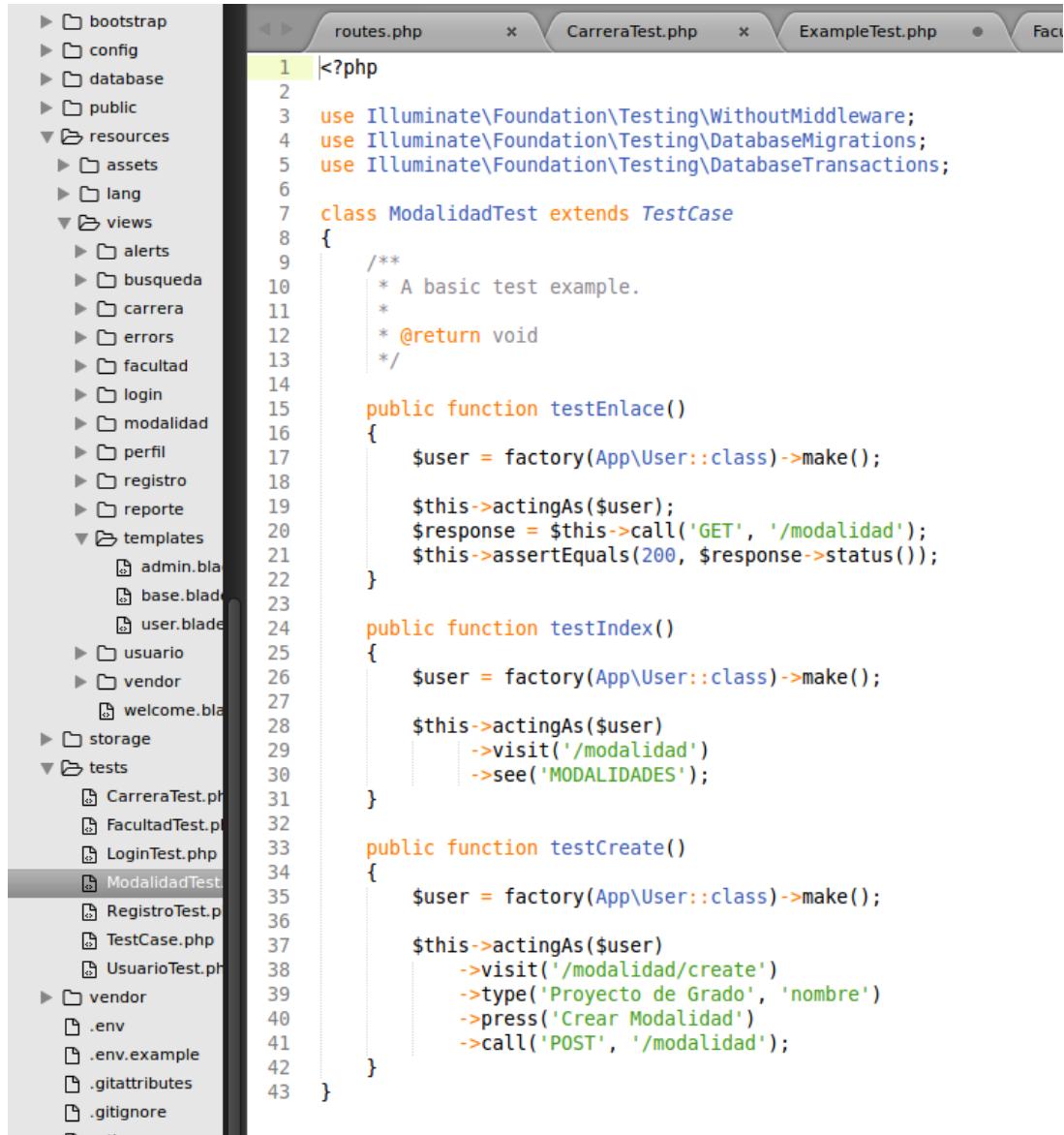
    public function testCreate()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/carrera/create')
            ->type('Informática', 'nombre')
            ->select('1', 'facultad_id')
            ->press('Crear Carrera')
            ->call('POST', '/carrera');
    }
}
```

Figura 3.23: Test Carrera
Fuente: (Elaboración propia)

3.5.1.5 TEST MODALIDAD

En la Figura 3.24 se realiza el test de la modalidad para su correcto funcionamiento en crear modalidad, existencia del enlace, y la página principal funcionando.



The figure shows a code editor interface with a sidebar containing a file tree and a main panel displaying a PHP test script. The file tree on the left lists several directories and files, including bootstrap, config, database, public, resources (with assets, lang, and views sub-directories), tests (with CarreraTest.php, FacultadTest.php, LoginTest.php, RegistroTest.php, TestCase.php, and UsuarioTest.php), vendor, .env, .env.example, .gitattributes, and .gitignore. The main panel contains the following code:

```
<?php
use Illuminate\Foundation\Testing\WithoutMiddleware;
use Illuminate\Foundation\Testing\DatabaseMigrations;
use Illuminate\Foundation\Testing\DatabaseTransactions;

class ModalidadTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */

    public function testEnlace()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user);
        $response = $this->call('GET', '/modalidad');
        $this->assertEquals(200, $response->status());
    }

    public function testIndex()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/modalidad')
            ->see('MODALIDADES');
    }

    public function testCreate()
    {
        $user = factory(App\User::class)->make();

        $this->actingAs($user)
            ->visit('/modalidad/create')
            ->type('Proyecto de Grado', 'nombre')
            ->press('Crear Modalidad')
            ->call('POST', '/modalidad');
    }
}
```

Figura 3.24: Test Modalidad

Fuente: (Elaboración propia)

3.5.1.6 TEST REGISTRO TESIS

En la Figura 3.25 se realiza el test de registro de tesis para su correcto funcionamiento en crear registro, existencia del enlace, y la página principal funcionando. Este es el modulo más importante para el llenado de todos los campos.



The screenshot shows a code editor with a sidebar displaying a file tree and the main pane showing a PHP test script. The file tree on the left includes bootstrap, config, database, public, resources (with assets, lang, views), templates (with admin.blade.php, base.blade.php, user.blade.php, usuario, vendor, welcome.blade.php), storage, tests (with CarreraTest.php, FacultadTest.php, LoginTest.php, ModalidadTest.php, RegistroTest.php, TestCase.php, UsuarioTest.php), vendor, .env, .env.example, .gitattributes, .gitignore, artisan, composer.json, composer.lock, gulpfile.js, package.json, phpspec.yml, phpunit.xml, readme.md, and server.php. The main pane contains the following code:

```
<?php  
use Illuminate\Foundation\Testing\WithoutMiddleware;  
use Illuminate\Foundation\Testing\DatabaseMigrations;  
use Illuminate\Foundation\Testing\DatabaseTransactions;  
  
class RegistroTest extends TestCase  
{  
    /**  
     * A basic test example.  
     *  
     * @return void  
     */  
  
    public function testEnlace()  
    {  
        $user = factory(App\User::class)->make();  
  
        $this->actingAs($user);  
        $response = $this->call('GET', '/registro');  
        $this->assertEquals(200, $response->status());  
    }  
  
    public function testIndex()  
    {  
        $user = factory(App\User::class)->make();  
  
        $this->actingAs($user)  
            ->visit('/registro')  
            ->see('REGISTRO');  
    }  
  
    public function testCreate()  
    {  
        $user = factory(App\User::class)->make();  
  
        $this->actingAs($user)  
            ->visit('/registro/create')  
            ->type('1234', 'codigo')  
            ->type('Mujica Villa', 'apellidos')  
            ->type('Alejandra', 'nombres')  
            ->type('Sistema Web', 'titulo')  
            ->type('M.Sc. Aldo Ramiro Alvarado valdez', 'tutor')  
            ->select('1', 'facultad')  
            ->select('', 'carrera')  
            ->type('2015', 'anio')  
            ->type('102', 'paginas')  
            ->type('95', 'nota')  
            ->select('1', 'modalidad')  
            ->press('Crear Registro')  
            ->call('POST', '/registro');  
    }  
}
```

Figura 3.25: Test Registro Tesis

Fuente: (Elaboración propia)

3.5.1.7 RESULTADO DE LAS PRUEBAS DE INTEGRACIÓN

La Figura 3.26 se puede observar el resultado de las pruebas, las cuales, después de realizado todos los test necesarios para comprobar el correcto funcionamiento basado en las pruebas de integración, muestran un resultado exitoso a lo largo del proceso.

