

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



PROYECTO DE GRADO

**“SISTEMA WEB DE DIGITALIZACIÓN DE LIBRO DE
ACTA DE NOTAS SIA-UMSA
CASO: DIVISIÓN DE SISTEMAS DE INFORMACIÓN Y
ESTADÍSTICA”**

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

POR: JOSÉ OCTAVIO TORREZ LAURA

TUTOR METODOLÓGICO: M. Sc. ALDO VALDEZ ALVARADO

ASESORA: Lic. CARMEN ROSA HUANCA QUISBERT

LA PAZ – BOLIVIA
2016



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



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

LICENCIA DE USO

El usuario está autorizado a:

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

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

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

DEDICATORIA

Ante todo a Dios por mostrarme el camino de la fe y ayudarme en los momentos más difíciles.

A mis padres Octavio y Lucia por el cariño y apoyo dedicado en todos estos años.

A mis Hermanos Juan Carlos, Juana Jeanette, Mary Luz por el apoyo y confianza brindado todo este tiempo.

A Claudia y a mí querida hija Carolina, que son el motivo de fuerza, cariño y superación en mi vida.

José Octavio Torrez Laura

AGRADECIMIENTOS

A Dios por haberme bendecido con salud, sabiduría y fortaleza para poder luchar con éxito en esta vida.

A él Jefe de la División de Sistemas de información y Estadística Lic. Nicanor Delgado Ecos por la confianza, colaboración brindada para el desarrollo de este proyecto y sus consejos de superación personal.

Al Lic. Porfirio Mamani, Zenón Huaygua, Waldo Berrios por la enseñanza constante en el desarrollo de proyectos, por la paciencia brindada hacia mi persona, por el conocimiento adquirido por su experiencia en el desarrollo de sistemas y por brindarme una amistad noble ante todo.

Al M. Sc. Aldo Valdez Alvarado Docente Tutor Metodológico por la paciencia y gracia que tuvo en la revisión, colaboración y recomendaciones para el desarrollo de este proyecto.

A Lic. Carmen Huanca Quisbert Docente Asesora por brindarme los consejos necesarios y la confianza depositada en mi persona para el desarrollo de este proyecto.

A mi familia por todo el apoyo moral y constructivo que coadyuvaron en el desarrollo de mi vida.

A todos mis compañeros de trabajo, de la universidad, amigos que me brindaron su apoyo incondicional en las buenas y en las malas.

Bendiciones y muchas gracias.

RESÚMEN

El desplazamiento de las distintas organizaciones hacia la Web ha traído en la actualidad una constante evolución en las aplicaciones Web.

La transferencia de información confidencial y ejecución de procesos en línea, entre otros, exigen funcionalidad, confiabilidad, usabilidad y eficiencia como algunas de las características principales de calidad.

En este sentido la División de Sistemas de información y Estadística realiza diferentes procesos para la certificación a los estudiantes cuando estos concluyen su plan de estudios a través del Sistema de Información Académica en La universidad Mayor de San Andrés.

Entre otras tareas también se tiene la organización de otros tipos de documentos, circulares, actas, informes académicos, entre otros, mismos que pasan a las diferentes unidades de la de Kardex Académico.

El presente proyecto fue desarrollado para la Universidad Mayor de San Andrés a través de la División de Sistemas de Información y Estadística, con el denominativo de “Sistema Web de Digitalización de Libro de acta de Notas SIA-UMSA (SISACTAS)”

Para el análisis diseño e implementación se aplicó la metodología SCRUM y el modelado WebML, utilizado para apoyar las actividades de diseño de aplicaciones Web.

Para el desarrollo del Software se utilizó lenguajes de programación Java, Ajax, JavaScript con la plataforma NetBeans IDE 8.1, gestor de base de datos PostgreSQL 9.4 y el uso de Framework Hibernate, Spring, Bootstrap.

ABSTRACT

The displacement of the different organizations to the Web has now brought a constant evolution in the Web applications.

The transfer of confidential information and execution of online processes, among others, demand functionality, reliability, usability and efficiency as some of the main characteristics of quality.

In this sense, the Division of Information Systems and Statistics performs different processes for the certification of students when they complete their curriculum through the Academic Information System at the Universidad Mayor de San Andrés.

Among other tasks is also the organization of other types of documents, circulars, minutes, academic reports, among others, which pass to the different units of the Kardex Academic.

The present project was developed for the Greater University of San Andrés through the Division of Information Systems and Statistics, with the denomination of "Web System of Digitalization of Book of SIA-UMSA Notes (SISACTAS)".

For the design and implementation analysis, the SCRUM methodology and WebML modeling were used to support Web application design activities.

Java, Ajax, JavaScript programming languages with the NetBeans IDE 8.1 platform, PostgreSQL 9.4 database manager and the Hibernate, Spring, Bootstrap Framework were used for the development of the Software.

ÍNDICE

CAPÍTULO I.....	1
MARCO INTRODUCTORIO	1
INTRODUCCIÓN.....	1
1.1 ANTECEDENTES.....	2
1.1.1 ANTECEDENTES INSTITUCIONALES.....	2
1.1.2 ANTECEDENTES DE PROYECTOS SIMILARES.....	5
1.2 PLANTEAMIENTO DEL PROBLEMA	5
1.2.1 PROBLEMA CENTRAL.....	5
1.2.2 PROBLEMAS SECUNDARIOS.....	6
1.3 DEFINICIÓN DE OBJETIVOS.....	7
1.3.1 OBJETIVO GENERAL.....	7
1.3.2 OBJETIVOS ESPECÍFICOS.....	7
1.4 JUSTIFICACIÓN.....	8
1.4.1 JUSTIFICACIÓN ECONÓMICA.....	8
1.4.2 JUSTIFICACIÓN SOCIAL.....	8
1.4.3 JUSTIFICACIÓN TECNOLÓGICA.....	9
1.5 ALCANCES Y LÍMITES	10
1.5.1 ALCANCES	10
1.5.2 LÍMITES	11
1.6 APORTES	11
1.6.1 APORTE PRÁCTICO.....	11
1.6.2 APORTE TEÓRICO.....	12
1.7 METODOLOGÍA	13
CAPÍTULO II.....	15
MARCO TEÓRICO.....	15
INTRODUCCIÓN.....	15
2.1 INGENIERÍA DE SOFTWARE	15
2.2 MODELO DE DESARROLLO DE SOFTWARE.....	16
2.2.1 MODELO ITERATIVO E INCREMENTAL.....	17
2.2.2 MODELO EN CASCADA.....	18
2.2.3 MODELO EN ESPIRAL.....	19

2.2.4 MANIFIESTO ÁGIL	21
2.2.4.1 VALORES.....	21
2.2.4.2 PRINCIPIOS.....	22
2.3 METODOLOGÍA DE DESARROLLO ÁGIL SCRUM	23
2.3.1 DOMINIO SIMPLE.....	24
2.3.2 DOMINIO COMPLICADO.....	24
2.3.3 DOMINIO COMPLEJO	24
2.3.4 DOMINIO CAÓTICO.....	25
2.3.5 DOMINIO DESORDENADO	25
2.3.6 PRINCIPIOS DE SCRUM.....	26
2.3.7 VALORES DE SCRUM.	26
2.3.8 ROLES DE SCRUM.....	27
2.3.8.1 PRODUCT OWNER (DUEÑO DEL PRODUCTO)	27
2.2.8.2 EQUIPO DE DESARROLLO	28
2.3.8.3 SCRUM MASTER	29
2.3.9 ELEMENTOS DE SCRUM.....	30
2.3.9.1 PRODUCT BACKLOG (BITÁCORA DEL PRODUCTO).....	30
2.3.9.2 SPRINT BACKLOG (BITÁCORA DE SPRINT).....	31
2.3.9.3. INCREMENTO FUNCIONAL	31
2.3.10 DINÁMICA (FLUJO DE TRABAJO).....	32
2.3.10.1 SPRINT (ITERACIÓN)	32
2.3.10.2 SPRINT PLANNING MEETING (PLANIFICACIÓN DEL SPRINT).....	32
2.3.10.3 SCRUM DIARIO.....	32
2.3.10.4 REVISIÓN DEL SPRINT	33
2.3.11 HISTORIAS DE USUARIO.....	33
2.3.12 MODELO DE PROCESO.....	34
2.3.13 PRE-GAME	34
2.3.14 GAME.....	35
2.3.15 POST-GAME.....	35
2.4 INGENIERÍA WEB.....	36
2.4.1 INGENIERÍA DE REQUISITOS PARA LAS APLICACIONES WEB	37
2.4.2 MARCO DE TRABAJO PARA LA INGENIERÍA WEB	38

2.4.3 FORMULACIÓN.....	38
2.4.4 ANÁLISIS	39
2.4.5 DISEÑO	39
2.4.6 PRUEBAS	40
2.5 WEBML (WEB MODELING LANGUAGE)	41
2.5.1 MODELO ESTRUCTURAL	42
2.5.2 MODELO DE COMPOSICIÓN	43
2.5.3 MODELO DE NAVEGACIÓN	45
2.5.4 MODELO DE PRESENTACIÓN	46
2.5.5 MODELO DE PERSONALIZACIÓN	46
2.6 REGLAMENTO DE KARDEX ACADÉMICO.....	47
2.6.1 MOVIMIENTO DE EXPEDIENTES Y ACTA DE NOTAS.....	48
2.6.2 SEGURIDAD DEL S.I.A. – U.M.S.A.....	49
CAPÍTULO III.....	50
MARCO APLICATIVO	50
INTRODUCCIÓN.....	50
3.1 ITERACIONES DEL SISTEMA WEB	50
3.2 PRE GAME	51
3.2.1 RECOPIACIÓN DE REQUERIMIENTOS	51
3.2.2 DEFINICIÓN DEL CRONOGRAMA DE TRABAJO.....	53
3.2.3 ANÁLISIS DE RIESGOS.....	54
3.3 GAME	55
3.3.1 PRIMERA ITERACIÓN.....	55
3.3.2 SEGUNDA ITERACIÓN.....	57
3.3.3 TERCERA ITERACIÓN	58
3.3.4 CUARTA ITERACIÓN	59
3.4 MODELADO DEL SISTEMA.....	60
3.4.1 HISTORIAS DE USUARIO	60
3.4.2 MODELO ESTRUCTURAL.....	64
3.4.3 MODELO DE USUARIO.....	65
3.4.4 MODELO DE NAVEGACIÓN	66
3.4.5 DIAGRAMA DE COMPONENTES.....	67

3.4.5.1 BÚSQUEDA DE ESTUDIANTE.....	67
3.4.5.2 ALMACENAMIENTO EN FORMATO DIGITAL.....	68
3.4.5.3 EMISIÓN DE REPORTES DE INFORMACIÓN.....	68
3.4.6 MODELO DE PRESENTACIÓN	69
3.5 POST GAME	72
3.6 MODELO DE IMPLEMENTACIÓN DEL SISTEMA	72
3.6.1 DISEÑO DE INTERFACES.....	72
CAPÍTULO IV.....	76
CALIDAD Y SEGURIDAD DEL SOFTWARE.....	76
4.1 CALIDAD DE SOFTWARE.....	76
4.2 CALIDAD DE SOFTWARE (NORMA ISO-9126)	76
4.2.1 FUNCIONALIDAD.....	77
4.2.2 FIABILIDAD.....	80
4.2.3 USABILIDAD.....	81
4.2.4 EFICIENCIA	83
4.2.5 MANTENIBILIDAD	83
4.2.6 PORTABILIDAD.....	84
4.3 CALIDAD TOTAL	86
4.4 SEGURIDAD DEL SOFTWARE	86
4.4.1 POLÍTICAS DE SEGURIDAD (USUARIOS).....	86
4.4.2 POLÍTICAS DE SEGURIDAD DE ACCESO AL SOFTWARE	87
CAPÍTULO V	88
ANÁLISIS COSTO - BENEFICIO.....	88
5.1 ANÁLISIS DE COSTOS.....	88
5.2 COCOMO.....	88
5.3 VALOR ACTUAL NETO.....	92
5.3.1 COSTO – BENEFICIO	94
5.4 TASA INTERNA DE RETORNO.....	95
CAPÍTULO VI.....	96
CONCLUSIONES Y RECOMENDACIONES	96
6.1 CONCLUSIONES.....	96
6.2 RECOMENDACIONES.....	97

ÍNDICE DE FIGURAS

Figura 1. 1 Organigrama de la División de Sistemas de Información y Estadística.....	4
Figura 2. 1 Desarrollo Iterativo e Incremental.....	17
Figura 2. 2 Estructura Modelo en Cascada.....	18
Figura 2. 3 Modelo en Espiral.....	20
Figura 2. 4 Marco Cynefin.....	23
Figura 2. 5 Ejemplo Product Backlog.....	30
Figura 2. 6 Ejemplo Sprint Backlog.....	31
Figura 2. 7 Historias de Usuario.....	33
Figura 2. 8 Proceso Scrum.....	36
Figura 2. 9 Enfoque WebML.....	42
Figura 2. 10 Modelo Estructural.....	43
Figura 2. 11 Unidad de datos.....	43
Figura 2. 12 Unidad Multidatos.....	44
Figura 2. 13 Unidad Índice.....	44
Figura 2. 14 Unidad Scroller.....	44
Figura 2. 15 Unidad Filtro.....	45
Figura 2. 16 Modelo de navegación.....	46
Figura 2. 17 Modelo de Personalización.....	47
Figura 3. 1 Etapas de Sprint.....	51
Figura 3. 2 Modelo Estructural del Software.....	65
Figura 3. 3 Contexto de Sesión.....	66
Figura 3. 4 Modelo de Navegación Control Administrativo y Estudiantil.....	67
Figura 3. 5 Modelo de Componente Búsqueda Estudiante.....	67
Figura 3. 6 Modelo de Componente Almacenamiento.....	68
Figura 3. 7 Modelo de Componente Reporte.....	68
Figura 3. 8 Modelo de Presentación Login.....	69
Figura 3. 9 Modelo de Presentación Pantalla inicial.....	69
Figura 3. 10 Modelo de presentación Datos Estudiante.....	70
Figura 3. 11 Modelo de presentación Datos Docente.....	70
Figura 3. 12 Modelo de presentación Libro de actas.....	71
Figura 3. 13 Elaboración propia Despliegue de Actas.....	71
Figura 3. 14 Modelo de presentación detalle actas.....	72
Figura 3. 15 Ingreso al Software.....	73
Figura 3. 16 Pantalla de Bienvenida.....	73
Figura 3. 17 Ventana de búsqueda ESTUDIANTE-DOCENTE.....	74
Figura 3. 18 Ventana de Listado Docente.....	74
Figura 3. 19 Registro Docente.....	75
Figura 3. 20 Panel de Reporte.....	75
Figura 4. 1 Características ISO 9126.....	77

ÍNDICE DE TABLAS

Tabla 3.1 Requerimientos Institucionales.....	53
Tabla 3.2 Análisis de Riesgo.....	55
Tabla 3.3 BackLog del Primer Sprint.....	56
Tabla 3.4 BackLog del Segundo Sprint.....	58
Tabla 3.5 BackLog del Tercer Sprint	59
Tabla 3.6 BackLog de la Cuarta Iteración.....	60
Tabla 3.7 Historia de Usuario Registro	61
Tabla 3.8 Historia de Usuario Almacenamiento	62
Tabla 3.9 Historia de Usuario Búsqueda.....	63
Tabla 3.10 Historia de Usuario Reportes	64
Tabla 4. 1 Calculo Punto Función.....	78
Tabla 4. 2 Valores de Ajuste de Complejidad	78
Tabla 4. 3 Ajuste de Complejidad Punto.....	79
Tabla 4. 4 Escala de Ajuste de Usabilidad	81
Tabla 4. 5 Evaluación de Usabilidad	82
Tabla 4. 6 Valor de Ajuste según Pressman	83
Tabla 4. 7 Factores de Ajuste de Mantenibilidad.....	84
Tabla 4. 8 Resultado calidad Total Norma ISO-9126.....	86
Tabla 5. 1 Modelo COCOMO	89
Tabla 5. 2 Conversión de Puntos de Función	90
Tabla 5. 3 Calculo de VAN	93
Tabla 5. 4 Interpretación del VAN.....	94
Tabla 5. 5 Interpretación del TIR.....	95

CAPÍTULO I

MARCO INTRODUCTORIO

INTRODUCCIÓN

En el proceso de administrar una organización se generan una serie de documentos, cuya conservación es de vital importancia, los mismos que son testimonio y garantía de una institución, para tal efecto los mismos deben estar organizados debidamente desde el momento que se genera la información documental hasta ser archivada y depositada.

El desarrollo de software de las distintas organizaciones están orientadas hacia la Web lo que ha traído en la actualidad una constante evolución en las aplicaciones Web. La transferencia de información confidencial y ejecución de procesos en línea, los cuales requiere funcionalidad, confiabilidad, usabilidad y eficiencia como algunas de las características principales.

Los sistemas de información están presentes en casi todas las áreas de las organizaciones y ampliamente relacionadas con el desarrollo de las actividades que implica su ciclo productivo y a su vez relacionadas con los problemas de la generación de documentación masiva.

Por esta razón el uso de herramientas web para el desarrollo de sistemas en organizaciones se ha vuelto en un recurso indispensable, tanto para la imagen de la institución puesto que muchos proyectos presentan información estática y poco amigables, como para el usuario pudiendo interactuar dinámicamente con los sistemas, mientras que prestan servicios con mayor comodidad, versatilidad y seguridad a la hora de poder realizar un servicio.

Tales son los beneficios que usuarios finales ya no tienen la necesidad de acudir a las instituciones directamente, basta con tener acceso al servidor de datos de la institución

para poder obtener los servicios requeridos por el usuario con cualquier dispositivo que tenga acceso a la red, de tal forma que facilita la obtención de información al instante.

En consecuencia la Universidad Mayor de San Andrés (U.M.S.A.) realiza diferentes procesos para resguardar las notas de los estudiantes cuando estos concluyen su periodo académico ya sea este semestral o anual e incluso cursos de temporada. Son los Docentes que dan valor agregado con la firma correspondiente, presentado las notas de los inscritos a su asignatura a la unidad de Kardex de la Carrera correspondiente, los cuales realizan la recepción de las notas de los docentes y estas notas son almacenadas en el sistema S.S.A.¹ que dará curso a los estudiantes regúlales a avanzar a los niveles siguientes, es así que los documentos que los docentes presenta con las notas de los estudiantes y con la firma correspondiente conformara el Libro de acta de notas de una gestión académica dada de materias en actas de acuerdo al plan de estudios actual.

El proyecto tiene como objeto presentar un Sistema de información, registro y organización de estos archivos con la implementación de la digitalización de los mismos, que resguarde de manera digital y que permita a los usuarios hacer búsquedas de manera ágil y oportuna según estudiante, docente, asignatura paralelos, mismos documentos que son de vital importancia en el continuo desenvolvimiento de parte de los proceso de enseñanza aprendizaje, el mismo que reflejara la veracidad de las notas obtenidas por los estudiantes.

1.1 ANTECEDENTES

1.1.2 ANTECEDENTES INSTITUCIONALES

La División de Sistemas de Información y Estadísticas (DSIE) es una unidad que presta servicios para la Universidad Mayor de San Andrés (UMSA) desde la década de los 80 como referente en la historia Informática. Junto al Centro de Cálculo dependiente de la facultad de ingeniería, desde entonces conocido como Centro de Procesamiento de Datos e

¹ Sistema de Seguimiento Académico

Información (CPDI), fueron las unidades responsables del desarrollo e implementación de sistemas informáticos permitiendo la sistematización, manejo ágil y oportuno de la información académico financiera de la universidad.

DSIE encara plenamente el desafío tecnológico y bajo los tres estamentos de la estructura universitaria dispongan de acceso oportuno a la información, mediante el acceso al Sistema de Información Académica de la Universidad Mayor De San Andrés (SIA-UMSA), cuyo principal propósito persigue la administración y control de la información académica, ofreciendo a la comunidad universitaria informes y datos estadísticos para la toma de decisiones.

Durante la gestión 2013-2014 el SIA-UMSA se implementó en nueve de las trece facultades, con total aceptación, confirmando en este tiempo resultado satisfactorios, dando conformidad y actualmente cuenta con nuevos registros de 115.166 estudiantes.

La División de Sistemas de Información y Estadística dependiente del Departamento de Tecnología de Información y Comunicación (DTIC) de la Universidad Mayor de San Andrés, cuenta con una sola base de datos de 281.741 estudiantes matriculados en los niveles de pregrado y de postgrado, administrada desde el año 1990 con datos parametrizados para la emisión de informes en línea [DTIC-Memoria institucional, 2013-2014].

La misión del DSIE es brindar información de los estudiantes en línea referentes a los datos de identificación personal, rendimiento académico, número de estudiantes, por carreras y sedes dentro de la universidad además de procesar y difundir la información estadística de la Universidad Mayor de San Andrés [DTIC-Memoria institucional, 2013-2014].

Para una mejor comprensión de las áreas que conforman, el Organigrama de la División de Sistemas de Información y Estadística a ser Aprobado tiene la siguiente estructura orgánica (ver Figura 1.1).

