

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



PROYECTO DE GRADO

**“SISTEMA WEB DE REGISTRO, SEGUIMIENTO Y CONTROL DE
CORRESPONDENCIA BASADO EN BPM
CASO: CARRERA DE INFORMÁTICA”**

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

POSTULANTE: ALAN RODRIGO CORINI GUARACHI
TUTOR METODOLÓGICO: M. Sc. JORGE TERÁN POMIER
ASESOR: M. Sc. CARLOS MULLISACA CHOQUE

LA PAZ – BOLIVIA

2014



**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 DERIVARÁ EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

. A Dios.

Por haberme permitido llegar a este momento tan especial de mi vida, por haberme dado salud, dándome fortaleza y sabiduría para continuar y lograr mis objetivos, además de su infinita bondad y amor.

A mi mamá Teresa y a mi papá Juan.

Por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles, me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos.

Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y mi capacidad. Es por ello que soy lo que soy ahora.

A mi familia.

Por brindarme su apoyo incondicional y por compartir conmigo buenos y malos momentos

AGRADECIMIENTOS

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Le doy gracias a mis padres Juan y Teresa por apoyarme en todo momento.

A mi Tutor M. Sc. Jorge Terán Pomier, por brindar su conocimiento, experiencia y sobre todo el apoyo incondicional para la conclusión del proyecto de grado.

A mi Asesor M. Sc. Carlos Mullisaca Choque, por su colaboración y orientación con sus conocimientos pude llegar a esta etapa de la vida y además de poder brindarme su amistad.

Al M. Sc. Aldo Valdez Alvarado, por brindar conocimiento, experiencia.

Al Lic. José Luis Zeballos y Lic. Reynaldo Escobar por su apoyo desinteresado, y su conocimiento compartido que me ayudo a construir este proyecto.

Al Personal de la Carrera de Informática, en especial a la Lic. Menfy Morales Ríos por brindarme su apoyo y tiempo para las primeras fases del proyecto como también en las últimas.

A todos los compañeros de la carrera, por compartir esos buenos momentos de estudio dentro y fuera de la universidad.

Gracias también a mis queridos amigos (as), que me apoyaron y me permitieron entrar en su vida durante estos años de convivir dentro y fuera de la universidad.

A todos gracias.

Resumen

El proyecto de grado titulado: "SISTEMA WEB DE REGISTRO, SEGUIMIENTO Y CONTROL DE CORRESPONDENCIA BASADO EN BPM CASO: CARRERA DE INFORMÁTICA" fue desarrollado para la jefatura de carrera, con el propósito de brindar una herramienta que coadyuve con las tareas de registro, control y seguimiento de la correspondencia, interno para la carrera de informática.

El capítulo I trata sobre la parte introductoria de presente proyecto de grado identificando los problemas y objetivos del sistema de correspondencia.

El capítulo II muestra información teórica que ayudara a comprender el ámbito y la base teórica del sistema.

El capítulo III muestra las fases de análisis del sistema mediante la metodología ágil *AUP*, fase de inicio que es el modelo de negocio y el modelo de requisitos, La Fase de Elaboración donde se elaboraron los diagramas de casos de usos, la funcionalidad del sistema, la fase de elaboración y el flujo de procesos que son pilares fundamentales para el funcionamiento del sistema, La fase de construcción el cual contiene la estructura del sistema lógica y física, y por último la fase de Transición la que se encarga de realizar pruebas de requerimientos y pruebas de estrés para el buen funcionamiento del sistema.

El capítulo IV describe el análisis de seguridad en torno a tres ámbitos importantes como ser riegos, seguridad por acceso y seguridad por a nivel de base de datos.

La Calidad de Software se basa en la tesis doctoral de OLSINA(OLSINA, 1999)

El capítulo V realiza cálculo del Sistema de Correspondencia basado en el análisis COCOMO II.

Por último, el capítulo VI brindar las respectivas conclusiones y recomendaciones de para el mejoramiento del sistema en un futuro.

ÍNDICE

CAPÍTULO I MARCO INTRODUCTORIO	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. ANTECEDENTES DE LA INSTITUCIÓN	2
1.2.2. PROYECTOS SIMILARES	3
1.3. PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1. PROBLEMA CENTRAL.....	4
1.3.2. PROBLEMAS SECUNDARIOS	5
1.4. DEFINICIÓN DE OBJETIVOS.....	5
1.4.1. OBJETIVO GENERAL	5
1.4.2. OBJETIVOS ESPECÍFICOS	5
1.5. JUSTIFICACIÓN	6
1.5.1. JUSTIFICACIÓN ECONÓMICA	6
1.5.2. JUSTIFICACIÓN SOCIAL.....	6
1.5.3. JUSTIFICACIÓN TECNOLÓGICA.....	7
1.6. ALCANCES Y LIMITES	7
1.6.1. ALCANCES	7
1.6.2. LIMITES	8
1.7. APORTES	8
1.7.1. PRÁCTICOS	8
1.7.2. TEÓRICOS	8
1.8. METODOLOGÍAS	9
1.8.1. MÉTODO CIENTÍFICO	9
1.8.2. METODOLOGÍA DE INGENIERÍA	10
1.8.3. HERRAMIENTAS	11
CAPÍTULO II MARCO TEÓRICO.....	12
2.1. INGENIERÍA DEL SOFTWARE.....	12
2.2. METODOLOGÍAS DE DESARROLLO DE SOFTWARE	12
2.2.1. METODOLOGÍAS ESTRUCTURADAS.....	12
2.2.2. METODOLOGÍAS TRADICIONALES	13

2.2.3.	METODOLOGÍAS AGILES.....	13
2.3.	AUP	13
2.3.1.	INTRODUCCIÓN	13
2.3.2.	PROCESO DE DESARROLLO ÁGIL.....	13
2.3.3.	FASES DEL AUP	14
2.3.4.	DISCIPLINAS DEL AUP	16
2.3.5.	INCREMENTO Y DESARROLLO DE AUP.-.....	19
2.3.6.	PRINCIPIOS DE LA AUP.-	19
2.4.	ARQUITECTURA MVC.....	20
2.4.1.	MODELO.....	20
2.4.2.	VISTA.....	21
2.4.3.	CONTROLADOR	21
2.5.	INGENIERÍA WEB	23
2.6.	UWE.....	24
2.6.1.	MODELO DE REQUERIMIENTOS	24
2.6.2.	FASES DE LA METODOLOGÍA UWE	28
2.7.	BPM.....	29
2.7.1.	CONCEPTOS BPM.....	30
2.7.2.	BPM PRACTICA	30
	CAPÍTULO III MARCO APLICATIVO.....	33
3.	INTRODUCCIÓN.....	33
3.1.	FASE DE INICIO.....	34
3.1.1.	MODELO DE NEGOCIO	34
3.1.2.	MODELO DE REQUISITOS.....	35
3.2.	FASE DE ELABORACIÓN	41
3.2.1.	DIAGRAMA DE CASOS DE USO	41
3.2.2.	MODELO CONCEPTUAL	44
3.2.3.	DIAGRAMA DE NAVEGACIÓN	44
3.2.4.	MODELO DE PRESENTACIÓN	45
3.2.5.	MODELO DE PROCESO	51
3.2.6.	MODELO DE ESTRUCTURA DE PROCESO.....	52
3.2.7.	MODELO DE FLUJO DE PROCESO	52

3.3.	FASE DE CONSTRUCCIÓN.....	55
3.3.1.	DIAGRAMA DE CLASES	55
3.3.2.	DIAGRAMA DE PAQUETES.....	56
3.3.3.	DISEÑO DE LA ARQUITECTURA DEL SISTEMA.....	58
3.4.	FASE DE TRANSICIÓN	60
3.4.1.	FASE DE PRUEBAS.....	60
3.4.2.	FASE DE IMPLEMENTACIÓN	63
CAPÍTULO IV SEGURIDAD Y CALIDAD		65
4.	INTRODUCCIÓN.....	65
4.1.	SEGURIDAD.....	65
4.1.1.	INTRODUCCIÓN.....	65
4.1.2.	RIESGOS	65
4.1.3.	SEGURIDAD DE ACCESO	66
4.1.4.	SEGURIDAD A NIVEL DE BASE DE DATOS	67
4.2.	CALIDAD DE SOFTWARE	68
4.2.1.	INTRODUCCIÓN.....	68
4.2.2.	DEFINICIÓN Y ESPECIFICACIÓN DE REQUERIMIENTOS DE CALIDAD	68
4.2.3.	DEFINICIÓN E IMPLEMENTACIÓN DE LA EVALUACIÓN ELEMENTAL	71
4.2.4.	DEFINICIÓN E IMPLEMENTACIÓN DE LA DEFINICIÓN GLOBAL.....	74
4.2.5.	RESULTADOS DE LA CALIDAD DEL SISTEMA	83
CAPÍTULO V ANÁLISIS DE COSTOS Y BENEFICIOS.....		85
5.	INTRODUCCIÓN.....	85
5.1.	ANÁLISIS DEL DISEÑO TEMPRANO DE COSTO.....	85
5.2.	ANÁLISIS DE COSTO POST ARQUITECTURA.	87
5.3.	ANÁLISIS DE LOS BENEFICIOS CON EL VAN Y EL TIR	91
CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES.....		94
6.	INTRODUCCIÓN.....	94
6.1.	CONCLUSIONES	94
6.2.	RECOMENDACIONES	94
BIBLIOGRAFÍA.....		96

GLOSARIO	99
ANEXOS.....	104

ÍNDICE DE FIGURAS

FIGURA 1.1. ORGANIGRAMA DE LA CARRERA DE INFORMÁTICA	3
FIGURA 2.1. ARQUITECTURA MVC	28
FIGURA 2.2. DIAGRAMA DE SECUENCIA	30
FIGURA 2.3. BPM	37
FIGURA 2.4. BPM ARTICULA LA ESTRATEGIA, LOS PROCESOS Y LA TECNOLOGÍA DE UNA ORGANIZACIÓN.....	31
FIGURA 2.5. BPMS.....	32
FIGURA 3.1. AUP FASES	33
FIGURA 3.2. CASOS DE USO ACTUAL.....	35
FIGURA 3.3. RECEPCIÓN DE CORRESPONDENCIA.....	40
FIGURA 3.4. FLUJO ACTUAL DE LA CARRERA DE INFORMÁTICA	41
FIGURA 3.5. MODELO DE CASOS DE USO DE RECEPCIÓN Y ELABORACIÓN DE DOCUMENTOS ..	42
FIGURA 3.6. CASOS DE USO DE REGISTRO Y ASIGNACIÓN.....	43
FIGURA 3. 7.CASO DE USO DE REGISTRO DE CORRESPONDENCIA.....	44
FIGURA 3.8. DIAGRAMA DE NAVEGACIÓN	45
FIGURA 3.9. COMPONENTES DEL MODELO DE PRESENTACIÓN.....	46
FIGURA 3 10. AUTENTIFICACIÓN DE USUARIO	46
FIGURA 3.11.CUERPO DEL SITIO WEB	47
FIGURA 3.12.CABECERA DE LA PAGINA	47
FIGURA 3.13. MENÚ DEL SISTEMA	48
FIGURA 3.14. PIE DE PAGINA	48
FIGURA 3.15. PÁGINA DE CONTENIDO.....	49
FIGURA 3.16. CONSULTA DE CORRESPONDENCIA	49
FIGURA 3.17. REGISTRO DE CORRESPONDENCIA EN EL MODELO DE PRESENTACIÓN	50
FIGURA 3.18.ELABORACION DE RESPUESTA DE LA CORRESPONDENCIA.....	50
FIGURA 3.19. ASIGNACIÓN DE DOCUMENTOS	51
FIGURA 3.20. BÚSQUEDA DE CORRESPONDENCIA.....	51
FIGURA 3.21. REGISTRO DE CORRESPONDENCIA.....	52
FIGURA 3.22. ACCESO AL SISTEMA.....	53
FIGURA 3.23. REGISTRO DE NUEVA CORRESPONDENCIA	54
FIGURA 3.24. PROCESO DE GESTIÓN DE CORRESPONDENCIA.....	55

FIGURA 3.25. DIAGRAMA DE CLASES DE SISTEMA DE CORRESPONDENCIA	56
FIGURA 3.26. PAQUETE DE REGISTRO DE CORRESPONDENCIA.....	56
FIGURA 3.27. PAQUETE DE ENVÍO Y ASIGNACIÓN DE CORRESPONDENCIA	57
FIGURA 3.28. PAQUETE DE SEGUIMIENTO DE CORRESPONDENCIA	57
FIGURA 3.29. DIAGRAMA DE COMPONENTES DE LA ESTRUCTURA MVC.....	58
FIGURA 3.30. DIAGRAMA DE COMPONENTES DEL SISTEMA DE CORRESPONDENCIA	59
FIGURA 3.31. DIAGRAMA DE DESPLIEGUE.....	60
FIGURA 3.32. ELABORACIÓN DE DOCUMENTO RESPUESTA.....	63
FIGURA 3.33. REGISTRO DE CORRESPONDENCIA.	64
FIGURA 3.34. PARÁMETROS DE INSTITUCIÓN	64
FIGURA 4.1. FUNCIONES SIMPLES DE AGREGACIÓN DE PREFERENCIAS	74
FIGURA 4.2. EJEMPLO DE ESTRUCTURA DE AGREGACIÓN DE PREFERENCIAS PARCIALES PARA LA USABILIDAD Y FUNCIONALIDAD	75
FIGURA 4.3. EJEMPLO DE ESTRUCTURA DE AGREGACIÓN DE PREFERENCIAS PARCIALES PARA LA CONFIABILIDAD Y EFICIENCIA.....	76
FIGURA 4.4. PREFERENCIAS PARCIALES PARA LAS CARACTERÍSTICAS DE MÁS ALTO NIVEL...	83
FIGURA 5.1. ESCALA DE FACTORES.....	89

ÍNDICE DE TABLAS

TABLA 1.1. REQUERIMIENTOS GENERALES	6
TABLA 2.1. ESTEREOTIPOS E ICONOS DE LOS CASOS DE USO	24
TABLA 2.2. ESTEREOTIPOS E ICONOS DE LAS ACTIVIDADES	25
TABLA 2.3. ESTEREOTIPOS E ICONOS DE NAVEGACIÓN	26
TABLA 2.4. ESTEREOTIPOS E ICONOS DE PRESENTACIÓN	27
TABLA 2.5. ESTEREOTIPOS E ICONOS DEL FLUJO DE PROCESO	28
TABLA 3.1. IDENTIFICACIÓN DE LOS USUARIOS Y ACTORES	35
TABLA 3.2. TIPO DE DOCUMENTO Y SU ESTADO DE IMPLEMENTACIÓN	39
TABLA 3.3. DESCRIPCIÓN DEL CASO DE USO GENERAL	43
TABLA 3.4. DESCRIPCIÓN DEL CASO DE USO DE REGISTRO Y ASIGNACIÓN	43
TABLA 3.5. TABLA DE REQUERIMIENTOS	61
TABLA 3.6. TABLA DE RESISTENCIA	62
TABLA 3.7. TABLA DE RENDIMIENTO	63
TABLA 4.1. SUB- CARACTERÍSTICAS DE USABILIDAD	70
TABLA 4.2. SUB- CARACTERÍSTICAS DE FUNCIONALIDAD	70
TABLA 4.3. SUB- CARACTERÍSTICAS DE CONFIABILIDAD	71
TABLA 4.4. SUB- CARACTERÍSTICAS DE EFICIENCIA	71
TABLA 4.5. CRITERIOS DE RECOLECCIÓN DE DATOS DE MÁS ALTO NIVEL PARA USABILIDAD	72
TABLA 4.6. CRITERIOS DE RECOLECCIÓN DE DATOS DE MÁS ALTO NIVEL PARA FUNCIONALIDAD	72
TABLA 4.7. CRITERIOS DE RECOLECCIÓN DE DATOS DE MÁS ALTO NIVEL PARA CONFIABILIDAD	73
TABLA 4.8. CRITERIOS DE RECOLECCIÓN DE DATOS DE MÁS ALTO NIVEL PARA EFICIENCIA	74
TABLA 4.9. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA ATRIBUTOS DE USABILIDAD	77
TABLA 4.10. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA SUB- CARACTERÍSTICAS DE USABILIDAD	78
TABLA 4.11. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA ATRIBUTOS DE FUNCIONALIDAD	78
TABLA 4.12. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA ATRIBUTOS DE FUNCIONALIDAD	79

TABLA 4.13. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA ATRIBUTOS DE FUNCIONALIDAD.....	79
TABLA 4.14. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA CARACTERÍSTICAS DE FUNCIONALIDAD.....	80
TABLA 4.15. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA ATRIBUTOS DE CONFIABILIDAD	80
TABLA 4.16. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA SUB - CARACTERÍSTICAS DE CONFIABILIDAD	81
TABLA 4.17. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA CARACTERÍSTICAS DE CONFIABILIDAD	81
TABLA 4.18. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA ATRIBUTOS DE EFICIENCIA	82
TABLA 4.19. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA SUB - CARACTERÍSTICAS DE EFICIENCIA	82
TABLA 4.20. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA CARACTERÍSTICAS DE EFICIENCIA	83
TABLA 4.21. RESULTADO DE LAS PREFERENCIAS DE CALIDAD PARA CARACTERÍSTICAS DE EFICIENCIA	84
TABLA 5.1. CONVERSIÓN DE PUNTOS DE FUNCIÓN.....	85
TABLA 5.2. MÓDULOS Y TAMAÑO	86
TABLA 5.3. COSTOS POR MODULO	87
TABLA 5.4. COSTOS TOTAL DEL PROYECTO A PRIORI.....	87
TABLA 5.5. PESO DE FACTOR DE COMPLEJIDAD POR CADA SUB MÓDULO.....	88
TABLA 5.6. COSTO TOTAL DEL SOFTWARE.	90
TABLA 5.7. COSTO TOTAL DEL SISTEMA.	90
TABLA 5.8. INGRESO ESTIMADOS.....	91
TABLA 5.9. EGRESOS ESTIMADOS.....	92
TABLA 5.10. FLUJO DE CAJA NETO.	92

CAPÍTULO I MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

La palabra carta se deriva del latín *charta* y significa, papel escrito que se manda a una persona para darle cuenta de algo. Se llama también epístola, misiva o comunicación. Tomando en cuenta la forma de su redacción, la carta es realmente una conversación. Por eso, todas las normas y requisitos exigidos por esta, son también necesarios para la carta. La Correspondencia: es el trato recíproco entre dos personas mediante el intercambio de cartas, esquelas, tarjetas, telegramas, catálogos, folletos, etc. En las empresas se considera que la correspondencia es el alma del comercio y de la industria. (Avilez, s.f.)