```
danny@danny-desktop: /var/www/html/BCTesis2
danny@danny-desktop:/var/www/html/BCTesis2$ vendor/bin/phpunit
PHPUnit 4.8.19 by Sebastian Bergmann and contributors.

.
.
.
Time: 4.26 seconds, Memory: 19.25Mb

OK (17 tests, 36 assertions)
danny@danny-desktop:/var/www/html/BCTesis2$ █
```

Figura 3.26: Resultado de las Pruebas de Integración
Fuente: (Elaboración propia)

3.5.2 PRUEBAS DE ESTRÉS

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el nro de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema. Ver Figura 3.27.

Informe Agregado											
Nombre: Informe Agregado											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo	Navegar...	Log/Mostrar sólo:	<input type="checkbox"/> Escribir en Log Sólo Errores	<input type="checkbox"/> Éxitos	<input type="checkbox"/> Configurar						
Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec
Petición HTTP	500	22338	21115	47897	51205	53600	257	54163	21,60%	8,1/sec	30,0
Total	500	22338	21115	47897	51205	53600	257	54163	21,60%	8,1/sec	30,0

Figura 3.27: Reporte de resumen
Fuente: (Elaboración propia)

Como puede verse, el tiempo promedio para acceder a una página es 22338 segundos, realizándose un total de 500 requerimientos al servidor.

El tiempo total utilizado para los 500 threads se puede calcular con la siguiente fórmula:

$$\text{Tiempo Total} = \#Muestras * Media = 500 * 22338 = 11169000 \text{ milisegundos}$$

El tiempo promedio total requerido por cada thread, se puede calcular de la siguiente manera:

$$((\text{Tiempo Total} / 1000) / 60) / \text{cantidad de Thread} = ((11169000 / 1000) / 60) / 500 \\ = 0,3726 \text{ minutos}$$

Para ver los resultados de manera gráfica puede observar la Figura 3.28.

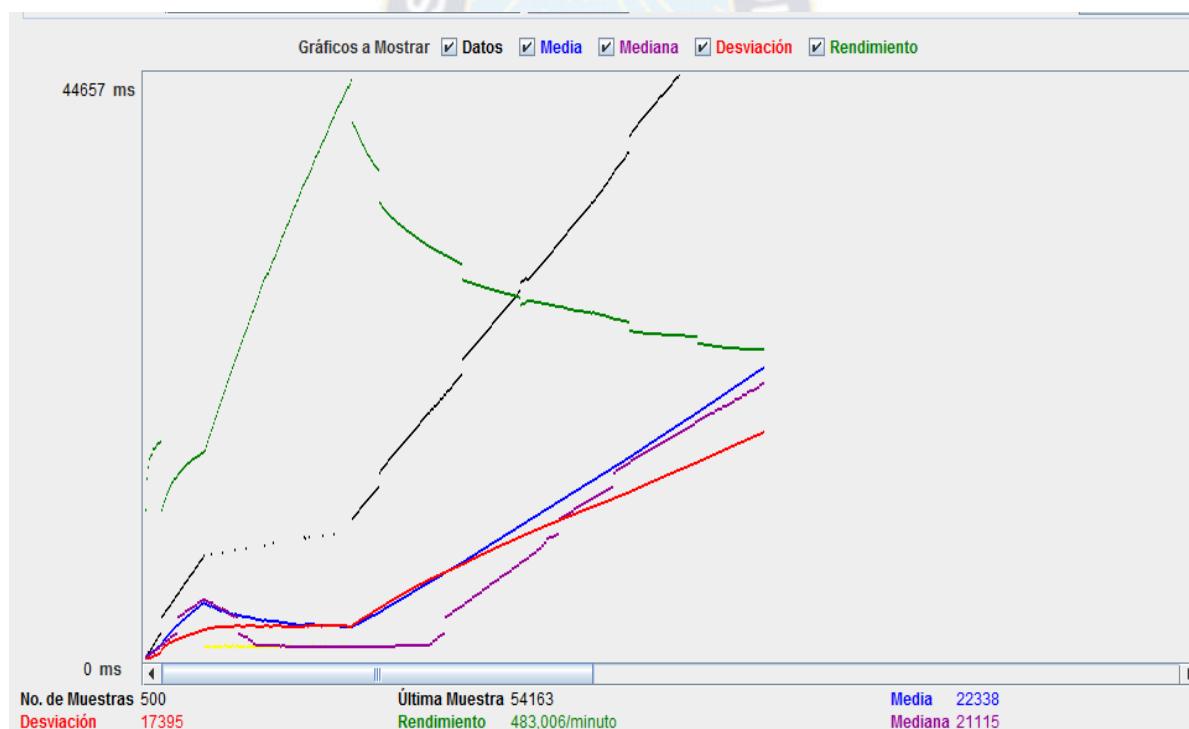


Figura 3.28: Gráfico del resultado
Fuente: (Elaboración propia)

3.6 FASE DE MANTENIMIENTO

Dado que durante esta fase si se encuentra algún nuevo requerimiento que implementar o algo que modificar se comienza todo el ciclo nuevamente.

Las historias de usuario presentaron algunas alteraciones durante las iteraciones, y las pruebas de aceptación fueron mejorando, por ejemplo los reportes de cada facultad ordenados alfabéticamente por apellidos y nombres, además se agregaron interfaces como Administración de Facultades, Carreras y Modalidades.

3.7 DISEÑO DE LA ARQUITECTURA DEL SISTEMA

La arquitectura del sistema está comprendida tanto en la descripción del diseño y contenido, incluye información sobre la organización fundamental del sistema, que incluye sus componentes, las relaciones entre sí y el ambiente, y los principios que gobiernan su diseño y su distribución. (Ver Figura 3.29)

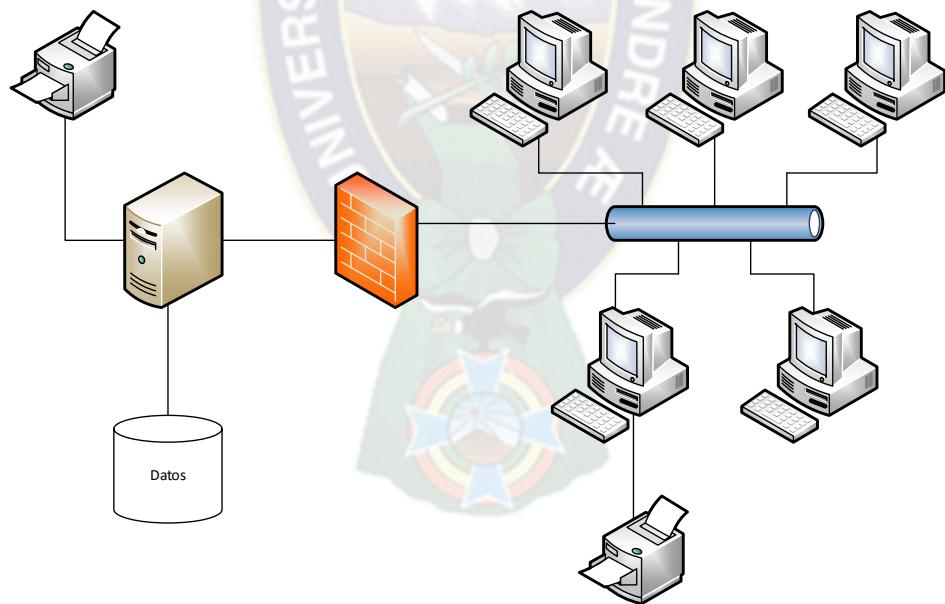


Figura 3.29. Arquitectura del Sistema

[Elaboración propia]

3.8 FASE CIERRE DE PROYECTO

El sistema web para la Biblioteca Central UMSA, se concluirá exitosamente gracias a las prácticas del método XP y OOHDMD, adicional a este proyecto se adjuntará el “**Manual de Usuario**”, además se capacito al personal, explicándoles paso a paso cómo funciona el Sistema y facilitándoles el manual de usuario del “Sistema de Registro, Control y Seguimiento de Tesis”.

3.9 VALORES

Para el desarrollo del presente sistema el equipo (1 integrante) debe tomar en cuenta cuatro valores que inspira XP.

3.9.1 COMUNICACIÓN

Se mantendrá comunicación directa con los clientes. El encargado de las Tesis se integra al equipo para establecer prioridades y resolver dudas. De esta forma con el desarrollo del proyecto.

3.9.2 SIMPLICIDAD

Se analiza y resuelve solamente los problemas y necesidades del cliente en cada historia de usuario, no se adiciona funcionalidades que el cliente no requiere.

3.9.3 VALENTÍA

Ya que el equipo consta de una persona, es necesario tener mucha *actitud* para estar sujeto a cambios o modificaciones, comprender los requerimientos del cliente, esclarecer y manejar los *tiempos* de desarrollo siendo este oponente principal para el equipo.

3.9.4 FEEDBACK

El flujo de retro-alimentación muy valioso para detectar los problemas o desviaciones, en cada iteración se mejora la comprensión y necesidades del cliente, adicionando al mismo tiempo nuevas funcionalidades en alguna de las iteraciones.

Con una *retroalimentación* constante se garantiza un alto grado de satisfacción del cliente.

CAPÍTULO IV CALIDAD DE SOFTWARE Y SEGURIDAD

4.1 INTRODUCCIÓN

Los desarrolladores de software más hastiados estarán de acuerdo que el software de alta calidad es una de las metas más importantes. La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según el cliente que las pida. (Pressman, 2010)

La calidad del software se define también, en términos de ausencia de errores en el funcionamiento del sistema. El ajuste a las necesidades del usuario, el sistema debe ser flexible u susceptible a modificaciones que se puedan realizar de manera rápida y oportuna. El sistema debe alcanzar un desempeño apropiado en términos de tiempo, volumen y espacio. Un sistema debe cumplir de la mejor forma los estándares internacionales establecidos, en lo que a la calidad de software se refiere