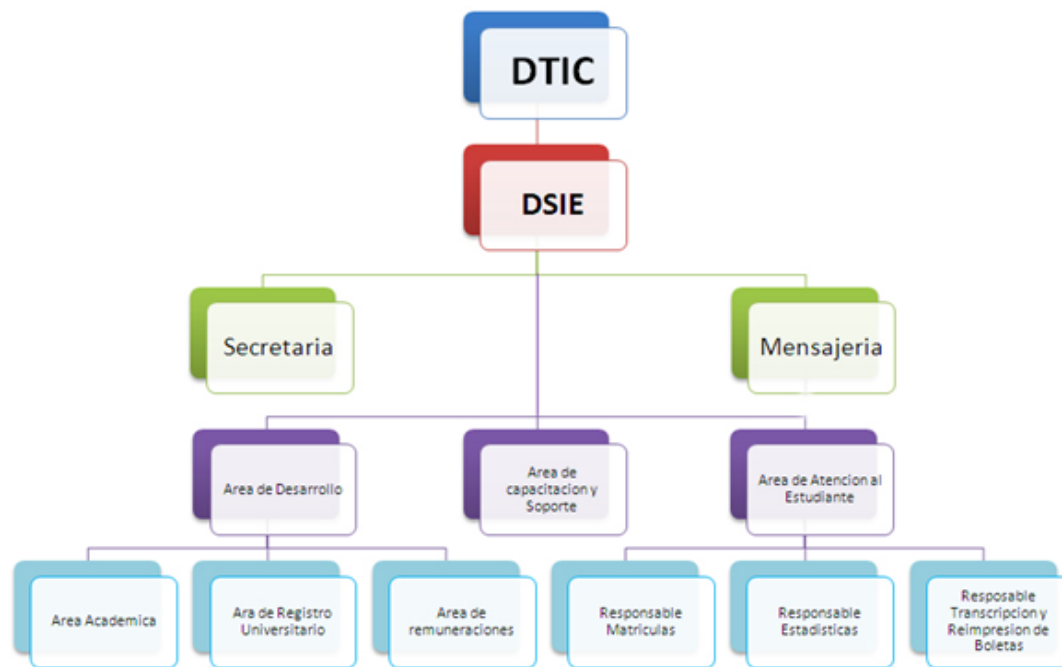


Figura 1. 1 Organigrama de la División de Sistemas de Información y Estadística.
Fuente: [Portal web DSIE, 2015]

El proyecto SIA-UMSA cuenta con dos Subsistemas consolidados y en pleno funcionamiento los cuales son:

Sistema de Registro Universitario (SRU) el cual captura la Información de los estudiantes desde el ingreso a la Universidad Mayor de San Andrés (UMSA), el cual hace un control de pago de matrículas de estudiantes de pregrado y postgrado, emisión de reportes de certificación de matrícula, estadísticas entre otras funcionalidades.

Sistema de Seguimiento Académico (SSA) el cual está en funcionamiento en nueve de las trece facultades que comprende la Universidad Mayor de San Andrés, el sistema permite hacer seguimiento académico de los estudiantes, brindando reportes de notas, historial académico, gestión de inscripción, reporte de boletas de inscripción, convalidaciones de materias, entre otras funcionalidades.

1.1.3 ANTECEDENTES DE PROYECTOS SIMILARES

Frente al incremento del volumen de documentos e información en Instituciones de prestigio surge la necesidad de tener un buen control y seguimiento de los documentos académicos y administrativos.

Tomando en cuenta Proyectos similares que fueron desarrollados con similitud y anterioridad, podemos dar a conocer algunos Proyectos que fueron desarrollados y proporcionados a la Biblioteca de la carrera de Informática de la Universidad Mayor de San Andrés que fueron consultadas como referencia para el desarrollo de la documentación para el proyecto que se propone, entre estas tenemos a las siguientes:

“Sistema de Registro y Administración de Archivos”, Mary Isabel Rosas Mamani, 2013, Universidad Mayor de San Andrés, carrera de Informática, Sistema que mejora de forma eficaz el control y organización de los archivos, en base a un seguimiento de estado de la documentación mediante reportes y escaneado de documentación física.

“Sistema de Envío y Recepción de Documentos Para la Fuerza Aérea Boliviana”, Gabriela Surco Aguilar, 2009, Universidad Mayor de San Andrés, Carrera de Informática, Sistema que Organiza, Almacena, y Controla el flujo de correspondencia dando prioridad de atención a los documentos pendientes, con el uso de Sistemas Manejadores de Procesos de Negocio, cuyo propósito es acercar personas, procesos y maquinas, con el objeto de reducir tiempos y acelerar los flujos de trabajo en equipo.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 PROBLEMA CENTRAL

Las Unidades de Kardex de la Universidad Mayor de San Andrés que trabajan con el Proyecto SIA-UMSA realiza trabajos Académicos y Administrativos, entre éstas tenemos a Informes "Conclusión de Estudios, Títulos Académicos, Circulares, Conservar los Libros de Actas de notas, entre otros", que involucran la actividad académica de estudiantes de

acuerdo a Pensum de su carrera o en su defecto algún administrativo necesita uno de estos informes.

Los Docentes designados para impartir una asignatura son los directos responsables de validar las calificaciones obtenida por los estudiantes, quienes registran notas en el Sistema de Seguimiento Académico S.S.A con su cuenta Personal, o en formato físico en Kardex de la carrera para que el kardex proceda a la migración, el Docente está obligado a realizar la verificación correspondiente antes de firmar la acta, esto debe realizarlo en los plazos establecidos por la carrera. Es así que se conforman las actas de calificaciones de estudiantes.

En estos últimos tiempos existen denuncias acerca de la enmienda de calificaciones y/o alteraciones de las mismas, incluso de falsificaciones que acarrear problemas legales de supuestos profesionales que falsificaron su título académico, pero para detectarlos se debe realizar una búsqueda exhaustiva que en casos las unidades de Kardex cambian de personal en el transcurso del tiempo no se agiliza tal búsqueda, que posteriormente será verificado las notas de los denunciados. Y en otros tiempos existían perdidas de expedientes y de notas de los estudiantes que fueron perjudicados en el tiempo, por la inexistencia de respaldo que corrobore.

Es en ese sentido que surge la siguiente interrogante:

¿De qué manera se podrá resguardar la información de las actas de notas, de manera que permita realizar búsquedas oportunas, precisas e inmediatas además de mostrar las actas de nota para verificación en las Carreras que trabajan en el proyecto SIA-UMSA?

1.2.2 PROBLEMAS SECUNDARIOS

Como consecuencia a lo descrito anteriormente se pueden identificar los siguientes problemas secundarios:

- Demora en la búsqueda física de los libros de acta de notas, lo que causa retraso en la verificación correspondiente.
- Deterioro del material que fue elaborado en gestiones anteriores, lo que causa incongruencias de la información obtenida.
- Perdida de la información de las actas de notas, por falta de organización.
- Espacio reducido para salvaguardar la documentación física generada lo que causa confusión.
- Control en la entrega y registro oportuno de las calificaciones de docentes, lo que deriva en el retraso de la elaboración de los libros de acta de notas.
- No se puede proporcionar actas de notas por asignaturas de docentes, lo que conlleva a no tener un control por docente.

1.3 DEFINICIÓN DE OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar un SISTEMA WEB DE DIGITALIZACIÓN DE LIBRO DE ACTA DE NOTAS SIA-UMSA para la División de Sistemas de Información y Estadística mediante nuevas plataformas tecnológicas, permitiendo almacenar documentación de resguardo en formato digital, búsqueda oportuna, precisa e inmediata de tal forma que permita verificar la veracidad de las notas almacenadas por las carreras que trabajan con el proyecto SIA-UMSA.

1.3.2 OBJETIVOS ESPECÍFICOS

Puntualmente puede describirse los siguientes objetivos específicos:

- Recuperar la documentación describiendo hora y fecha de almacenamiento.
- Digitalizar la documentación bajo la premisa de resguardo y disponibilidad de consulta.
- Controlar la información por orden según pensum académico.

- Almacenar en el sistema la documentación en formato digital.
- Emitir historial de documentación almacenada en el sistema.
- Elaborar reportes de carácter descriptivo por docente.

1.4 JUSTIFICACIÓN

1.4.1 JUSTIFICACIÓN ECONÓMICA

Con el nuevo software, la sistematización de los procesos que actualmente son realizados de forma manual, se reducirá en relación al tiempo y por ende los costos que involucran en cuanto a papeleos, envío manual a otras unidades o verificación de autoridades dentro de las carreras, entre otros se minimizan considerablemente ya que estos documentos serán respaldados de forma digital.

En relación al costo por el desarrollo e implementación del nuevo sistema, estos serán mínimos puesto que la propuesta hecha anteriormente se enfoca en trabajar con herramientas libres y certificadas.

Prácticamente todas las unidades de Kardex administrarán eficientemente la información que manipulan en el sistema, dando lugar a priorizar otras funciones académicas y administrativas que anteriormente estaban rezagadas.

En cuanto a requerimiento de hardware y software, las facultades cuentan con equipos y tecnología de computación necesaria y como se mencionó anteriormente, las plataformas a ser utilizadas para el desarrollo son de libre uso y además de ser una institución con fines educativos.

1.4.2 JUSTIFICACIÓN SOCIAL

El Proyecto en desarrollo beneficiará a las personas de las unidades de Kardex de la Universidad Mayor de San Andrés las mismas que contarán con un sistema que

coadyuvaran con el buen desempeño de sus labores diarias, contarán con información precisa, oportuna e inmediata cuando así lo requieran.

Así mismo se beneficiará la población estudiantil los mismos que necesitan una pronta respuesta a trámites de convalidación de notas o referirnos a estudiantes antiguos aleguen pérdida de notas, elevación de informes para fines jurídicos según requerimientos.

En cuanto a imagen se beneficiará la Universidad Mayor de San Andrés a través de la División de Sistemas de Información y Estadística, Facultades y carreras que trabajan con el proyecto SIA-UMSA, por la tecnología de punta que se utilizará para el desarrollo del sistema.

1.4.3 JUSTIFICACIÓN TECNOLÓGICA

La División de Sistemas de Información y Estadística cuenta con la tecnología necesaria suficiente y actualizada en cuanto a hardware y software se refiere.

La División cuenta con tres computadores de escritorio con las siguientes características básicas para llevar a cabo seguimiento y mantenimiento del presente proyecto a desarrollar:

Procesador: Intel(R) Core(TM) i5 3.2 Ghz

Memoria: 4 GB.

Sistema Operativo: Windows 8 Enterprise 64 bits

Cuenta con tres Scanners HP Scanjet G3110.

Tipo de escáner: Cama plana

Resolución de escaneo, óptica: Hasta 4800 ppp

Profundidad en bits: 48 bits

Niveles de escala de grises: 256

Formato del archivo de digitalización: PDF que permite búsquedas, TIFF, TIFF comprimido, JPG, BMP, PNG, FPX, GIF, PCX, RTF, TXT, HTML.

Así mismo las unidades de Kardex de las diferentes facultades que trabajan con el proyecto SIA-UMSA cuentan con la tecnología requerida para el funcionamiento y buen desarrollo.

Se puede evidenciar que cuentan con computadores de escritorio con las siguientes características básicas:

Procesador: Intel Core 2 Duo Processor E8400 3.1 Ghz.

Memoria: 4 GB.

Sistema Operativo: Windows 7 Enterprise 64 bits

Gracias al Departamento de tecnologías de Información y Comunicación (DTIC) de la UMSA, las respectivas unidades cuentan con el servicio de internet con una velocidad de 15mbps y el cableado estructurado respectivo.

Por todo lo anteriormente señalado se justifica el desarrollo del presente proyecto.

1.5 ALCANCES Y LÍMITES

1.5.1 ALCANCES

De acuerdo a las funcionalidades requeridas para el sistema descrito en la especificación de los requerimientos a coordinación con la jefatura de la División de Sistemas de Información y Estadística.

Las principales tareas se resumen en los siguientes módulos:

- Módulo de seguimiento documental de cada libro de acta de notas
- Módulo de acceso de usuarios autorizados.
- Módulo de emisión de reportes de información.
- Módulo de consultas en línea.

- Módulo de almacenamiento de archivos.
- Módulo de tratamiento de imágenes y optimización de memoria en almacenamiento.

1.5.2 LÍMITES

Los límites del proyecto según un análisis de requerimientos en relación a él almacenamiento de la información:

- No gestiona las notas de los estudiantes.
- Se relaciona al área Académica.
- No registra, no elimina ni modifica datos de los estudiantes.
- No gestiona los datos del Sistema Seguimiento Académico S.S.A.

1.6 APORTES

1.6.1 APORTE PRÁCTICO

La implementación del Software proporcionará ciertos aportes a la Universidad Mayor de San Andrés a través de la División de Sistemas de Información y Estadística más propiamente a las diferentes unidades de Kardex, entre éstas podemos mencionar:

Construir un software parametrizable en el tiempo.

Contribuir con los Kardixtas en agilizar Búsquedas de Actas para la formulación de informes según requerimientos.

Reducir espacios innecesarios para resguardar la documentación elaborada.

Uso de Nuevas herramientas de desarrollo ágil, como ser el Framework para JEE, Spring, además de ORM Hibernate entre otros.

Integración a Sistemas en Funcionamiento dentro de la Universidad Mayor de San Andrés, como ser el sistema de Registro Universitario y el Sistema de Seguimiento Académico.

1.6.2 APORTE TEÓRICO

El Software será desarrollado bajo la Metodología Ágil SCRUM, puesto que se acomoda al proceso de desarrollo del proyecto, surge como una manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en los siguientes principios ágiles:

- Colaboración estrecha con el cliente.
- Predisposición y respuesta al cambio.
- Desarrollo incremental con entregas frecuentes de funcionalidad.
- Comunicación verbal directa.
- Simplicidad, solo los artefactos necesarios.
- Motivación, compromiso y responsabilidad del equipo por la auto-gestión y auto-organización.

Scrum es una de las metodologías ágiles más populares del momento, se caracteriza por su simplicidad, seguimiento continuo y efectividad, motivo por el cual es utilizada por algunas de las empresas como: Google, Sun, Microsoft, Siemens, entre otros.

Para el modelado del Sistema Web se usará WebML (Lenguaje de Modelado Web), es un robusto y fuerte sistema que nos ayudará en el orden y mantenimiento del ciclo de vida del desarrollo del Software.

Hoy en día, las páginas Web son portales poderosos, que llevan empresas, gobiernos e industrias, a la sociedad y los consumidores, por lo cual, su complejidad nos obliga a tomar las mejores herramientas para garantizar su éxito.

Una de estas herramientas es WebML que nos permitirá saber que va a suceder, cómo y cuándo predecir el funcionamiento del sistema web. De esta manera poder obtener mejor calidad en los datos, evitar procesos redundantes y lograr una amplia visión sistémica que permita tener claridad y un completo manejo de la información.

1.7 METODOLOGÍA

El presente proyecto se sostiene bajo el método científico, nos basaremos en la investigación de tipo exploratorio, puesto que se emplea la búsqueda de conocimiento sobre el problema, sus alternativas de decisión y las variables que se deben considerar, para lograr un conocimiento previo en el cual debemos apoyarnos.

Por otra parte se concluirá la investigación de tipo Descriptivo, que comprenderá la recolección de datos, en relación a los beneficios que conlleva la elaboración del producto software que pueda proporcionar a las unidades de Kardex del proyecto SIA-UMSA.

La metodología ágil a aplicarse será SCRUM, la cual está diseñada para entregas de software que los clientes requieren en momentos precisos.

Scrum motiva a los desarrolladores a responder a requerimientos cambiantes de los clientes, aun en fases tardías del ciclo de vida del desarrollo del software.

Scrum se basa en la teoría del control de procesos empíricos o empirismo. Lo cual asegura que el conocimiento procede de la experiencia de tomar decisiones basándose en la información que se conoce.

Además la metodología emplea una óptica iterativa e incremental para optimizar, predecir y tomar control de futuros riesgos.

Son tres los preceptos que dan soporte a la implementación de control de procesos empíricos, los cuales son: transparencia, inspección y adaptación.

Para el lenguaje de modelado, está basado en la experiencia de WebML, puesto que las pequeñas ventanas de información que comprendían hace bastante tiempo atrás ya no están en vigencia y actualmente, gracias al avance tecnológico los motores web, utilizan mecanismos de información multiusuario para acceder a sus sitios web favoritos, como google, Facebook, ebay, entre otros.

En este caso se perfilan modelos de diseño web adaptados a las características básicas de internet en la actualidad, es el caso de Web Modeling Language o lenguaje de modelado web.

En cuanto a herramientas de desarrollo de software se establece:

- Gestor de base de datos PostgreSQL 9.4.
- Plataforma de desarrollo Netbeans IDE 8.0
- Frameworks spring, Hibernate, Bootstrap, Ajax.
- Lenguaje de Programación Java, JavaScript.

La evaluación del Producto software no es tarea sencilla, es difícil de considerar todas las características y los atributos deseables y obligatorios de una aplicación o sitio web si no se encuentra con un modelo de calidad que permita su evaluación, es así que se aplicara el estándar ISO 9126, es decir que la calidad está definida por un alto nivel de abstracción por las características denominadas usabilidad, funcionalidad, confiabilidad, eficiencia, mantenibilidad y portabilidad.

CAPÍTULO II

MARCO TEÓRICO

INTRODUCCIÓN

Los Sistemas de Información, para su desarrollo deben basarse en métodos y técnicas para llevar a cabo su propósito, es en ese sentido que en este capítulo se presenta la teoría necesaria para elaborar el presente proyecto de grado.

2.1 INGENIERÍA DE SOFTWARE

La ingeniería de Software es una disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad, que resuelven temas de todo tipo. Esta definición permite incluir dentro de la ingeniería de software a un gran número de áreas en la informática, desde desarrollos de sistemas operativos, construcción de compiladores hasta los nuevos desarrollos de software para aplicaciones basadas en aplicaciones enriquecidas en la web. [Pressman, 2002]

Los métodos de la ingeniería de software proporcionan la experiencia técnica necesaria para elaborar software, que incluye un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo.

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semi-automatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información obtenida pueda ser utilizada por otros actores, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software; a continuación se detallan las herramientas más importantes: [Van Vliet, 2002].

- **Requerimientos de Software.** Un requerimiento se define como una exigencia que debe ser cumplida para dar solución a un problema específico. Existen diferentes

áreas de análisis, como ser la especificación de requerimientos, análisis, validación, clasificación, negociación, etc.

- **Diseño del Software.** Es el proceso en el cual se define la arquitectura, componentes, interfaces y otras características relativas al desarrollo del software mismo.
- **Construcción del Software.** Referida a la creación en detalle del Software a través de la combinación de las diferentes herramientas de codificación, verificación, pruebas de unidad, pruebas de integración y depuración.
- **Pruebas del Software.** Proceso de verificación dinámica del comportamiento del Software ante un conjunto limitado de casos de prueba.
- **Mantenimiento del Software.** Estas actividades de mantenimiento comienzan generalmente cuando el Software entra en la etapa de pruebas de funcionalidad con datos reales, pruebas que son realizadas por parte de los usuarios finales con el propósito de que estos puedan adaptarse a los cambios, así mismo puedan sugerir cambios para su posterior adaptabilidad.

2.2 MODELO DE DESARROLLO DE SOFTWARE

El proceso de desarrollo de software permite, mediante el uso del modelo de gestión, ordenar las actividades y realizar una planificación de flujo para la construcción de software.

La gestión ágil de proyectos tiene como objetivo dar garantías a las demandas principales de la industria actual en relación a los valores, reducción en relación del tiempo de desarrollo, agilidad, flexibilidad y fiabilidad.

Definamos algunos conceptos ya mencionados [Palacio & Ruata, 2011]

- **Agilidad.** Es la capacidad de producir partes completas del producto en periodos breves de tiempo.

- **Flexibilidad.** Es la forma de adaptar el desarrollo de software a las características del proyecto y a la evolución de los requisitos.

La Industria del Software es la primera en adoptar este tipo de metodología y muchos de sus profesionales documentan y propagan las formas particulares de implementación de los principios de la agilidad en sus equipos de trabajo.

2.2.1 MODELO ITERATIVO E INCREMENTAL

El desarrollo iterativo e incremental es una parte esencial de SCRUM, RUP, DSDM, XP, principales metodologías que generalmente son usadas por desarrolladores en el marco de trabajo de desarrollo ágil. [Kendall & Kendall, 1995]

Para reflejar gráficamente la Figura 2.1 nos presenta este proceso.

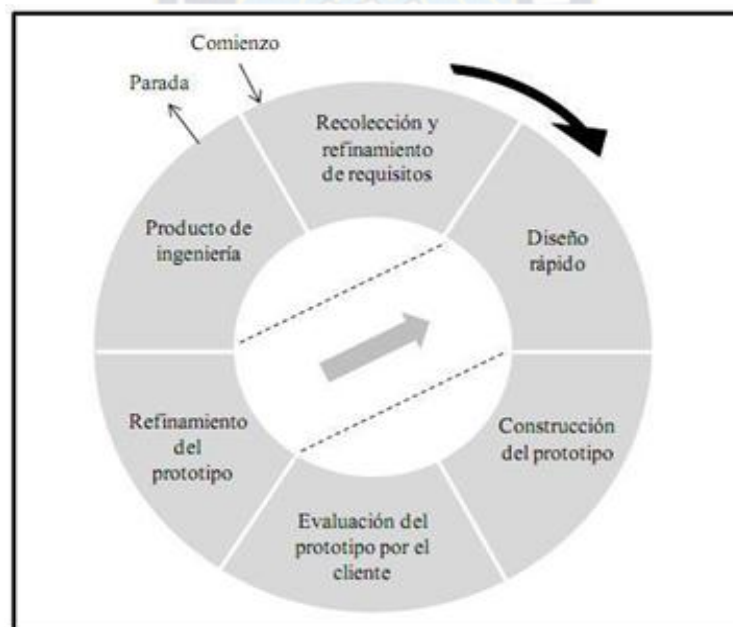


Figura 2. 1 Desarrollo Iterativo e Incremental

Fuente: [Kendall & Kendall, 1995]

El desarrollo incremental es una estrategia programada con etapas, en la cual las diferentes partes o módulos del sistema pueden ser desarrollados en diferentes momentos o diferentes velocidades y se integran a medida que se completan.

2.2.2 MODELO EN CASCADA

Este modelo admite la posibilidad de hacer iteraciones, es decir, durante las modificaciones que se hacen en el mantenimiento se puede ver por ejemplo la necesidad de cambiar algo en el diseño, lo cual significa que se harán los cambios necesarios en la codificación.

Después de cada etapa se realiza una revisión para comprobar si se puede pasar a la siguiente etapa. [Pressman, 2002]

Como se muestra en la Figura 2.2 acerca del modelo en cascada.