La expresión o relación entre las personas es común que charlemos y en algunos casos lleguemos a ciertos compromisos, más sin embargo en algunos casos suele darse que olvidamos lo que hicimos o ciertos compromisos contraídos los negamos, por no convenir a nuestros intereses.

En la empresa, industria o dependencias gubernamentales no podemos llegar y decir “se me olvidó lo que dije o a lo que me comprometí”, razón por la cual en todo tipo de compromisos relacionados con las empresas, industrias o dependencias gubernamentales se elaboran por escrito, y a esto suele llamarse correspondencia. Al interior de las organizaciones se trabaja con un flujo de trabajo (workflow en inglés), las cuales cuentan con una gran cantidad de información y son necesarias para diferentes procesos.

La carrera de informática no puede estar al margen de usar esta tecnología por eso tiene que contar con un sistema propio de gestión de correspondencia, el cual podrá automatizar los procesos de control, seguimiento y envío de diferentes trámites dentro de la carrera de informática entre las diferentes unidades administrativas y académicas pertenecientes a la carrera de informática, de tal manera mejorar el acceso a la información y el flujo de esta, para este fin se utilizarán herramientas estables, con el propósito fundamental de servir a la comunidad estudiantil como a personas ajenas a nuestra institución.

En la universidad se cuenta con un BPM llamado LDAP del cual no hay información que se pueda solicitar por ser muy privada para la institución.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES DE LA INSTITUCIÓN

La Carrera de Informática fue creada en 1974, como una mención del Departamento de Matemáticas, en 1984 se independiza administrativamente de éste, convirtiéndose posteriormente en Carrera de Informática con aproximadamente 300 alumnos. En 1987 se crea el Instituto de Investigaciones en Informática que luego de su fundación emprende las importantes labores de investigación entre las cuales se destacan proyectos en áreas como Redes de computadoras, Inteligencia artificial, Informática teórica, Desarrollo de Sistemas y otras. Siguiendo el modelo académico planteado por la UMSA se crea la subdirección académica y la de interacción social.

El actual plan de estudios está dividido en tres niveles. El primer nivel, nivel básico, tiene como fundamento la lógica matemática pero también un alto contenido en prácticas de laboratorio de computación. El segundo nivel, nivel formación, contiene en su generalidad diversas asignaturas referidas a las ciencias de la computación. El tercer nivel, nivel de especialidad, se dirige a orientar a los estudiantes a una de las dos menciones: Ingeniería de Sistemas Informáticos y Ciencias de la Computación. Este plan, cumpliendo con la sociedad, provee tres grados académicos: Técnico Superior en Informática, Bachiller en Ciencias Informáticas y Licenciatura en Informática. Actualmente la Carrera cuenta con una matrícula de 3600 alumnos convirtiéndola en la tercera carrera más numerosa de la UMSA. La renovación constante de los avances científicos y tecnológicos genera la necesidad de adaptar el plan de estudios a estas innovaciones. También los equipos de laboratorio son objeto de actualización permanente. Sin embargo la rapidez de los cambios y el crecimiento vegetativo traen consigo problemas de infraestructura en aulas y laboratorios así como en la obsolescencia de software y hardware.

El proyecto UMSANet del Instituto de Investigaciones en Informática iniciado en 1994 con la conformación de una pequeña red de computadoras que luego fue ampliado a otras

unidades y se convierte en 1996 en el nodo Web UMSANet reconocido como el sitio Web oficial de la UMSA. Este proyecto sirve de base en la creación del proyecto UMSATIC que es la explotación de las tecnologías de información y comunicación y los servicios Internet bajo políticas institucionales que coadyuven en la consecución de los fines y objetivos de la MSA.

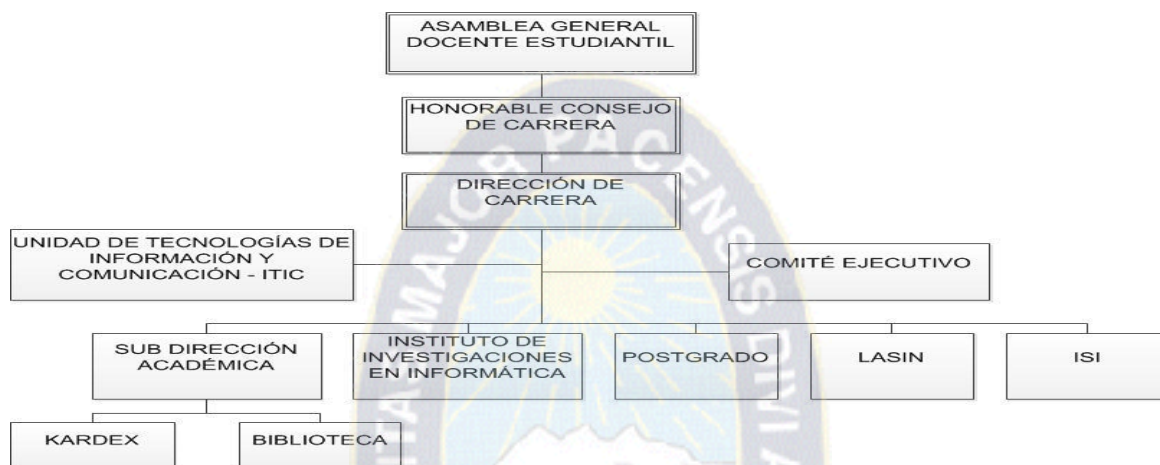


Figura 1.1. Organigrama de la Carrera de Informática

Fuente: (Carrera de Informatica, 2010)

1.2.2. PROYECTOS SIMILARES

A continuación se describirá proyectos de grado.

“Chasqui Digital” E – CORRESPONDENCIA, Caso Facultad de Ciencias Puras y Naturales, elaborado por (Ibañez, 2009), Universidad Mayor de San Andrés, Carrera de Informática. La automatización de la información, es considerada muy importante y dar una herramienta de apoyo a las necesidades de los funcionarios de las diferentes instituciones es un logro más. Por esta razón .En el mismo se ha realizado un análisis de requerimientos, en los cuales se vio la necesidad de automatizar el flujo de la correspondencia.

Sistema de registro y seguimiento de correspondencia, Caso: Servicio departamental de gestión social SEDEGES, elaborado por (Huanca M. , 2008), Universidad Mayor de San Andrés, Carrera de informática. Este presente proyecto muestra el desarrollo e implementación del sistema de Registro y seguimiento de correspondencia para el Servicio Departamental de Gestión Social (SEDEGES) en la

intranet de la institución.

Sistema de correspondencia para la empresa S.I.M.S.A., elaborado por (Vargas, 2009), **Universidad mayor de San Andrés, Carrera de Informática.** La implementación del presente trabajo se realizó bajo el estudio detallado de los procesos y sus reglas en tiempos específicos, para automatizar actividades que se presenten diariamente

Sistema de registro y seguimiento de correspondencia y asignación de aulas para el centro de multiservicios educativos CEMSE - CERPI (Huanca S. , 2008), **Universidad Mayor de San Andrés, Carrera de Informática,** El presente proyecto de grado, realizó los estudios en una solución de la denominada Gestión Documental el cual nos permite conservar el patrimonio documental de la institución.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. PROBLEMA CENTRAL

En la carrera de informática llegan 3 tipos de documentos de diferentes fuentes.

- Documentos provenientes de la Universidad (sin pasar por el conducto regular), generalmente documentos provenientes de Rectorado y Vicerrectorado, o alguna unidad administrativa dependiente de rectorado.
- Documento proveniente de decanatura (conducto regular), estos documentos provienen de decano y vicedecano alguna unidad administrativa dependientes de decanatura.
- Documentos Terceros, estos documentos son directamente depositados en la dirección de carrera como ser (cartas de solicitud de pasantía, solicitud de laboratorio, memorándum de felicitaciones, etc.)

Según observación y por el control con el que cuenta la carrera de informática podemos llegar al planteamiento de tener las siguientes dificultades; de no contar con un sistema informático de Información y seguimiento de correspondencia , el manejo de la correspondencia se realiza de forma manual el cual es registrado en 2 libros , los cuales son libro de recibidos y de despachos , tampoco cuenta con un sistema que de reportes estadísticos de la información que es ingresada a la carrera de informática.

¿Cómo se mejorará el manejo, registro, seguimiento y control de correspondencia dentro de la carrera de informática?

1.3.2. PROBLEMAS SECUNDARIOS

A continuación se mencionan algunos de los problemas más importantes que se pudieron Identificar:

- Desconocimiento del estado de la correspondencia. lo que puede ocasionar perdidas de documentos
- Cómo se podría mejorar el envío de la correspondencia dentro de la carrera de informática. lo que ocasionaría lentitud en el proceso de despacho
- Cómo se podría obtener información oportuna, precisa e inmediata sobre la correspondencia. lo que generar reportes fiables
- Desconocimiento de la prioridad de documentos importantes. lo que provoca lentitud en procesos importantes o de suma urgencia
- Manejo manual de la documentación que ingresa y sale. lo que ocasiona lentitud en el proceso del *workflow*.
- Falta de comunicación entre áreas. lo que ocasionaría lentitud en el proceso del *workflow*.

1.4. DEFINICIÓN DE OBJETIVOS

1.4.1. OBJETIVO GENERAL

Diseñar, desarrollar e implementar un SISTEMA WEB DE REGISTRO, SEGUIMIENTO Y CONTROL DE CORRESPONDENCIA basado EN *BPM* interno para la carrera de informática.

1.4.2. OBJETIVOS ESPECÍFICOS

- Emitir alerta tempranas sobre las correspondencia
- Crear opciones de búsquedas avanzadas para diferente tipo de correspondencia
- Registrar la llegada y salida de la correspondencia
- Socializar a los usuarios de la prioridad de documentos que ingresan
- Informar a los usuarios de la prioridad de los documentos

1.5. JUSTIFICACIÓN

1.5.1. JUSTIFICACIÓN ECONÓMICA

Cada organización pública o privada tiene la necesidad de abaratar costos administrativos, maximizar ganancias con pocos recursos, como ahorrar en papel, en este sentido proporcionara beneficios en cuanto a la reducción de tiempo, esfuerzos y gastos que se realizara en costo del papel, así reduciendo el costo económico en el manejo de los libros de recepción y devolución, en los cuales se hace difícil el almacenamiento. Para el desarrollo del presente sistema los gastos económicos serán reducidos ya que las diferentes unidades de la carrera de informática cuentan con equipos de computación de escritorio y cuentan con licencias de software libre en las maquinas. La licencia del desarrollo del sistema web estará bajo los estándares libre y bajo licencias libres. La carrera de informática cuenta con una red propia de intranet y servidores propios para alojar el sistema.

El beneficio que trae este sistema web para los administrativos de la carrera de informática así también a la comunidad estudiantil y público en general, puesto que ya no se realizaran búsquedas innecesarias en los libros de registro y despacho, así también ya no tendrá errores en la generación de los números correlativos de cada documento ya que el sistema lo hace automático, como también se podrá saber en dónde está el documento.

1.5.2. JUSTIFICACIÓN SOCIAL

En una investigación exploratoria realizada en la carrera de informática, se detectó la necesidad de contar con un sistema interno de correspondencia para la carrera de informática que ayude con el registro, control y seguimiento de la documentación, desde que ingresa por la ventanilla única, proceso por diferentes unidades, y sea despachado por las mismas unidades. Por lo tanto el sistema ayudara al procesamiento de la información que se realizara dentro de la carrera de informática, para así tener un mejor rendimiento del manejo de la correspondencia, así también de las mismas unidades académicas, mejorando la eficiencia y eficacia y lograra contribuir una imagen buena de la carrera de informática.

Este sistema web beneficiara no solo a los administrativos, secretarias, personal de apoyo sino será de gran ayuda a la comunidad estudiantil y usuarios externos quien podrá hacer seguimiento de su documento y el estado en el que se encuentra con solo acceder a la web.

1.5.3. JUSTIFICACIÓN TECNOLÓGICA

Considerando que la carrera de informática cuenta con los equipos necesarios y recursos necesarios para la implementación del sistema de correspondencia se describen en la siguiente tabla de requerimientos generales.

Material	Descripción	Estado actual
Una Pc. Desktop	Para el uso deservidor	Existente
Instalación de red interna	Intranet	Existente
Impresoras	Para procederá la impresión de los documentos existentes	Existente
Sistema operativo	Equipo configurado con S. O. Libre	Existente
Scanner	Para la digitalización de toda la documentación a ingresar	Existente
Impresora	Para el recibió de la correspondencia	Existente

Tabla 1.1. Requerimientos Generales

Fuente: [Elaboración propia]

En el presente proyecto coadyuvara en el manejo de la tecnología tanto en software como en hardware, que proporcionara un buen manejo de la información con nuevas tecnologías que beneficiara el manejo de la correspondencia dentro de la carrera de informática.

En cuanto a lo que se refiere a software y desarrollo del sistema, se trabajara completamente con software libre como ser Java y open source, en cuanto se refiere al servidor trabajara con sistema operativo Linux, por lo que no se pagara por costo de licencias, la red de internet que cuenta la carrera de informática proviene de UMSATIC, el servicio de la señal es estable.

1.6. ALCANCES Y LIMITES

1.6.1. ALCANCES

El sistema podría ser implementado en las diversas carreras, institutos, etc. que cuenten con el flujo que existe en la carrera de informática los cuales serán analizados por áreas.

- El sistema contara un módulo de registro.
- El sistema contara con un módulo que permitirá digitalizar la correspondencia.

- el sistema contara con un módulo que permita ver la existencia de documentos.
- el sistema contara con un motor de búsquedas.
- El sistema contara con un módulo de reportes el cual mostrar la eficiencia y eficacia del personal.
- El sistema contara con un módulo de alerta temprana el cual ayudara a los administrativos de diferentes aéreas cuando un documento se está por vencer el plazo

1.6.2. LIMITES

El sistema no podrá ser implementado en organizaciones que no cuenten con el flujo de la información que tiene la carrera de informática. Ya que en otras instituciones manejan otro flujo de información.

- Los usuarios no podrán modificar fechas
- El sistema no brindara un modo de almacenamiento de imágenes superior a 2 Mb.
- El sistema no brindara la ubicación exacta de la correspondencia requerida, solo dará fe de la existencia y como se procesó.
- Las consultas de la existente de la correspondencia solo podrán ser consultadas por los propietarios de las mismas.
- El sistema se basara en patrones parametrizable de correspondencia estándar no se podrá añadir otra característica si no estaba definida anteriormente.

1.7. APORTES

1.7.1. PRÁCTICOS

Como aportes el sistema brindara la automatización y velocidad en el procesamiento de información de la correspondencia dentro de la carrera de informática, obteniendo informes periódicos para ver la eficiencia, los cuales podrán obtener información oportuna de toda la documentación existente.

1.7.2. TEÓRICOS

Entre los aportes teóricos que brinda este documento podemos mencionarlos siguientes:

- El uso de metodologías ágiles tales como AUP para el desarrollo del sistema.
- Uso de tecnología para modelado como UWE.

- El uso de herramientas tecnológicas como el Framework Java Server Faces que nos permitan el desarrollo e implementación del sistema en base a Java versión 7.
- El uso de LeanOffice, el cual coadyuvare en la organización del personal dependiente de la dirección de carrera.

1.8. METODOLOGÍAS

1.8.1. MÉTODO CIENTÍFICO

El método científico es una serie ordenada de procedimientos de que hace uso la investigación científica para observar la extensión de nuestros conocimientos. Podemos concebir el método científico como una estructura, un armazón formado por reglas y principios coherentemente concatenados. Es quizás uno de los más útiles o adecuados, capaz de proporcionarnos respuesta a nuestras interrogantes. Respuestas que no se obtienen de inmediato de forma verdadera, pura y completa, sin antes haber pasado por el error. Esto significa que el método científico llega a nosotros como un proceso, no como un acto donde se pasa de inmediato de la ignorancia a la verdad. Este es quizás el método más útil o adecuado, ya que es el único que posee las características y la capacidad para auto corregirse y superarse, pero no el único. El método científico es la conquista máxima obtenida por el intelecto para descifrar y ordenar los conocimientos. Tiene los siguientes pasos a seguir:

Observación del Fenómeno, consiste en la recopilación de hechos acerca de un problema o fenómeno natural que despierta nuestra curiosidad. Las observaciones deben ser lo más claras y numerosas posible, porque han de servir como base de partida para la solución.

- El planteamiento del problema.
- Formulación de la Hipótesis.
- Planteamiento de objetivos.
- Diseño del experimento.
- Obtención de resultados.
- Conclusión.

1.8.2. METODOLOGÍA DE INGENIERÍA

1.8.2.1. AUP

Según Scott Ambler El Proceso Unificado Ágil es una versión simplificada del RUP, la cual describe en una forma simple, fácil de entender y brinda un enfoque de desarrollo de software utilizando técnicas ágiles y conceptos del RUP. En comparación de las disciplinas del RUP que son 9, el AUP tiene solamente 7 las cuáles algunos son combinaciones de dos disciplinas del RUP.