4.2 CALIDAD DE SOFTWARE

La calidad de software no es necesariamente llegar al producto perfecto aunque sea lo deseable, las métricas de calidad no son absolutas, nos proporcionan una manera sistemática de valorar la calidad basándose en un conjunto de «reglas claramente definidas». La ISO 9126 identifica los siguientes atributos:

4.2.1 FUNCIONALIDAD

La funcionalidad no se mide directamente, por tanto no es necesario evaluar un conjunto de características y capacidades del sistema.

Para el cálculo de la funcionalidad utilizaremos la métrica de “**Punto Función**” (**PF**), para esto se debe determinar cinco características de dominios de información necesarias para el cálculo de la misma.

- i. **Número de Entradas del Usuario.** Se cuenta cada entrada del usuario que proporciona diferentes datos al sistema, en el caso del sistema se identificaron **10 entradas** del usuario.
- ii. **Número de Salidas del Usuario.** Se cuenta cada salida que proporciona la información del usuario, estas pueden ser informes, reportes, etc. Se identificaron **8 salidas** del usuario en el sistema.
- iii. **Número de Peticiones del Usuario.** Se cuentan la cantidad de entradas interactivas que producen la generación de respuestas (salidas) inmediatas del sistema. Se apreciaron **8 peticiones** de usuario.
- iv. **Número de Archivos.** Se cuenta cada archivo maestro lógico es decir un grupo lógico de datos que sean parte de la base de datos, o archivos independientes. Se contaron **18 archivos**.
- v. **Número de Interfaces Externas.** Se cuenta todas las interfaces legibles por la maquina (por ejemplo: archivos de datos de disco) que se utilizan para transmitir información a otros sistemas. En este caso existen **3 interfaces externas**.

La siguiente tabla muestra las cinco características con factor de ponderación medio para el cálculo de punto función

Parámetros de medición	Factor de Ponderación				Resultado
	Cuenta	Simple	Medio	complejo	
Nro. de entradas de usuario	10	3	4	6	40
Nro. de salidas de usuario	8	4	5	7	40
Nro. de peticiones de usuario	8	3	4	6	32
Nro. de archivos	18	7	10	15	180
Nro. de interfaces externas	3	5	7	10	21
	Cuenta Total				313

Tabla 4.1 Cálculo de Punto Función
Fuente: (Elaboración Propia)

ESCALA DE AJUSTE		Importancia	Incremental	Moderado	Medio	Significativo	Esencial	F_i
	FACTOR	0	1	2	3	4	5	-
1	¿Requiere el sistema copias de seguridad y recuperación flexible?						X	5
2	¿Se requiere comunicación de datos?						X	5
3	¿Existen funciones del procedimiento distribuido?				X			3
4	¿Es crítico el rendimiento?				X			3
5	¿Se ejecutara el sistema en un entorno operativo existente y fuertemente utilizado?						X	5
6	¿Requiere el sistema entrada interactiva?					X		4
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?				X			3
8	¿Se actualiza los archivos maestros de forma interactiva?						X	5
9	¿Son complejos las entradas, las salidas, los archivos y las peticiones?				X			3
10	¿Es complejo el procedimiento interno?					X		4
11	¿Se ha diseñado el código para ser reutilizable?						X	5
12	¿Están incluidas en el diseño la conversión y la instalación?					X		4

13	¿Se ha desarrollado el sistema para soportar múltiples instalaciones en diferentes organizaciones?						X	5
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizado por el usuario?						X	5
	FACTOR DE COMPLEJIDAD TOTAL (FCT) $\sum F_i$							59

Tabla 4.2 Ajuste de Complejidad

Fuente: (Elaboración Propia)

La relación que permite calcular los puntos función es la siguiente:

$$PF = Cuenta\ Total * \left(0.65 + 0.01 * \sum F_i \right)$$

$$PF = 313 * (0.65 + 0.01 * 59)$$

$$PF = 388.12$$

Posteriormente hallamos el punto función máximo para comparar los valores de funcionalidad del Sistema, considerando el máximo valor de ajuste de complejidad como

$$\sum F_i = 70: PF_{max} = 313 * (0.65 + 0.01 * 70)$$

$$PF_{max} = 422.55$$

Entonces:

$$Funcionalidad = \left(\frac{PF}{PF_{max}} \right) * 100\%$$

$$Funcionalidad = \left(\frac{388.12}{422.55} \right) * 100\%$$

$$Funcionalidad = (0.9185) * 100\%$$

$$\text{Funcionalidad} = 91.85\%$$

Por lo tanto el Sistema de Registro, Seguimiento y Control de Tesis tiene una funcionalidad de 91.85%.

4.2.2 CONFIABILIDAD

La confiabilidad está definida como la probabilidad del sistema libre de fallos en un contexto determinado y durante un periodo de tiempo.

$$\text{Probabilidad de hallar una falla: } P(T \leq t) = F(t)$$

$$\text{Probabilidad de no hallar una falla: } P(T > t) = 1 - F(t)$$

$$\text{La confiabilidad es } F(t) = Fc * e^{(-\frac{\lambda}{4}t)}$$

Dónde:

$Fc = 0.9185$: Funcionalidad del sistema.

$\lambda = 1$: Tasa de fallos en 4 ejecuciones dentro de un mes.

Hallando confiabilidad:

$$F(t) = Fc * e^{(-\frac{\lambda}{4}t)}$$

$$F(12) = Fc * e^{(-\frac{1}{4}12)}$$

$$F(t) = 0.04$$

Hallando probabilidad de falla:

$$P(T \leq t) = F(t)$$

$$F(t) = 0.04$$

Hallando probabilidad de no hallar una falla:

$$P(T > t) = 1 - F(t)$$

$$1 - F(t) = 1 - 0.04$$

$$1 - F(t) = 0.96$$