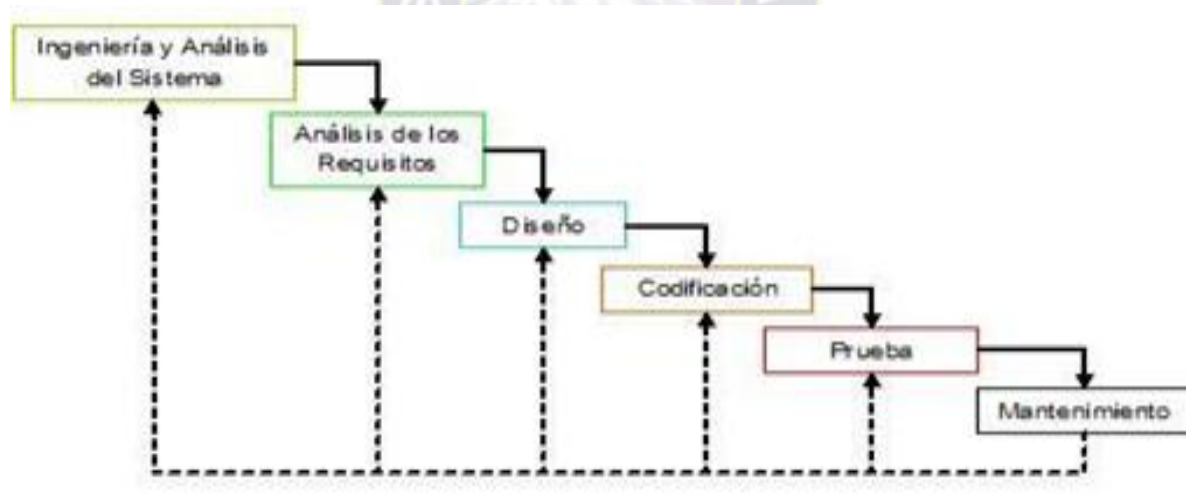


Figura 2. 2 Estructura Modelo en Cascada
Fuente: [Pressman, 2002]

La estructura del modelo en cascada es el más conocido y está basado en el ciclo convencional de la ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades: [Pressman, 2002]

- **Ingeniería y análisis del sistema.** Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignado algún subconjunto de esos requisitos al software.
- **Análisis de los requisitos del Software.** El proceso de recopilación de requisitos se basa en el desarrollo del software, el mismo que debe comprender al ámbito de la información que se maneja, la función que se desempeña, el rendimiento y las interfaces requeridas.
- **Diseño.** El diseño del software se enfoca en cuatro atributos distintos del programa; La estructura de datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz.
- **Codificación.** El diseño debe traducirse en lenguaje semántico que será interpretado por el computador.
- **Prueba.** Las pruebas se centran en la lógica interna del software y en las funciones externas (backend, frontend), realizando pruebas que aseguren que la entrada definida produce los resultados esperados.
- **Mantenimiento.** El software deberá de sufrir cambios luego de su implementación final, los cambios ocurrirán debido a que se hayan encontrado errores, que el software deba adaptarse a cambios en el entorno externo (Sistema operativo, Navegadores, etc.), o debido a que el usuario final requerirá ciertas modificaciones funcionales de carácter específico o de rendimiento.

2.2.3 MODELO EN ESPIRAL

El desarrollo en espiral es un modelo de ciclo de vida del software definido por las actividades que se conforman en una especie de espiral, en la que cada bucle o iteración

representa un conjunto de actividades. Las mismas que no están fijadas a prioridad, si no que se eligen en función al análisis de riesgo, comenzando por el bucle interior.

En cada iteración cuenta los siguientes elementos: [Kendall & Kendall, 1995]

- **Los objetivos.** Necesidad que debe cubrir el producto software.
- **Alternativas.** La diferentes formas de conseguir el objetivo de manera exitosa, desde diferentes puntos de vista; experiencia personal, requisitos a cumplir, formas de gestión del sistema, riesgos asumidos con cada alternativa, entre otros.
- **Desarrollar y verificar.** Programar el software y probar su funcionamiento.

Existen cuatro actividades o tareas para cada ciclo que se ven en la Figura 2.3:

- Determinar Objetivos.
- Análisis de Riesgo.
- Planificación.
- Desarrollar y Probar.



Figura 2. 3 Modelo en Espiral
Fuente: [Kendall & Kendall, 1995]

2.2.4 MANIFIESTO ÁGIL

El Manifiesto ágil se compone de 4 valores y 12 principios. [Alaimo, 2013]

2.2.4.1 VALORES

Pueden describirse los siguientes valores: [Alaimo, 2013]

- Valorar a las Personas y las Iteraciones entre ellas por sobre los procesos y las herramientas.

Las personas son el principal factor de éxito de un proyecto de software. Es más importante construir un buen equipo que construir el contexto. Muchas veces se comete el error de construir primero el entorno de trabajo y separar que el equipo se adapte automáticamente.

- Valorar el software funcionando por sobre la documentación detallada.

La regla a seguir es “no producir documentos a menos que sean necesario de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo esencial.

- Valorar la colaboración con el cliente por sobre la negociación de contratos.

Se propone que haya una interacción constante con el cliente y el equipo de desarrollo. Esta mutua colaboración será la que pondrá en marcha el proyecto y asegure su éxito.

- Valorar las respuestas a los cambios por sobre el seguimiento estricto de los planes.

Las habilidades responden a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, entre otros) determina también su éxito o fracaso. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta a cualquier equipo.

2.2.4.2 PRINCIPIOS

Los valores anteriores son los pilares sobre los cuales se construyen los doce principios del manifiesto ágil. De estos doce principios, los dos primeros son generales y resumen de gran parte del espíritu ágil de desarrollo de software, mientras que los siguientes son más específicos y los mismos están orientados a equipos de desarrollo: [Alaimo, 2013]

- Nuestra mayor prioridad es satisfacer al cliente a través de entregas tempranas y frecuentes de software con valor.
- Aceptar el cambio incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan los cambios para darles al cliente ventajas competitivas.
- Entregar software en funcionamiento de forma frecuente, desde un par de semanas a un par de meses, prefiriendo el periodo de tiempo más corto.
- Expertos del negocio y desarrolladores deben trabajar juntos diariamente durante la ejecución del proyecto.
- Construir proyectos entorno a personas motivadas, generándoles el ambiente necesario, atendiendo necesidades confiando en que ellos van a poder hacer el trabajo.
- La manera más eficiente y efectiva de compartir la información dentro de un equipo de desarrollo es la conversación cara a cara.
- El software funcionando es la principal métrica de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los sponsors, desarrolladores y usuarios deben poder mantener un ritmo constante indefinidamente.

- La atención continua a la excelencia técnica y buenos diseños incrementan la agilidad.
- La simplicidad (El arte de maximizar la cantidad de trabajo no realizado) es esencial.
- Las mejores arquitecturas, requerimientos y diseños emergen de equipos auto-organizados.
- A intervalos regulares, el equipo reflexiona acerca de cómo convertirse en más efectivos, luego mejora y ajusta su comportamiento adecuadamente.

2.3 METODOLOGÍA DE DESARROLLO ÁGIL SCRUM

Veamos el contexto para el cual Scrum es más eficiente. Podemos utilizar el marco Cynefin para poder comprender las diferentes situaciones en las que nos podemos encontrar operando y cuál es, según este enfoque, la manera más eficiente de responder a cada una de ellas. El marco de Cynefin compara las características de cinco dominios de complejidad, diferentes: simple, complicado, complejo, caótico y desordenado, ver Figura 2.4. [Alaimo, 2013]

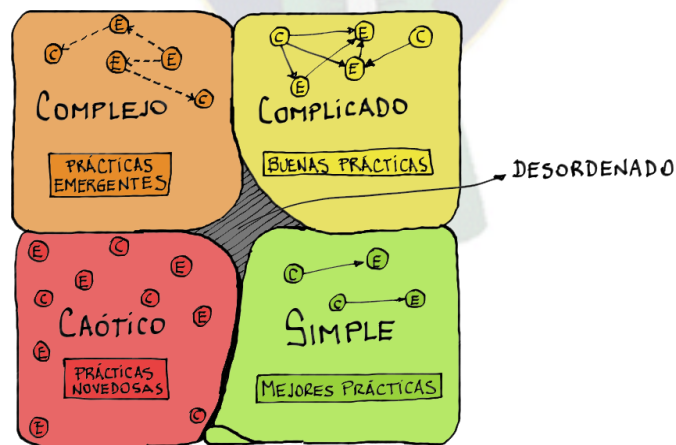


Figura 2. 4 Marco Cynefin
Fuente: [Alaimo, 2013]

2.3.1 DOMINIO SIMPLE

En este dominio se opera con problemáticas simples. Es muy fácil identificar las causas y efectos. Por lo general, la propuesta correcta es clara, conocida por todos e indiscutible. En este dominio existen las mejores prácticas, soluciones conocidas para problemas conocidos. Los procesos más eficientes en este dominio son aquellos que especifican una serie lógica de pasos y se ejecutan de manera repetitiva, una y otra vez. Ejemplos de este dominio son la construcción en serie de un mismo producto, la instalación en muchos clientes de un mismo sistema. Si bien Scrum puede funcionar en este contexto, los procesos compuestos por pasos bien definidos son mucho más eficientes. [Alaimo, 2013]

2.3.2 DOMINIO COMPLICADO

En este dominio encontramos problemas complejos, buenas prácticas y perfiles expertos. Para encarar puede haber infinidad de formas correctas de soluciones para la misma problemática, pero se requiere la intervención de expertos para poder identificarlas. Si bien Scrum podría emplearse, no necesariamente sea la forma más eficiente de resolver estas situaciones, donde funcionara mejor un conjunto de expertos en la materia que revelen la situación, investiguen diferentes alternativas y planteen la solución en base a las buenas prácticas. Una práctica habitual de este dominio es el mantenimiento de sistemas y soporte técnico. [Alaimo, 2013]

2.3.3 DOMINIO COMPLEJO

Cuando nos enfrentamos a problemas complejos, los resultados se vuelven más impredecibles. No existen ni mejores ni buenas prácticas catalogadas para las situaciones frente a las cuales nos podemos encontrar. Simplemente no sabemos con anticipación si una determinada situación va a funcionar. Solo podemos examinar los resultados y adaptarnos. Las soluciones encontradas rara vez son replicables, con los mismos resultados, a otros problemas similares. Para poder operar en la complejidad necesitamos generar contextos donde haya lugar para la experimentación y donde el fallo sea de bajo impacto. Se requiere

niveles altos de creatividad, innovación, interacción y comunicación. El desarrollo de nuevos productos o incorporaciones de nuevas características en productos existentes es un contexto complejo en el que Scrum se utiliza mucho para actuar, inspeccionar y adaptar las prácticas emergentes en un equipo de trabajo. [Alaimo, 2013]

2.3.4 DOMINIO CAÓTICO

Los problemas caóticos requieren una respuesta inmediata. Estamos en crisis y necesitamos actuar de inmediato para restablecer cierto orden. Imaginemos que el sistema de despacho de vuelos en un aeropuerto de alto tráfico deja de funcionar. Este no sería un escenario para utilizar Scrum, aquí debemos actuar de inmediato, alguien debe tomar el control y mover la situación fuera del caos. Por ejemplo, solucionar el problema inmediatamente (sin importar la forma técnica), para luego fuera del caos, evaluar y aplicar una solución más robusta, siendo este el dominio de la improvisación. [Alaimo, 2013]

2.3.5 DOMINIO DESORDENADO

Nos movemos en un espacio desordenado cuando no sabemos en qué dominio estamos. Se clasifica como zona peligrosa, ya que no podemos medir las situaciones ni determinar la forma de actuar. Es muy típico en estas situaciones que las personas interpreten estas y actúen en base a preferencias personales. El gran peligro del dominio desordenado es actuar de manera diferente a la que se necesite para resolver ciertos problemas. Por ejemplo, mucha gente en el ámbito del desarrollo del software está acostumbrada al desarrollo secuencial, por fases, detalladamente planificado utilizado las mejores prácticas de la industria y en este enfoque, que corresponde al dominio simple, muchas veces se aplica en el dominio complejo. Si nos encontramos en el espacio desordenado, todo lo que hagamos debemos estar enfocados netamente a salirnos de ese espacio hacia uno mejor identificado, para luego actuar de la manera en que dicho dominio lo requiera. [Alaimo, 2013]

2.3.6 PRINCIPIOS DE SCRUM

Scrum es el modelo más utilizado dentro de las metodologías ágiles. Muchos de los valores y principios de manifiesto ágil tienen su origen en Scrum, desde la perspectiva de Scrum, los valores de manifiesto ágil: [Alaimo, 2013]

- **Individuos en interacciones por sobre procesos y herramientas.** Scrum se apoya en la confianza hacia las personas, sus interacciones y los equipos.
- **Software funcionando por sobre documentación exhaustiva.** Scrum requiere que al final de cada sprint se entregue un producto funcionando. La documentación es entendida, en Scrum, como un producto intermedio sin valor de negocio.
- **Colaboración con el cliente por sobre la negociación de contratos.** El Scrum Product Owner es el responsable de la relación que existe con los usuarios finales, y áreas de la organización que van a obtener el beneficio del producto.
- **Respuesta al cambio por sobre el seguimiento de un plan.** Scrum, por diseño, se asegura que todo el mundo dentro de un equipo tenga toda la información necesaria para poder tomar decisiones basadas proyecto que se realizara. Fomentar el cambio es una ventaja competitiva de Scrum.

2.3.7 VALORES DE SCRUM.

Además de los cuatro principios mencionados, Scrum se construye sobre cinco pilares, sus valores: [Alaimo, 2013]

- **Foco.** Los Equipos Scrum se enfocan en un conjunto acotado de características por vez. Esto permite que al final de cada sprint se entregue un producto de alta calidad.
- **Coraje.** Debido a que los equipos Scrum trabajan como verdaderos equipos, pueden apoyarse entre compañeros, y así tener el coraje de asumir compromiso desafiante que les permita crecer como profesionales y como equipo.

- **Apertura.** Los equipos Scrum privilegian la transparencia y la discusión abierta de los problemas. No hay agendas ocultas ni triangulación de conflictos. La sinceridad se agradece y la información está disponible para todos todo el tiempo.
- **Compromiso.** Los equipos Scrum tienen mayor control sobre sus actividades, por eso se espera de su parte el compromiso profesional para el logro del éxito.
- **Respeto.** Debido a los miembros de un equipo Scrum trabaja de forma conjunta, compartiendo éxitos y fracasos, se fomenta el respeto mutuo, y la ayuda entre pares es una cuestión a respetar.

2.3.8 ROLES DE SCRUM

En un equipo Scrum se espera que intervenga tres roles: Product Owner, equipo de desarrollo y Scrum master. [Alaimo, 2013]

2.3.8.1 PRODUCT OWNER (DUEÑO DEL PRODUCTO)

El Product Owner es la persona responsable del éxito del producto desde el punto de vista de los stakeholders. Sus principales responsabilidades son: [Alaimo, 2013]

- Determinar la visión del producto, hacia dónde va el equipo de desarrollo.
- Gestionar las expectativas de los stakeholders.
- Recolectar los requerimientos.
- Determinar y conocer en detalle las características funcionales de alto y bajo nivel.
- Generar y mantener el plan de entregas (realse place): fechas de entregas y contenidos de cada una.
- Maximizar la rentabilidad del producto.
- Determinar las prioridades de cada una de las características por sobre el resto.
- Cambiar las prioridades de las características según avanza el proyecto acompañado así los cambios en el negocio.
- Aceptar/ Rechazar el producto construido durante el Sprint y proveer feedback valioso para su evolución.

- Participar de la evolución del sprint junto a los miembros del equipo de desarrollo para obtener feedback de los stakeholders.

El Product Owner se focaliza en maximizar la rentabilidad del producto. La principal herramienta con que cuenta para poder realizar esta tarea es la priorización. De esta manera puede reordenar la cola de trabajo del equipo de desarrollo para que este construya con mayor anticipación las características o funcionalidades más requeridas por el mercado. [Alaimo, 2013]

2.2.8.2 EQUIPO DE DESARROLLO

El equipo de desarrollo está formado por todos los individuos necesarios para la construcción del producto en cuestión. Es el único responsable por la construcción y calidad del producto. [Alaimo, 2013]

El equipo de desarrollo es auto-organizado. Esto significa que no existe un líder externo que asigne las tareas ni que determine la forma en la que serán resueltos los problemas. Es el mismo equipo quien determinara la forma en que realizara el trabajo y como resolverá cada problema que se presente. [Alaimo, 2013]

La contención de esta auto-organización está dada por el objetivo a cumplir; transformar las funcionalidades comprometidas en software funcional y con calidad productiva, es decir producir un incremento funcional potencialmente entregable. Lo que se espera de un miembro de un equipo de desarrollo es que no solamente realice las tareas en las cuales se especializa sino también todo lo que este a su alcance para colaborar con el éxito del equipo. [Alaimo, 2013]

El equipo de desarrollo tiene tres responsabilidades tan fundamentales como indelegables. La primera es proveer las estimaciones de cuanto esfuerzo será requerido para cada una de las características del producto. La segunda responsabilidad es comprometerse en cada sprint a construir un conjunto determinado de características en el tiempo que dura el

mismo. Y finalmente, también es responsable por la entrega del producto terminado al finalizar cada sprint. [Alaimo, 2013]

2.3.8.3 SCRUM MASTER

El Scrum Master es el líder del equipo y es quien ayuda a alcanzar su máximo nivel de productividad posible. Tomando algunas referencias de Leonard Wolk podemos decir que el Scrum Master, es tanto a ser líder, es facilitador, provocador, detective y soplador de brazas. [Alaimo, 2013]

Líder. Por ser un ejemplo a seguir.

Facilitador. Por fomentar contextos de apertura y discusión donde todos pueden expresar sus opiniones y lograr consensos comunes.

Provocador. Por desafilar las estructuras rígidas y las antiguas concepciones sobre cómo deben hacerse las cosas.

Detective. Por involucrarse activamente en la búsqueda e identificación de indicios y pistas en la narrativa del equipo y los individuos.

Soplador de brazas. Un socio facilitador de aprendizaje, que acompañe en la búsqueda de aprendizaje, capacidad de generar nuevas respuestas de sus otros integrantes del equipo. Soplar brazas para reconectar a las personas en desacuerdos que se verán perjudicados por ciertos retrasos en la construcción del producto software que llevan a cado como equipo de desarrollo.

Se espera además, que el Scrum Master acompañe al equipo de trabajo en su día a día y garantice que todos, incluyendo al Product owner, comprendan y utilicen Scrum de forma correcta. [Alaimo, 2013]

2.3.9 ELEMENTOS DE SCRUM

El proceso de Scrum posee una mínima cantidad necesaria de elementos formales para poder llevar adelante un proyecto de desarrollo. [Alaimo, 2013]

2.3.9.1 PRODUCT BACKLOG (BITÁCORA DEL PRODUCTO)

El primer de los elementos, y principal de Scrum, es el Backlog del producto o también conocido como pila del producto como se ve en la Figura 2.5. [Alaimo, 2013]

Nº	Descripción del requerimiento	Prioridad	Estimación Esfuerzo días	Sprint
2	Tener disponible la información de los insumos médicos	1	4	1
1	Registrar los datos personales de los promotores de ventas	2	3	1
3	Definir las áreas de ventas	3	3	1
4	Tener un kardex de clientes (consultorios, farmacia) por áreas	4	3	1
5	Anotar los pedidos de los clientes en un formulario	5	4	1
16	Enviar pedido de insumos al área de Despachos	5	2	1
7	Definir planes de pagos por clientes y por pedido	6	4	2
9	Actualizar los pagos de los clientes	7	4	2
11	Emitir recibos por pagos de los clientes	8	2	2
8	Tener un control de las ventas por promotor	9	3	2
10	Calcular las comisiones por ventas de los promotores	10	3	2
18	Enviar detalle de comisiones al Depto. De Personal	10	2	2
12	Controlar las moras de los clientes	11	4	3
13	Emitir listas de morosos	11	1	3
17	Emitir facturas a clientes	13	1	3
14	Emitir lista de productos vendidos	14	5	3
15	Emitir detalle de ingresos	15	5	3

Figura 2. 5 Ejemplo Product Backlog

Fuente: [Alaimo, 2013]

El Backlog del producto es básicamente un listado de ítems o características del producto a construir, manteniendo y priorizando por el Product Owner. Es importante que exista una clara priorización, ya que es esta priorización la que determinara el orden en el que el equipo de desarrollo transformara las características (ítems) en un producto funcional acabado. [Alaimo, 2013]

Una forma de priorizar los ítems del Product Backlog es según su valor de negocio. Podemos entender el valor de negocio como la relevancia que un ítem tiene para el cumplimiento del objetivo de negocio que estamos buscando. [Alaimo, 2013]

2.3.9.2 SPRINT BACKLOG (BITÁCORA DE SPRINT)

El Sprint Backlog es el conjunto de ítems que fueron seleccionados para trabajar en ellos durante un cierto Sprint, conjuntamente con las tareas que el equipo de desarrollo ha identificado que debe realizar para poder crear un incremento funcional potencialmente entregable al finalizar el sprint además que las tareas tendrán un cierto estado hasta la conclusión como se puede visualizar en la Figura 2.6. [Alaimo, 2013]

PBI	Todo	In Progress	Done

Figura 2. 6 Ejemplo Sprint Backlog
Fuente: [Alaimo, 2013]

2.3.9.3. INCREMENTO FUNCIONAL

Es el resultado de cada Sprint el cual debe ser un incremento funcional potencialmente entregable, pues una característica funcional nueva de un producto que está siendo construido de manera evolutiva. El producto crece con cada Sprint. A su vez cada una de estas características se encuentra lo suficientemente valida o verificada como para poder ser desplegada en producción(o entrega a usuarios finales) si así el negocio lo permite o el cliente lo desea. [Alaimo, 2013]

2.3.10 DINÁMICA (FLUJO DE TRABAJO)

Antes de describir en detalle la dinámica de Scrum, recordemos el mecanismo de Timeboxing² promovido por Scrum y los principios de ritmo sostenible, entrega frecuente de software valioso y adaptación constante que encontramos en el manifiesto ágil. La razón es que en conjunto constituyen la piedra angular de la dinámica de Scrum: aprendizaje, inspección y adaptación. [Alaimo, 2013]

2.3.10.1 SPRINT (ITERACIÓN)

Las iteraciones en Scrum se conocen como Sprint. Scrum como todos los enfoques ágiles, es un proceso de desarrollo incremental e iterativo. Esto significa que el producto se construye en incrementos funcionales entregados en periodos cortos para obtener feedback frecuente. [Alaimo, 2013]

2.3.10.2 SPRINT PLANNING MEETING (PLANIFICACIÓN DEL SPRINT)

Al comienzo de cada sprint se realiza una reunión de planificación de sprint donde serán generados los acuerdos y compromisos entre el equipo de desarrollo y el Product Owner sobre el alcance del sprint. Esta reunión de planificación habitualmente se divide en dos partes con finalidades diferentes; una primera parte estratégica y enfocada en el “que” y una segunda parte táctica cuyo hilo conductor principal es el “como”. [Alaimo, 2013]

2.3.10.3 SCRUM DIARIO.

Uno de los beneficios de Scrum está dado por el incremento de la comunicación dentro del equipo de proyecto. Esto facilita la coordinación de acciones entre los miembros del equipo de proyecto. Esto facilita la coordinación de acciones entre los miembros del equipo de desarrollo y conocimiento “en vivo” de las dependencias de las actividades que realizan. Por otro lado, se requiere además aumentar y explicar los compromisos asumidos entre los miembros del equipo de desarrollo y dar visibilidad a los impedimentos que surjan del

² Mecanismo que consiste en poner límites de tiempo para una actividad. Fuente: Alaimo, 2013.

trabajo que está siendo realizado y que muchas veces nos impide lograr los objetivos. [Alaimo, 2013]

2.3.10.4 REVISIÓN DEL SPRINT

Al finalizar cada sprint se realizara una reunión de revisión de sprint, donde se evaluara el incremento funcional potencialmente entregable construido por el equipo de desarrollo. En esta reunión el equipo Scrum y los Stackholder revisan el resultado (Producto utilizable y potencialmente entregable) del sprint. [Alaimo, 2013]

2.3.11 HISTORIAS DE USUARIO

Técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el desarrollador describe brevemente las características de los módulos que el software debe poseer, dando prioridad a los que el cliente necesita con anticipación, estos requerimientos pueden ser funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. [Alaimo, 2013]

Como se puede observar en la Figura 2.7.