- Modelo, entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio.
- Implementación, transformar el modelo en código ejecutable y realizar un nivel básico de pruebas individuales.
- Prueba, realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificar que se cumplan los requisitos.
- Despliegue, realizar un plan para la presentación del sistema y ejecutarlo para hacer que el sistema se encuentre a disposición de los usuarios finales.
- Gestión de Configuración, realizar la gestión de acceso a artefactos de su proyecto. Esto incluye no sólo el seguimiento de las versiones del artefacto en el tiempo, sino también el control y la gestión de cambios para ellos.
- Gestión del Proyecto, dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de los riesgos.
- Ambiente, apoyar el resto de los esfuerzos por garantizar que el proceso adecuado, la orientación y herramientas. (Ambler, 2013)

1.8.2.2. UWE

Según Ludwig Maximilians de la Universidad de München cito “UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las

técnicas de UML, la notación de UML y los mecanismos de extensión de UML. Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo de navegación, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad. UWE define vistas especiales representadas gráficamente por diagramas en UML. Además UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. (Wikipedia, 2006).

1.8.3. HERRAMIENTAS

1.8.3.1. GESTOR DE BASE DE DATOS

Para el desarrollo e implementación del gestor de base de datos se realizara con Postgresql y pgpsql como lenguaje de programación, para evitar el uso de licencias de funcionamiento la aplicación será desarrollada en base a software libre, los cuales no implican costo de uso

1.8.3.2. LENGUAJES DE PROGRAMACIÓN

Para el desarrollo e implementación de la aplicación web se lo realizara bajo entorno JAVA y JavaScript como lenguaje de programación, para evitar el uso de licencias de funcionamiento la aplicación será desarrollada en base a software libre, los cuales no implican costo de uso.

CAPÍTULO II MARCO TEÓRICO

2.1. INGENIERÍA DEL SOFTWARE

Ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, la aplicación de la ingeniería al software. Es la aplicación de la ingeniería al software, ya que integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería. Se pueden citar otras definiciones enunciadas por prestigiosos autores:

- Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software
- Ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales.

2.2. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo de software se refiere a un *framework* que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad. El *framework* para metodología de desarrollo de software consiste en "Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software", Estos *framework* son a menudo vinculados a algún tipo de organización, que además desarrolla, apoya el uso y promueve la metodología

2.2.1. METODOLOGÍAS ESTRUCTURADAS

Los métodos estructurados comenzaron a desarrollarse a fines de los 70 con la Programación Estructurada, luego a mediados de los 70 aparecieron técnicas para el diseño primero y posteriormente para el análisis. Estas metodologías son particularmente apropiadas en proyectos que utilizan para la implementación lenguajes de 3 y 4 generación.

2.2.2. METODOLOGÍAS TRADICIONALES

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo, llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales.

2.2.3. METODOLOGÍAS ÁGILES

Un proceso es ágil cuando el desarrollo de software es incremental, cooperativo, sencillo y adaptable. Entre las metodologías ágiles identificadas son: *Extreme Programming*, *Scrum*, *Open Source*, *Software Development*, etc.

2.3. AUP

2.3.1. INTRODUCCIÓN

Los procesos ágiles de desarrollo de software, conocidos anteriormente como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados. Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil. La mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto. (Bolivariana, 2013)

2.3.2. PROCESO DE DESARROLLO ÁGIL

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

Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad. (Bolivariana, 2013). El proceso unificado es un marco de desarrollo software iterativo e incremental. AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo.

2.3.3. FASES DEL AUP

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP. A continuación se detallara las fases del AUP. (Bolivariana, 2013)

2.3.3.1. FASE DE INCEPCIÓN

Las actividades de esta fase son (Bolivariana, 2013):

- Definir el ámbito del proyecto, incluye definir en alto nivel lo que el sistema hará. Es también importante definir lo que el sistema no hará. Esto establece los límites en los que el equipo trabajará.
- Estimar el coste y el calendario, se estima en alto nivel el calendario y el coste para el proyecto. Las estimaciones generales son usadas en iteraciones en las últimas fases.
- Definir riesgos, los riesgos del proyecto se definen aquí. La gestión de riesgos es una parte importante de un proyecto AUP. La lista de riesgos cambia con el tiempo.
- Estudio de viabilidad, el proyecto debe tener sentido desde las perspectivas técnicas, operacionales y de negocio.
- Preparar el entorno, esto incluye reservar un espacio de trabajo para el equipo, pedir lo que se necesite, obtener hardware y software que se necesite inmediatamente.

Para salir de esta fase, necesitaremos completar los hitos de esta fase:

- Conseguir que los stakeholders estén de acuerdo con el ámbito acordado del proyecto.
- Lista inicial de requisitos.
- Los stakeholders están de acuerdo con el coste inicial y el calendario estimado.
- Aceptación de riesgos.

- Aceptación de los procesos de AUP.
- El proyecto es viable.
- Tener un plan de proyecto para las siguientes fases.

2.3.3.2. FASE DE ELABORACIÓN

El principal objetivo es probar la arquitectura del sistema a desarrollar. Hay que asegurarse de que el equipo puede desarrollar un sistema que satisfaga los requisitos, y la mejor manera de hacerlo es construir un esqueleto funcional del sistema, llamado prototipo arquitectónico. El objetivo de esto es escribir un software funcional de alta calidad. Es importante darse cuenta de que los requisitos no se especifican completamente en este punto. Son detallados solo lo suficiente para entender los riesgos arquitectónicos y para asegurarse de que se entiende el ámbito de cada requisito. Durante esta fase también se va preparando la siguiente. A medida que se va ganando manejo con la arquitectura del sistema, se va preparando el entorno para la construcción, adquiriendo hardware, software y herramientas. La lista de hitos a cumplir para terminar esta fase es la siguiente (Bolivariana, 2013):

- La visión del proyecto es estable y realista.
- Estabilidad de la arquitectura. Satisface los requisitos.
- Aceptación de riesgos.
- Viabilidad.
- Plan de proyecto para las siguientes iteraciones de construcción.
- Ver si la arquitectura del sistema refleja la realidad de la arquitectura de la empresa.

2.3.3.3. FASE DE CONSTRUCCIÓN

El objetivo de esta fase es desarrollar el sistema hasta el punto de que esté listo para hacer pruebas de pre-producción. En las fases anteriores, la mayoría de los requisitos habían sido identificados y la arquitectura del sistema como punto de partida. Para finalizar esta fase debes de realizar los siguientes hitos (Bolivariana, 2013):

- Analizar y diseñar el modelo.
- Documentar las decisiones críticas de diseño.
- Construirlo.

- Ir evolucionando el dominio lógico, las interfaces de usuario y el esquema de datos.
- Probar el software.
- Desarrollar scripts para su instalación.
- Desarrollar una documentación inicial.
- Poner todos los productos bajo el CM (mantenimiento de configuración).

2.3.3.4. FASE DE TRANSICIÓN

Tiene como objetivo poner el sistema en producción. Se debe de incrementar las pruebas en esta fase y probar una versión beta. Los hitos a realizar son los siguientes (Bolivariana, 2013):

- Finalizar la documentación.
- Detectar fallos.
- Validar el sistema.
- Validar la documentación.
- Finalizar el modelo de prueba.
- Finalizar la documentación.
- Finalizar el desarrollo del paquete.
- Colocar el sistema en producción.
- Capacitación al personal.

2.3.4. DISCIPLINAS DEL AUP

Las disciplinas se llevan a cabo de manera sistemática, a la definición de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar, y entregar el software de trabajo que responda a las necesidades de sus interlocutores.

2.3.4.1. MODELADO

Su objeto es entender la lógica de negocio de la aplicación, el dominio del problema del proyecto e identificar una solución viable para el dominio del problema. Las sugerencias para esta disciplina son:

- No es necesario crear todos los modelos que figuran en el diagrama de flujo de trabajo.
- Usted no necesita una gran cantidad de datos durante las fases Incepción y Elaboración, así que no invertirá mucho tiempo en el modelado y la documentación detallada.

- Su objetivo es crear modelos que son solamente lo suficientemente buenos para su situación a la mano.
- La mayoría de los modelos son descartados después de que son usados, una pequeña minoría son mantenidos.
- Existe una amplia gama de enfoques para modelado.
- Buscar oportunidades para su reutilización, usted puede reutilizar más que código.
- Cuanto más amplia sea la gama de modelos que entienda, mayor será la probabilidad de que usted sea capaz de aplicar el producto de trabajo adecuado a su situación.
- La participación activa de los interesados para el éxito de su negocio.
- Use una arquitectura por capas.

2.3.4.2. IMPLEMENTACIÓN

Transformar los modelos en código ejecutable y realizar pruebas básicas, en particular pruebas unitarias. En esta disciplina se sugiere:

- Programación en Parejas, es donde dos desarrolladores trabajan juntos en un mismo equipo, pueden mejorar la calidad de su trabajo, su experiencia de aprendizaje.
- Desarrollo Dirigido por Pruebas, cuando se escribe una prueba con sólo la producción de código suficiente para satisfacer las pruebas. Trabaja increíblemente bien.
- Un dibujo puede valer más que mil líneas de código.
- Adopte y siga un conjunto de directrices de codificación.

2.3.4.3. PRUEBAS

Realizar una evaluación de los objetivos para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema funciona como fue diseñado y verificar que los requisitos se cumplen. En esta disciplina se sugiere:

- Pruebas a lo largo del ciclo de vida.
- Desarrollo Controlado de Pruebas es una implementación técnica que promueve el desarrollo de software de alta calidad.
- Usted necesita automatizar su paquete de pruebas para verdaderamente soportar el desarrollo evolutivo.

- Pruebas de aceptación deben cumplir "doble función" tanto con los requerimientos como con las pruebas de los productos finales.

2.3.4.4. DESPLIEGUE

Planear la entrega del sistema y ejecutar el plan para hacer que el sistema quede disponible para los usuarios finales. En esta disciplina se sugiere:

- Desarrolle y mejore los códigos fuente durante la fase de construcción.
- Disponga de una preproducción de una caja de arena de pruebas, donde puede comprobar que su sistema funcione antes de la implementación y puesta en ejecución.
- Su organización debe tener "períodos de suspensión", cuando no es permitido entregar nuevo software para la puesta en producción.
- Tenga puntos de decisión "continuar / detenerse" en el proceso de desarrollo.

2.3.4.5. GESTIÓN DE LA CONFIGURACIÓN

Gestionar el acceso a los artefactos del proyecto. Esto incluye, además de la traza de versiones de los artefactos, el control de cambios y la gestión de los mismos. En esta disciplina se sugiere:

- Use una estructura de directorios estándar, para sus productos del proyecto.
- Si vale la pena crearlo vale la pena ponerlo bajo el control de la CM.
- Proporcionar capacitaciones en CM, ya que muchas personas encontrarán el concepto muy extraño a primera vista.
- Los usuarios deben también poner sus productos del trabajo bajo el control de la CM.

2.3.4.6. GESTIÓN DEL PROYECTO

Dirige las actividades que tienen lugar dentro del proyecto, incluyendo gestión de riesgos, dirección del personal y coordinación. En esta disciplina se sugiere:

- Manejo de las personas en su equipo, los equipos necesitan liderazgo, no administración.
- Usted sólo puede planificar unas pocas semanas de antemano en detalle.
- La persona que hace el trabajo es la mejor preparada para planearlo, no el administrador del proyecto.

- La flexibilidad es crítica para sus éxitos, sus planes y estimaciones que deben cambiar con el tiempo para reflejar su cambiante ambiente.
- Iteraciones cortas, una o dos semanas en tiempo, requiere que usted se concentre en actividades de alto nivel.

2.3.4.7. ENTORNO

Apoyar el resto del esfuerzo asegurando que los procesos, métodos y herramientas están disponibles para el equipo cuando los necesitan. En esta disciplina se sugiere:

- Centrarse en el desarrollo de plantillas y guías ajustadas a las necesidades del equipo.
- Las personas probablemente no leerán los procedimientos, pero le pedirán ayuda.
- Hacer lo posible por usar licencias corporativas de herramientas de software.
- Ser flexible, los nuevos equipos de proyecto pueden requerir nuevos tipos de software.

2.3.5. INCREMENTO Y DESARROLLO DE AUP.-

Los equipos de AUP suelen ofrecer versiones de desarrollo al final de cada iteración en preproducción, una versión de desarrollo de una aplicación es algo que podrían ser liberados en la producción si se ponen a través de su pre-producción de garantía de calidad, las pruebas y los procesos de despliegue. La primera producción de liberación a menudo toma más tiempo para entregar versiones posteriores. La primera producción de liberación puede tomar doce meses para entregar la segunda versión de nueve meses, y luego otras liberaciones se entregan cada seis meses. Una de las primeras se centra en cuestiones de despliegue, no sólo permite evitar los problemas, sino que también permite tomar ventaja de sus experiencias durante el desarrollo.

2.3.6. PRINCIPIOS DE LA AUP.-

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

- 1) El personal sabe lo que está haciendo. La gente no va a leer detallado el proceso de documentación, el AUP producto proporciona enlaces a muchos de los detalles, si usted está interesado, pero no obliga a aquellos que no lo deseen.
- 2) Simplicidad. Todo se describe concisamente utilizando un puñado de páginas.
- 3) Agilidad. Ágil arriba el ajuste a los valores y principios de la Alianza Ágil.

- 4) Centrarse en actividades de alto valor. La atención se centra en las actividades que se ve que son esenciales para el de desarrollo.
- 5) Herramienta de la independencia. Usted puede usar cualquier conjunto de herramientas que usted desea con el ágil UP. Lo aconsejable es utilizar las herramientas que son las más adecuadas para el trabajo.
- 6) Adaptación de este producto para satisfacer sus propias necesidades. La AUP producto es de fácil acomodo común a través de cualquier herramienta de edición de HTML.

2.4. ARQUITECTURA MVC

El patrón de arquitectura MVC es un patrón que define la organización independiente del modelo, la vista y el controlador.

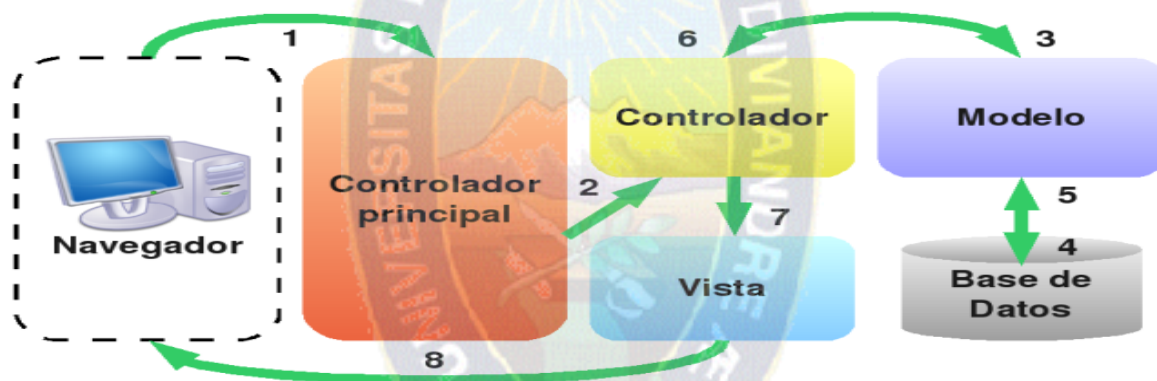


Figura 2.1. Arquitectura MVC

Fuente: (MARCADÉ, 2009)

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

2.4.1. MODELO

Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos. Esto quiere decir que aquí se operan los datos y las reglas de negocio asociadas al sistema, incluyendo el análisis sintáctico y el procesamiento de los datos de entrada y de los datos de salida.

El Modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio. se define la funcionalidad del sistema.
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

2.4.2. VISTA

Este presenta el Modelo, usualmente la interfaz de usuario. La vista es la capa de la aplicación que ve el usuario en un formato adecuado para interactuar, en otras palabras, es nuestra interfaz gráfica. Las vistas son responsables de:

- Recibir datos del modelo y la muestra al usuario.
- Tienen un registro de su controlador asociado.
- Pueden dar el servicio de “Actualización ()”, para que sea invocado por el controlador o por el modelo.

2.4.3. CONTROLADOR

El Controlador es la capa que controla todo lo que puede realizar nuestra aplicación. Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Está compuesto por acciones que se representan con funciones en una clase. Por ejemplo, yo tengo mi controlador llamado Clientes, y este controlador puede realizar las acciones Crear, Editar, Listar entre otras. (prestashop5estrellas, 2010). El controlador es responsable de:

- Recibe los eventos de entrada.
- Contiene reglas de gestión de eventos, del tipo “SI Evento Z, entonces Acción W”. Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método “Actualizar ()”. Una petición al modelo puede ser “Obtener_tiempo_de_entrega (nueva_orden_de_venta)”.

Para entender cómo funciona nuestro patrón modelo, vista y controlador, se debe entender la división a través del conjunto de estos tres elementos y como estos componentes se comunican unos con los otros y con otras vistas y controladores externos al modelo principal.

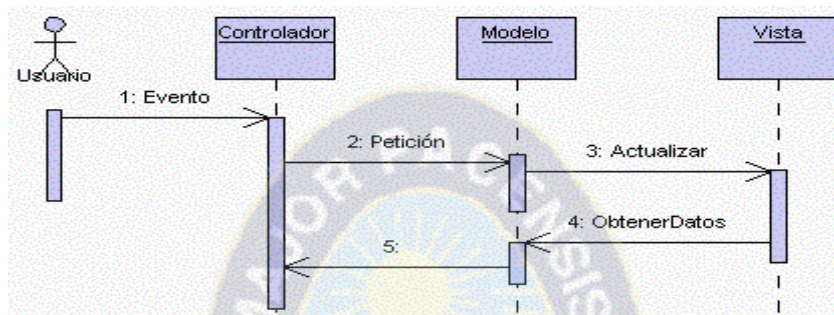


Figura 2.2 Diagrama de Secuencia

Fuente: (prestashop5estrellas, 2010)

Comunicación.- El modelo, la vista y el controlador deben comunicarse de una manera estable los unos con los otros, de manera que sea coherente con las iteraciones que el usuario realizara. Como es lógico la comunicación entre la vista y el controlador es bastante básica pues están diseñados para operar juntos, pero los modelos se comunican de una manera diferente, un poco más sutil

Modelo pasivo.- No es necesario para el modelo hacer ninguna tener alguna disposición a él, simplemente basta con tener en cuenta su existencia, por lo que esta función la llevara a cabo el controlador porque será el que interprete las ordenes de este usuario debido a que solo debe comunicar que algo ha cambiado.