Por lo tanto, el Sistema presenta una **Confiabilidad del 0.96**, lo que quiere decir que el **96% de las ocasiones**, el sistema funciona sin presentar fallos y el resto (4%), presenta fallos que no afectan de sobremanera el desempeño global del sistema.

4.2.3 PORTABILIDAD

La Portabilidad, es la posibilidad de que el software pueda ser llevado de un entorno a otro (hardware/software). (Pressman, 2010)

- **Adaptabilidad:** El software es apto para funcionar bajo las plataformas de: Windows en sus versiones 98/XP/Vista/Seven, y también es posible adaptar a algunas distribuciones de Linux.
- **Facilidad de Instalación:** El código del sistema está hecho en PHP un lenguaje de servidor, por lo que no se necesitará instalar, pues simplemente se tendrá acceso a través de la red “Internet” por medio de diferentes exploradores como Explorer Internet, Mozilla Firefox, Google Chrome y otros.

4.2.4 MANTENIBILIDAD

- **Mantenimiento Adaptivo:**

El mantenimiento adaptativo ocurrirá cuando se cambien las políticas o cuando se cambie la estructura organizacional, o cambie el personal de alguna de las Unidades, modificaciones que harán que el sistema cambie en poca o gran medida, cambios para los cuales el sistema está preparado en adaptarse a algunos de estos casos, pero para otros más complejos se deberá hacer revisión de los procesos y su adaptación con los nuevos cambios que se generen.

- **Mantenimiento Perfectivo:**

El sistema está completamente abierto a añadir o adicionar nuevas funcionalidades de acuerdo a los nuevos requerimientos del cliente, siempre y cuando sean relacionados con el servicio e información que brinda el sistema.

El índice de madurez de software (IMS) proporciona una indicación de la estabilidad de un producto de software y nos sirve de métrica de calidad del mantenimiento del sistema.

Calculamos el índice de madurez del software con la siguiente relación:

$$IMS = \frac{[M_t - (F_c + F_a + F_e)]}{M_t}$$

Dónde:

M_t : número de módulos en la versión actual

F_c : número de módulos en la versión actual que se han cambiado

F_a : número de módulos en la versión actual que se han añadido

F_e : número de módulos en la versión anterior que se han eliminado en la versión actual

Se tiene que el índice de madurez del software es:

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

$$IMS = \frac{[6]}{6}$$

$$IMS = 1$$

Por tanto el sistema empieza a **estabilizarse en un 100%**.

4.2.5 RESULTADOS

En la siguiente tabla se muestra los resultados obtenidos de la calidad de software utilizando la ISO 9126.

FUNCIONALIDAD	91.85%
CONFIABILIDAD	96%
PORTABILIDAD	100%
MANTENIBILIDAD	100%
TOTAL	96.96%

Tabla 4.3: Tabla de resultados de calidad

Fuente: (Elaboración propia)

4.3 SEGURIDAD

4.3.1 SEGURIDAD DEL SISTEMA

Dado que el sistema corre bajo una intranet se debe considerar por lo menos algunas reglas básicas a cumplirse para que la seguridad del sistema no sea una preocupación en las Unidades de la Biblioteca Central UMSA.

Política de contraseñas. El sistema es capaz de realizar la comprobación de contraseñas para los 2 tipos de usuarios que se tiene actualmente, también es capaz de asignar nuevos usuarios con niveles de acceso.

Si el usuario requiere cambiar de contraseña, deberá solicitarlo al administrador.

Política de respaldo. El administrador del sistema es el responsable de realizar respaldos de la información periódicamente. Cada treinta días deberá efectuarse un respaldo completo del sistema y también deberán ser respaldados todos los archivos que fueron modificados o creados.

Seguridad en el servidor. La validación del lado del cliente no es suficiente, también tienen que realizarse otro tipo de controles por el lado del servidor:

- **Backups:** Se debe contar con copias de seguridad, almacenados en dispositivos externos para cualquier contingencia.
- Se debe contar con un **firewall** para contrarrestar amenazas externas.
- El servidor cuenta con una clave de seguridad, puesto que es restringido y solamente puede ingresar el administrador de sistemas o personal autorizado.

4.3.2 SEGURIDAD DE INFORMACIÓN

Confidencialidad. Los registros de las tesis solamente podrán ser vistos por los usuarios que solicitaron.

Integridad. Los archivos no sufrirán alteraciones ni suplantaciones ya que están encriptados en la base de datos.

Autenticación. Cuando se realiza el registro de las tesis el usuario tiene que estar previamente registrado para realizar la acción.

No Repudio. Existen dos posibilidades:

- No repudio en origen: El emisor no puede negar que envío porque el destinatario tiene pruebas del envío, el receptor recibe una prueba infalsificable del origen del envío, lo cual evita que el emisor, de negar tal envío, tenga éxito ante el juicio de terceros. En este caso la prueba la crea el propio emisor y la recibe el destinatario
- No repudio en destino: El receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción. Este servicio proporciona al emisor la prueba de que el destinatario legítimo de un envío, realmente lo recibió, evitando que el receptor lo niegue posteriormente. En este caso la prueba irrefutable la crea el receptor y la recibe el emisor.