Entrega 1 - Comercializar Eventos				
Prio.	Como ...	Necesito ...	Para ...	Criterios de Aceptación
1	Comercial	Crear un evento confirmado	Hacer el seguimiento del mismo	- Debe tener Nombre, Fecha, Descripción, Destinatarios, Programa, Instructor, Lugar, Ciudad, País, Capacidad, Precios y Promociones: SEB (Super Early Bird), EB (Early Bird), dto. en % para 2 personas y dto. en % para 3 o más personas. - Las promociones son opcionales. - Las fechas de SEB y EB deben ser anteriores a la fecha del evento - Por defecto SEB=30 días antes, EB=10 días antes, 2personas=-10%, 3+ personas=-15%. - Un evento puede ser público o privado
2	Comercial	Ver listado de eventos confirmados	No superponer eventos	- Mostrar Nombre, Ciudad y País - Muestra solo los futuros - Ordenado por fecha ascendente
3	Comercial	Modificar evento confirmado	Corregir cualquier error o re programarlo	- Permite modificar todos los campos.

Figura 2. 7 Historias de Usuario
Fuente: [Alaimo, 2013]

2.3.12 MODELO DE PROCESO

Scrum es un método de desarrollo muy simple, que requiere mayor trabajo ya que no se basa en el seguimiento de un plan, sino en la adaptación continua de las circunstancias de la evolución del proyecto. Scrum es una metodología ágil, y como tal emplea la estructura incremental basada en iteraciones y revisiones. El ciclo de vida Scrum está compuesto por tres tareas o etapas, Pre-Game, Game y Post-Game. [Alaimo, 2013]

2.3.13 PRE-GAME

Antes de llevar a cabo el desarrollo del proyecto, se especifica lo que se va a realizar en las iteraciones, además de la prioridad con la que se lo hará, esta fase consta de dos puntos importantes: [Alaimo, 2013]

Planeación. Durante la planeación todos los miembros del equipo incluyendo el cliente contribuyen a la creación de una lista de características del software, para el análisis y conceptualización del problema.

Las tareas que se realizan en esta primera etapa son:

- Recopilación de requerimientos para conformar el backlog del producto, priorizando de acuerdo a una evaluación del cliente.
- Definición de las fechas de entrega de los sprints y sus funcionalidades.
- Análisis de riesgo y controles para los riesgos.
- Selección de las herramientas y la infraestructura de desarrollo.
- Cálculo y estimación de costos por cada iteración.

Arquitectura. El Objetivo de esta etapa es diseñar elementos de backlog del producto que serán puestos en ejecución. En esta fase incluyendo una revisión de la arquitectura del software tanto en diseño de alto nivel.

En esta etapa se realiza las tareas de:

- Revisión de los ítems del backlog del producto.
- Análisis del dominio para reflejar el nuevo contexto y requisitos del sistema.
- Revisión de la arquitectura del software de acuerdo a los requisitos definidos.
- Revisión del diseño de alto nivel.

2.3.14 GAME

Una vez realizada la especificación correspondiente, se lleva a cabo la elaboración correspondiente con el seguimiento a cargo del mismo equipo de desarrollo.

En cada iteración del game se realiza las siguientes tareas: [Alaimo, 2013]

- **Planeación del Sprint.** Antes de iniciar el sprint debe llevarse a cabo dos reuniones la primera será para mejorar y priorizar el backlog del producto, además de elegir las metas de la iteración. En la segunda reunión se deben considerar el cómo alcanzar los requerimientos y crear el backlog del sprint.
- **Desarrollo del sprint.** El trabajo generalmente se organiza en iteraciones de 30 días (Sprint). El Sprint es el desarrollo de la nueva funcionalidad para el producto. Esta fase provee la siguiente documentación. Backlog del sprint con las actividades desarrolladas, los responsables y la duración de cada actividad.
- **Revisión del sprint.** Al final de cada iteración se lleva a cabo una reunión de revisión, donde se presenta la nueva funcionalidad del software, las metas e inclusive la información de las funcionalidades, diseño ventajas, inconvenientes y esfuerzos del equipo.

2.3.15 POST-GAME

Luego de haber culminado todas las iteraciones, solo resta la revisión final denominada según la metodología de Scrum, el cierre. [Alaimo, 2013]

Cierre. En esta última etapa se realiza la preparación operacional, incluyendo la documentación final necesaria para la presentación. También es en esta etapa que se debe realizar dependiendo del tipo de producto, ya sea el entretenimiento del usuario o el marketing para la venta de nuevo software. [Alaimo, 2013]

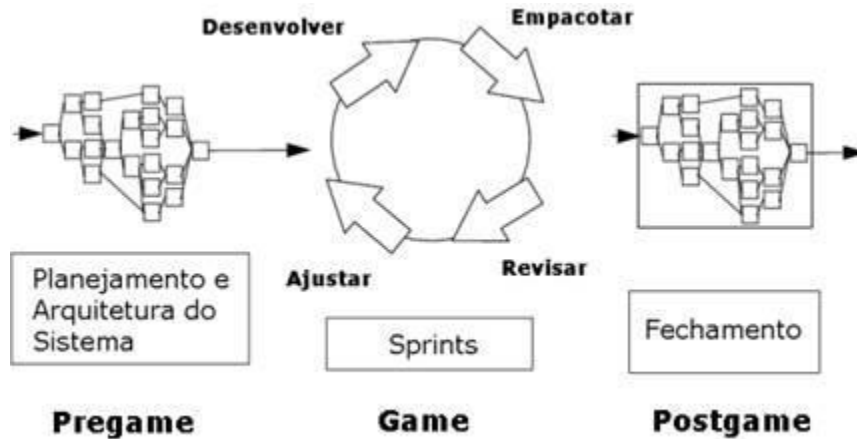


Figura 2. 8 Proceso Scrum

Fuente: [Alaimo, 2013]

2.4 INGENIERÍA WEB

La ingeniería web es el proceso utilizado para crear, implantar y mantener aplicación y sistemas web de alta calidad. El proceso de ingeniería web está relacionado con la inmediatez y evolución, estos crecen continuamente para llevar un proceso incremental y evolutivo que permite al usuario involucrarse activamente, a facilitar el desarrollo de productos que se ajustan a lo que se está buscando en cuanto a la funcionalidad del producto. [Pressman, 2002]

El proceso de desarrollo de aplicaciones hipermedia con lleva a la realización de una serie de actividades que son aplicables a cualquier sistema web, independientemente del tamaño y complejidad del mismo. La ingeniería web aplica el desarrollo, operación y mantenimiento de todo el ciclo de vida del desarrollo de software.

El importante mencionar que el impacto de la sistemas y aplicaciones basados en metodologías web es el suceso más significativo en el área de la tecnología de información y comunicación. Conforme a la importancia de las aplicaciones web que crece, surge la necesidad de enfocar la ingeniería web a un entorno disciplinado y bien estructurado. [Escalona, 2002]

Visualizando la evolución para las personas que están involucradas en el área de la informática, es significativamente grande y en ocasiones muy confusas; pasmos de páginas estáticas hechas con HTML, JavaScript, CMS y estándares como XHTML, CSS, P3P, por nombrar alguna de ellas. [Escalona, 2002]

La Web crece a medida que los usuarios colaboran usando estas plataformas tecnológicas. Esta evolución tecnológica es lo que muchos llaman la web 2.0; que no es una versión sino un concepto y una forma diferente de ver las cosas. [Escalona, 2002]

2.4.1 INGENIERÍA DE REQUISITOS PARA LAS APLICACIONES WEB

Un requerimiento es una especificación de lo que debería ser implementado. Son descripciones de cómo el sistema debe comportarse, es una prioridad o un atributo y son sobre las restricciones en el proceso de desarrollo del sistema [Bosh, 2008].

Las especificaciones de los requerimientos no incluyen detalles de diseño e implementación, información sobre la planificación del proyecto o planes de pruebas.

Los requerimientos de aplicaciones al igual que los de software tradicional, comprenden 3 niveles distintos como ser Requerimientos del negocio, Requerimientos del usuario y Requerimientos funcionales. [Brambilla & Manolescu, 2002]

- **Requerimientos del Negocio.** Representan los objetivos de la empresa, institución o usuario que solicita el software.
- **Requerimientos de Usuario.** Describen lo que el usuario de la aplicación web será capaz de realizar un vez que se haya implementado el mismo.

- **Requerimientos Funcionales.** Especifican las funcionalidades del software, que los desarrolladores deben construir para que el usuario logre cumplir sus objetivos planeados.

2.4.2 MARCO DE TRABAJO PARA LA INGENIERÍA WEB

A medida de la evolución de las Web Apps, pasa de utilizar recursos estáticos de información controlada por el contenido, a utilizar entornos de aplicaciones dinámicas controladas por el usuario, para lo que es importante aplicar una gestión sólida, por lo que se necesita un marco de trabajo eficaz. Aplicaciones que manejan grandes cantidades de datos las cuales deben ser gestionadas eficaz y eficientemente por los administradores del sistema. [Brambilla & Manolescu, 2002]

2.4.3 FORMULACIÓN

Para comenzar esta etapa se deben plantear las siguientes preguntas: [Brambilla & Manolescu, 2002]

¿Cuál es la motivación principal para la Web App?

¿Por qué es necesaria la Web App?

¿Quiénes van a utilizar la Web App?

Las respuestas deben ser lo más sencillas posibles. El Objetivo es delimitar la intención global del sitio Web. Para luego definir metas: [Brambilla & Manolescu, 2002]

Metas informativas: Indican la intención de proporcionar el contenido y/o información específica para el usuario final.

Metas aplicables: Indican la habilidad de realizar algunas tareas dentro de las Web App.

Una vez identificado todas las metas aplicables e informativas se desarrolla el perfil del cliente. El perfil del usuario recoge las características relevantes de los usuarios potenciales incluyendo antecedentes, conocimientos, preferencias, entre otros.

2.4.4 ANÁLISIS

Durante el análisis de la ingeniería web surgen cuatro tipos de análisis diferentes: [Brambilla & Manolescu, 2002]

- **Análisis de contenido.** Se trata de identificar el contenido que se va proporcionar, incluyendo datos de texto, gráficos, video y sonido.
- **Análisis de la Interacción.** Se trata de la descripción detallada de la interacción del usuario con la aplicación web, basados en casos prácticos de la aplicación.
- **Análisis Funcional.** Los escenarios de utilización (casos de uso) credos como parte del análisis de interacción que definen las operaciones que se aplicaran en el contenido de la Web App e implicara otras funciones de procesamiento, es en este análisis que se definen todas las funciones y operaciones.
- **Análisis de la Configuración.** Se efectúa una descripción detallada del entorno y de la infraestructura donde se aplicara la Web App. La aplicación web no necesariamente debe ser alojada en un servidor de internet, puede estar en una intranet. Además se deberá identificar la estructura y composición en relación a los componentes y el grado de utilización de la base de datos, con el fin de generar contenido preciso y eficaz en la Web App.

2.4.5 DISEÑO

La Modularidad eficaz (exhibida con la cohesión alta, acoplamiento bajo), la elaboración paso a paso, y cualquier otra heurística de diseño del software conducirá a diseñar aplicaciones basadas en la Web más fáciles de adaptar, probar y utilizar.

En este sentido tomaremos en cuenta: [Brambilla & Manolescu, 2002]

- **Configuraciones de diseño.** Las configuraciones de diseño se pueden aplicar no solo a los elementos funcionales de una aplicación sino también a los documentos, gráficos y estética general de un sitio Web.
- **Plantillas.** Las plantillas se pueden utilizar para proporcionar un marco de trabajo esquemático de cualquier configuración de diseño o documento a utilizar dentro de una Web App.
- **Diseño Arquitectónico.** El diseño arquitectónico para el desarrollo de software y aplicaciones basados en la web se centra en la definición de la estructura global hipertexto para las Web App. Y en aplicación de las configuraciones de diseño y plantillas constructivas para estandarizar la estructura con el fin de lograr la reutilización.
- **Diseño de Navegación.** Una vez establecida una arquitectura e identificados los componentes, el diseñador deberá definir las rutas de navegación (url's) que permita al usuario acceder al contenido y a los servicios de la Web App.
- **Diseño de Interfaz.** La interfaz de usuario de Web App es la primera experiencia interactiva entre el usuario y la Web App, por lo cual los desarrolladores Web generalmente utilizan lenguajes interpretados (Script) en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que requiera recargar la página cada vez (lo que suele resultar molesto para los usuarios).

2.4.6 PRUEBAS

El enfoque de las pruebas del Web App adopta los principios básicos de todas las pruebas del software y aplica estrategias y tácticas que ya han sido testeadas y recomendadas para los sistemas basados al enfoque web, entre estas tenemos las siguientes: [Pressman, 2002]

- El modelo de diseño para la Web App es revisado para describir errores de navegación. Los casos prácticos derivados como parte de la actividad de análisis

permiten que un desarrollador web ejercite cada escenario de utilización frente al diseño arquitectónico y de navegación.

- Se construye la arquitectura, se realizan las pruebas de integración. La estrategia para la prueba de integración depende de la arquitectura que se haya elegido para la Web App.
- La Web App ensamblada se prueba para conseguir funcionalidad global y un contenido. Al igual que la validación convencional, la validación de los sistemas y aplicaciones basados en la web se centran en acciones visibles de los usuarios finales y en salidas reconocibles para los usuarios.
- La Web App se implementa en una variedad de configuraciones diferentes en relación a los diferentes entornos de navegación para poder comprobar la compatibilidad con cada configuración. Se crea una matriz de referencia cruzadas que define todos los sistemas operativos posibles, plataformas de hardware para navegadores y protocolos de comunicación. Entonces se lleva a cabo pruebas para describir todas las incompatibilidades generadas por el uso de las herramientas.
- La Web App se comprueba con la población de usuarios finales controlada y monitorizada.

Se selecciona un grupo de usuarios que abarque todos los roles posibles de usuarios. La Web App se pone en ejecución con esto los usuarios son los que evalúan los resultados de la interacción con el software para ver los errores de contenido y de navegación los intereses en usabilidad, compatibilidad, fiabilidad y rendimiento de la Web App. [Pressman, 2002]

2.5 WEBML (WEB MODELING LANGUAGE)

WebML es un lenguaje de modelado gráfico para apoyar las actividades del diseño de sitios web. Define diagramas: estructura, composición, navegación, Hipertexto, presentación, personalización. WebML es el lenguaje conceptual para diseño de alto nivel de sitios web

con datos intensivo, y no así para sitios web pequeños o estáticos. En la figura 2.9 se puede apreciar la estructura asociada al modelo WebML. [Ceri, 2000].

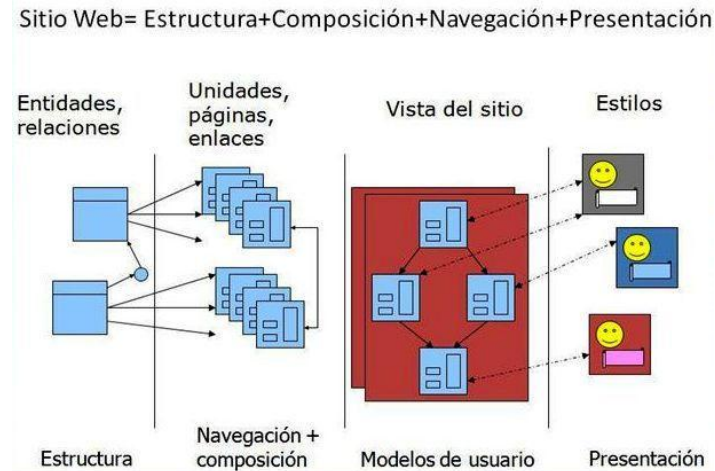


Figura 2. 9 Enfoque WebML
Fuente: [Ceri, 2000]

2.5.1 MODELO ESTRUCTURAL

Cuando se trabaja con WebML el proceso de desarrollo comienza con la descripción conceptual del sistema, en la cual, utilizando una herramienta CASE para modelado mediante la definición de entidades o contenedores de datos y sus relaciones que deben tener una cantidad de elementos y un rol asociado. [Ceri, 2000]

En este sentido el modelo estructural describe la organización conceptual de datos, este modelo es compatible con:

- Modelo entidad-relación.
- Diagrama de clases UML

En la Figura 2.10 se muestra un ejemplo de diagrama de estructura, el cual contiene cinco entidades: [Ceri, 2000]

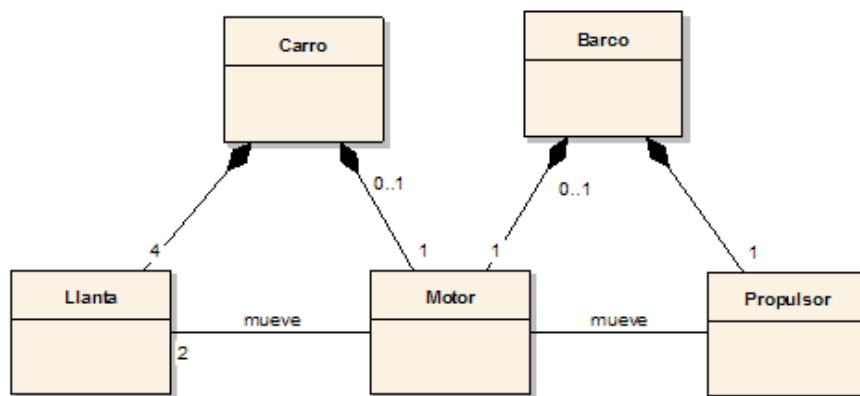


Figura 2. 10 Modelo Estructural

Fuente: [Ceri, 2000]

Una característica a destacar de WebML es que no exige ninguna herramienta específica para hacer este tipo de modelo.

2.5.2 MODELO DE COMPOSICIÓN

El propósito del diagrama de composición es definir los nodos que forman parte del hipertexto contenido en el sitio Web, es decir se especifican las páginas y las unidades (elementos atómicos de información que deben aparecer en sitio web) que componen el sitio web. WebML soporta seis tipos de unidades para componer el hipertexto: [Ceri, 2002]

- **Unidad de datos.** Muestran información sobre un solo objeto, son definidas para seleccionar una mezcla de información. Para definir una unidad de datos se requiere la designación del concepto al cual se refiere la unidad y la selección de los atributos de la unidad como se ve en la Figura 2.11.

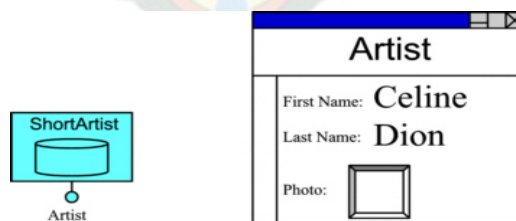


Figura 2. 11 Unidad de datos

Fuente: [Ceri, 2000]

- **Unidad Multidatos.** Muestra información sobre un conjunto de objetos, presenta múltiples instancias de una entidad o componente como se ven en la Figura 2.12.

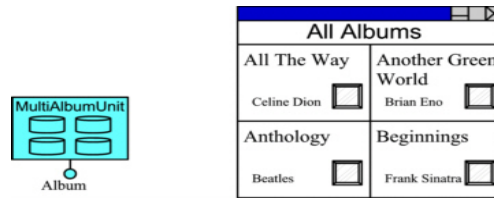


Figura 2. 12 Unidad Multidatos
Fuente: [Ceri, 2000]

- **Unidad Índice.** Presenta múltiples instancias de la unidad o componentes una lista, esta unidad tiene dos partes principales: el contenedor que incluye las instancias que se desean mostrar y los atributos usados como clave índice, ver la Figura 2.13.



Figura 2. 13 Unidad Índice
Fuente: [Ceri, 2000]

- **Unidad Scroller.** Provee comandos para desplazarse a través de los objetos en un contenedor. Esta unidad es normalmente usada conjuntamente con la unidad de datos, la cual representa el elemento actual visualizado del contenedor, como puede verse en la Figura 2.14.



Figura 2. 14 Unidad Scroller
Fuente: [Ceri, 2000]

- **Unidad filtro.** Provee campos de entrada para buscar los objetos en un contenedor, esta unidad es normalmente usada con la unidad índice o multidatos, la cual muestra los objetos que coinciden con las condiciones de búsqueda, ver la Figura 2.15.