Unión del modelo con la vista y el controlador.- Como no todos los modelos pueden ser pasivos, necesitamos algo que comunique al controlador y a la vista, por lo que en este caso, sí que necesitamos el modelo, ya que solo este puede llevar a cabo los cambios necesarios al estado actual en el que estos se encuentran.

Al contrario que el modelo, que puede ser asociado a múltiples asociaciones con otras vistas y controladores. (ejemplosTIW, 2013)

2.5. INGENIERÍA WEB

Uno de los aspectos más tenidos en cuenta, en el desarrollo de sitios web es sin duda alguna el diseño gráfico y la organización estructural del contenido. En la actualidad la web está sufriendo grandes cambios, que han obligado a expertos en el tema a utilizar herramientas y técnicas basadas en la ingeniería del software, para poder garantizar el buen funcionamiento y administración de los sitios web. Para garantizar el buen funcionamiento y mantenimiento de los sitios web, este debe contar con ciertos atributos y características que en conjunto forman un concepto muy importante, para alcanzar el éxito en cualquier organización, herramienta, y todo aquello que se pueda considerar como servicio. Dicho concepto es la calidad, que con atributos como, usabilidad, navegabilidad, seguridad, mantenibilidad.

Pero para tener artefactos de calidad, a esa misma se le debe planificar, programar y controlar, es decir la calidad no podrá ser agregada a un artefacto web o a cualquier otro producto, al final del proceso de desarrollo, si no que se deberá implementar durante todo el ciclo de vida del desarrollo. Para finalizar el resultado de un proceso de calidad, podría arrojar recomendaciones para introducir mejoras, y la decisión final podría consistir en lanzar una nueva versión del sitio web o en modificar algunos atributos ausentes o pobremente diseñados. Cabe destacar que la ingeniería web hace una diferencia entre un webSite y una aplicación, ya que la ingeniería de la web no se dedica a la construcción de sitios web si no a la construcción de aplicaciones web la principal característica que los distingue que los sitios web son sitios en la web en donde se publica contenido generalmente estático o un muy bajo nivel de interactividad con el usuario, mientras que las aplicaciones son lugares con alto contenido de interactividad y funcionalidades que bien podrían ser de un software convencional, la aplicación web más sencillo seria uno que contenga formularios y subiendo de nivel encontramos los que realizan conexión con bases de datos remotas, y administradores de contenidos entre otras. Entonces la ingeniería de la Web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente.

2.5.1. MODELADO WEB

Las aplicaciones web es un tipo particular de software, por ello se puede modelar con diagramas *UML*. Muchas aplicaciones telemáticas son un caso particular de aplicaciones web. Las aplicaciones web tienen particularidades, lo que hace que se puedan plantear modelos específicos o la forma de realizar el proceso de modelado para ser más precisos y tener más ventajas. Dependiendo de la sintaxis del lenguaje se clasifican en:

- Nuevos lenguajes con diferentes elementos respecto a *UML*: *WebML*.
- Extensiones de *UML*: *UWE*.
- *UML* sin extensiones: *OOHDM*.

2.6. UWE

UWE es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización. UWE está basada en UP y UML pero adaptados a la web. Los requisitos separan en fases (UWE – UML-based Web Engineering, 2012).

2.6.1. MODELO DE REQUERIMIENTOS

En UWE el modelado de requisitos consiste de dos partes (UWE – UML-based Web Engineering, 2012):

- Casos de uso de la aplicación y sus relaciones.
- Actividades describiendo los casos de uso en detalle.

i) Casos de Uso

En UWE se distinguen casos de uso estereotipados con «browsing» y con «processing» para ilustrar si los datos persistentes de la aplicación son modificados o no. En la tabla 2.1 se pueden ver los estereotipos y los iconos de los casos de uso.




nombres de estereotipos y sus iconos	
 browsing	 processing
 webUseCase	

Tabla 2.1. Estereotipos e iconos de los casos de uso

Fuente: (UWE – UML-based Web Engineering, 2012)

ii) Actividades

Como con casos de uso solamente es posible capturar poca información, cada caso de uso puede ser descrito más detalladamente mediante un proceso. Es decir, las acciones que son parte de un caso de uso, así como los datos presentados al usuario y aquellos requeridos como entrada de datos pueden ser modelados con precisión como actividades.







nombres de estereotipos y sus iconos	
 » <i>userAction</i>	 » <i>systemAction</i>
 » <i>displayAction</i>	 » <i>navigationAction</i>
 » <i>displayPin</i>	 » <i>interactionPin</i>

Tabla 2.2. Estereotipos e iconos de las actividades

Fuente: (UWE – UML-based Web Engineering, 2012)

Los dos estereotipos «user Action» y «system Action» pueden ser usados análogamente en el flujo de procesos. El estereotipo «user Action» es usado para indicar interacciones de usuario en la página web iniciando un proceso o respondiendo a un requisito explícito de información. Por lo contrario, «system Action» describe acciones que son ejecutadas por el sistema. Los detalles de las estructuras de datos usadas pueden ser representados por nodos de objeto y pines de acción. El nodo de objeto es usado para modelar clases de contenido y los pines sus atributos.

Durante la ingeniería de requisitos es usual determinar qué datos son representados, donde y cuando. Para modelar grupos de presentación en UWE son usados el estereotipo «display Action», mientras que los dos pines de acción estereotipados «interaction Pin» y «display Pin» son usados para modelar la entrada y la salida de datos. Finalmente el estereotipo «navigationAction», puede ser usado para modelar opciones de navegación y los elementos asociados de presentación (UWE – UML-based Web Engineering, 2012).

iii) Transformaciones

Una vez que los requisitos han sido modelados, hay dos maneras de simplificar los pasos siguientes en el modelado de contenido, navegación, presentación y procesos:

- En vez de crear un modelo y el diagrama correspondiente manualmente, el mismo puede ser generado con una transformación de los datos del modelo de requisitos.
- Adicionalmente, un modelo previamente generado puede ser extendido por nuevas clases transformando desde el modelo de requisitos o agregando a las clases existentes nuevos datos que son dependientes del modelo.

2.6.1.1. MODELO CONCEPTUAL

Este es un diagrama UML normal de clases. UWE apunta a construir un modelo conceptual de una aplicación Web, procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web. La construcción de este modelo conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos.

2.6.1.2. MODELO DE NAVEGACIÓN

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos y enlaces. Los nodos son unidades de navegación y están conectados por medio de enlaces. Pueden ser presentados en diferentes páginas o en una misma página. UWE provee diferentes estereotipos. La forma más simple de obtener un Diagrama de Navegación básico es utilizando la Transformación Para los nodos y enlaces son usados los estereotipos «navigationClass» and «navigationLink». El objetivo es una aplicación en la cual se puede acceder a las operaciones de nuestro diagrama de casos de uso.








nombres de estereotipos y sus iconos	
 clase de navegación	 menú
 índice	 pregunta
 visita guiada	 clase de proceso
 nodo externo	

Tabla 2.3. Estereotipos e iconos de navegación

Fuente: (UWE – UML-based Web Engineering, 2012)

2.6.1.3. MODELO DE PRESENTACIÓN

El Modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenecen a una página web. Podemos usar un Diagrama de Presentación con el fin de proveer esta información. Agrega una clase «presentationPage» y agrega las propiedades con los estereotipos de UWE en ellos para expresar, que el elemento está ubicado en una página web. Las propiedades pueden anidarse.













nombres de estereotipos y sus iconos	
 grupo de presentación	 página de presentación
 texto	 entrada de texto
 ancla	 fileUpload
 botón	 imagen
 formulario	 componente de cliente
 alternativas de presentación	 selección

Tabla 2.4. Estereotipos e iconos de presentación.

Fuente: (UWE – UML-based Web Engineering, 2012)

Mensaje, confirmación y error de validación son modelados aquí, tan sólo para mayor claridad aunque en la visión general de nuestro Diagrama de Navegación no son visibles. Ellos son páginas simples visualizando un mensaje o una pregunta.

2.6.1.4. MODELO DE PROCESO

Hasta ahora podemos modelar muchos aspectos de nuestro sitio web. Pero no hemos hablado en ningún momento de que aspecto tienen las acciones de nuestras clases de proceso. El Modelo de Proceso comprende:

- El modelo de estructura del proceso que describe las relaciones entre las diferentes clases de proceso.
- El modelo de flujo del proceso que especifica las actividades conectadas con cada.

i) Modelo de Estructura del Proceso

Con el fin de describir las relaciones entre las diferentes clases de proceso, creamos un diagrama de clases, usando la transformación de navegación a estructura de proceso (Navigation to Process Structure Transformation).

ii) Modelo de flujo del proceso

Un flujo del proceso (flujo de trabajo) es representado como un diagrama de actividades, describiendo el comportamiento de una clase de proceso, por ejemplo que sucede en detalle, cuando el usuario navega a una clase de proceso. En la tabla 2.5 se tiene a los estereotipos e iconos del flujo de proceso.



nombres de estereotipos y sus iconos	
 acción de usuario	 acción de sistema

Tabla 2.5. Estereotipos e iconos del flujo de proceso.

Fuente: (UWE – UML-based Web Engineering, 2012)

Podemos seleccionar nuestro diagrama de navegación y ejecutar la transformación de navegación a flujo del proceso (Navigation to Process Flows Transformation). El estereotipo «user Action» es usado para indicar iteraciones de usuario con la página web iniciando un proceso o respondiendo a un requerimiento explícito de información. Por el contrario, «system Action» describe acciones, que son ejecutadas por el sistema.

2.6.2. FASES DE LA METODOLOGÍA UWE

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

Las fases o etapas a utilizar son:

- Captura, análisis y especificación de requisitos, en simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipo de la interfaz de usuario.
- Diseño del sistema, se basa en la especificación de requisitos producido por el análisis de los requerimientos, el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

- Codificación del software, durante esta etapa se realizan las tareas que comúnmente se conocen como programación. que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- Pruebas, las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- La Instalación o Fase de Implementación, es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados y eventualmente, configurados. todo ello con el propósito de ser ya utilizados por el usuario final.
- El Mantenimiento, es el proceso de control, mejora y optimización del software ya desarrollado e instalada.

2.7. BPM

BPM, o *BPMS* es el conjunto de servicios y herramientas que facilitan la administración de procesos de negocio. Por administración de procesos entendemos: análisis, definición, ejecución, monitoreo, y control de los procesos. *BPM* además contempla soporte para interacción humana, e integración de aplicaciones, y es aquí la diferencia fundamental con la tecnología de *WorkFlow* existente, que es que *BPM* integra en los flujos a los sistemas.

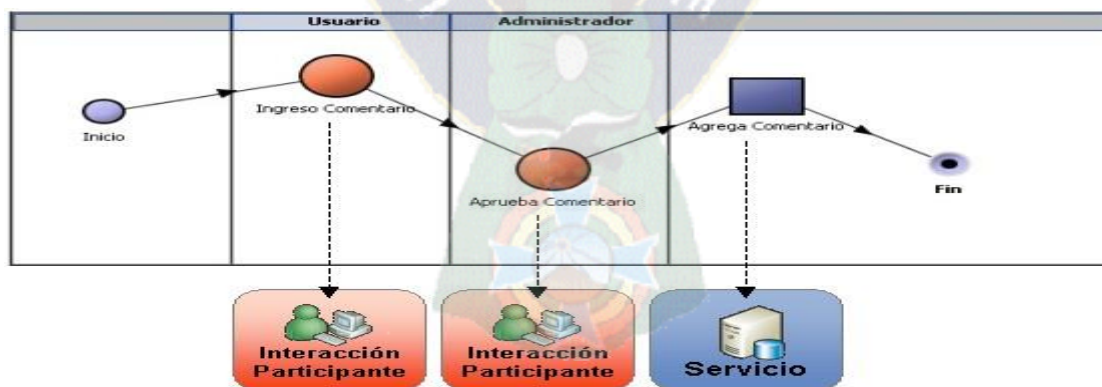


Figura 2.3. BPM

Fuente: (www.soaagenda.com, 2013)

Las soluciones del tipo *WorkFlow* solo se limitaban a definir el flujo de actividades humanas, o de documentos, y con esto obtener el seguimiento de los procesos, pero en estos casos si un participante del proceso requería como parte de sus actividades ingresar datos

en una aplicación, entonces debía salir del ambiente del *WorkFlow*, levantar la aplicación, y luego de terminada su operación volver al *WorkFlow* y registrar el cambio de estado, o término de la actividad. En *BPM* todo está integrado en el mismo flujo lo que es más natural para un participante, el completa su actividad dentro del flujo *BPM*. En la práctica el modelo de proceso *BPM* visualmente es muy parecido a un *WorkFlow*, la diferencia está en que uno puede notar que ciertas actividades son realizadas por personas, y otras son actividades sistematizadas y ambas aparecen en el flujo.

2.7.1. CONCEPTOS *BPM*

La decisión de mejorar los procesos de negocio como un camino para lograr mayor desempeño en los resultados de las compañías no es un tema nuevo. Desde principios de siglo se han venido abordando los procesos con diferentes metodologías con el fin de elevar los resultados financieros de las firmas. En la actualidad asistimos a un escenario de gestión en el cual los procesos requieren de ser gestionados independientemente de un dominio específico de un sistema. Ellos, constituyen el foco y la unidad primaria de iniciativas de automatización e integración de información, necesarios para responder ágilmente a los cambios exigidos por la dinámica del mercado. La gestión de procesos de negocio en estas condiciones ha dado origen a una nueva etapa en la gestión de procesos denominada *BPM*.

2.7.2. *BPM* PRACTICA

La implementación de *BPM* involucra la articulación de la estrategia, los procesos y la tecnología de una empresa para generar valor al negocio. A diferencia de los modelos de gestión anteriores, *BPM* se concentra en la articulación de las iniciativas estratégicas con los procesos de negocio, apalancados en estándares tecnológicos que facilitan su despliegue alineado en las operaciones diarias de la organización.

Para lograr esta articulación es necesario desarrollar una serie de procesos que permiten alinear de manera controlada, los aspectos estratégicos del negocio, a través de la identificación y articulación de los conceptos claves del proceso y la asociación de los componentes tecnológicos que permitan flexibilizar los cambios en la cotidianidad empresarial. En la práctica la implantación de esta disciplina de mejoramiento requiere por

parte de la empresa, una dosis de pensamiento en procesos de negocio y la utilización de tecnologías de Información centradas en procesos.



Figura 2.4. *BPM* articula la estrategia, los procesos y la tecnología de una organización

Fuente: (sanchez, 2004)

Pensar en procesos de negocio significa que las acciones de cambio que se ejercen sobre el proceso, son evaluadas y planeadas teniendo en cuenta las diferentes dimensiones que juegan en la dinámica del mismo. Esto quiere decir que el proceso se evalúa revisando las actividades que se llevan a cabo, buscando eliminar aquellas que no adicionan valor e identificando las políticas, reglas de negocio y normas que determinan las decisiones que la organización toma sobre el proceso. De igual manera se examinan los trabajos y roles que la empresa destina a la realización del proceso, con el fin de gestionar las barreras culturales, paradigmas, conocimientos y competencias requeridas para su realización.

2.7.3. LA TECNOLOGÍA DE *BPM*

La tecnología que posibilita la implantación y adopción de *BPM* constituye una categoría nueva de sistemas de información denominada *BPMS*. Inicialmente y de manera general un *BPMS* puede ser definido como un conjunto de utilidades de software para definir, implementar y mejorar procesos de negocio que cumplen con un grupo de características técnicas necesarias para aplicar el concepto de *BPM*. Estos sistemas permiten manejar el ciclo de vida del proceso a través de características funcionales y no funcionales que

posibilitan definir, modelar, implementar y mejorar el proceso durante su operación. Un sistema *BPMS* está en capacidad de realizar las siguientes operaciones:

- Modelado de procesos de negocio
- Provee entornos de desarrollo de aplicaciones para colaboración entre procesos de negocio.
- Generación, actualización y publicación de documentación de procesos.
- Simulación de procesos de negocio para evaluar su comportamiento en situaciones de carga exigidas en determinados momentos del proceso.
- Integración de información proveniente de otros sistemas de negocio.
- Automatización de procesos.
- Colaboración entre las empresas que participan en la cadena productiva.
- Despliegue de aplicaciones que soportan el proceso en condiciones tales que no se requiere mayor conocimiento y experiencia de un usuario final.
- Análisis de procesos y comportamiento de la operación.
- Gestión de ciclo de generación publicación y consumo del conocimiento generado en la operación del proceso.

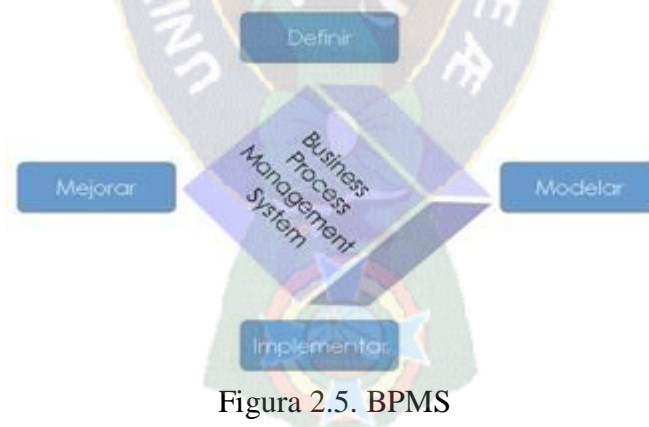


Figura 2.5. BPMS

Fuente: (sanchez, 2004)

Estas características constituyen la base sobre la cual se desarrolla el modelado, simulación e implementación de procesos. La flexibilidad y agilidad en el diseño de procesos, se basan en la abstracción de la realidad que plasma el arquitecto de negocio y las posibilidades del sistema para representar esta realidad de manera gráfica.