CAPÍTULO V ANÁLISIS, COSTO/BENEFICIO

5.1 ANÁLISIS DE COSTOS

Se debe calcular todos los costos anticipados asociados con el sistema. Para determinar el costo total del proyecto se tomará en cuenta los siguientes costos:

- Costo del software desarrollado
- Costo de la implementación del Sistema
- Costo de la elaboración del proyecto.

5.2 COSTO DEL SOFTWARE DESARROLLADO

El modelo COCOMO II (COnstructive COst MOdel) desarrollado por Barry M. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto. Por un lado COCOMO II define tres modos de desarrollo o tipos de proyectos:

- **Orgánico:** proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- **Empotrado:** proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Y por otro lado existen diferentes modelos que define COCOMO II:

- **Modelo básico:** Se basa exclusivamente en el tamaño expresado en LDC.

- **Modelo intermedio:** Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- **Modelo avanzado:** Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

Para nuestro caso el modelo intermedio será el que usaremos, dado que realiza las estimaciones con bastante precisión.

Así pues las fórmulas serán las siguientes:

- $E = \text{Esfuerzo} = a * KLDC^e * FAE$ (persona x mes)
- $T = \text{Tiempo de duración del desarrollo} = c * E^d$ (meses)
- $P = \text{Personal} = \frac{E}{T}$ (personas)

Para determinar el costo del software desarrollado, se utilizará el modelo constructivo COCOMO II, orientados a los puntos función, que se calcularon en la parte de funcionalidad. Ahora convertimos los puntos función a número estimado de líneas de código distribuidas. Para ello se tiene la siguiente tabla:

LENGUAJE	NIVEL	FACTOR LDC
C	2.5	128
ANSI BASIC	5	64
JAVA	6	53
COBOL	3	107
VISUAL BASIC	7	46
ASP	9	36
PHP	11	29
VISUAL C++	9.5	34

Tabla 5.1 Conversión de puntos función

$$LDC = PF * factorLDC$$

$$LDC = 388.12 * 29 = 11255,48$$

Por tanto, el número estimado de líneas de código distribuidas en miles

$$KLDC = 11.25$$

Así pues, en nuestro caso el tipo orgánico será el más apropiado ya que el número de líneas de código no supera los 50 KLDC, y además el proyecto no es muy complejo. Y por otro lado también hemos de hallar la variable FAE, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observan en la siguiente tabla:

Conductores de coste	VALORACIÓN					
	<i>Muy bajo</i>	<i>Bajo</i>	<i>Nominal</i>	<i>Alto</i>	<i>Muy alto</i>	<i>Extr. Alto</i>
Fiabilidad requerida del software	0,75	0,88	1.00	1,15	1,40	-
Tamaño de la base de datos	-	0,94	1.00	1,08	1,16	-
Complejidad del producto	0,70	0,85	1.00	1,15	1,30	1,65
Restricciones del tiempo de ejecución	-	-	1.00	1,11	1,30	1,66
Restricciones del almacenamiento principal	-	-	1.00	1,06	1,21	1,56
Volatilidad de la máquina virtual	-	0,87	1.00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1.00	1,07	1,15	-

Capacidad del analista	1,46	1,19	1.00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1.00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1.00	0,86	0,70	-
Experiencia en S.O. utilizado	1,21	1,10	1.00	0,90	-	-
Experiencia en el lenguaje de programación	1,14	1,07	1.00	0,95	-	-
Prácticas de programación modernas	1,24	1,10	1.00	0,91	0,82	-
Utilización de herramientas software	1,24	1,10	1.00	0,91	0,83	-
Limitaciones de planificación del proyecto	1,23	1,08	1.00	1,04	1,10	-

Tabla 5.2 Conductores de coste

$$FAE = 1,15 * 1,00 * 0,85 * 1,11 * 1,00 * 1,00 * 1,07 * 0,86 * 0,82 * 0,70 * 1,00 * 0,95 \\ * 1,00 * 0,91 * 1,08 = 0,53508480$$

Justificación de los valores:

Atributos de software

- **Fiabilidad requerida del software:** Si se produce un fallo por el pago de un pedido, o fallo en alguna reserva, etc... puede ocasionar grandes pérdidas a la empresa (Valoración Alta).
- **Tamaño de la base de datos:** La base de datos de nuestro producto será de tipo estándar (Valoración Nominal).

- **Complejidad del producto:** La aplicación no va a realizar cálculos complejos (Valoración Baja).

Atributos de hardware

- **Restricciones del tiempo de ejecución:** En los requerimientos se exige alto rendimiento (Valoración Alta).
- **Restricciones del almacenamiento principal:** No hay restricciones al respecto (Valoración Nominal).
- **Volatilidad de la máquina virtual:** Se usarán sistemas de la “Familia Windows” (Valoración Nominal).
- **Tiempo de respuesta del ordenador:** Deberá ser interactivo con el usuario (Valoración Alta).

Atributos del personal

- **Capacidad del analista:** Capacidad alta relativamente, debido a la experiencia en análisis en proyecto similar (Valoración Alta)
- **Experiencia en la aplicación:** Se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración muy alta).
- **Capacidad de los programadores:** Teóricamente deberá tenerse una capacidad muy alta por la experiencia en anteriores proyectos similares (Valoración muy alta).
- **Experiencia en S.O. utilizado:** Con Windows 2000 Professional la experiencia es a nivel usuario (Valoración Nominal).