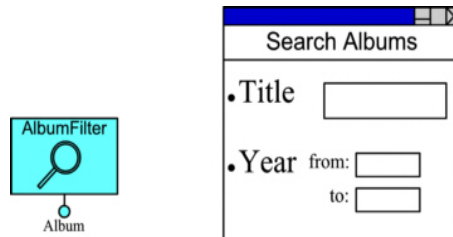


Figura 2. 15 Unidad Filtro
Fuente: [Ceri, 2000]

- **Unidad directa.** Expresa un tipo particular de índice, el cual contiene un solo objeto asociado a otros objetos por una relación uno a uno, ver la Figura 2.16.

2.5.3 MODELO DE NAVEGACIÓN

El propósito de este modelo es poder especificar la forma en la cual las unidades y las páginas son conectadas para formar un hipertexto, para este contenido WebML provee la noción de enlaces (url's), de los cuales hay dos tipos: [Ceri, 2000]

Enlaces contextuales. Conectan unidades de una forma coherente a la semántica expresada por el modelo de estructura de la Web App. Un enlace contextual lleva información (de contexto) de la unidad de origen a la unidad destino, esta información es usada para determinar el objeto o conjunto de objetos a ser mostrados en la unidad destino.

Enlaces no contextuales. Conectan paginas libremente, independiente del contexto, como se refleja en la Figura 2.16.

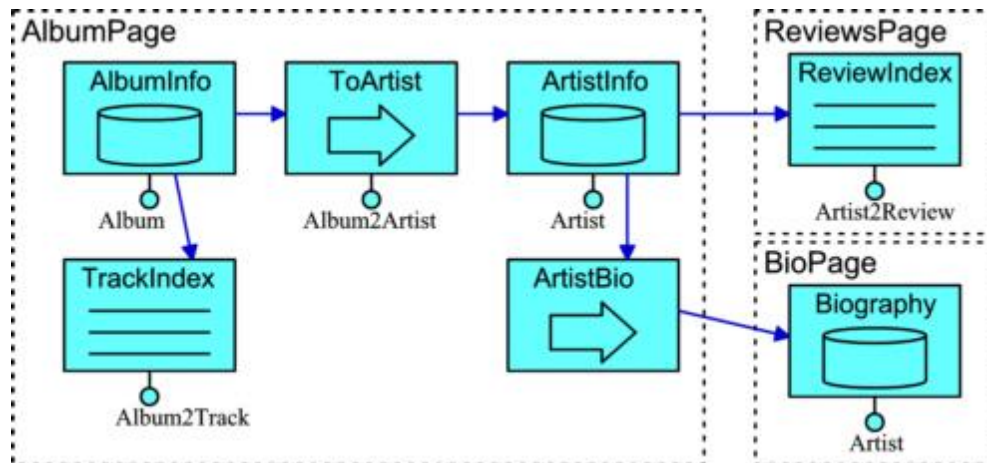


Figura 2. 16 Modelo de navegación
Fuente: [Ceri, 2000]

2.5.4 MODELO DE PRESENTACIÓN

En esta fase se define la apariencia gráfica de cada una de las páginas que conforman el proyecto. Permite definir la posición de las unidades en la página. WebML no incluye un modelo específico para establecer la presentación a nivel conceptual. [Ceri, 2000]

2.5.5 MODELO DE PERSONALIZACIÓN

Este modelo comprende de las siguientes fases: [Ceri, 2000]

- **Control de acceso.** Login / Logout operaciones para el reconocimiento de usuarios.
- **Asignación del Site View.** Basado en el grupo al que el usuario pertenece.
- **Personalización de la página.** Usuarios y grupos dependiendo del contenido.

En principio un rol de grupo definido para todos (default) y estos usuarios no necesitan “login”. Cada grupo puede contener uno o más usuarios, estos si necesitan tener un “login”.

Cada usuario puede estar uno o más grupos y tiene un grupo por defecto y cada grupo está asociado a una vista en la Web App, una forma de representarla por ejemplo es como se puede observar en la figura 2.17.

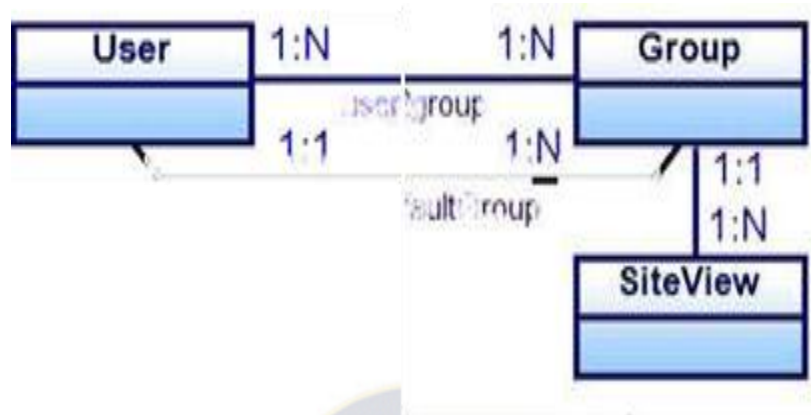


Figura 2. 17 Modelo de Personalización

Fuente: [Ceri, 2000]

2.6 REGLAMENTO DE KARDEX ACADÉMICO

El reglamento de Kardex presenta una normativa, es en ese sentido que el presente proyecto dará cumplimiento al Apartado “Resguardo y Responsabilidad de Kardex”, Artículo 17. “La Unidad de Kardex académico será responsable de realizar el escaneo de las Actas finales de notas firmadas por el docente para ser almacenadas en el Sistema de Información Académica (SIA – UMSA).” [Reglamento de Kardex Académico, 2016].

La reglamentación de Kardex en la universidad mayor de san Andrés tiene como objetivo principal Uniformar el tratamiento archivístico y digitalización de los documentos académicos del estudiante desde su ingreso hasta su titulación [Reglamento de Kardex Académico, 2016].

Para tener un concepto más claro acerca de la definición de lo que es un acta de notas el reglamento presenta una definición formal, en el Artículo 13. “Es un documento original, único y útil con valor administrativo, legal, e histórico que refleja las calificaciones finales de las materias que el estudiante ha cursado durante la gestión semestral o anual en la Carrera, estos documentos deben llevar las firmas del docente, Director de Carrera, Vicedecano y/o Decano.

Estas actas de calificaciones finales deben ser digitalizadas para el control de las notas de los estudiantes de una determinada materia en una gestión académica, a través de la visualización de dichas actas firmadas por el docente.” [Reglamento de Kardex Académico, 2016].

Básicamente el Sistema que se elabora debe dar soluciones a los siguientes artículos que presenta el reglamento de Kardex académico:

- “**Artículo 39.** Los encargados de las Unidades de Kardex, deben almacenar el acta de notas debidamente firmadas y selladas, al final de cada gestión académica (semestral, anual, cursos de temporada), deberán digitalizarlas para su almacenamiento en el sistema y almacenarlas donde corresponda y previa ordenación y depuración de los expedientes académicos.” [Reglamento de Kardex Académico, 2016].
- “**Artículo 40.**
 - a) **Búsqueda de expedientes estudiantiles.-** El Sistema de Kardex Digital permitirá el acceso fácil a la información de cada uno de los estudiantes, el encargado de Kardex podrá visualizar los documentos de cada expediente así como la verificación de su nota en las actas finales. La búsqueda física, si así lo requiere, se realizara viendo las características definidas en el Tratamiento Archivístico.
 - b) **Búsqueda de nota en las actas.-** El encargado podrá visualizar los expedientes realizando la búsqueda por gestión académica materia y paralelo. Si se requiere la información en físico se lo podrá ubicar de la misma manera.” [Reglamento de Kardex Académico, 2016].

2.6.1 MOVIMIENTO DE EXPEDIENTES Y ACTA DE NOTAS

Con relación al movimiento de los expedientes podemos citar lo siguiente:

- “**Artículo 41.** Si por alguna razón un expediente o acta de notas tenga cierto movimiento a otras instancias debe ser registrado en el sistema, para ser visualizado en el estado de cada expediente para una mejor administración de los documentos.”
[Reglamento de Kardex Académico, 2016].

2.6.2 SEGURIDAD DEL S.I.A. – U.M.S.A.

En cuanto a seguridad de la documentación y en referencia a la elaboración de los sistemas que contemplara el Sistema de Información Académica en la Universidad Mayor de San Andrés (SIA-UMSA) denota la siguiente normativa, la podemos describir en el artículo siguiente:

- “**Artículo 35.-** La División de Sistemas de Información y Estadísticas establecerá:
 - a) Niveles de seguridad
 - b) Niveles de consulta
 - c) Respaldo de datos
 - d) Impresión de reportes estandarizados.”

[Reglamento de Kardex Académico, 2016].

Es en ese entendido que el presente proyecto tiene una base reglamentada bajo una normativa denominada Reglamento de Kardex, ya que justamente serán las unidades de Kardex de la Universidad Mayor de San Andrés quienes serán el motor fundamental que darán dinamismo al manejo de la información y la explotación del sistema que se plantea en el presente proyecto.

CAPÍTULO III

MARCO APLICATIVO

INTRODUCCIÓN

En el presente capítulo se dará énfasis al desarrollo del proyecto propuesto de acuerdo a los requisitos funcionales, tareas del sistema, modelado y construcción de incrementos a través de iteraciones, para el software denominado Sistema Web de Digitalización de Libro de Acta de Notas (SISACTAS).

Para tal desarrollo el proyecto utiliza la metodología ágil Scrum, la cual recurre a un proceso iterativo e incremental que se apoya con el uso del modelado WebML, que está orientado a una alta iteración con los datos que se gestionan en el uso de almacenamiento y consulta de la información.

Cada Iteración (Sprint) se determina que partes se van a construir, tomando como criterios las necesidades de la institución y la cantidad de trabajo que se abordara durante la iteración.

3.1 ITERACIONES DEL SISTEMA WEB

En respuesta a los requerimientos de la institución se realizaron cuatro iteraciones, que se desarrolla en el GAME del proyecto.

Primera iteración. Registro y recepción de documentos.

Segunda iteración. Almacenamiento de la información en formato digital.

Tercera iteración. Búsqueda de actas de notas según parámetros.

Cuarta iteración. Reportes del Sistema y listado de actas almacenadas.

Las iteraciones se muestran en la Figura 3.1.

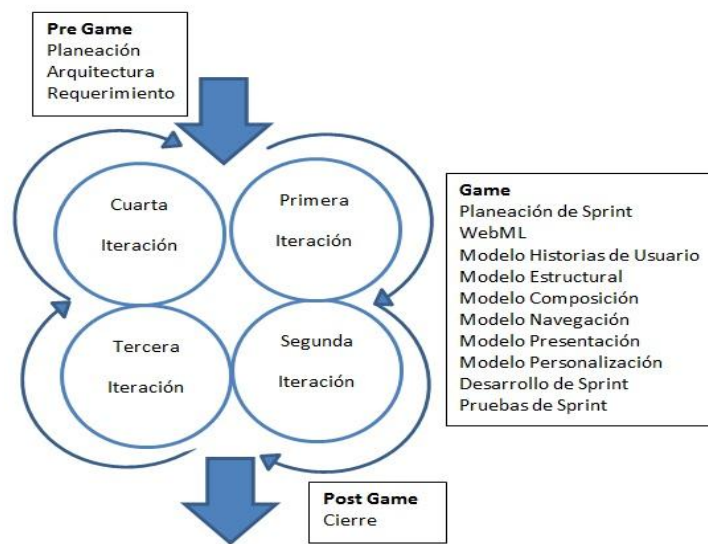


Figura 3. 1 Etapas de Sprint
Fuente: [Elaboración propia]

3.2 PRE GAME

3.2.1 RECOPIACIÓN DE REQUERIMIENTOS

El Backlog del producto, contempla los requerimientos y características finales del sistema web.

- **Reuniones.** Durante las reuniones se genera una lista que abstrae los requerimientos funcionales del sistema web.
- **Requisitos para el desarrollo del software.** Se propone técnicas de programación para apoyar a los resultados que se desean obtener, de manera que puedan evolucionar en las iteraciones y durante el proceso de desarrollo.

A continuación mencionaremos algunas características de flujo de trabajo en relación a la elaboración de los libros de actas de notas de parte de las unidades de Kardex.

Los Kardixtas son responsables de las actas de calificaciones finales el cual es un documento original, único y útil con valor administrativo, legal e histórico que refleja calificaciones finales de las materias que el estudiante ha cursado durante el periodo académico respectivo, estos documentos deben llevar las firmas del Docente, Director de Carrera, Vicedecano y/o Decano.

Estas actas de calificaciones finales deben ser digitalizadas para el control de las notas de los estudiantes de una determinada materia en una gestión académica.

El docente designado para impartir una asignatura es el único responsable de validar las calificaciones finales de los estudiantes.

El encargado de kardex académico será responsable de realizar el escaneo de las actas finales de notas firmadas por el docente y las autoridades correspondientes.

Una vez que se coleccionen los documentos correspondientes el encargado de kardex se encargará del empaste del libro de acta de notas de una gestión.

Se evidenció que la elaboración de los libros de acta de notas es de forma manual, lo que conllevaría a errores de ordenación y mezcla de documentación, las entregas de las actas de calificaciones de parte de los docentes no tienen un control de entrega, además que la información que proporcionan los docentes, es necesaria para la elaboración del certificado de notas, para cuando los estudiantes así lo soliciten, la documentación debe estar resguardada en formato físico y en el formato digital, de tal manera que sea más práctica cualquier verificación pertinente.

En la Tabla 3.1 se detalla los requerimientos establecidos y solicitados de acuerdo a las necesidades establecidas para la aplicación del sistema web, previamente evaluados en las reuniones de cada sprint.

ID	Descripción	Módulo	prioridad	estado
R1	Base de datos para el sistema "SISACTAS"	Plataforma	Alta	Terminado
R2	Automatizar Registro de Docentes	Contenido	Alta	Terminado
R3	Almacenamiento de las actas de notas impresas y firmadas con las autoridades correspondientes	contenido	Alta	Terminado
R4	Almacenamiento de los Índices Correspondientes de las actas de notas	Contenido	Alta	Terminado
R5	Control de acceso en las cuentas de usuario	Plataforma	Alta	Terminado
R6	Actualización Frecuente de la Base de Datos	Plataforma	Media	Terminado
R7	Control y asignación de Usuarios	Plataforma	Media	Terminado
R8	Control y Asignación de roles de Autoridades	Plataforma	Media	Terminado
R9	Interfaz Amigable	Contenido, Comunicación	Media	Terminado

Tabla 3.1 Requerimientos Institucionales
Fuente: [Elaboración Propia]

3.2.2 DEFINICIÓN DEL CRONOGRAMA DE TRABAJO

El cronograma de trabajo se definió en base al ciclo de vida de la metodología Scrum, y se identifican tres etapas principales, mencionadas anteriormente (Pre-Game, Game, Post-Game), este cronograma de trabajo se puede apreciar en un diagrama Gantt del Anexo A del presente Proyecto de grado.

3.2.3 ANÁLISIS DE RIESGOS

Es la probabilidad de ocurrencia de un acontecimiento adverso, problema o daño y las consecuencias del mismo, por lo cual se aplica tres tipos de riesgo:

- **Riesgo del proyecto.** Que afecta a la ejecución de los módulos en ciertos periodos de tiempo ya definimos por el grupo de trabajo.
- **Riesgo del producto.** Afectan a la calidad o rendimiento del sistema web que se está desarrollando.
- **Riesgo de negocio.** Afectan a la Organización o Institución que desarrolla o suministra el sistema web.

En la Tabla 3.2 detallamos el análisis de riesgo del presente proyecto:

Riesgo	Tipo	Descripción	Probabilidad	Efecto	Estrategia
No se cumplen las fechas establecidas en el cronograma	Proyecto	Es probable que las fechas descritas en el diagrama de Gantt no se cumplan	Alta	Tolerable	Realizar un Cronograma flexible
Cambios frecuentes en los requerimientos	Proyecto Producto	El riesgo más común es generalmente el miedo al cambio en relación a los usuarios finales	Alta	Serio	Realizar reuniones de coordinación constante
No se cumplen los plazos de entrega del software	Producto	Los plazos de entrega están sujeto al Scrum Master del Proyecto	moderada	Serio	Agilizar el proceso de desarrollo

No existe el incentivo adecuado para la elaboración del proyecto	Producto Proyecto	pagos retrasados, disminución de activos, presión innecesaria	baja	Tolerable	Entregas de versiones funcionales en cortos periodos de tiempo
--	-------------------	---	------	-----------	--

Tabla 3.2 Análisis de Riesgo
Fuente: [Elaboración Propia]

3.3 GAME

Durante esta etapa del proyecto, cada iteración adopta la metodología WebML que son implementados basándose principalmente en la base de datos del sistema web (gestor de base de datos PostgreSQL).

En cuanto a la persistencia de objetos se prefirió por hacer corresponder cada clase del modelo conceptual con las tablas de la base de datos, donde las filas correspondientes representarían las instancias de los objetos y las columnas los atributos de las clases.

Las clases y sus métodos son implementados en el lenguaje de programación Java.

Finalmente se desarrolla las vistas en el entorno Web en base a los modelos de presentación y navegación, al mismo las vistas tienen un diseño adaptable a diferentes resoluciones de pantalla que facilita la navegación en diferentes tipos de dispositivos.

3.3.1 PRIMERA ITERACIÓN

Durante la primera iteración se desarrolla los elementos pertinentes a la plataforma de registro de docentes y los índices correspondientes a una gestión académica. La actividad realizada en esta iteración se observa en la Tabla 3.3, que establece el BackLog del Sprint.

		SPRINT	INICIO	DURACIÓN
		1	09/07/2016	17 días
ID	TAREAS	TIPO	DIAS DE TRABAJO	ESTADO
1.1	Realizar la Planificación de la iteración	Planificación	2	Terminado
1.2	Analizar los Requerimientos del BackLog del Producto	Planificación	2	Terminado
1.3	Analizar los Requerimientos de la iteración con Historias de Usuario	Desarrollo	2	Terminado
1.4	Construir el modelo de contenido	Desarrollo	2	Terminado
1.5	Diseñar la base de Datos	Desarrollo	2	Terminado
1.6	Construir el Modelo de Usuario	Desarrollo	2	Terminado
1.7	Construir el Modelo de Navegación	Desarrollo	2	Terminado
1.8	Construir el Modelo de Proceso	Desarrollo	2	Terminado
1.9	Construir el Modelo de Presentación	Desarrollo	2	Terminado
1.10	Desarrollar el módulo de registro de Docentes e Índices	Desarrollo	6	Terminado
1.11	Probar los Módulos Desarrollados en la Iteración	Desarrollo	1	Terminado

Tabla 3.3 BackLog del Primer Sprint
Fuente: [Elaboración Propia]

Las Funcionalidades correspondientes al incremento de esta iteración son:

- Base de datos funcional de acuerdo a los requerimientos del desarrollo del Software.
- Ingreso de datos de Docentes mediante Número de Ítem, Carnet de identidad, Nombres, Apellidos.
- Despliegue en pantalla del historial de académico del estudiante.

3.3.2 SEGUNDA ITERACIÓN

En esta iteración se desarrolla el módulo de almacenamiento de las actas de notas en formato digital. La actividad realizada en esta iteración se observa en la Tabla 3.4

		SPRINT	INICIO	DURACIÓN
		2	26/07/2016	17 días
ID	TAREAS	TIPO	DIAS DE TRABAJO	ESTADO
2.1	Realizar la Planificación de la iteración	Planificación	2	Terminado
2.2	Analizar los Requerimientos del BackLog del Producto	Planificación	2	Terminado
2.3	Analizar los Requerimientos de la iteración con Historias de Usuario	Desarrollo	2	Terminado
2.4	Construir el modelo de contenido	Desarrollo	2	Terminado
2.5	Diseñar la base de Datos	Desarrollo	2	Terminado
2.6	Construir el Modelo de Usuario	Desarrollo	2	Terminado
2.7	Construir el Modelo de Navegación	Desarrollo	2	Terminado
2.8	Construir el Modelo de Proceso	Desarrollo	2	Terminado
2.9	Construir el Modelo de Presentación	Desarrollo	2	Terminado
2.10	Desarrollar el módulo de almacenamiento de las actas de notas en	Desarrollo	6	Terminado

	formato digital.			
2.11	Probar los Módulos Desarrollados en la Iteración	Desarrollo	1	Terminado

Tabla 3.4 BackLog del Segundo Sprint
Fuente: [Elaboración Propia]

Las Funcionalidades correspondientes al incremento de esta segunda iteración son:

- Listado de índices almacenados según el pensum.
- Almacenamiento desde el cliente al lado del servidor de acta de notas en formato digital.
- Visualización de la documentación almacenada.

3.3.3 TERCERA ITERACIÓN

En esta iteración se desarrolla los módulos de búsqueda para las actas de notas correspondientes a los datos almacenados según parámetros, esta iteración se observa en la Tabla 3.4.

		SPRINT	INICIO	DURACIÓN
		3	13/08/2016	17 días
ID	TAREAS	TIPO	DIAS DE TRABAJO	ESTADO
2.1	Realizar la Planificación de la iteración	Planificación	2	Terminado
2.2	Analizar los Requerimientos del BackLog del Producto	Planificación	2	Terminado
2.3	Analizar los Requerimientos de la iteración con Historias de Usuario	Desarrollo	2	Terminado
2.4	Construir el modelo de contenido	Desarrollo	2	Terminado
2.5	Diseñar la base de Datos	Desarrollo	2	Terminado

2.6	Construir el Modelo de Usuario	Desarrollo	2	Terminado
2.7	Construir el Modelo de Navegación	Desarrollo	2	Terminado
2.8	Construir el Modelo de Proceso	Desarrollo	2	Terminado
2.9	Construir el Modelo de Presentación	Desarrollo	2	Terminado
2.10	Desarrollar el desarrolló los módulos de búsqueda para las actas de notas.	Desarrollo	6	Terminado
2.11	Probar los Módulos Desarrollados en la Iteración	Desarrollo	1	Terminado

Tabla 3.5 BackLog del Tercer Sprint

Fuente: [Elaboración Propia]

Las Funcionalidades correspondientes al incremento de esta tercera iteración son:

- Módulo de búsqueda por base de datos.
- Módulo de búsqueda por Web Services al Sistema de Seguimiento Académico.
- Módulo de Verificación de la información con los anteriores módulos.