CAPÍTULO III MARCO APLICATIVO

3. INTRODUCCIÓN

Para el desarrollo del SISTEMA WEB DE REGISTRO, SEGUIMIENTO Y CONTROL DE CORRESPONDENCIA BASADO EN *BPM* CASO: CARRERA DE INFORMÁTICA se utilizara un enfoque de análisis y diseño de una metodología ágil como es *AUP*, que se utilizara con la herramienta de modelado que es *UWE*, los cuales tienen un flujo de actividades y requerimientos de usuario en el software.

El desarrollo de software está bajo un ciclo de vida iterativo incremental el cual tiene las siguientes ventajas:

- Reduce el riesgo de tener un producto ineficiente
- permite controlar los tiempos previstos del producto

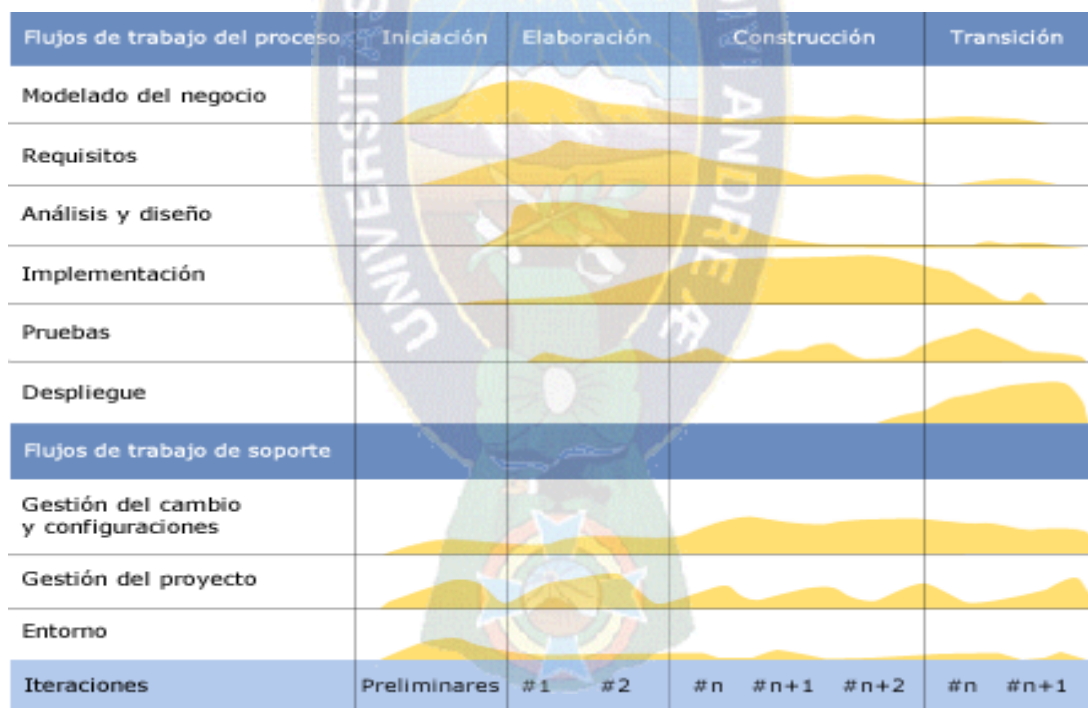


Figura 3.1. AUP fases

Fuente: (Bolivariana, 2013)

Se definirá las fases en cuales se trabajara:

Fase de inicio:

En la fase de inicio entraran los requisitos, las especificaciones de los requerimientos para el sistema web, el modelado anterior y actual de la institución, el flujo de proceso de correspondencia *BPM* actual.

Fase de Elaboración:

En la Fase de elaboración se trabajara con el desarrollo de modelado *UWE*, *BPM* y *MVC*, cual abarcara el modelo conceptual, modelo de presentación.

Fase de Construcción y Fase de Transición:

En esta dos fases se unirán por que se hará en una iteración se colocara los diagramas de despliegue, diagrama de componentes, se adicionara también las pruebas de estrés y de aceptación y pantallas del sistema



3.1. FASE DE INICIO

En la fase de inicio estableceremos los pilares fundamentales en los cuales se basara el proyecto

3.1.1. MODELO DE NEGOCIO

El modelo de la organización consta de un modelo de casos de uso de la organización y un modelo de objetos de la organización. El modelo de casos de uso de la organización consiste en un conjunto de diagramas de casos de uso, contruidos con base *UWE* para el modelado que se basa en *UML*.

El modelo de objetos de la organización consiste en un conjunto de diagramas de objetos opcionalmente explicados mediante diagramas de secuencia contruidos con *UWE* que está en base de *UML* para el modelado de la organización.

ACTOR	DESCRIPCIÓN
 <p>Jefe de Unidad</p>	<p>Es el encargado que recibe un documento y da una respuesta.</p>
 <p>Administrativo</p>	<p>Este actor se encarga de realizar tratamiento respecto a los tramites asignados a su persona</p>



 Secretaria	Actor encargado de centralizar el movimiento de correspondencia dentro de la unidad, se encarga de registro, despacho y elaboración de documentos
 Remitente	Es aquel usuario que quiere averiguar en qué estado esta su trámite.

Tabla 3.1. Identificación de los Usuarios y Actores

Fuente: [Elaboración propia]

En la carrera de informática se muestra una deficiencia en el control de correspondencia como se mostrara en el siguiente diagrama de caso de uso. El proceso actual de registro de la correspondencia, el problema es en la demora de entrega y recepción de la correspondencia, como en la figura 3.2.



Figura 3.2. Casos de uso actual

Fuente: [Elaboración propia]

3.1.2. MODELO DE REQUISITOS

Los requisitos del sistema serán:

- Identificación del producto

El software lleva como nombre "sistema web de registro, seguimiento y control de correspondencia basado en *BPM* caso: carrera de informática"

- ¿Qué hará el sistema?

El sistema tendrá las operaciones básicas como ser listar, adición, modificación y eliminación de la correspondencia, además de contar con las operaciones de control y seguimiento, El sistema utilizar niveles de usuario para así poder evitar la intromisión de usuarios a páginas no autorizadas

El sistema contara con un motor de búsqueda para agilizar, controlar y ubicar la documentación.

El sistema tendrá un módulo de auditoría el cual facilitara a los auditores para realizar pruebas.

- ¿Que no hará el sistema?

El sistema no realizar aquellos procesos en los cuales intervenga el razonamiento lógico personal.

- Beneficios

El sistema beneficiara a las unidades y subunidades dentro de la carrera de informática, facilitara el acceso de la correspondencia, también coadyuvara en el almacenamiento correcto, recepción, distribución de los documentos entrantes y salientes.

- Riesgos del Proyecto

Para la realización del presente sistema se ha identificado riesgos los cuales, y al mismo tiempo se propuso acciones a seguir:

- Una mala aplicación de la metodología propuesta

Acción: Proveer información sobre la metodología AUP, proyecto ya realizados, libros, contenido multimedia o de internet que coadyuven con la metodología

- Poco conocimiento en las herramientas y plataforma de desarrollo, lo cual ocasionaría retrasos y alteración en el cronograma de actividades inicialmente programadas

Acción: tomar en cuenta el aprendizaje y dominio de las herramientas y plataformas de desarrollo del sistema, haciendo un estudio de factibilidad el cual coadyuvara para evitar cualquier contingencia.

- Que las actividades planificadas no se lleven a cabo en el tiempo establecido para la entrega.

Acción: Realizar un calendario flexible a cambios y especifique tiempos máximos para la demora.

A continuación se hará una descripción detallada de cada tipo de documento que llega a la Carrera de Informática y su implementación.

Nº	Nombre	Implementación
1	otro	Implementado
2	Convocatoria de Ayudantía	Falta detalle
3	Convocatoria de Docente	Falta detalle
4	Acreditación	Falta detalle
5	Acreditación Asociación Docente	Falta detalle
6	Acreditación Asociación Estudiantil	Falta detalle
7	Actas de Calificaciones	Falta detalle
8	Becas	Implementado
9	Boleta de Garantía	Falta detalle
10	Certificado de Conclusión de Materias	Implementado
11	Certificación Presupuestaria	Falta detalle
12	Certificado de Notas	Falta detalle
13	Circular	Implementado
14	Citación	Implementado
15	Comunicado	Implementado
16	Contratación	Falta detalle
17	Contrato Beca Trabajo	Falta detalle
18	Contrato Docentes	Falta detalle
19	Convalidación de Materias	Falta detalle
20	Convenio	Falta detalle
21	Convocatoria	Falta detalle
22	Defensa de Tesis	Falta detalle
23	Descargo de Viajes	Falta detalle
24	Designación de Tribunal	Falta detalle
25	Designación Docente	Falta detalle
26	Disponibilidad Financiera	Falta detalle
27	Evaluación Docente	Falta detalle

28	Informe de Avance de Tareas	Falta detalle
29	Informe de Proyecto	Falta detalle
30	Informe Jurídico	Falta detalle
31	Informes de Legalización	Falta detalle
32	Invitaciones	Falta detalle
33	Licencias	Falta detalle
34	Memo de Felicitación	Implementado
35	Memo de Llamada de Atención	Implementado
36	Modificación Presupuestaria	Falta detalle
37	Nombramiento Coordinador de Postgrado	Falta detalle
38	Nombramiento Director Académico	Falta detalle
39	Nombramiento Director de Carrera	Falta detalle
40	Nombramiento Director de Instituto	Falta detalle
41	Nombramiento de Autoridades	Falta detalle
42	Partes de Asistencia	Implementado
43	Partes de Asistencia Auxiliares	Falta detalle
44	Partes de Asistencia Docentes	Falta detalle
45	Prácticas Profesionales	Falta detalle
46	Presupuesto POA - Agenda	Falta detalle
47	Reglamento	Falta detalle
48	Reincorporación Docente	Falta detalle
49	Resolución Administrativa	Falta detalle
50	Resolución HCU	Falta detalle
51	Resolución HCF	Falta detalle
52	Resolución HCC	Falta detalle
53	Revalidación de Cheques	Falta detalle
54	Revalidación de Títulos	Falta detalle
55	Solicitud de año Sabático	Falta detalle
56	Solicitud Caja Chica	Falta detalle
57	Solicitud Compra	Falta detalle
58	Solicitud de Fondo en Avance	Falta detalle
59	Solicitud de Pago	Falta detalle
60	Traspaso de Carrera	Falta detalle
61	Traspaso Presupuestario	Falta detalle
62	Otros Ingresos	Falta detalle
63	Certificado de alumno regular	Falta detalle
64	practicar pre profesionales	Falta detalle

65	Implementación de proyecto	Falta detalle
66	Solicitud de matriculación 2009	Implementado
67	Aprobación de calendario académico	Falta detalle
68	Remisión de copia	Falta detalle
69	Solicitud de emisión de matrícula	Falta detalle
70	Acreditación facultativa	Falta detalle
71	Anulación de materias Repetidas	Falta detalle
72	Solicitud de matriculación rezagado	Falta detalle
73	Pasantía	Implementado
74	Préstamo sala audiovisual	Falta detalle
75	Solicitud alumno regular	Falta detalle
76	Solicitud de acreditación de Célula Académica	No implementado
77	Carta para Pasantía	No implementado
78	Solicitud de Aula para Ayudantía	No implementado
79	Solicitud de Aula para Célula Académica	No implementado
80	Solicitud de Laboratorio para Ayudantía	No implementado
81	Solicitud de Laboratorio para Célula Académica	No implementado
82	Solicitud de Laboratorio para seminario	No implementado
83	Solicitud de Aula varios.	No implementado
84	Solicitud de Laboratorio varios.	No implementado
85	Solicitud para romper pre requisitos	No implementado
86	Presentación de perfil de tesis taller 1	No implementado
87	Presentación de documentos para defensa de Tesis	No implementado
88	Solicitud de firma de record Académico	No implementado
89	Solicitud de firma para tramites varios	No implementado
90	Solicitud Open Info	No implementado
91	Solicitudes de Alumnos Varias	No implementado

Tabla 3.2. Tipo de Documento y su estado de implementación.

Fuente: [Elaboración propia]

En la tabla 3.2 hace referencia al estado de cada tipo de documento que llega a la carrera de informática, cada documento tiene un proceso, cada proceso tiene un principio y fin, pero hay algunos documentos que no cuentan con un específico detalle del proceso que debe seguir, otros se encuentran implementado y con su respectivo proceso.

Hay algunos tipos de documentos que no están en la lista y eso se debe a que cada vez hay un nuevo proceso de correspondencia que se debe añadir al sistema con su respectivo proceso.

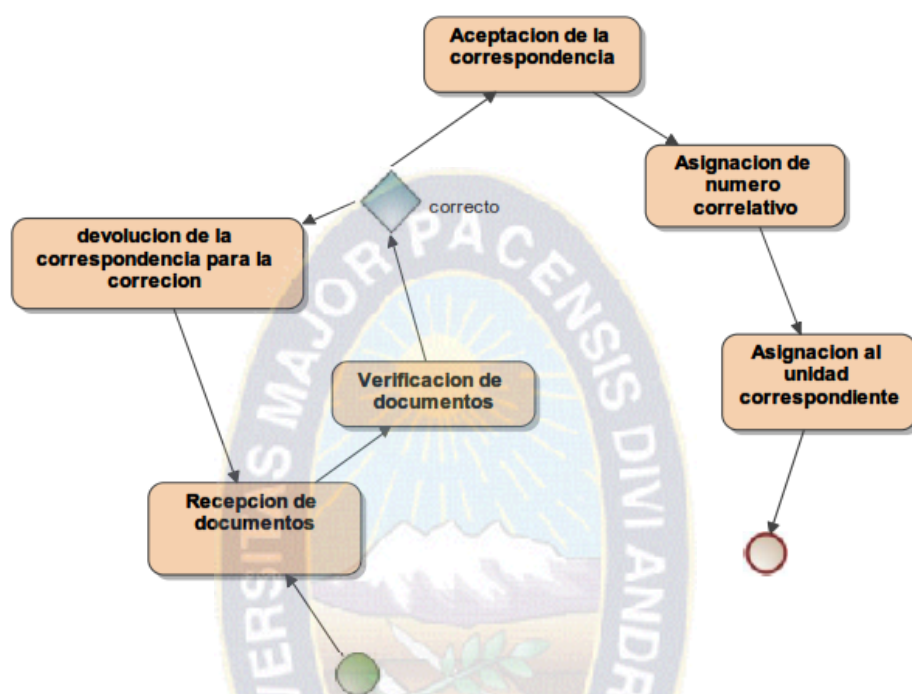


Figura 3.3. Recepción de correspondencia

Fuente: [Elaboración propia]

Mediante los diagramas *BPM* se puede observar en la figura 3.3. El flujo de proceso de la información al momento de recepción de la correspondencia dentro de la carrera de informática, en la figura 3.4 se puede observar en un diagrama *BPM* el flujo de procesamiento que sigue la correspondencia dentro la dirección y su archivado o despacho.

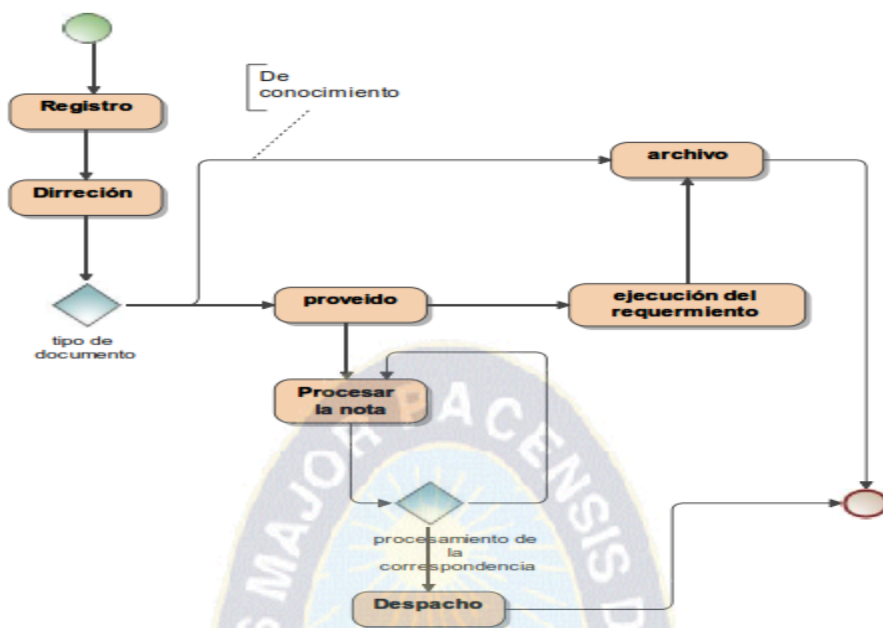


Figura 3.4. Flujo actual de la carrera de informática

Fuente: [Elaboración propia]

El sistema tendrá que satisfacer los requerimientos y los diagramas de flujo *BPM* que se están utilizando en la primera fase.

3.2. FASE DE ELABORACIÓN

El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.

3.2.1. DIAGRAMA DE CASOS DE USO

El diagrama de caso de uso general muestra una parte de la funcionalidad del sistema, ya que los procesos más importantes estarán a automatizados.