- **Experiencia en el lenguaje de programación:** Es relativamente alta, dado que se controlan las nociones básicas y las propias del proyecto (Valoración Alta).

Atributos del proyecto

- **Prácticas de programación modernas:** Se usarán prácticas de programación mayormente convencional (Valoración Nominal).

- **Utilización de herramientas software:** Se usarán herramientas estándar que no exigirán apenas formación, de las cuales se tiene cierta experiencia (Valoración Alta).
- **Limitaciones de planificación del proyecto:** Existen pocos límites de planificación. (Valoración Baja).

Ahora aplicaremos las formulas básicas del esfuerzo, tiempo calendario y personal requerido.

Proyecto de Software	a	B	c	D
Orgánico	2.4	1.05	2.5	0.38
Semi-acoplado	3	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Tabla 5.3 Coeficientes

Como este proyecto es un proyecto intermedio en tamaño y complejidad se eligió los coeficientes de la opción Semi-acoplado.

$$E = a * (KLCD)^b * FAE$$

$$D = c * (E)^d$$

$$E = 2,4 * (11.25)^{1,05} * 0,53508480$$

$$D = 2.5 * (45.12)^{0.38}$$

$$E = 16,31$$

$$D = 7,22$$

Cálculo del esfuerzo del desarrollo:

$$E = \mathbf{16,31 \text{ personas /mes}}$$

Cálculo tiempo de desarrollo:

$$D = \mathbf{7,22 \text{ meses}}$$

Personal promedio:

$$P = \frac{E}{D} = \frac{16,31}{7,22} = 2,25 \text{ personas}$$

Número de programadores = 2,25 ≈ 3

El número de programadores requerido es de 3 aproximadamente.

El salario de un programador es aproximadamente 500 \$us, entonces se estima que el costo del software es el siguiente:

*Costo de Software = Número de programadores * Salario de Programador*

*Costo de Software = 3 * 500 \$us*

Costo de Software = 1500 \$us/mes

Costo total del software por 7 meses = 10500 \$us

5.3 COSTO DE IMPLEMENTACIÓN

La Biblioteca Central de la Universidad Mayor de San Andrés cuenta con un servidor de aplicaciones y la Base de datos instalados y configurados, por lo que el costo de implementación es cero.

5.4 COSTO DE LA ELABORACIÓN DEL PROYECTO

Esta parte se refiere a los costos de estudio del sistema representados en la siguiente tabla:

Descripción	Costo \$us
Impresiones	25
Bibliografía	30
Fotocopias	20
Internet	100
Otros	50
Total	225 \$us

Tabla 5.4 Detalle de Elaboración del proyecto

5.5 COSTO TOTAL

El costo total es la suma de: El costo del software desarrollado, el costo de la implementación del Sistema y el Costo de la elaboración del proyecto.

Descripción	Costo \$us
Costo del software desarrollado	10500
Costo de la implementación del Sistema	0
Costo de la elaboración del proyecto	225
Costo Total	10725 \$us

Tabla 5.5 Detalle del Costo Total

5.6 VALOR ACTUAL NETO (VAN)

Este método es un clásico de la valoración de inversiones en activos fijos, proporcionando una valoración financiera en el momento actual de los flujos de caja netos proporcionados por la inversión.

Es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto.

Fórmula que nos permite calcular el Valor Actual Neto es:

$$\text{VAN} = \sum_{t=1}^n \frac{V_t}{(1+k)^t} - I_0$$

V_t = representa los flujos de caja en cada periodo t.

I_0 = es el valor del desembolso inicial de la inversión.

n = es el número de períodos considerado.

k = es el tipo de interés.

	MES 1	MES 2	MES 3	MES 4	MES 5	MES 6	MES 7
Flujo de caja neto	2500	2500	2500	2500	2500	2500	2500

Inversión: 10725 \$us

Beneficio Actual Neto: 17500 \$us

$$VAN = \frac{2500}{(1 + 0,04)^1} + \frac{2500}{(1 + 0,04)^2} + \frac{2500}{(1 + 0,04)^3} + \frac{2500}{(1 + 0,04)^4} + \frac{2500}{(1 + 0,04)^5} \\ + \frac{2500}{(1 + 0,04)^6} + \frac{2500}{(1 + 0,04)^7} - 10725$$

$$VAN = 4280,14 \$us$$

5.7 TASA INTERNA DE RETORNO (TIR)

También conocida como tasa interna de rentabilidad de una inversión, está definida como la tasa de interés con la cual el valor actual neto o valor presente neto (VAN o VPN) es igual a cero. El VAN o VPN es calculado a partir del flujo de caja anual, trasladando todas las cantidades futuras al presente. Es un indicador de la rentabilidad de un proyecto, a mayor TIR, mayor rentabilidad.

Utilizando la función de Excel, se tiene el valor de TIR

$$TIR = 14\%$$

5.8 COSTO BENEFICIO

El análisis de costo beneficio es un proceso para encontrar buenas respuestas a las preguntas siguientes: Dada una evaluación de costos y beneficios realizada por un gerente, ¿Cuál sería la opción recomendada? El análisis de costo beneficio envuelve los siguientes pasos generales: Especificar una lista de todos los cursos de acción posibles. Asignar un valor (positivo o negativo) para los resultados de cada acción a tomar, y calcular la probabilidad de cada resultado. Calcular el resultado esperado para cada acción o procedimiento. Tomar el procedimiento que proporcione el mejor resultado esperado.

$$\frac{B}{C} = \frac{11796,11}{10725}$$

$$\frac{B}{C} = 1,1$$

La relación beneficio costo significa que con cada dólar que estamos invirtiendo en el proyecto, estamos obteniendo 1.1 dólares de ganancia.

Como puede verse el valor calculado de la relación calculado costo es 1.10, lo que indica que para este indicador el proyecto es factible.