3.3.4 CUARTA ITERACIÓN

En esta iteración se desarrolla los módulos para los reportes en relación a la información almacenada, además de la posibilidad de consumir servicios web de la base de datos del sistema de seguimiento académico, esta iteración se observa en la Tabla 3.4.

		SPRINT	INICIO	DURACIÓN
		4	28/08/2016	17 días
ID	TAREAS	TIPO	DIAS DE TRABAJO	ESTADO
2.1	Realizar la Planificación de la iteración	Planificación	2	Terminado
2.2	Analizar los Requerimientos del BackLog del Producto	Planificación	2	Terminado

2.3	Analizar los Requerimientos de la iteración con Historias de Usuario	Desarrollo	2	Terminado
2.4	Construir el modelo de contenido	Desarrollo	2	Terminado
2.5	Diseñar la base de Datos	Desarrollo	2	Terminado
2.6	Construir el Modelo de Usuario	Desarrollo	2	Terminado
2.7	Construir el Modelo de Navegación	Desarrollo	2	Terminado
2.8	Construir el Modelo de Proceso	Desarrollo	2	Terminado
2.9	Construir el Modelo de Presentación	Desarrollo	2	Terminado
2.10	Desarrollar el desarrolló los módulos de Reportes y consumo de Web Services de SSA.	Desarrollo	6	Terminado
2.11	Probar los Módulos Desarrollados en la Iteración	Desarrollo	1	Terminado

Tabla 3.6 BackLog de la Cuarta Iteración
Fuente: [Elaboración Propia]

Las Funcionalidades correspondientes al incremento de esta cuarta iteración son:

- Módulos de reportes extendidos a las unidades de kardex.
- Generación del Web Services, que consume los recursos de la base de datos del sistema de seguimiento académico que posteriormente almacena nuevos registros según el rango de fechas establecidas.

3.4 MODELADO DEL SISTEMA

3.4.1 HISTORIAS DE USUARIO

Las historias de usuario comprenden el Product Backlog del presente proyecto.

Dada la naturaleza evolutiva del alcance de un proyecto ágil, las historias de usuario de mayor prioridad se reflejan a más detalle que las de menor prioridad.

HISTORIA 1: Desarrollar el módulo de registro de docentes e índices				
PRI	Como...	Necesito...	Para...	Criterios de aceptación
1	ADMINISTRATIVO	Obtener la información de materias aprobadas	Tener en pantalla y verificar notas	Listar el contenido del plan de estudios actual
2	ADMINISTRATIVO	Aviso de habilitación para visualización	Poder acceder al archivo digital de nota	Se visualizara por pantalla
3	ADMINISTRATIVO	Un botón para subir un documento digital de acta de nota	Poder almacenar en el sistema	Se visualizara por pantalla los cambios efectuados
4	ADMINISTRATIVO	Generar formulario pre establecido y sincronizado	Minimizar errores de ingreso manual de datos	Generación de referencias a la base de datos del software

Tabla 3.7 Historia de Usuario Registro
Fuente: [Elaboración Propia]

En la Tabla 3.7 se puede observar la historia de usuario referente al módulo de seguimiento documental de cada libro de acta de notas y registro de docentes, lo que deriva en tener una concepción de datos que se debe almacenar, además que en relación a los datos de notas de estudiantes deben ser consumidos del Sistema Seguimiento Académico.

HISTORIA 2: Desarrollar el módulo de almacenamiento de las actas de notas en formato digital				
PRI	Como...	Necesito...	Para...	Criterios de aceptación
1	ADMINISTRATIVO	Obtener la información de materias por pensum	Tener en pantalla y hacer operaciones	Listar el contenido de materias según plan de estudios.
2	ADMINISTRATIVO	Seleccionador de paralelo de materia	Poder hacer corresponder con el respectivo docente	Se visualizara por pantalla
3	ADMINISTRATIVO	Un botón para seleccionar el archivo a cargar	Poder almacenar en el sistema en el respectivo paralelo	Se visualizara por pantalla los cambios efectuados
4	ADMINISTRATIVO	Generar un detalle de almacenamiento	Poder controlar archivos almacenados	Generación de referencias a la base de datos hora y fecha

Tabla 3.8 Historia de Usuario Almacenamiento
Fuente: [Elaboración Propia]

En la Tabla 3.8 se puede observar la historia de usuario referente al almacenamiento digital de las actas de notas, las mismas que serán administradas por las unidades de Kardex que trabajan con el proyecto SIA-UMSA.

HISTORIA 3:				
Desarrollar el desarrolló los módulos de búsqueda para las actas de notas.				
PRI	Como...	Necesito...	Para...	Criterios de aceptación
1	ADMINISTRATIVO	Un formulario donde introducir el nombre de docente	Tener información de las materias del docente en pantalla y hacer operaciones	Listar el contenido de materias que dicta el docente.
2	ADMINISTRATIVO	Seleccionador de paralelo de materia	Poder hacer corresponder con el respectivo docente	Se visualizara por pantalla
3	ADMINISTRATIVO	Un Web Service que enlace a la base de datos del SSA	Poder obtener datos de estudiante, materias y paralelos aprobados	Se visualizara por pantalla los datos del estudiante

Tabla 3.9 Historia de Usuario Búsqueda
Fuente: [Elaboración Propia]

En la Tabla 3.9 se puede Observar la historia de usuario referente a la búsqueda de notas pertinentes a historial académico del estudiante, según pensum de materias y la relación materia docente.

HISTORIA 4:				
Desarrollar el desarrolló los módulos de Reportes y consumo de Web Services de SSA.				
PRI	Como...	Necesito...	Para...	Criterios de aceptación
1	ADMINISTRATIVO	Filtrar los diferentes datos que se envía en el Web Service del sistema SSA	Tener información oportuna	Listar el contenido de materias de estudiante

2	ADMINISTRATIVO	Permitir el acceso a usuarios autorizados	Tener el control de usuarios que se requieren	Se atenderá solicitudes según correspondencia y aceptación de jefatura
3	ADMINISTRATIVO	Imprimir informe de documentos almacenados por docentes	Saber que han sido almacenados los archivos en el sistema	Se genera un reporte detallando que será visualizado por pantalla para que este pueda ser impreso
4	ADMINISTRATIVO	Optimizar el espacio de almacenamiento en memoria	Tener espacio de almacenamiento disponible	Se elabora un algoritmo que reduzca en memoria física conservando la calidad de la imagen en servidor

Tabla 3.10 Historia de Usuario Reportes

Fuente: [Elaboración Propia]

En la tabla 3.10 se puede observar la historia de usuario referente a la emisión de reportes que tiene el sistema web, específicamente los reportes de almacenamiento de los documentos digitalizados.

3.4.2 MODELO ESTRUCTURAL

Se presenta como se encuentran relacionados los contenidos del sistema web, es decir la estructura de datos que se hallan alojados en el sitio Web, el contenido de la información relevante al almacenamiento en el sistema, como se estructura y como se relaciona, se muestra en la Figura 3.2

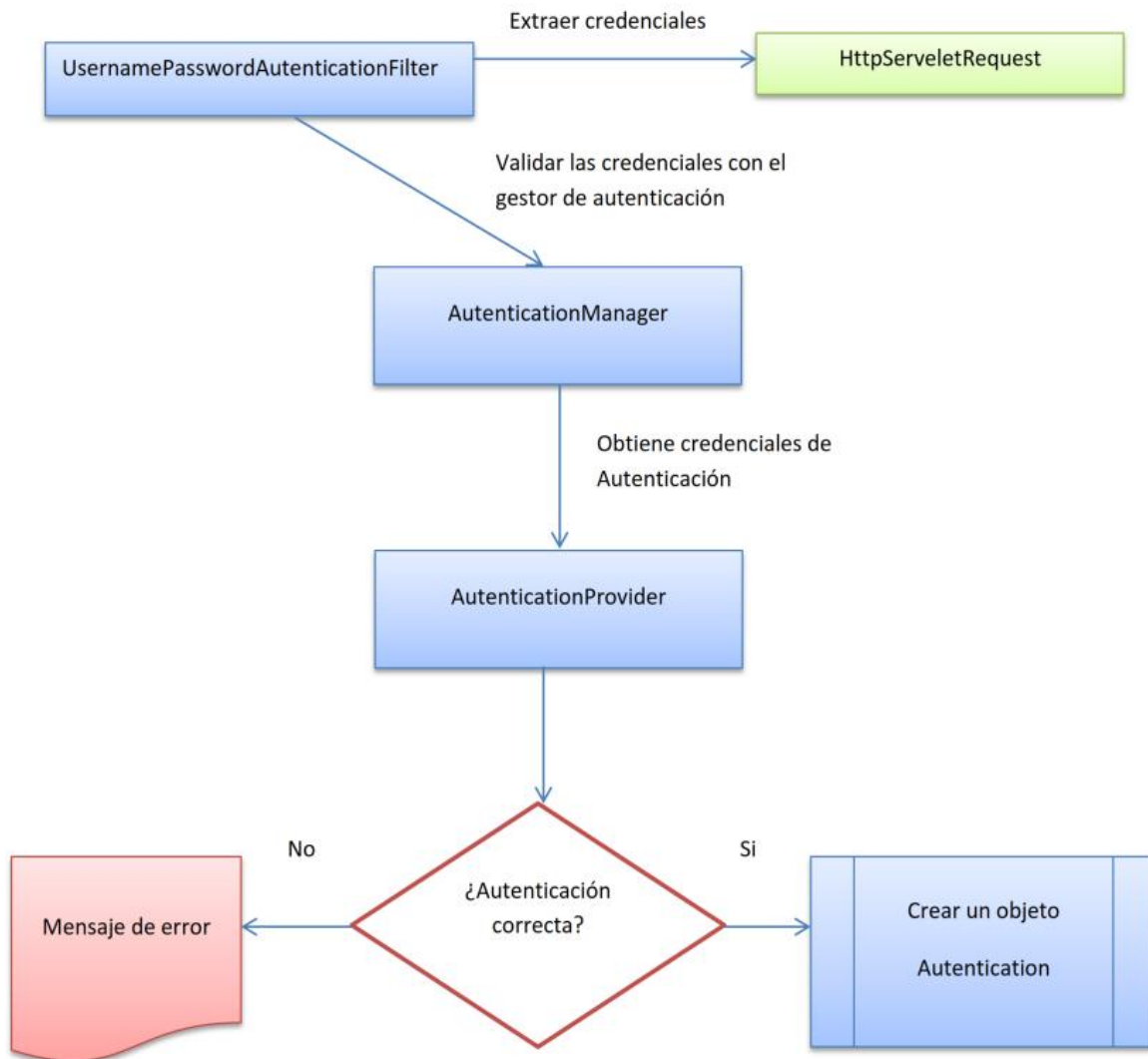


Figura 3. 3 Contexto de Sesión
Fuente: [Elaboración Propia]

3.4.4 MODELO DE NAVEGACIÓN

Representa la navegación en el sistema web, los cambios de vistas correspondientes a los módulos de solución, a continuación se muestra el modelo de navegación más representativo del sistema web en la Figura 3.4.

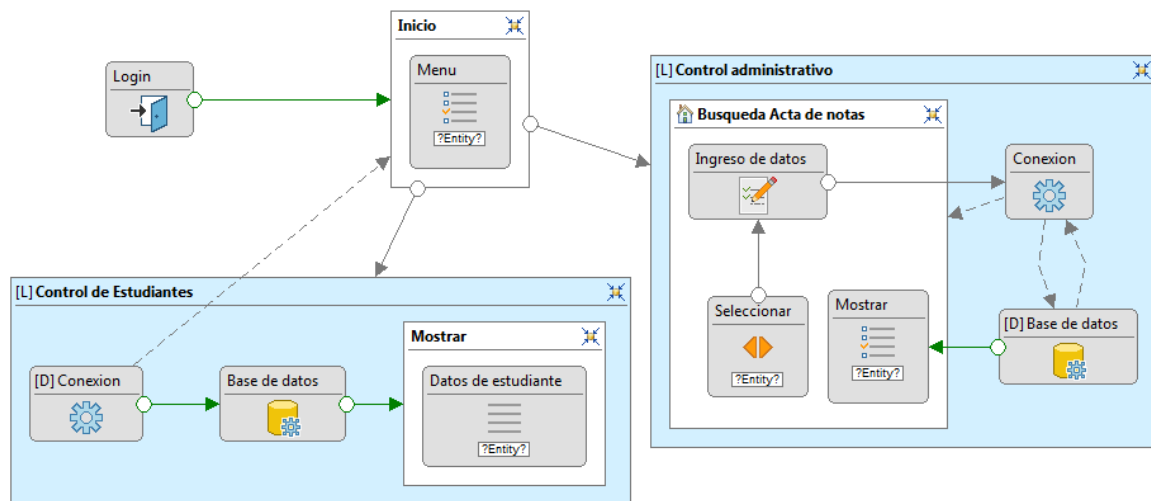


Figura 3. 4 Modelo de Navegación Control Administrativo y Estudiantil
Fuente: [Elaboración Propia]

3.4.5 DIAGRAMA DE COMPONENTES

Se definen los nodos que forman parte del hipertexto contenido del sistema web, es decir se especifica las páginas y las unidades (elementos atómicos de información) que componen el sistema.

3.4.5.1 BÚSQUEDA DE ESTUDIANTE

En la Figura 3.5 se muestra la búsqueda y filtrado de datos, como se utiliza estos para componer la interfaz de usuario detallada en el modelo de presentación.

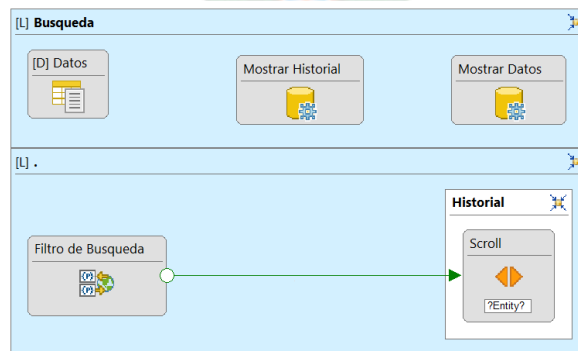


Figura 3. 5 Modelo de Componente Búsqueda Estudiante
Fuente: [Elaboración Propia]

3.4.5.2 ALMACENAMIENTO EN FORMATO DIGITAL

En la Figura 3.6 se muestra la asignación de un archivo en formato digital, a una correspondiente materia, con búsqueda previa.

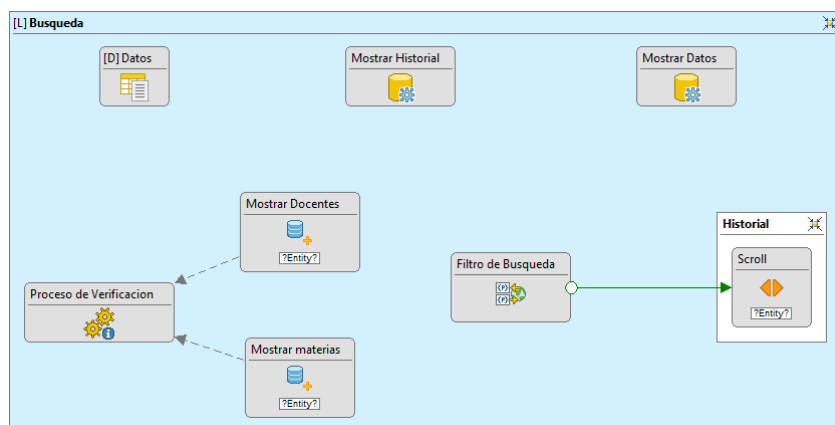


Figura 3. 6 Modelo de Componente Almacenamiento
Fuente: [Elaboración Propia]

3.4.5.3 EMISIÓN DE REPORTES DE INFORMACIÓN

En la Figura 3.7 se observa la composición de la interfaz en relación a la impresión de reportes de información.

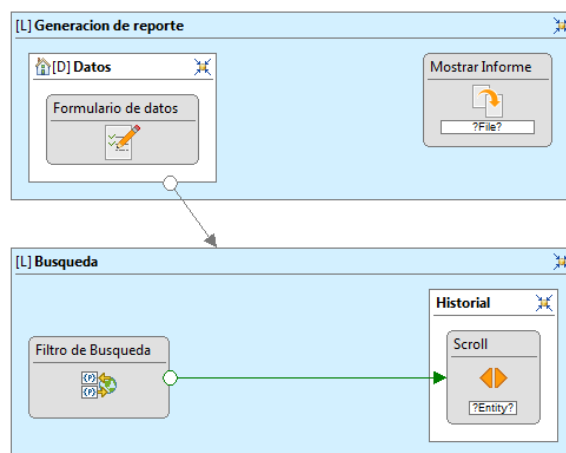
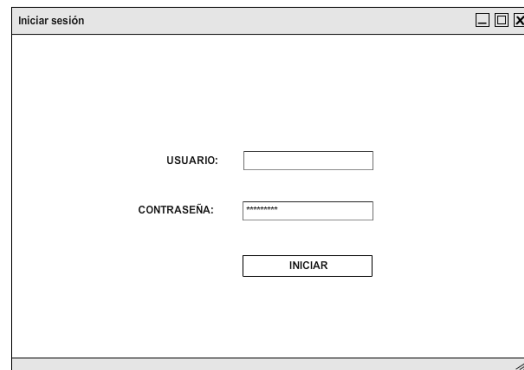


Figura 3. 7 Modelo de Componente Reporte
Fuente: [Elaboración Propia]

3.4.6 MODELO DE PRESENTACIÓN

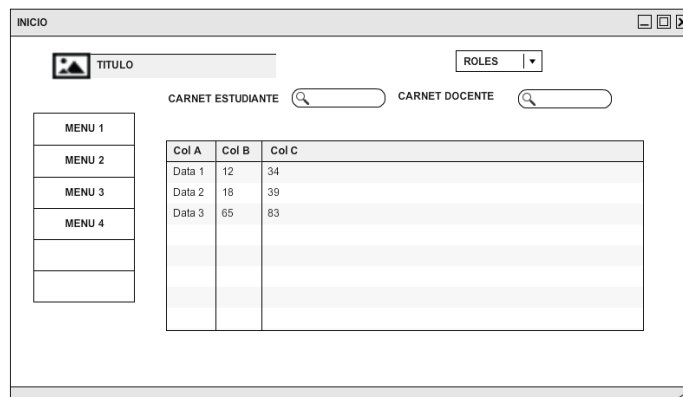
Se contempla las clases de navegación y procesos que pertenecen a cada página web. Las propiedades que contiene por composición se representan como un rectángulo en el que un elemento queda contenido con otro.



The screenshot shows a window titled "Iniciar sesión" with a standard Windows-style title bar. Inside the window, there are three main elements: a label "USUARIO:" followed by a text input field, a label "CONTRASEÑA:" followed by a password input field (indicated by asterisks), and a button labeled "INICIAR" centered below the input fields.

Figura 3. 8 Modelo de Presentación Login
Fuente: [Elaboración Propia]

En la Figura 3.8 se observa el modelo de presentación que hace referencia al ingreso de forma segura, puesto que el usuario debe contar con un usuario y una contraseña.



The screenshot shows a window titled "INICIO" with a standard Windows-style title bar. The interface includes a search bar labeled "TITULO" with a magnifying glass icon, a dropdown menu labeled "ROLES", and two search input fields labeled "CARNET ESTUDIANTE" and "CARNET DOCENTE" with magnifying glass icons. On the left side, there is a vertical menu with four items: "MENU 1", "MENU 2", "MENU 3", and "MENU 4". In the center, there is a table with three columns: "Col A", "Col B", and "Col C". The table contains three rows of data:

Col A	Col B	Col C
Data 1	12	34
Data 2	18	39
Data 3	65	83

Figura 3. 9 Modelo de Presentación Pantalla inicial
Fuente:[Elaboración Propia]

En la Figura 3.9 se observa el modelo de presentación que hace referencia a formulario de búsqueda de datos de estudiante y docentes, la cual es la pantalla inicial.

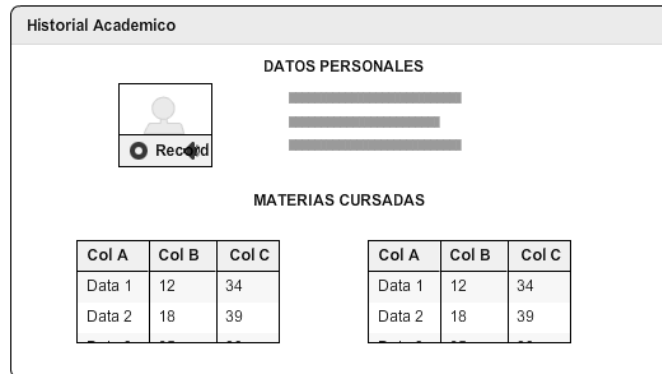


Figura 3. 10 Modelo de presentación Datos Estudiante

Fuente: [Elaboración propia]

En la Figura 3.10 observamos el modelo de presentación que hace referencia al panel que muestra datos del estudiante en una búsqueda previa.

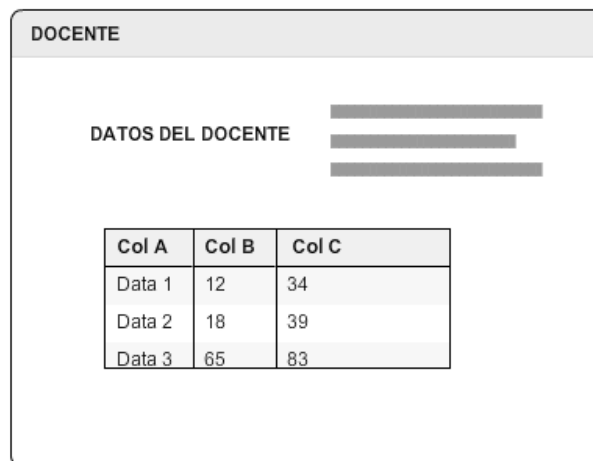


Figura 3. 11 Modelo de presentación Datos Docente

Fuente: [Elaboración Propia]

En la Figura 3.11 observamos el modelo de presentación que hace referencia al panel de datos del docente, materias y paralelos en una búsqueda previa.