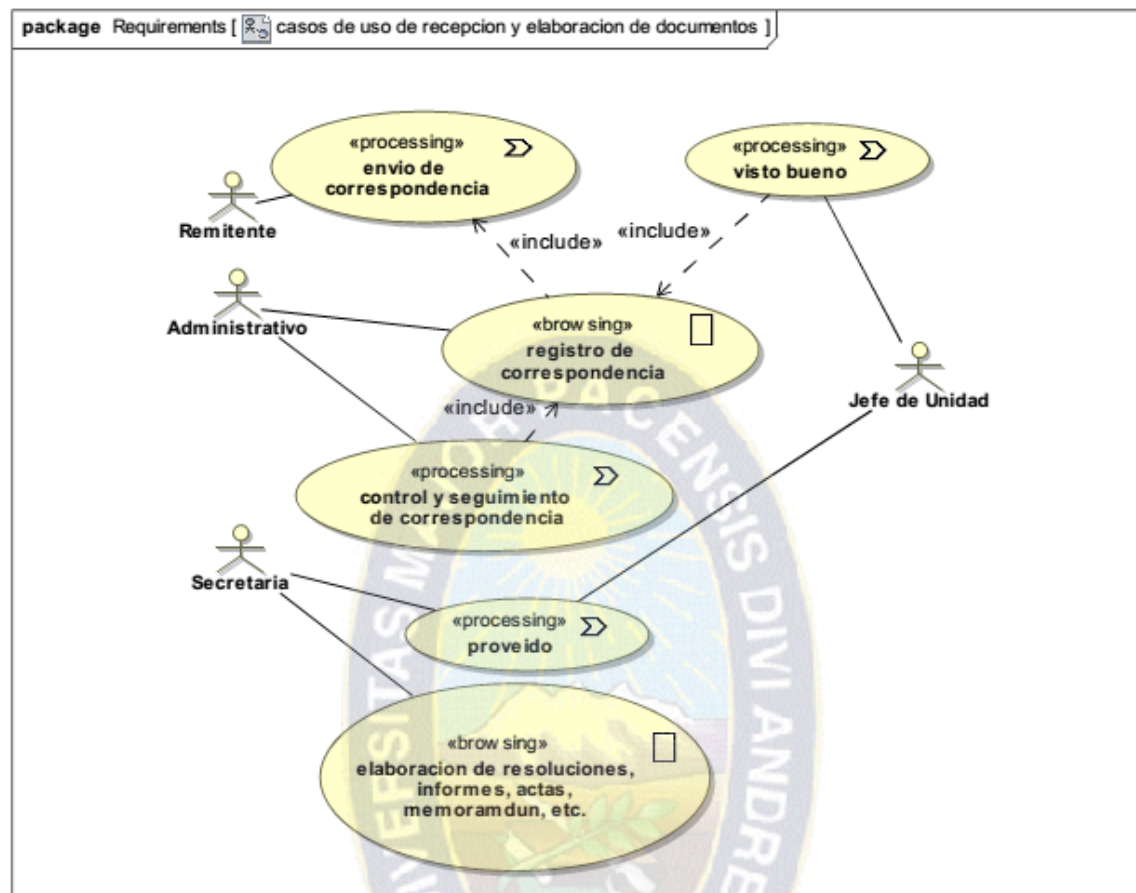


Figura 3.5. Modelo de casos de uso de recepción y elaboración de documentos

Fuente: [Elaboración propia]

En la figura 3.5. Y en la tabla 3.3 se plantea una solución al problema actual que tiene la carrera en cuanto a la elaboración y recepción de documentos.

Caso de uso:	Registra nueva correspondencia, elaboración de documentos.
Actores:	Remitente , administrativo y jefe de unidad, Secretaria
Propósito:	Registrar una nueva correspondencia, dar vistos bueno a la documentación y elaborar respuestas.
Descripción:	El remitente entregara un documento que será recibido por un administrativo el cual se encargara de la recepción y asignara a la unidad que corresponde.

	El jefe de unidad o administrativo dará un proveído a la secretaria para que elabore un documento de respuesta
Tipo:	Primario y esencial

Tabla 3.3. Descripción del caso de uso general

Fuente: [Elaboración propia]

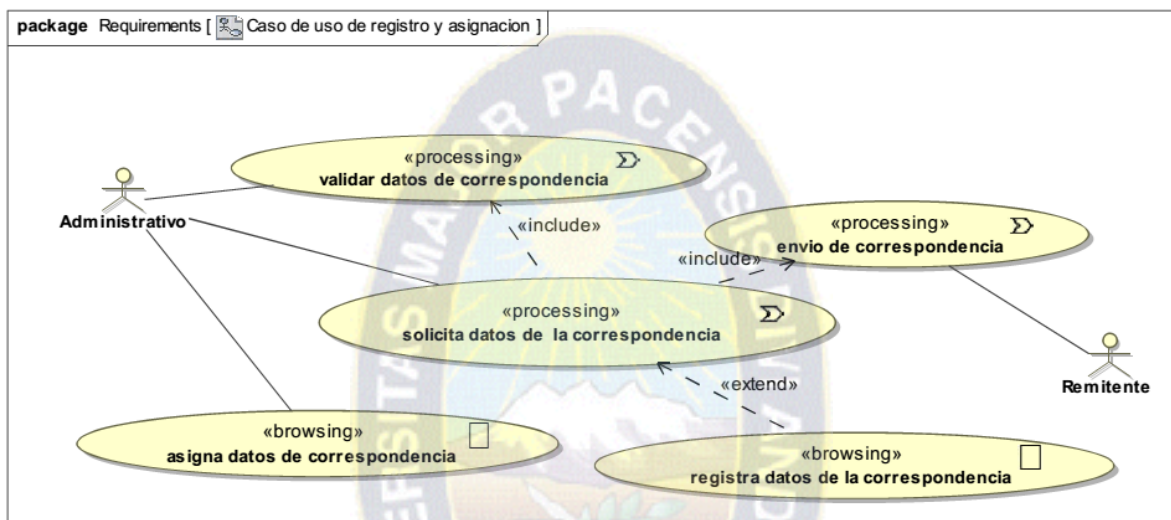


Figura 3.6. Casos de Uso de Registro y Asignación

Fuente: [Elaboración propia]

Caso de uso:	Registra y asigna nueva correspondencia
Actores:	Remitente , administrativo y jefe de unidad
Propósito:	Registrar y asignar detalladamente la nueva correspondencia
Descripción:	El remitente enviara la correspondencia el cual será recibido por un administrativo el que se encargara de solicitar y validara los datos necesarios para el registro de la nueva correspondencia, luego se le asignara al administrativo o personal destinatario.
Tipo:	Primario y esencial

Tabla 3.4. Descripción del caso de uso de Registro y asignación

Fuente: [Elaboración propia]

En la figura 3.6. Y en la tabla 3.4 se plantea una solución al problema actual que tiene la carrera en cuanto al registro y asignación.

3.2.2. MODELO CONCEPTUAL

En este modelo se puede ver el diseño conceptual del sistema donde se hace la relación de las clases involucradas en el sistema, así de esta manera podemos definir de manera correcta los espacios de aplicación del sistema. La construcción de este se lleva a cabo de acuerdo a los casos de uso descritos anteriormente. En la figura 3.7. Se tiene al diagrama del modelo conceptual que para UWE es el diagrama de clases que se utiliza en el sistema web.

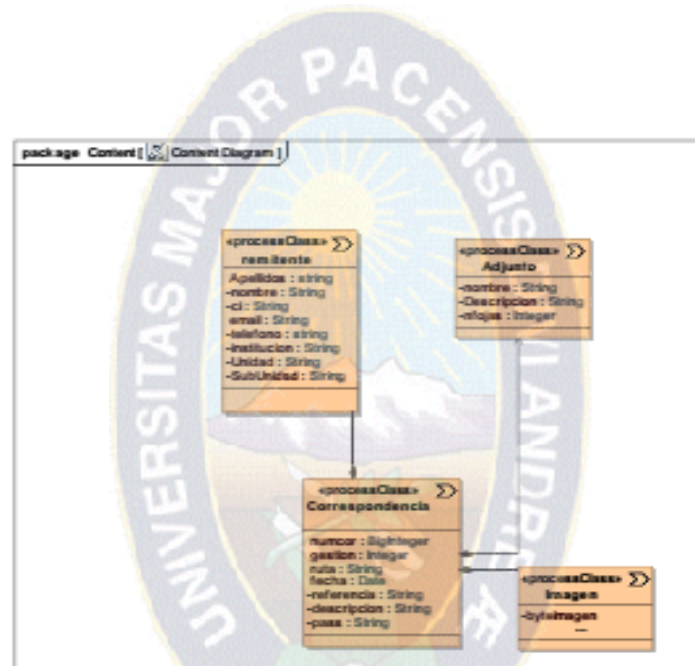


Figura 3. 7.Caso de Uso de Registro de Correspondencia

Fuente: [Elaboración propia]

3.2.3. DIAGRAMA DE NAVEGACIÓN

La Figura 3.8. De diagrama de navegación trata de escenificar el caso simple de un sistema web con muchos parámetros y el flujo de navegación que sigue.

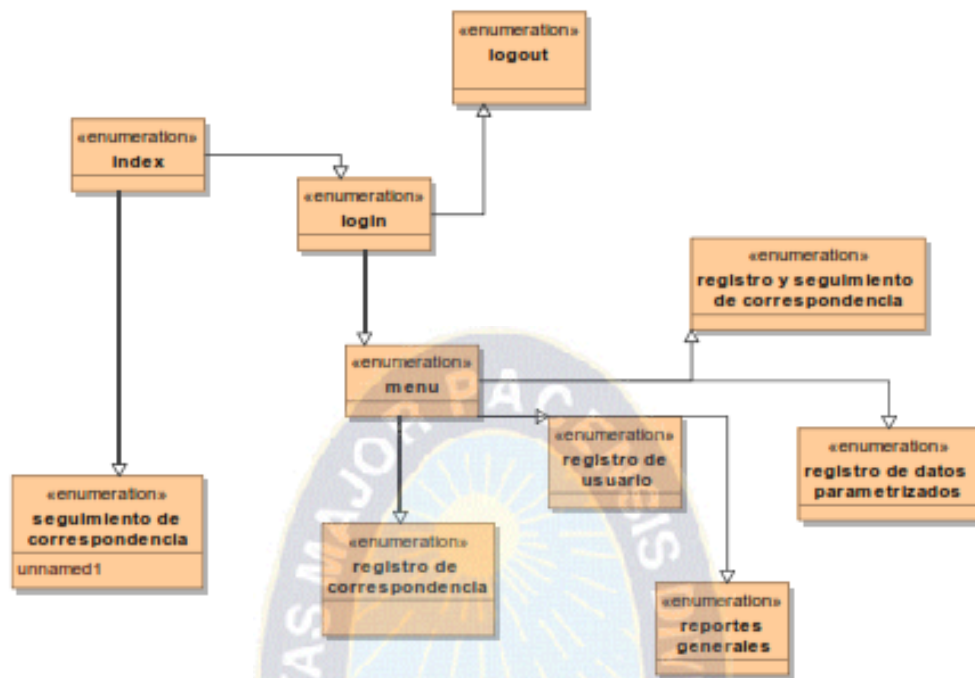


Figura 3.8. Diagrama de navegación

Fuente: [Elaboración propia]

3.2.4. MODELO DE PRESENTACIÓN

En este modelo se puede ver cómo se presenta el sistema web que tiene distintas vistas como se pudo ver en el diagrama de navegación de acuerdo al rol con el cual se ingresa. En este modelo se cuenta con una página de inicio que llega a ser la autenticación del usuario, una vez autenticado el usuario se presentan dos alternativas del área de contenido y así mismo se tienen distintos menús de navegación, en la figura 3.8. Diagrama se mostrara el análisis de componentes del modelo de presentación

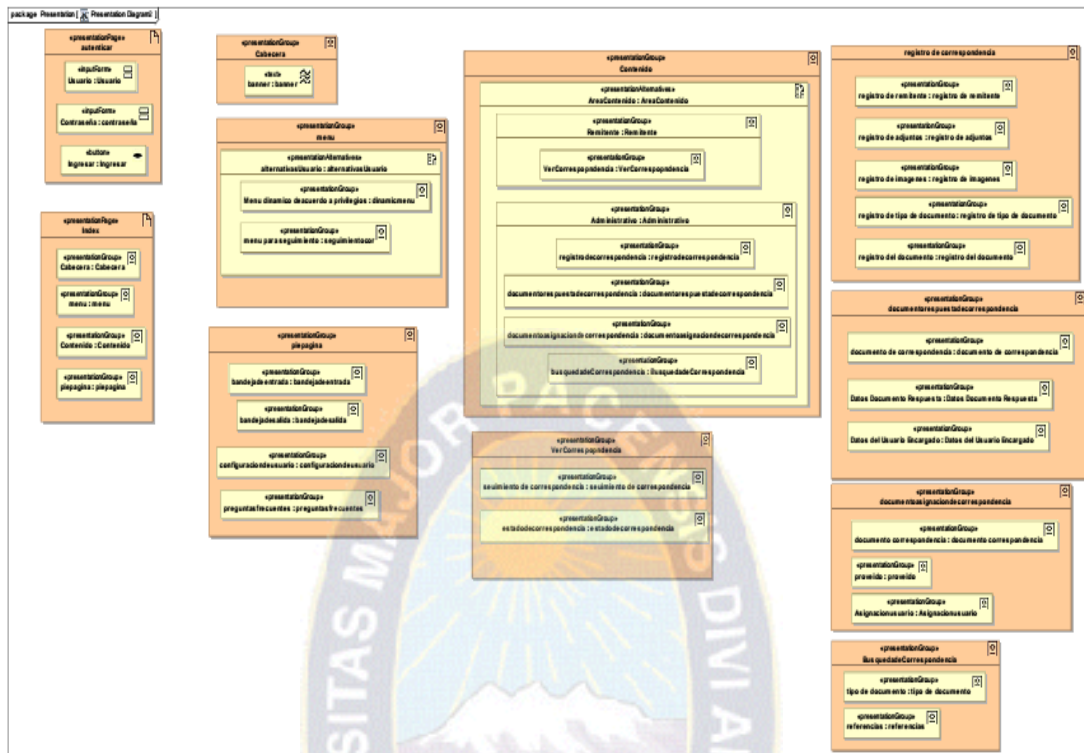


Figura 3.9. Componentes del modelo de presentación

Fuente: [Elaboración propia]

Como primer componente se tiene a la página de autenticación que nos dirá el tipo de usuario con el cual se está ingresando al sistema. En la figura 3.10 se tiene a este componente de autenticación.

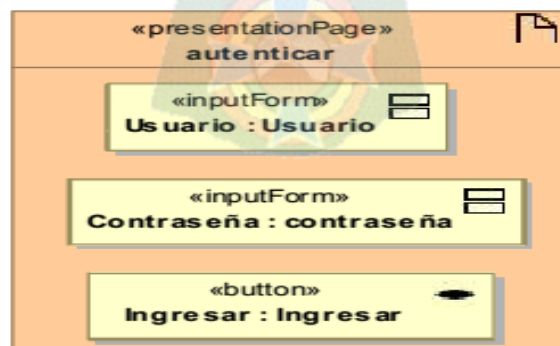


Figura 3 10. Autenticación de usuario

Fuente: [Elaboración propia]

La siguiente presentación una vez autenticado en el sistema es la página principal, en la figura 3.11 tenemos este componente



Figura 3.11. Cuerpo del sitio web

Fuente: [Elaboración propia]

Como se puede observar en la figura 3.11 se puede obtener el primer componente que es la cabecera que se observara en la figura 3.12

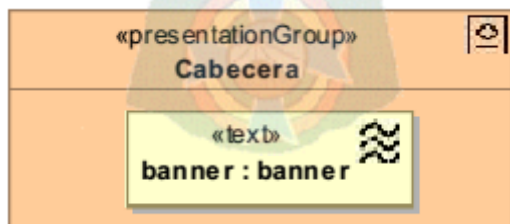


Figura 3.12. Cabecera de la Página

Fuente: [Elaboración propia]

En la figura 3.13. Se observara el menú dinámico que cuenta el sistema el cual se obtendrá de acuerdo a los permisos del sistema

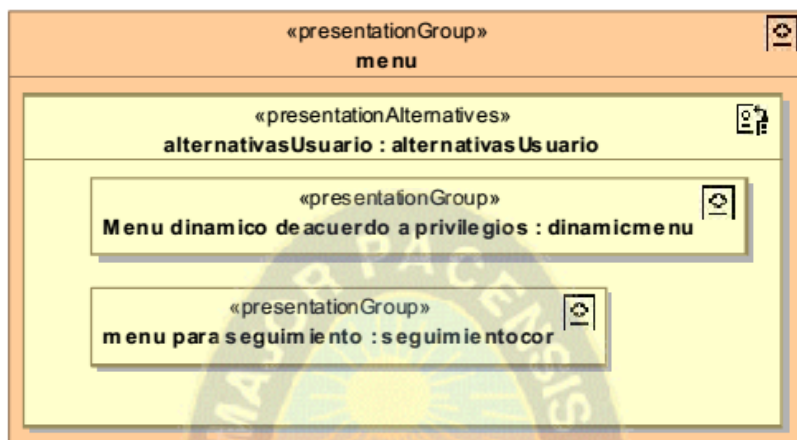


Figura 3.13. Menú del sistema

Fuente: [Elaboración propia]

En la figura 3.14 se observara el tercer componente que es el pie de página el cual proporcionara accesos directos a los procesos principales del sistema



Figura 3.14. Pie de Pagina

Fuente: [Elaboración propia]

En la figura 3.15 en la página de contenido en la cual se desplazara todos los documentos necesarios para el funcionamiento de la página.