5.9 RESULTADO VAN, TIR, B/C

En la siguiente tabla se detalla los resultados hallados de VAN, TIR, y de costo beneficio, indicando que el proyecto es factible. (Ver Tabla 5.6)

VAN	4280,14
TIR	14%
B/C	1,1

Tabla 5.6: Resultado de costos
Fuente: (Elaboración propia)

CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES

6.1 RESUMEN

Este Capítulo indica las conclusiones del proyecto y algunas recomendaciones necesarias para la aplicación y rendimiento del sistema.

6.2 CONCLUSIONES

La realización de este Proyecto, permitió elaborar un Sistema Automatizado para controlar el registro de las tesis que se procesan en la Biblioteca Central de la Universidad Mayor de San Andrés, fue desarrollado como una manera de darle solución a las fallas que se presentan al cumplir con esas funciones en forma manual, cuyo planteamiento del problema se originó de la observación directa de las tareas y actividades que se cumplen.

Entre los logros destacados tenemos:

- Se implementó el sistema de en la Biblioteca Central satisfactoriamente. Haciendo uso de metodologías orientadas a objetos para modelar el sistema, identificándose el medio, los actores y casos de uso para su implementación.
- Se implementó el módulo de impresión de certificados de las tesis, para facilitar la entrega al interesado.
- Se utilizó la tecnología AJAX para facilitar el llenado de los certificados seleccionando las Carreras según por las Facultades que cuenta la UMSA.
- Se implementó el módulo de búsqueda de tesis, para facilitar la búsqueda de los encargados responsables.
- Se realizó la búsqueda y la exportación de la tesis con toda información necesaria para la impresión del certificado.
- Se implementó el módulo para obtener reportes según a la petición del usuario en diferentes formatos.

- Se implementó el módulo para obtener reportes estadísticos para ver las tesis ingresadas por facultad, año.
- Con el módulo de búsqueda se puede realizar reportes fácilmente seleccionando el reporte que requiere para su impresión y almacenamiento en documento físico.
- El realizó el sistema con una interfaz sencilla y amigable para el llenado de las tesis y la vista previa del certificado antes de imprimir.

El proyecto fue desarrollado por módulos el cual puede ser susceptible de efectuarle ajustes, de acuerdo a los nuevos requerimientos que se tengan en el proceso de puesta en marcha del sistema.

6.3 RECOMENDACIONES

Una vez implementado el sistema y analizando los resultados que se obtuvieron se recomienda:

- Implementar el módulo para realizar el respaldo periódicamente de manera automática, para no tener pérdidas significativas por motivo de fallos.
- Implementar el módulo de impresión de certificados de no deuda de libros, porque se cuenta con la información requerida de los interesados que solicitan el certificado de las tesis.
- Implementar el módulo de búsqueda de tesis para los estudiantes, para poder realizar el prestado de las tesis.

BIBLIOGRAFÍA

- Álvarez Guerra, L. A. (2005). *Biblioteca Virtual Carrera de Informática*. La Paz, Bolivia.
- Apaza Guachalla, J. (2008). *Biblioteca Virtual basada en Web Semántica para el Postgrado en Informática*. La Paz - Bolivia.
- Beck, K. (2002). *Una explicación a la Programación Extrema*. Madrid - España: Iberoamericana.
- Carrasco Quino, G. (2009). *Sistema de Control y Gestión Hotelera*. La Paz.
- Del Valle Rodríguez, A. N. (2009). *Metodologías de Diseño usadas en Ingeniería Web, su vinculación con las NTICS*.
- Díaz, C. (11 de 2012). *Planificación de mi proyecto II*. Recuperado el 2 de Octubre de 2014, de <http://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>
- DTIC. (2005). *BIBLIOTECA CENTRAL UNIVERSITARIA - UMSA*. Recuperado el 29 de Septiembre de 2014, de Biblioteca Virtual de Salud: <http://pueblosindigenas.bvsp.org.bo/php/level.php?lang=es&component=19&item=29>
- Escalona, M. J. (2004). *Modelos y técnicas para la especificación y el análisis de la navegación en sistemas de software*. Sevilla - España.
- Fernández Escribano, G. (2002). *Introducción a Extreme Programming, Ingeniería del Software II*. Albacete - España.
- Garcia, R., & Platini, V. (2003). *Calidad en el Desarrollo y Mantenimiento del Software*.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la Investigación*. Mexico D.C.: McGRAW-HILL.
- Larman, C. (2004). *UML y Patrones*. México DF - México (Primera Edición): Prentice Hall.

Mamani Uturunco, M. (2013). Sistema de Control Automatizado de Inventarios de Insumos Médicos y Farmacia. La Paz: UMSA.

Martinez, J. C. (21 de Agosto de 2009). Recuperado el 15 de Septiembre de 2014, de <http://www.ci.ulsa.mx/~elinos/docencia/ctrldesa/ISO14764.pdf>, 11/8/201

Martínez, R., & Fernández, A. (2006). Obtenido de http://educacion.chihuahua.gob.mx/sites/default/files/Arbol_de_problemas_y_objetivos.pdf

Monasterio, A. (2012). *Sistema Gestión de Correspondencia. Caso: Carrera de Administración de Empresas*. La Paz.