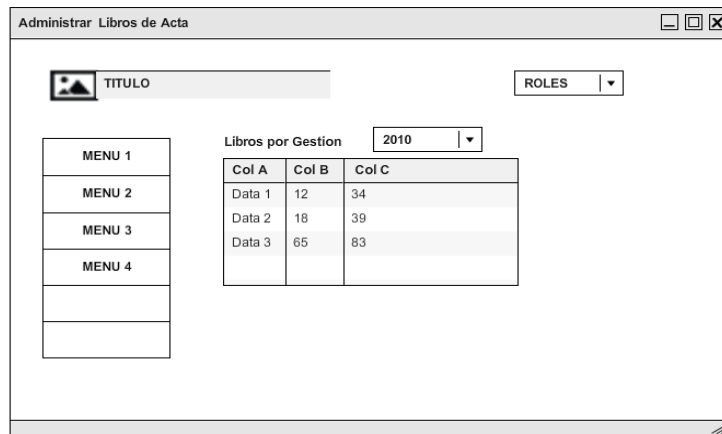


Figura 3. 12 Modelo de presentación Libro de actas

Fuente: [Elaboración Propia]

En la Figura 3.12 se observa el modelo de presentación que hace referencia al panel de Administrador de Libro de acta de Notas.

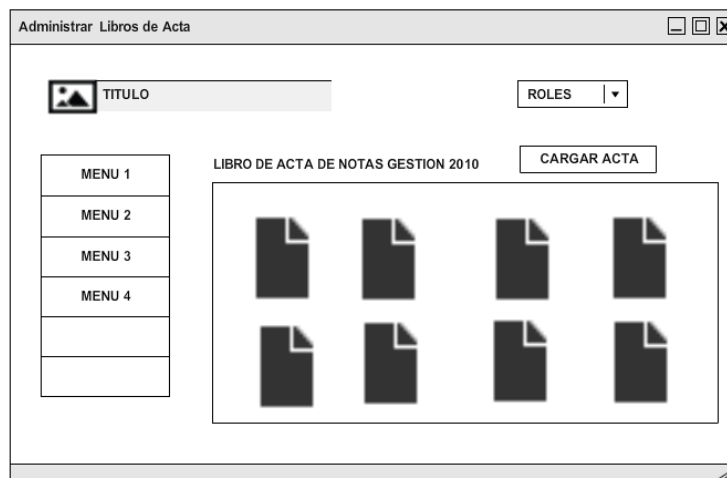


Figura 3. 13 Elaboración propia Despliegue de Actas

Fuente: [Elaboración propia]

En la Figura 3.13 se observa el modelo de presentación que hace referencia al panel de despliegue de la información contenida con los archivos que comprenderán Acta de Notas.

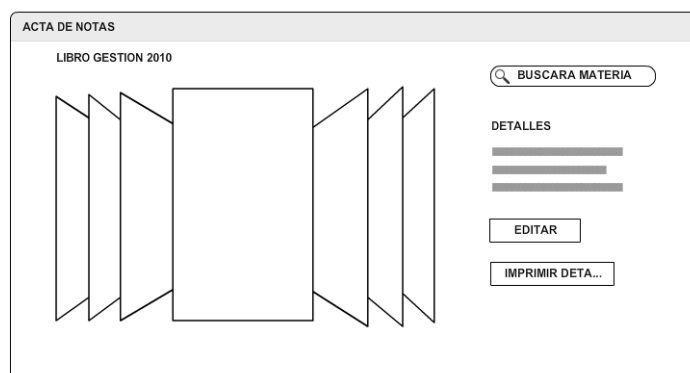


Figura 3. 14 Modelo de presentación detalle actas

Fuente: [Elaboracion propia]

En la Figura 3.11 se observa el modelo de presentación que hace referencia al panel de detalle de cada Archivo almacenado en el sistema, la descripción hora y fecha de almacenamiento.

3.5 POST GAME

En esta etapa final se realiza las actividades de prueba de software web, se propusieron métricas de calidad que se pueden apreciar en el capítulo IV de presente proyecto de grado.

3.6 MODELO DE IMPLEMENTACIÓN DEL SISTEMA

3.6.1 DISEÑO DE INTERFACES

A continuación se presenta las interfaces de usuarios los mismos que están distribuidos en los diferentes módulos, entre estos tenemos:

Ventana Login. Ventana de inicio y verificación de contraseña.

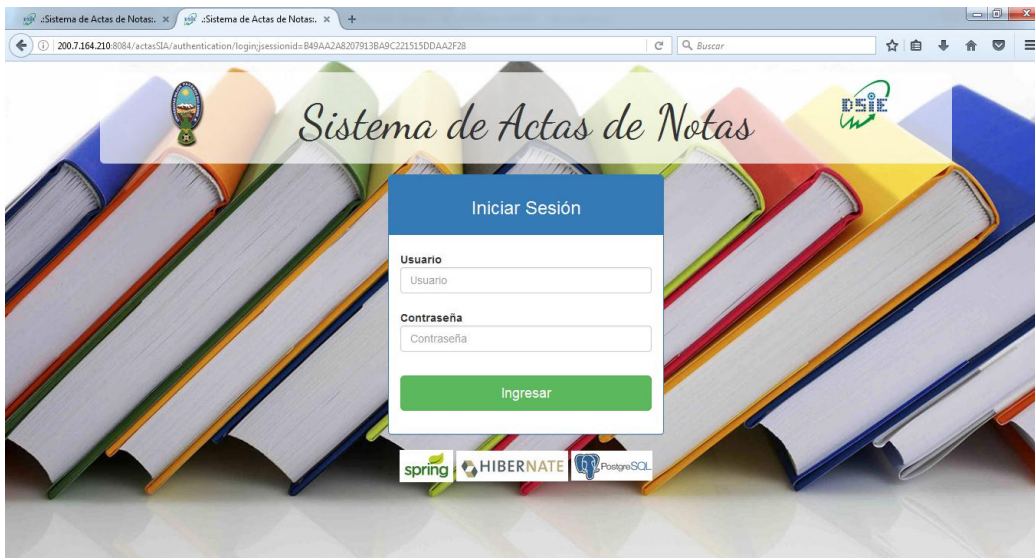


Figura 3. 15 Ingreso al Software
Fuente: [Elaboración Propia]

En la Figura 3.15 se presenta la pantalla de inicio de sesión en la que se toma encuentra el contexto de sesión vista en la Figura 3.3 anteriormente.

Ventana Roles. Ventana de selección de roles que se asigna a un usuario determinado como se ve en la Figura 3.16.

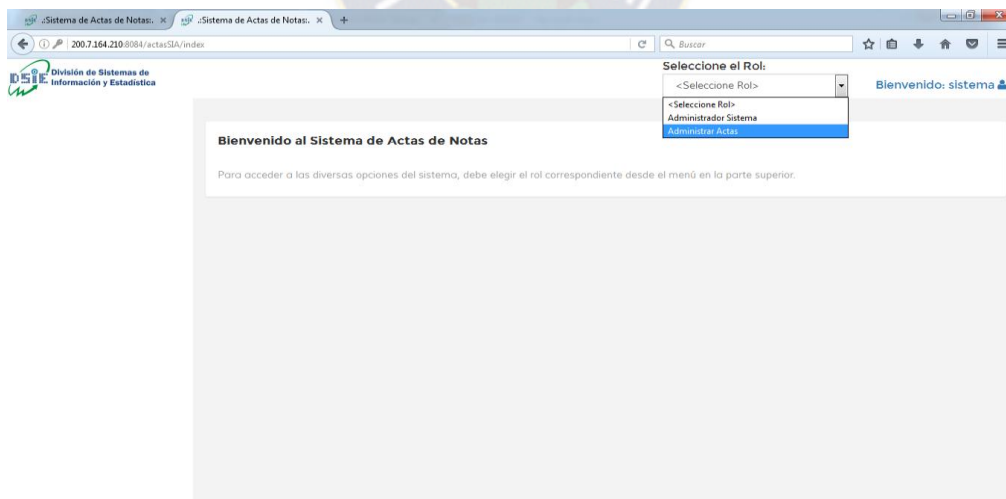


Figura 3. 16 Pantalla de Bienvenida
Fuente: [Elaboración Propia]

Ventana Búsqueda. Ventana comprende la interfaz de búsqueda de personas docentes y estudiantes como se ve en la Figura 3.17.

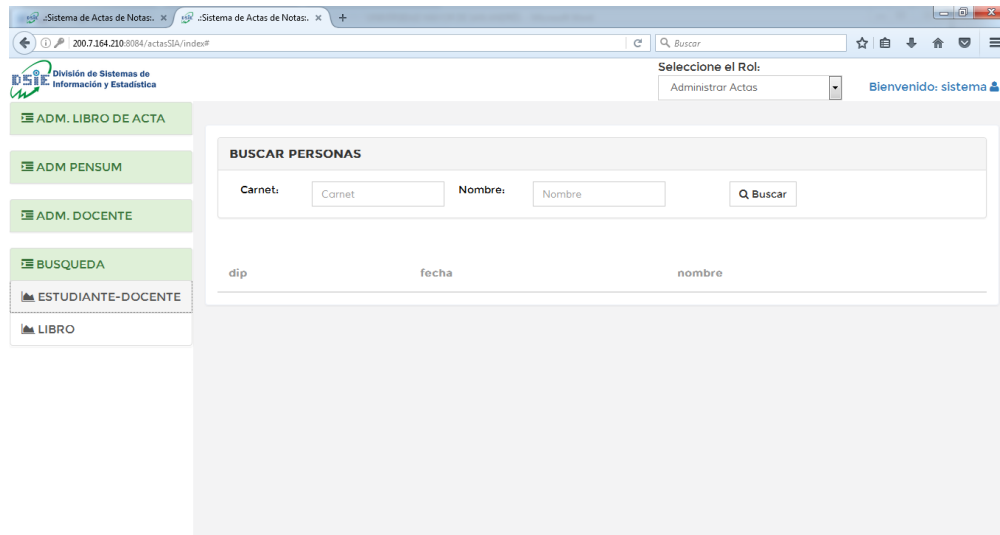


Figura 3. 17 Ventana de búsqueda ESTUDIANTE-DOCENTE

Fuente: [Elaboración Propia]

Ventana de Listado. Ventana en la que se detalla a docentes y la posibilidad de registrar nuevo Docente, ver la Figura 3.18.

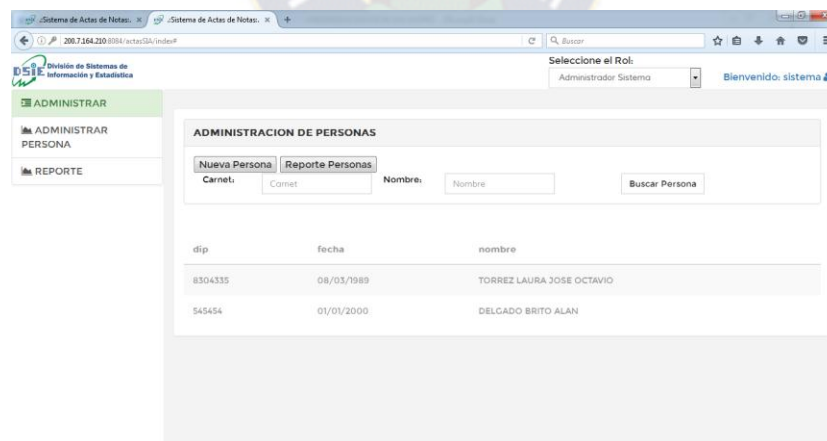


Figura 3. 18 Ventana de Listado Docente

Fuente: [Elaboración Propia]

Panel de Registro. Panel que permite el registro de nuevo docente como se muestra en la Figura 3.19.

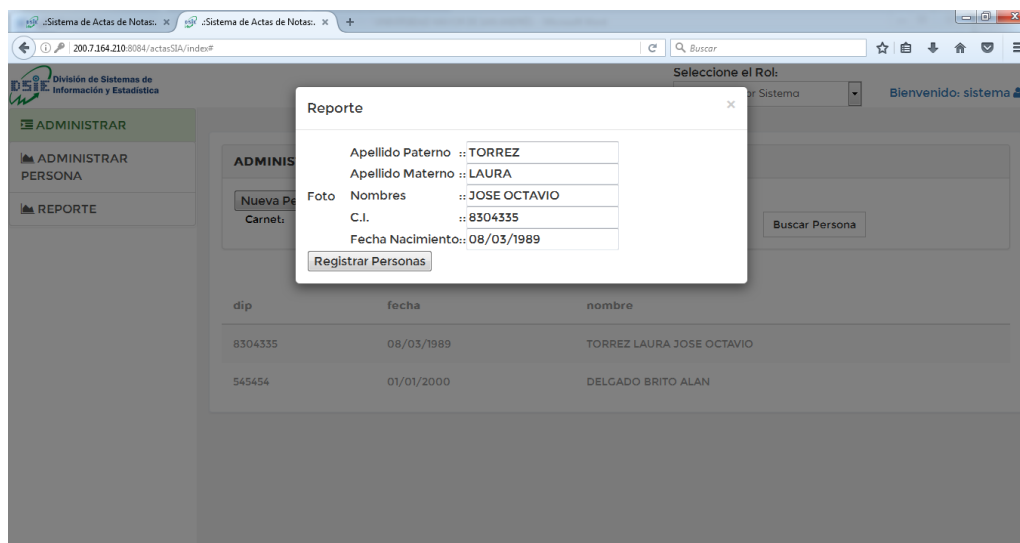


Figura 3. 19 Registro Docente
Fuente: [Elaboracion propia]

Panel de Reporte. Panel que muestra la generación de reporte correspondiente a lista de registro de docentes como se muestra en la Figura 3.20.

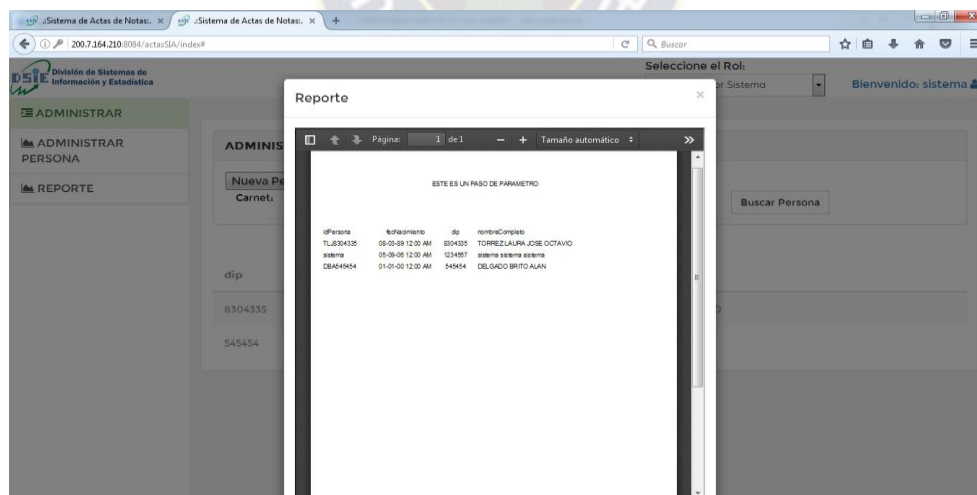


Figura 3. 20 Panel de Reporte
Fuente: [Elaboración Propia]

CAPÍTULO IV

CALIDAD Y SEGURIDAD DEL SOFTWARE

4.1 CALIDAD DE SOFTWARE

En gran parte de los Ingenieros de Sistemas, se suele estimar las funcionalidades, en relación al tamaño físico. El concepto de funcionalidad extrae relación a la cantidad de funciones contenidas en un producto entregado o en la descripción que se supone sería el producto [Fenton, 1997]. Las métricas orientadas a las funcionalidades son medidas indirectas del software y del proceso por el cual se desarrolla. Estas métricas se centran en la “Funcionalidad” o “utilidad” del software [Pressman, 2002].

En el presente proyecto se aplican métricas de calidad según el estándar ISO 9126.

4.2 CALIDAD DE SOFTWARE (NORMA ISO-9126)

La calidad de software puede ser evaluado midiendo atributos internos (medidas estáticas de productos intermedios), o puede ser evaluado midiendo atributos externos (medida de comportamiento del código cuando se está ejecutando).

La ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software, el cual fue publicado en 1992 con el nombre de “Information Technology – Software Product Evaluation: Quality characteristics and guide lines for their use”, en el cual establece las características básicas, que son: funcionalidad , confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detallan a través de un conjunto de sub características que permiten profundizar la evolución de la calidad de productos de software. [ABUD, (n.d.)]

Como se muestra ver en la Figura 4.1.

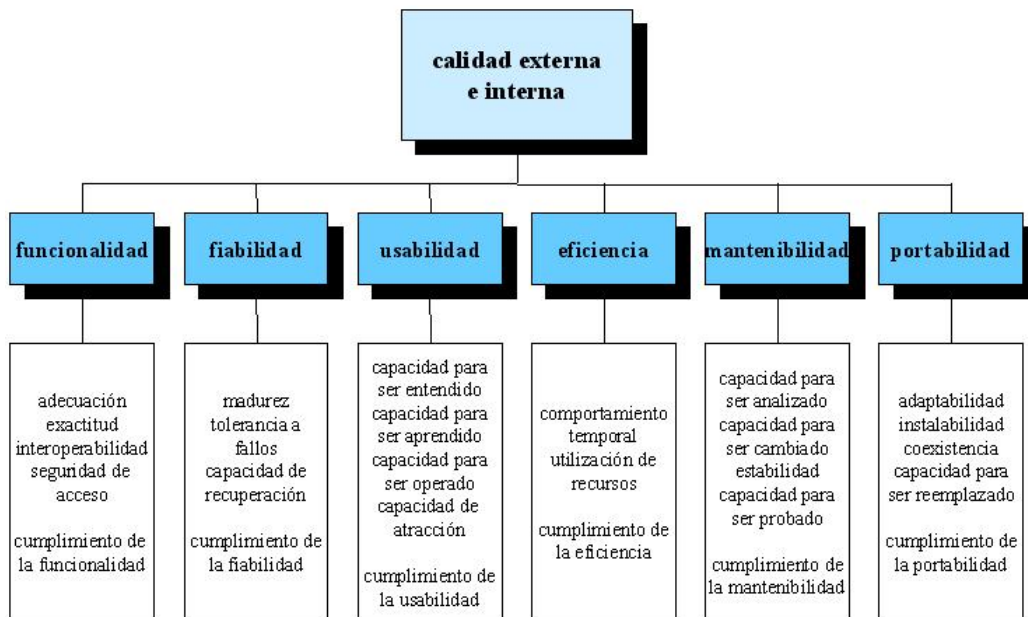


Figura 4. 1 Características ISO 9126
Fuente: [ABUD, (n.d.)]

4.2.1 FUNCIONALIDAD.

Las funcionalidades del sistema se mide a través de su complejidad, la funcionalidad no se puede ser medida directamente, entonces corresponde utilizar otras medidas directas como punto función, que permite un resultado medible y cuantificable a partir de la siguiente fórmula:

$$PF = CuentaTotal \times \left(0.65 + 0.01 \left(\sum F_i \right) \right)$$

Para la medición de la funcionalidad, se tomara en cuenta la siguiente característica, como se muestra en la Tabla 4.1:

Parámetros de Medida	Cuenta	Factor de peso			Total
		Simple	Medio	Complejo	
Número de entradas de usuario	62	3	4	6	248
Número de salidas de usuario	17	4	5	7	85
Número de peticiones de usuario	14	3	4	6	56
Número de Archivos	42	7	10	15	420
Número de Interfaces	0	5	7	10	0
				CUENTA TOTAL	809

Tabla 4. 1 Calculo Punto Función

Fuente: [Elaboración Proyecto]

Los valores de la variable Fi, se obtiene de los resultados que se aprecian en la Tabla 4.3 bajo la siguiente ponderación que se muestra en la Tabla 4.2:

Escala	Complejidad
1	Sin influencia
2	Incidental
3	Moderado
4	Medio
5	Significativo
6	Esencial

Tabla 4. 2 Valores de Ajuste de Complejidad

Fuente: [Elaboración Propia]

Pregunta	Ponderación
¿Requiere el sistema copias de seguridad y recuperación fiables?	5
¿Se requiere comunicación de datos?	5
¿Existen funciones de procesamiento distribuido?	3

¿Es crítico el rendimiento?	3
¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?	5
¿Requiere el sistema entrada de datos interactiva?	5
¿Requiere la entrada de datos interactiva par que estas transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	5
¿Se actualizan los archivos maestros de forma interactiva?	3
¿Son complejos las entradas, las salidas, los archivos de peticiones?	3
¿Es complejo el procedimiento interno?	5
¿Se ha diseñado el código para que sea reutilizable?	4
¿Están incluidas en el diseño la conversión y la instalación?	5
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
¿Se ha diseñado la aplicación Web para facilitar los cambios y para ser fácilmente utilizado por otro usuario?	5
$\sum(F_i)$ Total	61

**Tabla 4. 3 Ajuste de Complejidad Punto
Función Fuente:[Elaboración Propia]**

Empleando la fórmula para hallar el PF y PF(Máximo):

$$PF = 809 * [0.65 + 0.01 * 61]$$

$$PF = 1019.34$$

$$PF(Maximo) = 809 * [0.65 + 0.01 * 70]$$

$$PF(Maximo) = 1092.15$$

Con los valores máximos de ajuste de complejidad de Punto Función, se obtiene el siguiente resultado de Funcionalidad real:

$$Funcionalidad = \left(\frac{1019.34}{1092.15} \right) * 100$$

$$Funcionalidad = 93.33 \%$$

Por tanto la funcionalidad del sistema es de 93.33 por ciento, teniendo en cuenta el punto función máximo, lo que quiere decir que el sistema web cumple satisfactoriamente con los requerimientos funcionales de forma satisfactoria.

4.2.2 FIABILIDAD

La fiabilidad del Sistema se define como: “probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico” [Pressman, 2002]. Una medida de fiabilidad es el tiempo medio entre fallos, con la siguiente ecuación:

$$TMEF = TMDF + TMDR$$

Dónde:

TMEF: Tiempo medio entre fallos.