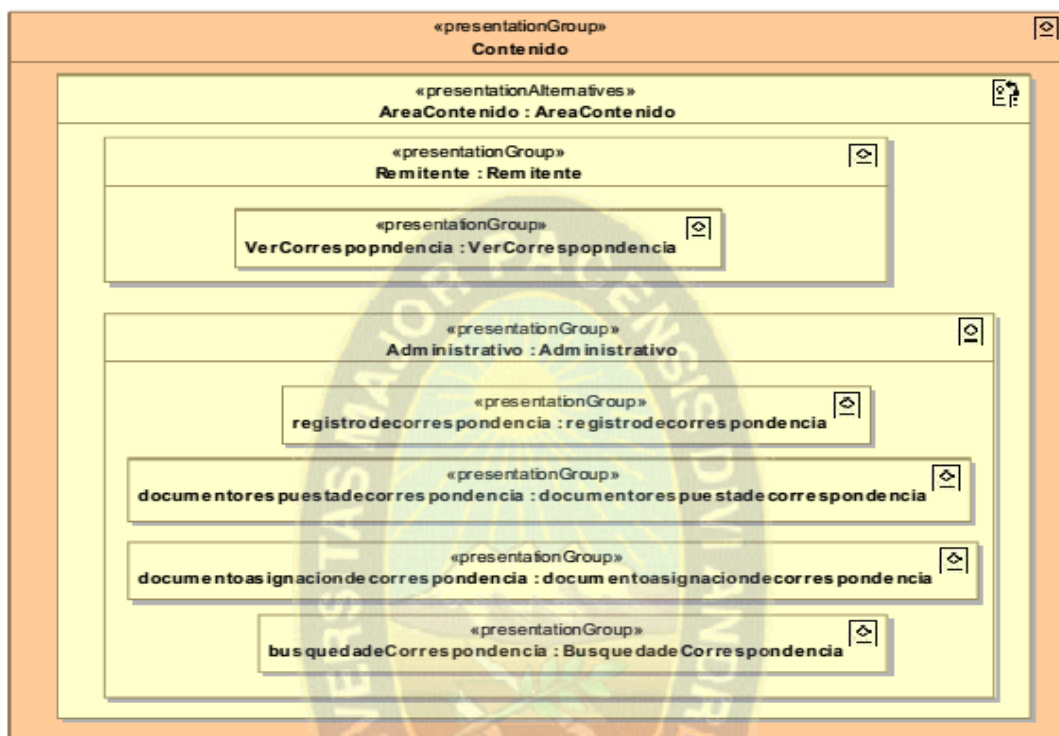


Figura 3.15. Página de contenido

Fuente: [Elaboración propia]

En la figura 3.16 se observara el proceso de consulta de la correspondencia y sus respectivos datos y en especial su estado.

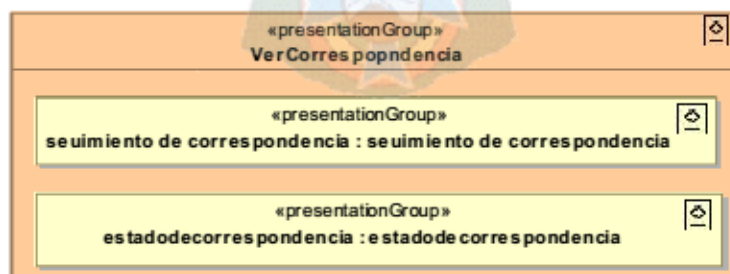


Figura 3.16. Consulta de Correspondencia

Fuente: [Elaboración propia]

En el proceso de registro de correspondencia de la figura 3.17 se muestra el modo de registro en la capa de presentación.

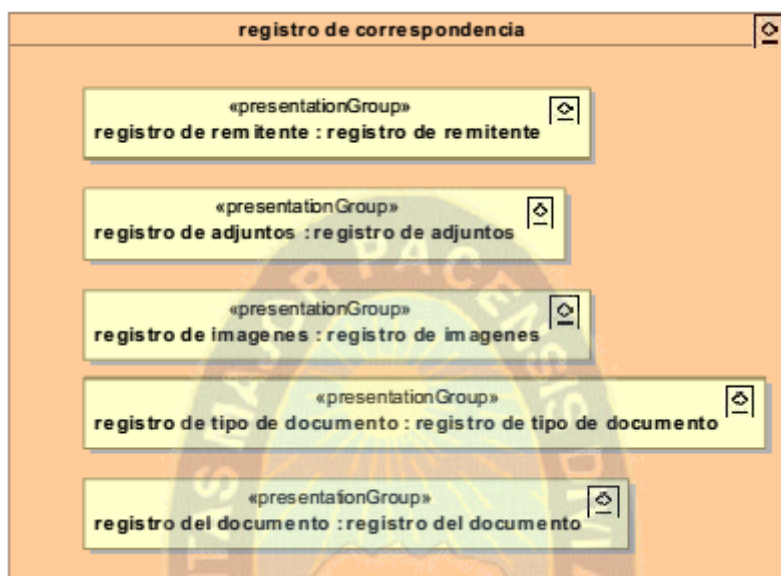


Figura 3.17. Registro de correspondencia en el modelo de presentación

Fuente: [Elaboración propia]

En la figura 3.18 se puede observar el modelo de elaboración de documento final el cual nos permitirá elaborar informes finales de la correspondencia recibida



Figura 3.18. Elaboración de Respuesta de la Correspondencia

Fuente: [Elaboración propia]

En la figura 3.19. Se puede observar cómo se asigna la correspondencia y se enviara a diferentes aéreas con el proveído.

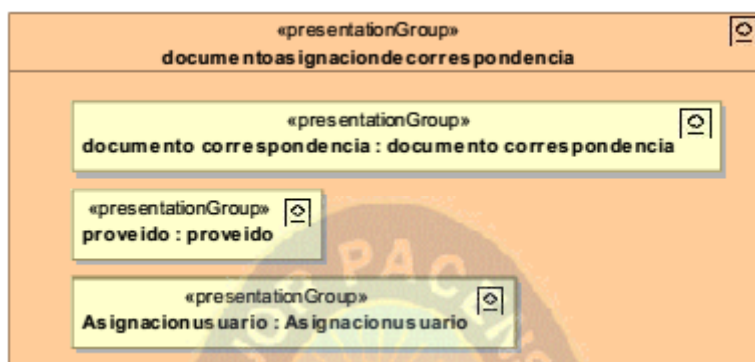


Figura 3.19. Asignación de documentos

Fuente: [Elaboración propia]

En la figura 3.20. Se podrá observar los parámetros para la búsqueda de documentación recibida y despachada.

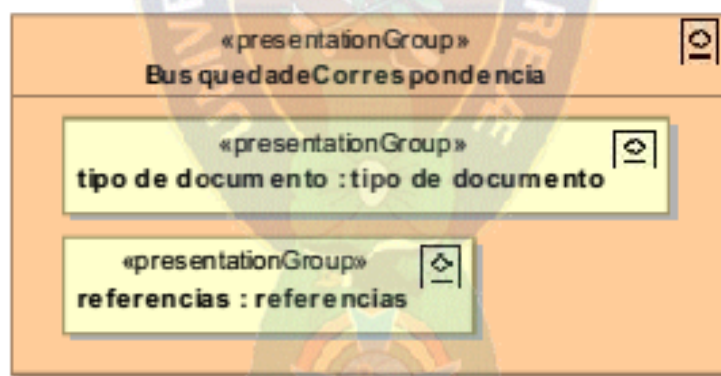


Figura 3.20. Búsqueda de Correspondencia

Fuente: [Elaboración propia]

3.2.5. MODELO DE PROCESO

El modelo de proceso cuenta con dos partes que son la estructura de procesos que será descrito por UWE, y los flujos de procesos que serán descritos por diagramas pertenecientes a *BPM*.

3.2.6. MODELO DE ESTRUCTURA DE PROCESO

En este modelo se ven como es que están estructurados los procesos, la mayoría de estos ya se conocen por los modelos de navegación y conceptual.

En el proceso de registro de la correspondencia de entrada se obtiene los procesos de digitalización de la correspondencia de entrada. Como se detallara el proceso de digitalización de la figura 3.21.

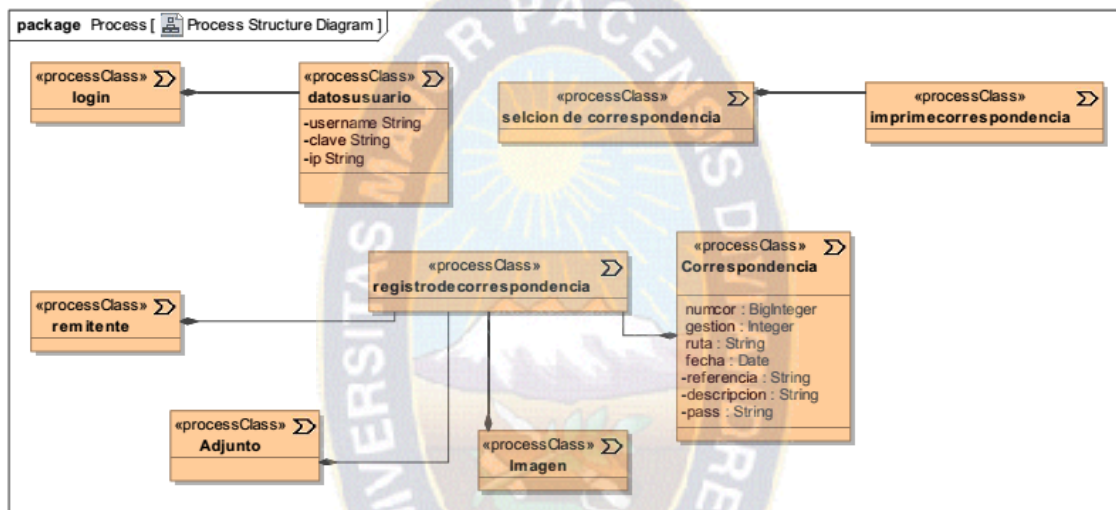


Figura 3.21. Registro de correspondencia

Fuente: [Elaboración propia]

Con los siguientes diagramas *BPM* se tratara de complementar los diagramas UWE, así también tratara de mejorar los flujos de trabajo que mantiene la carrera de informática.

3.2.7. MODELO DE FLUJO DE PROCESO

En el modelo de flujo de proceso se hace uso de los diagramas pertenecientes a *BPM* los cuales hacen reemplazo de los diagrama de actividades, una descripción de los procesos de autenticación, registro de nueva correspondencia y proceso de gestión de correspondencia.

3.2.7.1. AUTENTIFICACIÓN DE USUARIO

El diagrama *BPM* de autenticación de usuario reemplaza el diagrama de actividades, el cual especificara el acceso al sistema y el flujo que debe suceder cuando el usuario es identificado o no. como se muestra en la figura 3.22.

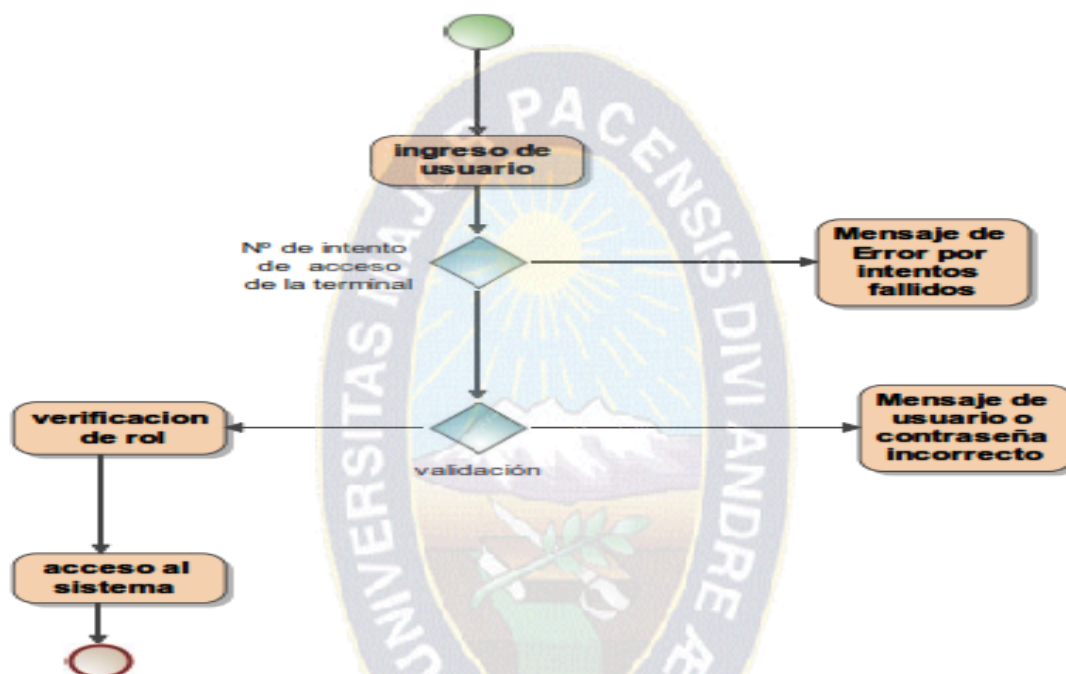


Figura 3.22. Acceso al sistema

Fuente: [Elaboración propia]

3.2.7.2. REGISTRO DE NUEVA CORRESPONDENCIA

El diagrama *BPM* de registro de nueva correspondencia reemplaza el diagrama de actividades, el cual especificara el flujo que tiene la nueva correspondencia, proceso por el cual debe seguir para ser recibida o rechazada la correspondencia, como se muestra en la figura 3.23

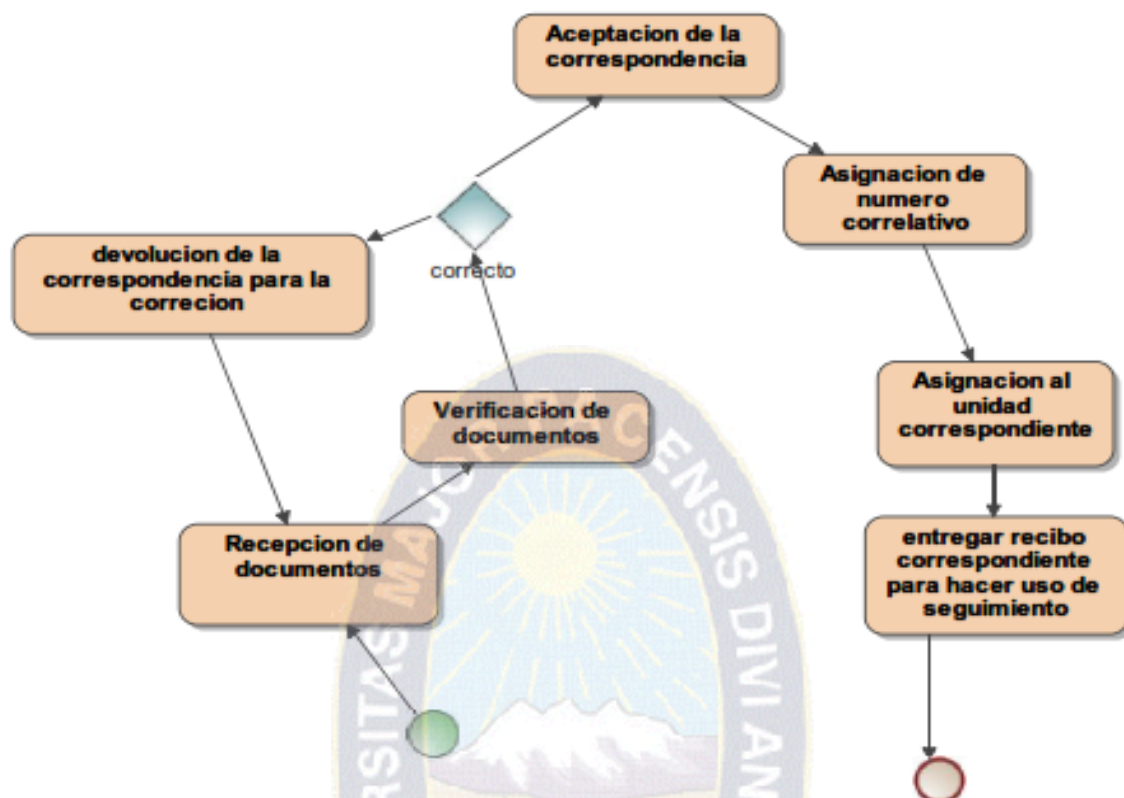


Figura 3.23. Registro de nueva correspondencia

Fuente: [Elaboración propia]

3.2.7.3. PROCESO DE GESTIÓN DE CORRESPONDENCIA

El diagrama *BPM* de gestión de correspondencia reemplaza el diagrama de actividades, el cual especificara el proceso de gestión que recibe la correspondencia hasta su culminación, como se muestra en la figura 3.24

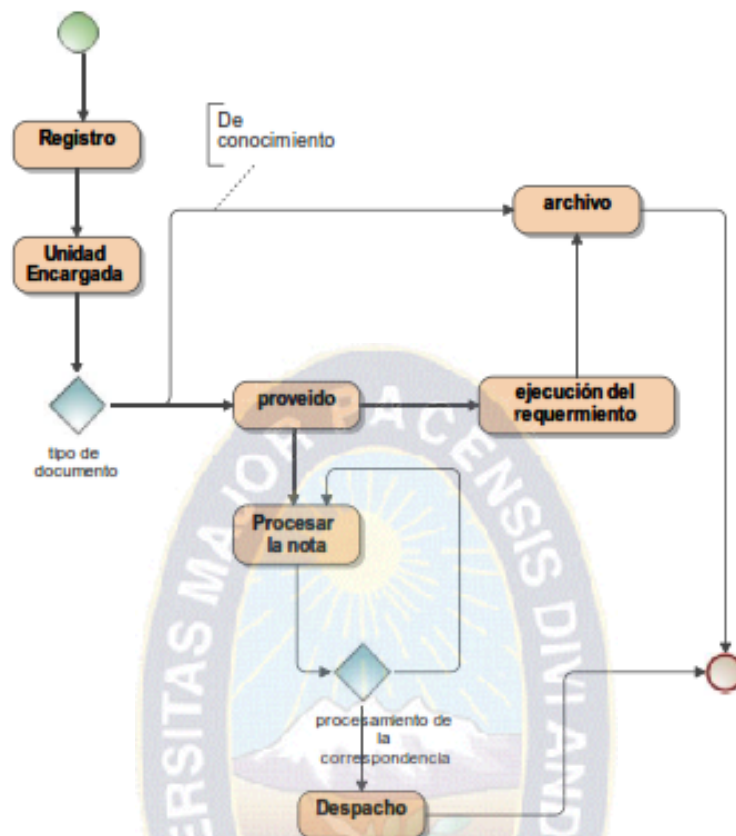


Figura 3.24. Proceso de gestión de correspondencia

Fuente: [Elaboración propia]

3.3. FASE DE CONSTRUCCIÓN

El objetivo es construir un software funcional, en unas bases incrementales que cumplan con las necesidades más prioritarias del cliente.