Peréz Meza, A. (2010). *Desarrollo e implementación de un Sistema Orientado a la Web para el Instituto Oftalmológico Gustavo Moreno J, Escuela Politécnica del Ejército*. Sangolqui - Ecuador.

Pressman, R. (2010). *Ingeniería de Software, 7º Edición*. Madrid-España: McGraw Hill.

Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. Prentice Hall.

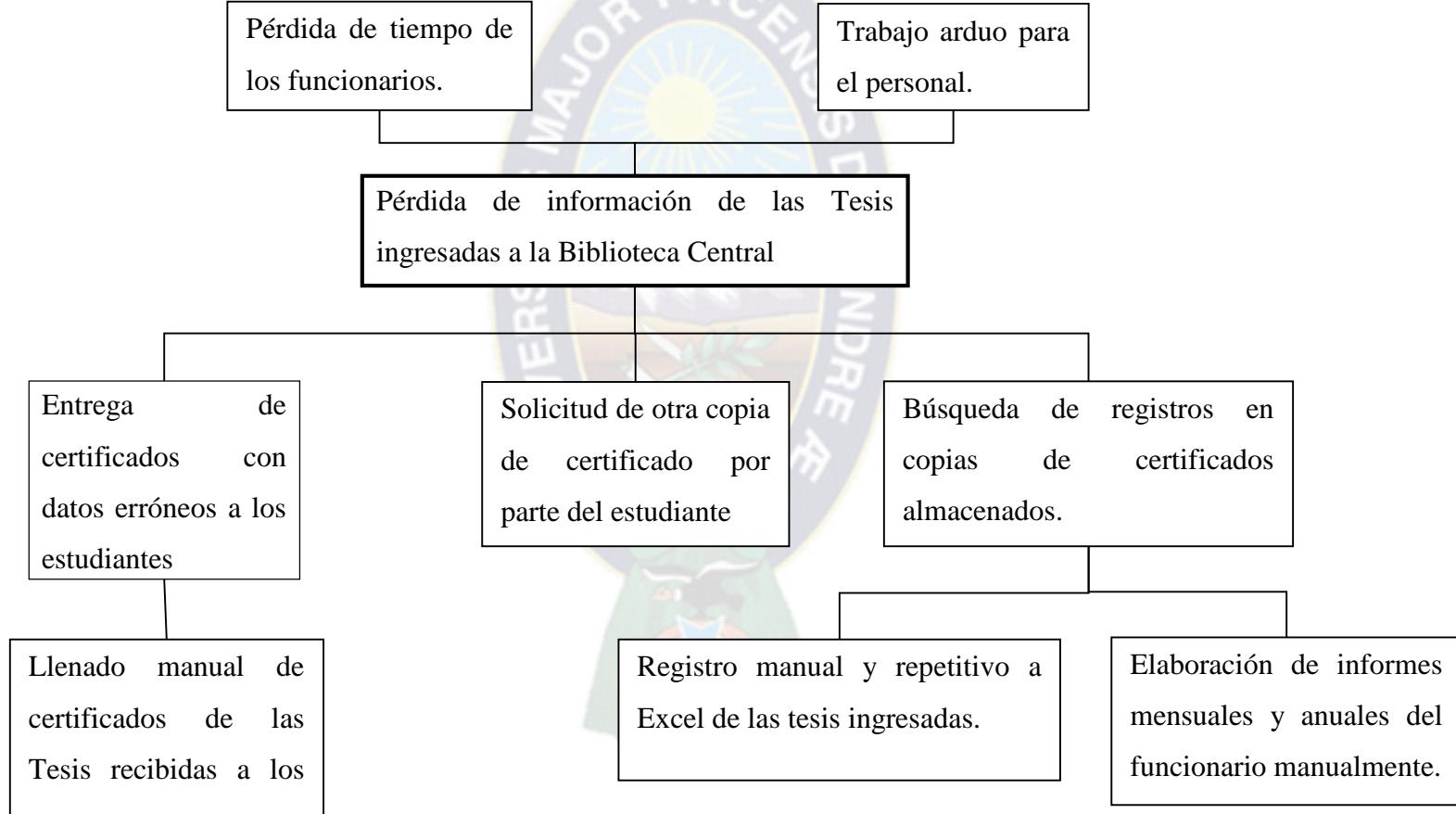
Solís Pineda, C. (2008). *Un Método de Desarrollo de Hipermédia Dirigido por Modelos*.

Verástegui Vilela, L. (1980). *50 Años de la Biblioteca Universitaria*. La Paz.

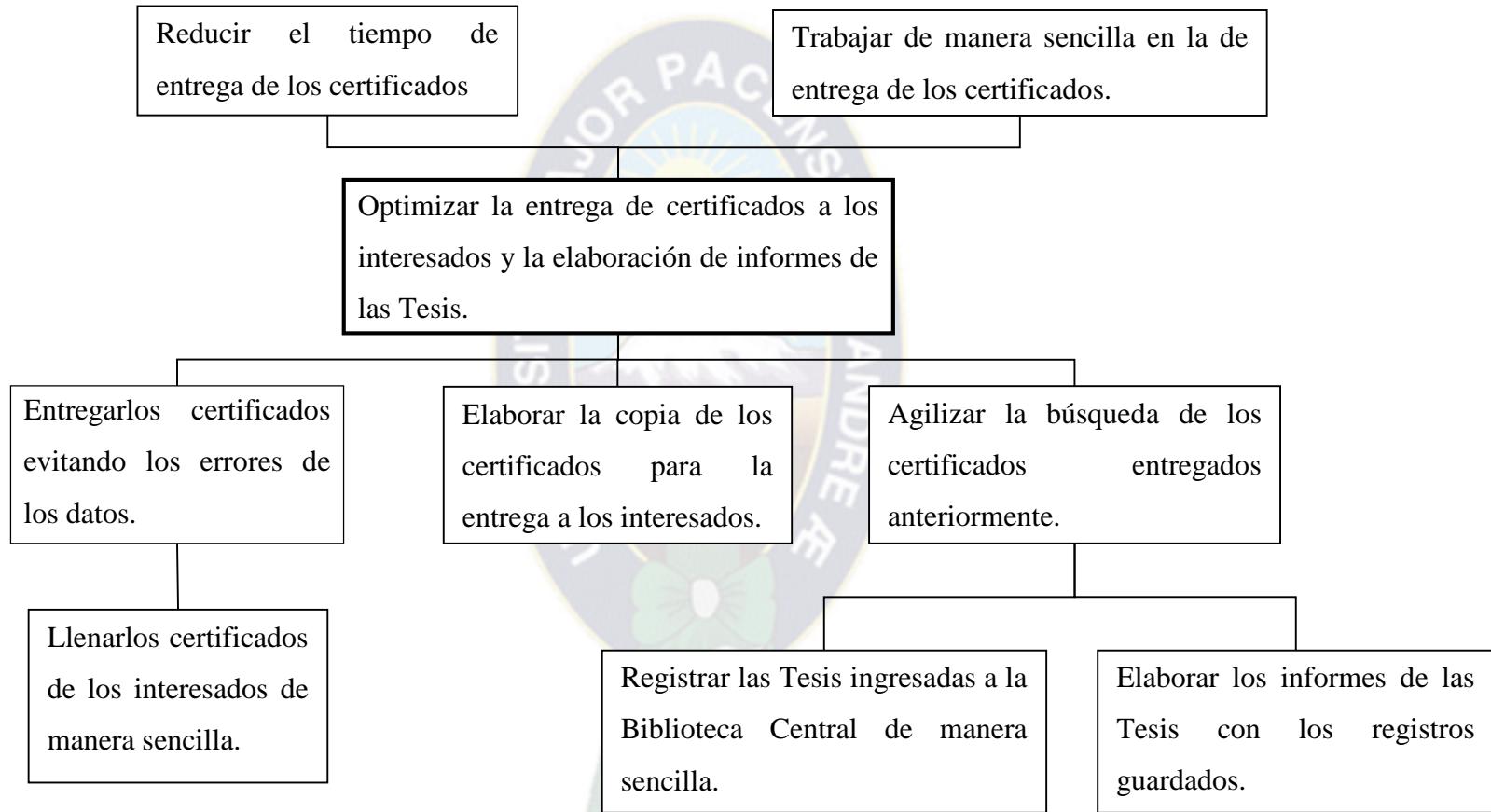
Villa Gutiérrez, M. A. (2007). Sistema para el control de ventas e inventarios de la empresa antiguo Arte Europeo S.A de C.V. Pachuca de Soto Hidalgo: Universidad Autónoma del Estado de Hidalgo.

ANEXOS

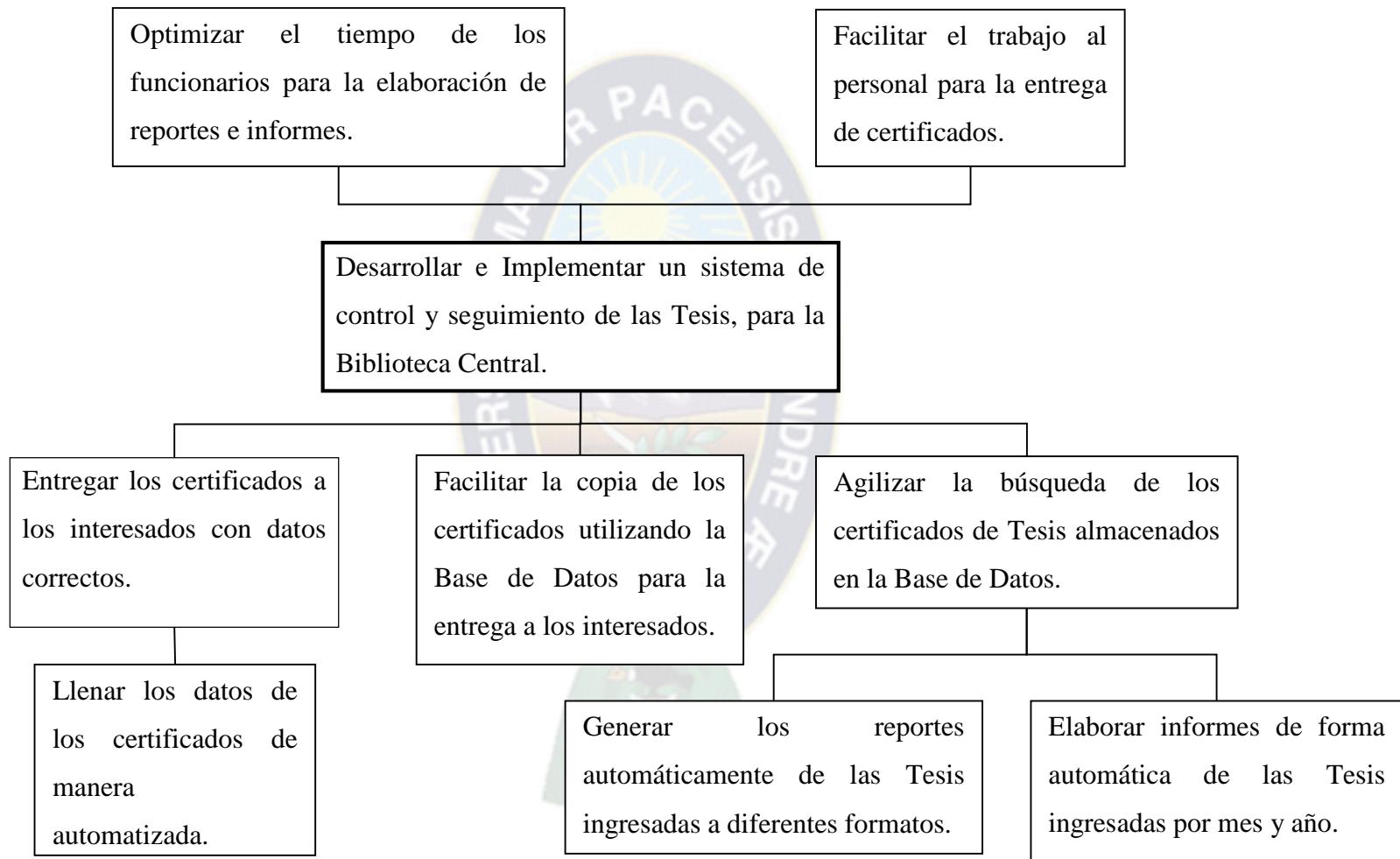
ANEXO A - ÁRBOL DE PROBLEMAS



ANEXO B - ÁRBOL DE OBJETIVOS



ANEXO C - ÁRBOL DE ALTERNATIVAS



ANEXO D - MARCO LÓGICO

Resumen Narrativo	Indicadores Objetivamente Verificable	Medios de Verificación	Supuestos
FIN Realizar un control eficiente de tesis registradas, emitiendo reportes e impresión de certificados en la Biblioteca Central.	<ul style="list-style-type: none"> • Reducir el trabajo manual del personal. • Mejorar el manejo de la información. • Reducir el tiempo en la entrega de informes y certificados. 	Entrevistas con el personal de la Biblioteca Central para la obtención de la información en cuanto a las tareas realizadas en la institución.	
PROPOSITO Diseñar e Implementar un sistema eficiente para el control de inventarios de Tesis en la Biblioteca Central.	3 semanas a partir del 16 de marzo de 2015 para implementar el sistema con informes y reportes confiables.	Entrevista realizadas al personal encargado en la recepción de las tesis.	Adaptación a los cambios en la Biblioteca Central donde se implementara el sistema.
PRODUCTO <ol style="list-style-type: none"> 1. Construir y diseñar la base de datos para el sistema. 2. Diseñar e implementar el sistema que funcionara 	<ul style="list-style-type: none"> • 2 semanas a partir de 17 de noviembre de 2014 para Modulo de registro de tesis ingresadas a la Biblioteca Central. • 2 semanas a partir del 24 de febrero de 2015 para 	<ul style="list-style-type: none"> • Carta de aceptación de la Biblioteca Central • Documentación del sistema. 	Contar con el apoyo de la Biblioteca Central para la realización del sistema.

bajo la red.	<p>Modulo de emisión de informes en tiempo real.</p> <ul style="list-style-type: none"> • 1 semana a partir de 9 de marzo de 2015 para Modulo de entrega de certificados a los estudiantes interesados. 		
ACTIVIDAD	<p>1. Realizar estudio preliminar, recopilando información actual sobre la recepción de las tesis/proyecto de Grado.</p> <p>2. Desarrollo.</p> <p>3. Documentación.</p> <p>4. Pruebas del sistema.</p> <p>5. Implementación.</p>	<ul style="list-style-type: none"> • El 27 de octubre de 2014 se realiza la recolección de los datos para el sistema. • El 28 de noviembre de 2014 obtener una PC con buenas características para instalar el servidor local. • 3 meses para el diseño y la documentación a partir de 14 de diciembre de 2014. • 3 semanas partir del 24 de febrero de 2015 para las pruebas del sistema y corrección de errores. • 1 semana de capacitación para el llenado de los registros planificado el 16 de marzo de 2015. • 16 de marzo de 2015 implementar el sistema. 	<ul style="list-style-type: none"> • Presentación de documentos del estudio preliminar. • Presentación de documentos del diseño y análisis del sistema. • Preparación de entrevista. • Manual de usuario. <ul style="list-style-type: none"> • Obtención de información útil. • Disposición de software para la realización de la programación. • Disposición del hardware adecuado para la instalación. • Contar con recursos humanos y económicos para la implementación.

ANEXO E - CRONOGRAMA DE DESARROLLO

Nro	Nombre	Comienzo	Fin	Duración	2015																			
					ENERO				FEBRERO				MARZO				ABRIL				MAYO			
					19	26	30	2	9	16	23	27	2	9	16	23	31	1	6	13	20	30	1	4
1	Análisis	15/01/2015	10/02/2015	26																				
2	Iteración 1	09/02/2015	25/02/2015	16																				
3	Iteración 2	27/02/2015	18/03/2015	19																				
4	Iteración 3	18/03/2015	27/03/2015	9																				
5	Iteración 4	27/03/2015	16/04/2015	20																				
6	Iteración 5	17/04/2015	10/05/2015	23																				

Figura 8.1: Cronograma de Desarrollo
[Elaboración propia]

ANEXO F – CERTIFICADO DE TESIS



UNIVERSIDAD BOLIVIANA
UNIVERSIDAD MAYOR DE SAN ANDRÉS
VICERRECTORADO
BIBLIOTECA CENTRAL
Telf. 2442505 • Fax: 2440232 • Casilla: 6548
<http://www.bc.umsa.bo>
La Paz - Bolivia



SERIE "A"

CERTIFICADO

Nº 005419

El suscrito Jefe de la Biblioteca Central de la Universidad Mayor de San Andrés:

CERTIFICA:

Que:

De la Carrera de:

De la Facultad de:

Hizo entrega a esta Unidad, un ejemplar de:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Es cuanto certifico para fines que convengan al interesado.

La Paz, de de 20.....

Figura 8.2: Certificado de Tesis
Fuente: Biblioteca Central