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

En una escala de trabajo de 8hrs el sistema web es capaz de ser operable y el tiempo de reparación de cuanto a fallo es de 42 minutos lo que es aproximadamente 0.7hrs.

Entonces la probabilidad de que el sistema funcione de acuerdo a los requisitos en un momento dado se define como:

$$Fiabilidad = \frac{T MDF}{T MDF + T MDR} = \frac{8 \text{ hrs}}{8 \text{ hrs} + 0.7 \text{ hrs}} = 0.9195 = 91.95\%$$

Por tanto el sistema web obtuvo una fiabilidad de 91.95%.

4.2.3 USABILIDAD

Cuando se habla de usabilidad, se espera que el sistema sea de fácil entendimiento y aprendizaje. Cabe resaltar que para la norma ISO 9126, la usabilidad no es afectada por la funcionalidad y eficiencia. La usabilidad está definida por los usuarios finales o personas afines al sistema. Para la medición de la usabilidad se tiene el siguiente cuestionario con los ajustes necesarios que pueden observar en las Tablas 4.4 y Tabla 4.5:

Descripción	Escala
Pésimo	1
Malo	2
Regular	3
Bueno	4
Muy Bueno	5

Tabla 4. 4 Escala de Ajuste de Usabilidad
Fuente: [Elaboración Propia]

Factor	Valor
¿Se ha cubierto los requerimientos establecidos por el sistema?	5
¿Es sencillo acceder a los datos?	5
¿Presenta suficiente ayuda durante el tiempo que accede al sistema web?	4
¿Los informes son suficientemente representativos?	5
¿El sistema tiene la seguridad necesaria?	5
¿Está de acuerdo con el funcionamiento del sistema?	4
¿El sistema facilita el trabajo que realiza?	4
TOTAL	32

Tabla 4. 5 Evaluación de Usabilidad
Fuente: [Elaboración Propia]

La Usabilidad se calcula por medio de la siguiente formula:

$$USABILIDAD = \left[\frac{\sum \text{valor}}{n} \times 100 \right] \div 5$$

$$USABILIDAD = \left[\frac{32}{7} \times 100 \right] \div 5$$

$$USABILIDAD = 91.42 \%$$

Por tanto la usabilidad del sistema web corresponde a un 92 por ciento que se interpreta, como la facilidad del usuario al interactuar con las interfaces. Es decir que de cada 100 usuarios 91 fueron capaces de entender y operar el sistema con un nivel de atracción alto.

4.2.4 EFICIENCIA

La eficiencia es el grado que la aplicación hace optimo el uso de recursos del sistema, indicada por los tiempos de uso y recursos utilizados. [Pressman, 2002]

Para la medición de la usabilidad se tiene el siguiente factor de ajuste en la Tabla 4.6:

Nro.	Factor de Ajuste	Valor %
1	Es la respuesta rápida al utilizar sus funciones	95
2	Tiene rendimiento de acuerdo a los factores que utiliza.	90
3	Responde adecuadamente cuando utiliza sus funciones.	92
4	El tiempo de respuesta a sus consultas es adecuado.	95
	Promedio	93

Tabla 4. 6 Valor de Ajuste según Pressman

Fuente: [Elaboración propia]

Por tanto, el sistema web obtuvo una eficiencia de 93%.

4.2.5 MANTENIBILIDAD

La mantenibilidad es la característica que representa la capacidad del sistema a ser modificado a nivel funcionalidad, correcciones de entorno cambios y mejoras del sistema web.

Para la puntuación de la mantenibilidad, se desarrollaron las siguientes preguntas, las mismas que fueron contestadas por el desarrollador del sistema web y son promediadas, como se muestra en la Tabla 4.7.

FACTOR DE AJUSTE	VALOR
¿Puede ser mejorado el sistema?	95
¿Deja identificar las partes que deben ser modificadas?	95
¿Permite implementar una modificación específica?	90
¿No presenta efectos inesperados en posibles errores?	95
TOTAL	93,75

Tabla 4. 7 Factores de Ajuste de Mantenibilidad

Fuente: [Elaboración propia]

$$MANTENIBILIDAD = 93.75 \%$$

Lo que refleja como resultado que un 94% aproximadamente en mantenibilidad, podemos asegurar que el sistema web contempla una cantidad de esfuerzo mínimo para cualquier cambio o modificación que requiera.

Por tanto existe una probabilidad de un 94% de que un desarrollador pueda realizar cambios con facilidad en el código o estructura interna del sistema propuesto.

4.2.6 PORTABILIDAD

El hecho de que el sistema web está desarrollado en el lenguaje de programación JAVA nos ayuda de sobre manera en relación a la portabilidad.

Hoy en día existen muchas computadoras conectadas a internet, con diferentes arquitecturas y sistemas operativos; es deseable tener un mecanismo que permita generar código y ser

interpretado en cualquier plataforma y/o sistema operativo. Para este contenido la máquina virtual de JAVA se encarga de interpretar y ejecutar instrucciones expresadas en un código binario especial conocido como ByteCode de JAVA.

Realizando instalación de la aplicación web en diferentes sistemas operativos se detalla de la siguiente manera:

- Windows XP, con éxito.
- Windows 8, con éxito.
- Windows 10, con éxito.
- Linux Ubuntu 12.0, con éxito.
- Linux Devian, con éxito.
- Linux lime, con éxito.
- Windows server 2008, con éxito.
- Windows vista, con éxito.
- Linux fedora, con éxito.
- Red hat, con fallo por términos de licencia.

Es por ello que la portabilidad del sistema viene dada de la siguiente forma:

$$x = \frac{\text{Numero de casos exitosos de instalacion}}{\text{Numero de casos de intento de instalacion}}$$

$$x = \frac{9}{10} = 0.9$$

Lo que indica que el sistema es portable con facilidad en un 90 % de los casos de instalación con los elementos necesarios para que el sistema este puesto en funcionamiento y pueda ejecutarse sin ningún problema.

4.3 CALIDAD TOTAL

Para obtener la calidad total del software, sacaremos la medida de todas las medidas expresadas en porcentaje, como se muestra en la Tabla 4.8.

CARACTERISTICAS	RESULTADOS EN %
FUNCIONALIDAD	93,33
FIABILIDAD	91,95
USABILIDAD	91,43
EFICIENCIA	93
MANTENIBILIDAD	93,75
PORTABILIDAD	90
TOTAL PROMEDIO	92,24

Tabla 4. 8 Resultado calidad Total Norma ISO-9126

Fuente: [Elaboración propia]

La calidad total del sistema web corresponde al 92.24%, lo que se interpreta como la satisfacción que tiene el usuario el interactuar con el sistema.

4.4 SEGURIDAD DEL SOFTWARE

4.4.1 POLÍTICAS DE SEGURIDAD (USUARIOS)

Administrador. Puede acceder a la información a través del gestor de base de datos del sistema, puede modificar, eliminar, depurar registros, administrar archivos almacenados, manejar las sesiones de usuarios.

Kardixta. Es el encargo de registra docentes, pensum y sus respectivas materias asignar paralelos con los respectivos docentes, almacenar los archivos de las actas de notas con las firmas respectivas por las autoridades pertinentes y final mente tener un registro de almacenamiento.

Autoridades Universitarias. Serán los encargados de consultar información académica del estudiante respecto a la carrera correspondiente a verificar, las autoridades con más alta jerarquía tendrán acceso a todas las carreras y facultades de la universidad que trabajan con el proyecto SIA-UMSA como herramienta de transparencia en el desarrollo académico de la comunidad estudiantil.

4.4.2 POLÍTICAS DE SEGURIDAD DE ACCESO AL SOFTWARE

El sistema verifica la autenticidad del usuario y su contraseña, los usuarios deben guardar la información de su cuenta de usuario y su respectiva contraseña, en caso de olvido se debe solicitar el restablecimiento de la cuenta de usuario al administrado del sistema.

El acceso a la base de datos es restringido para usuarios no autorizados, mediante permisos de acceso al servido y roles específico en la base de datos, además de usuario y contraseña para acceso al servidor.

CAPÍTULO V

ANÁLISIS COSTO - BENEFICIO

5.1 ANÁLISIS DE COSTOS

En este punto cuantificaremos la inversión de los recursos que se empleó en el desarrollo del sistema web. Para el cálculo de esfuerzo y costo de desarrollo se utilizara el modelo COCOMO.

5.2 COCOMO

El modelo constructivo de costes (COCOMO) es un modelo matemático de base empírica, utilizado para la estimación de costes de elaboración de Software. Incluye tres sub modelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo de Software: básico, intermedio y detallado. [COCOMO II, 2000]

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiendo detalladamente en su libro "Software Engineering Economics", COCOMO se apoya en tres modelos de estimación. En los tres modelos se utilizan las siguientes ecuaciones. [COCOMO II, 2000]

$$E = a(KLDC)^b, \text{ en personas} \times \text{mes}$$

$$D = c(E)^d, \text{ en meses}$$

$$P = \frac{E}{D}, \text{ en personas}$$

Dónde:

E: Esfuerzo requerido por el proyecto en meses.

D: Es tiempo requerido por el proyecto en meses.

P: Número de personas requerido por el proyecto en meses.

a, b, c, d: Constantes con valores definidos según modelo.

KLDC: Cantidad de líneas de código, en miles.

A su vez cada modelo se subdivide en modos, como ser:

Modo orgánico. Es un pequeño grupo de programadores experimentados desarrollando software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).

Modo semi-libre o semi-acoplado. Corresponde a un esquema intermedio entre el orgánico y rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

Modo rígido o empotrado. El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único.

Siendo difícil basarse en la experiencia puesto que esta no puede existir.

Para determinación del costo de software desarrollado utilizaremos la Tabla 5.1.

PROYECTO DE SOFTWARE	a^b	b^b	c^b	d^b
ORGANICO	2,4	1,05	2,5	0,38
SEMI - ACOPLADO	3	1,12	2,5	0,35
RIGIDO	3,6	1,2	2,5	0,32

Tabla 5. 1 Modelo COCOMO
Fuente: [Pressman, 2002]

También es necesario el punto función PF calculado en el capítulo anterior (PF=1019.34). Ahora se debe convertir los puntos función en miles de líneas de código (PF a KLCD), tomando en cuenta la tabla 5.2.

Language	Default SLOC / UFP	Language	Default SLOC / UFP
Access	38	Jovial	107
Ada 83	71	Lisp	64
Ada 95	49	Machine Code	640
AI Shell	49	Modula 2	80
APL	32	Pascal	91
Assembly - Basic	320	PERL	27
Assembly - Macro	213	PowerBuilder	16
Basic - ANSI	64	Prolog	64
Basic - Compiled	91	Query – Default	13
Basic - Visual	32	Report Generator	80
C	128	Second Generation Language	107
C++	55	Simulation – Default	46
Cobol (ANSI 85)	91	Spreadsheet	6
Database – Default	40	Third Generation Language	80
Fifth Generation Language	4	Unix Shell Scripts	107
First Generation Language	320	USR_1	1
Forth	64	USR_2	1
Fortran 77	107	USR_3	1
Fortran 95	71	USR_4	1
Fourth Generation Language	20	USR_5	1
High Level Language	64	Visual Basic 5.0	29
HTML 3.0	15	Visual C++	34
Java	53		

Tabla 5. 2 Conversión de Puntos de Función
Fuente: [COCOMO II, 2000]

Lo que lleva a hacer el cálculo de la siguiente ecuación:

$$LDC = PF \times Factor\ SLOC/UFP$$

El proyecto utiliza el lenguaje de programación JAVA, entonces reemplazando los valores quedara el siguiente cálculo:

$$LDC = 1019.34 \times 53$$

$$LDC = 54025,02$$

Las líneas de código en su totalidad son 54025,02 de las cuales se estima que el 95% del código es reutilizable, en ese cometido el KLDC será:

$$KLDC = (Total\ LDC - Total\ LDC\ reutilizable)/1000$$

$$KLDC = (54025.02 - (54025.02 \times 0.95))/1000$$

$$KLDC = 2.70$$

Por tanto existen 2.7 líneas de código distribuidas para el proyecto software. Ahora aplicándolas formulas básicas de esfuerzo, tiempo calendario y personal requerido.

$$E = a(KLDC)^{b^b}$$

$$D = c^b (E)^{a^b}$$

Remplazando tenemos:

$$E = 2.4 \times 2.7^{1.05}$$

$$E = 6.81 \text{ personas/mes}$$

Redondeando:

$$E = 7 \text{ personas/meses}$$

Estimación de tiempo necesario:

$$D = 2.5 \times 7^{0.38}$$

$$D = 5.23 \text{ meses}$$

Redondeando considerando que nos referimos a tiempo:

$$D = 6 \text{ meses}$$

Calculo de número de personas para el proyecto tenemos:

$$P = \frac{E}{D}$$

$$P = \frac{7}{6} = 1.1666$$

Redondeo considerando que se refiere a personas:

$$P = 2 \text{ personas}$$

Realizando un estudio de mercado laboral, se toma en cuenta que un ingeniero informático percibe un salario de 400 a 800\$us (Dólares Americanos), que expresado en moneda nacional será aproximadamente entre 3.500 a 8.600Bs (Peso Boliviano). Así podemos calcular el costo del sistema:

$$\text{costo de software} = \text{salarioMinimo(Media)} \times \text{Tiempo necesario} \times \# \text{ personas}$$

$$\text{costo de software} = \left(\frac{3500 + 8600}{2} \right) \times 6 \times 2$$

$$\text{costos de software} = 72600$$

5.3 VALOR ACTUAL NETO

El VAN o valor actual neto 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.

La fórmula que utilizaremos para hallar el valor actual neto será:

$$VAN = \sum \frac{Ganancias}{(1+k)^n} - \sum \frac{Costos}{(1+k)^n}$$

Dónde:

VAN: Valor Actual Neto.

Ganancias: Ingreso de Flujo anual.

Costos: Salidas de Flujo anual.

n: Periodo.

k: Tasa de descuento o tasa de interés al préstamo.

Los costos y ganancias que se estiman para un lapso de 4 años se evidencian en la Tabla 5.3, para este caso usaremos una tasa de descuento del 12% ya que esta tasa actual de interés de préstamos en entidades financieras.

AÑO	COSTOS	GANANCIA	COSTOS/(1+k)^n	GANANCIA/(1+k)^n	RESULTADO
1	72600	0	64821,43	0	-64821,4
2	12600	21000	10044,64	16741,1	6696,4
3	9350	42000	6655,15	29894,8	23239,6
4	4550	63000	2891,61	40037,6	37146,0
Σ	99100	126000	84412,82	86673,5	2260,7

Tabla 5. 3 Calculo de VAN
Fuente: [Elaboración propia]

$$VAN = \sum \frac{Ganancias}{(1 + 0.12)^n} - \sum \frac{Costos}{(1 + 0.12)^n} = 2260.7$$

La tabla 5.4 muestra si un proyecto es rentable y de acuerdo a ciertos criterios más el valor del VAN concluiremos si es rentable o no.

Valor del VAN	Interpretación
VAN>0	El proyecto es rentable
VAN=0	El proyecto es rentable, porque ya está incorporada la tasa de interés.
VAN<0	El proyecto no es rentable

Tabla 5. 4 Interpretación del VAN

Fuente: [Elaboración Propia]

De esta manera podemos considerar que el VAN = 2260.7 y siguiendo el criterio de la Tabla 5.4 se afirma que el presente proyecto es rentable ya que 2260 es mayor a 0.

5.3.1 COSTO – BENEFICIO

Para hallar el costo – beneficio aplicaremos la siguiente ecuación:

$$Costo\ Beneficio = \frac{\sum Beneficio}{\sum Costos}$$

$$Costo\ Beneficio = \frac{126000}{99100} = 1.27$$

Con este resultado se puede interpretar que por cada Peso Boliviano invertido en el proyecto de software, la institución genera una ganancia de 0.27 Bolivianos.

5.4 TASA INTERNA DE RETORNO

Cuando en la fórmula del VAN el valor de “k” es igual a “0” pasa a llamarse TIR (Tasa Interna de Retorno). La TIR es la rentabilidad que nos proporciona al proyecto.

La ecuación que utilizaremos es la siguiente:

$$TIR = \sum \frac{Ganancia - Costos}{(1 - i)^n}$$

Dónde:

TIR: Tasa interna de retorno.

Ganancia: Flujo de entrada de un periodo.

Costo: Flujo de salida de un periodo.

i: Tasa de interés.

n: Numero de periodos.

La Tabla 3.5 interpreta el cálculo del TIR:

Año	Costos	Ganancia	Ganancia-Costo/ (1-i)^n
1	72600	0	-82500
2	12600	21000	10847,1
3	9350	42000	47911,0
4	4550	63000	97466,2
		TIR	73724,3

Tabla 5. 5 Interpretación del TIR

Fuente: [Elaboración Propia]

Con esta información se concluye que el proyecto nos dará una rentabilidad de 73724.3 Bs durante 4 años.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

A partir del cumplimiento de los requerimientos establecidos por la institución, División de Sistemas de información y Estadística de la Universidad Mayor de San Andrés, se ha cumplido con los objetivos planteados por medio de la implementación del SISTEMA WEB DE DIGITALIZACION DE LIBRO DE ACTA DE NOTAS SIA-UMSA a través de los diferentes módulos desarrollados para el presente proyecto.

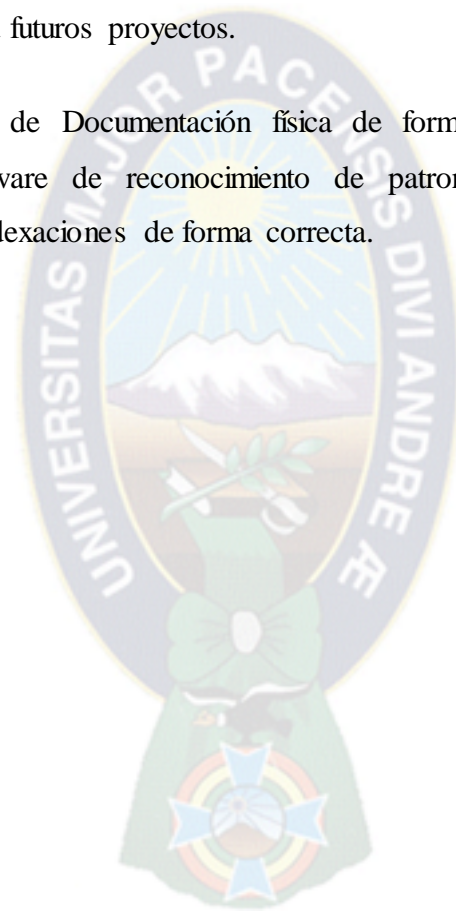
Por tanto se llega a las siguientes conclusiones:

- Se implementó el módulo de almacenamiento de archivos en formato digital de forma automática y directa.
- Se implementó módulo registro de docentes.
- Se implementó módulo de almacenamiento de pensum académico, materias y paralelos por docente.
- Se implementó el módulo de Búsquedas de actas según docente, pensum académico y de estudiante, describiendo hora y fecha de almacenamiento.
- Se proporcionó interfaz que muestra las actas de notas, lo que permite la verificación de los mismos para transparentar el desarrollo académico de la comunidad estudiantil y seguimiento de actas que acreditan los docentes.
- Se incorporó módulo de emisión de reportes de información.
- Mediante el uso de web Service desarrollado para el software web en cuestión, se mejoró el manejo de datos y archivos, mismos que son actualizados y almacenados periódicamente.
- Se contempló en el desarrollo del sistema web un diseño adaptativo, visualizando de forma elegante en los diferentes terminales.

6.2 RECOMENDACIONES

Para ampliar el presente proyecto de grado, se hacen las siguientes recomendaciones:

- Realizar ampliaciones en cuanto a funcionalidad del software a más procesos y extenderlos a más áreas de la comunidad universitaria.
- Aprovechar el uso de nuevas herramientas tecnológicas para el desarrollo aplicados a este proyecto para futuros proyectos.
- Debido al manejo de Documentación física de forma masiva, se recomienda el desarrollo de software de reconocimiento de patrones de tal forma que sean determinadas las indexaciones de forma correcta.



BIBLIOGRAFÍA

[Pressman, 2002] Pressman R. S. 2002 Título: Ingeniería de Software, un enfoque práctico.

[Ceri, 2000] Ceri, S., Fraternali, P, Bongio 2000 Título: Lenguaje de modelado WebML, un lenguaje para diseñar sitios Web Computer Networks USA.

[Van Vliet, 2002] Hans Van Vliet, 2002, Tercera Edición USA Título: Prácticas de la Ingeniería de Software.

[Escalona, 2002] María José Escalona, Sevilla, Diciembre de 2002 Título: Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo, recuperado de: www.lsi.us.es/docs/informes/LSI-2002-4.pdf

[Palacio & Ruata, 2011] Palacio, J. y Ruata, C. (2011). Título: Scrum Manager Projects. Web, recuperado de: www.scrummanager.net/files/sm_proyecto.pdf

[Brambilla & Manolescu, 2002] Brambilla & Manolescu, Ceri, Comai. “Encyclopedia of Information Science and Technology, First Edition”.

[Kendall & Kendall, 1995] KENDALL K. Y Kendall J. Análisis y diseño de sistemas de información, Mexico, Prentice Hall, 1995.

[Bosh, 2008] González Bosch, Verónica y Tamayo, Francisco, Título: "Blitz QFD: Un Vistazo Relámpago al Poder del QFD". Web, recuperado de: www.qfdlat.com

[Alaimo, 2013] Martín Alaimo, Ediciones Kleer 2013 Título: Proyectos ágiles con Scrum.

[Fenton, 1997] Fenton, N.1997. "Software quality assurance & Measurement. A worldwide perspective". Second edition. Chapman&Hall

[ABUD, (n.d)] Abud Figueroa, Maria Antonieta. Calidad en la Industria del Software. La Norma ISO-9126.

[DTIC-Memoria institucional, 2013-2014] Memoria Institucional Departamento de Tecnologías de Información y Comunicación 2013-2014, Gestión 2014.

[Portal web DSIE, 2015] Web recuperado de: <http://dsie.umsa.bo>

[COCOMO II, 2000] COCOMO II Model Definition Manual, Center for Software Engineering, USC Year 2000.