3.3.1. DIAGRAMA DE CLASES

El diagrama de clases se utilizara diagrama es el tipo de diagramas estáticos que se describe de un sistema mostrando sus clases orientadas a objetos mediante el modelado UWE. Como se muestra en la figura 3.25.

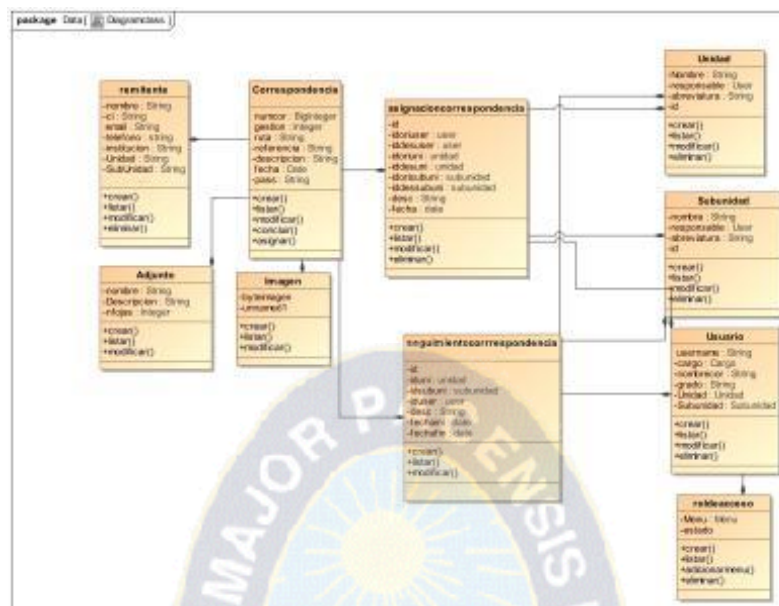


Figura 3.25. Diagrama de clases de sistema de correspondencia

Fuente: [Elaboración propia]

3.3.2. DIAGRAMA DE PAQUETES

En el diagrama de paquetes se mostrara como el sistema es dividido en agrupaciones lógicas mostrando sus dependencias entre esas agrupaciones para suministrar una descomposición lógica del sistema.

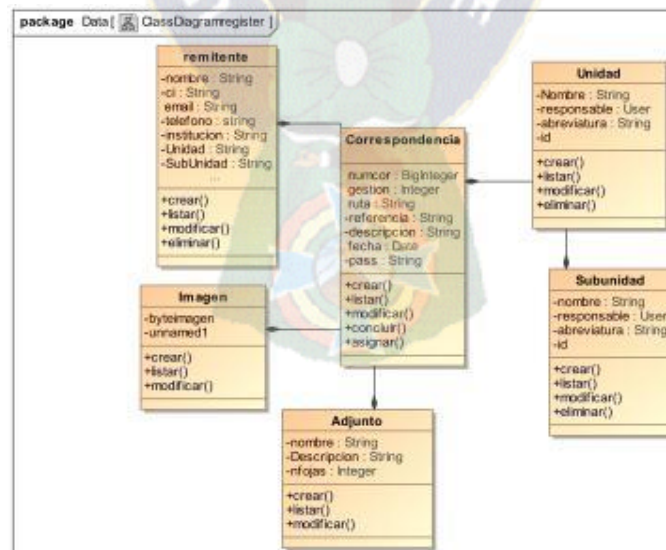


Figura 3.26. Paquete de registro de correspondencia

Fuente: [Elaboración propia]

En la figura 3.26. Podemos observar el diagrama de paquetes para el registro de correspondencia

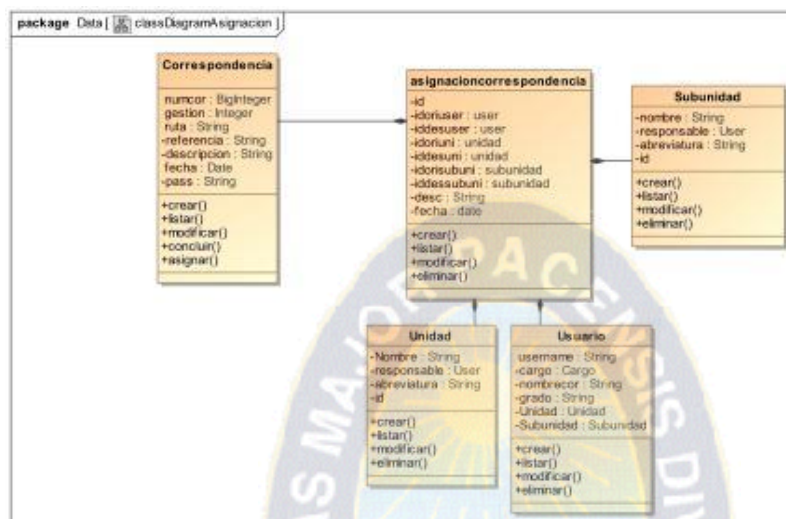


Figura 3.27. Paquete de envío y asignación de correspondencia

Fuente: [Elaboración propia]

En la figura 3.27. Podemos observar el diagrama de paquetes para el registro él envío y asignación de la correspondencia

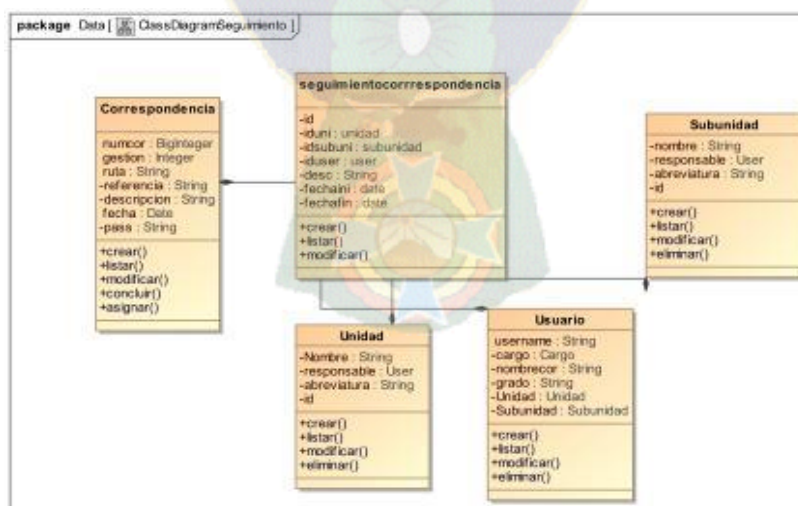


Figura 3.28. Paquete de Seguimiento de correspondencia

Fuente: [Elaboración propia]

En la figura 3.28. Se puede observar el paquete para el seguimiento de la correspondencia.

3.3.3. DISEÑO DE LA ARQUITECTURA DEL SISTEMA

3.3.3.1. DIAGRAMA DE COMPONENTES

El diagrama de componentes representara la arquitectura del software y es dividido en componentes y muestra sus dependencias entre los componentes,

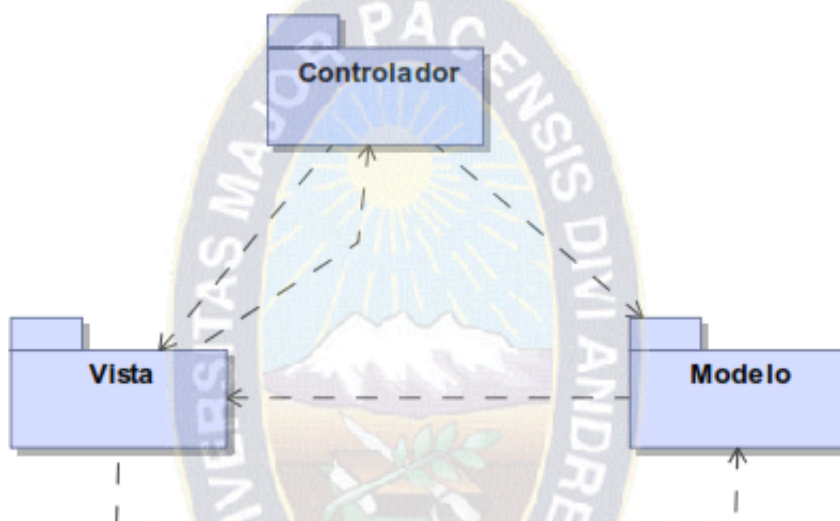


Figura 3.29. Diagrama de componentes de la estructura MVC

Fuente: [Elaboración propia]

En la figura 3.29 se obtiene la arquitectura MVC y en la figura 3.30 se mostraron las características principales del diagrama de componentes para el sistema web de correspondencia

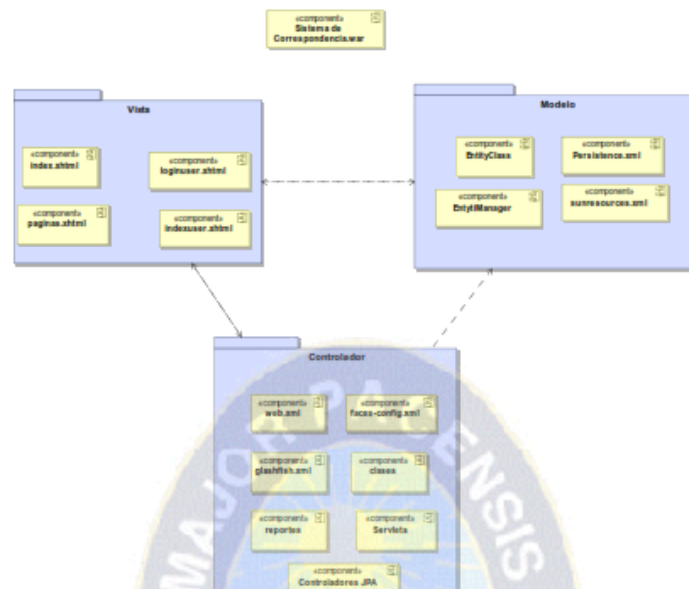


Figura 3.30. Diagrama de componentes del sistema de correspondencia

Fuente: [Elaboración propia]

3.3.3.2. DIAGRAMA DE DESPLIEGUE

En el diagrama de despliegue se diseña la parte lógica y física del sistema de correspondencia, en donde interactúan los componentes y hardware del sistema. Como se muestra en la figura 3.31.

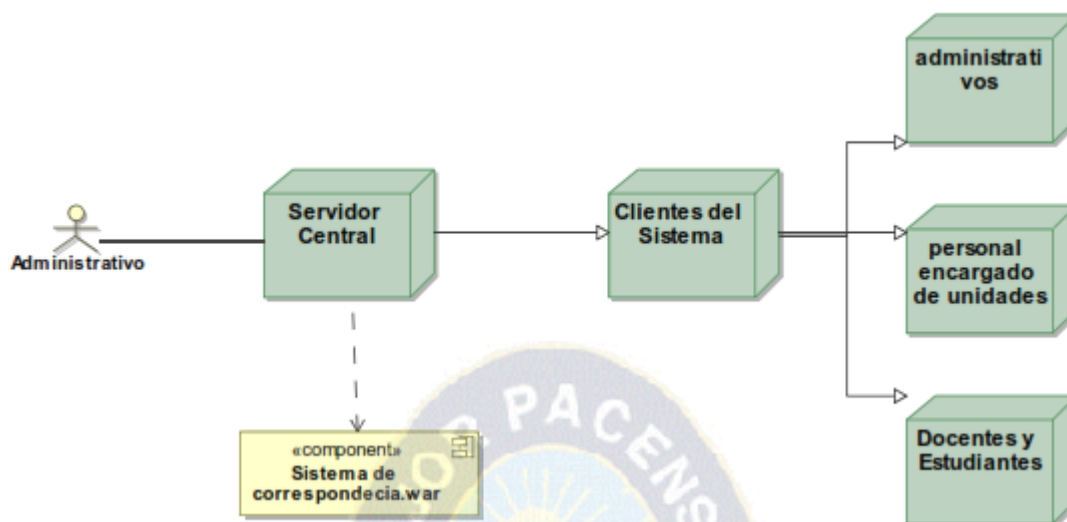


Figura 3.31. Diagrama de Despliegue

Fuente: [Elaboración propia]

3.4. FASE DE TRANSICIÓN

3.4.1. FASE DE PRUEBAS

Al ser la última iteración trabajaremos con dos tipos de pruebas las cuales son:

- Pruebas de requerimientos
- Pruebas de rendimiento o de estrés

3.4.1.1. PRUEBA DE REQUERIMIENTOS

La siguiente tabla describe la forma en que se satisficieron los resultados.

Requerimientos	Funciono correctamente SI/NO	Comentarios
El ingreso al sistema solo se puede si está registrado.	•	Si no se encuentra registrado el sistema lo devuelve al login.
Los Administradores tienen el control total del sistema.	•	

Los usuarios pueden hacer cambios en lo que necesiten	•	Los datos sensibles solo pueden ser modificados por un administrador
Los remitentes solo tienen la opción de ver el estado de la correspondencia	•	
Toda la información guardada se pudo verificar en la base de datos.	•	
Se pudo evidenciar todos los reportes generados por el sistema y se pudo observar la exportación en 4 formatos xls, cvs, xml y pdf.	•	Se realizó pruebas en el sistema con los distintos reportes que genera el sistema.

Tabla 3.5. Tabla de requerimientos

Fuente: [Elaboración propia]

Se observó en la tabla 3.4 que todos los requerimientos fueron satisfechos

3.4.1.2. PRUEBA DE ESTRÉS

Las pruebas de estrés que pueden aplicarse a un sistema web son las siguientes:

- Evaluar el desempeño del sistema al someterlo a cantidades superiores a las normales de interrupciones por segundo
- Elevar el volumen de datos de entrada buscando evaluar el comportamiento de las funciones de entrada
- Diseñar escenarios que necesiten niveles máximos de memoria.

Durante las pruebas de resistencia se incluyeron algunas pruebas de rendimiento para evaluar el desempeño del sistema en diferentes tipos de computadoras y con conexiones de internet variadas. Las pruebas de resistencia y rendimiento consistieron en evaluar los siguientes puntos:

- Comportamiento del sistema cuando múltiples instancias están siendo ejecutadas al mismo tiempo;
- Comportamiento del sistema cuando los recursos de memoria disponibles son mínimos.

- Comportamiento del sistema cuando ocurre una interrupción a la conexión a internet.
- Tiempo de descargas de los diferentes reportes generados.

Las pruebas fueron realizadas en computadoras con diferentes características, descritas a continuación:

- Windows XP, 2 GB RAM, Mozilla Firefox 24.0, conexión a Internet a 128 Kbps.
- Windows 7, 4 GB RAM, Google Chrome, conexión a internet de 512 Kbps.
- Linux Open Suse 13.1. 4 GB RAM, Chromium 29, conexión a internet de 1 Mb

Acciones	PC 1	PC 2	PC 3	Descripción
Comportamiento del sistema cuando 8 equipos están en el sistema	*	*	*	No se encontró ningún problema al sistema
Comportamiento del sistema cuando los recursos de memoria disponibles son mínimos.	*	*	*	El sistema tuvo lentitud al trabajar con Windows XP, debido a un desbordamiento de memoria de la maquina
Comportamiento del sistema cuando ocurre una interrupción a la conexión a internet o existente lentitud	*	*	*	Cuando el Sistema tuvo una conexión de 96 Kbps tuvo lentitud al ingresar y cuando se desconectó el internet ya no se tuvo acceso

Tabla 3.6. Tabla de resistencia

Fuente: [Elaboración propia]

En la tabla 3.5 se pudo observar lentitud en el S. O. Windows XP por fallas de diferentes índoles (bug y/o virus) el cual hacia lento al navegador.

Acciones	PC 1	PC 2	PC 3	Descripción
Tiempo transcurrido al insertar una correspondencia	5	4	3	
Tiempo transcurrido de ver la correspondencia de entrada	10	6	4	
Tiempo transcurrido al hacer un reporte de en pdf	10	6	4	
Tiempo de asignar una correspondencia	6	3	2	

Tabla 3.7. Tabla de rendimiento

Fuente: [Elaboración propia]

En la tabla 3.6 Se pudo observar lentitud en la máquina que estaba con Windows XP, a diferencia de la maquina con Windows 8 que mostro en algunas ocasiones mejor rendimiento que la maquina con GNU/Linux.

3.4.2. FASE DE IMPLEMENTACIÓN

En esta fase se mostraran capturas de pantalla del sistema en ejecución.

Figura 3.32. Elaboración de documento respuesta

Fuente: [Elaboración propia]

En la figura 3.32. Se muestra es registro del documento de correspondencia en una secuencia de pasos que debe seguir para un buen uso.

Documento Remitente Datos del Remitente Correspondencia Adjunto(s) Imagen(es) Confirmacion

Paso 1

Tipo de Documento: 37 -.- Modificacion Presupuestaria

Figura 3.33. Registro de Correspondencia.

Fuente: [Elaboración propia]

En la figura 3.33. Se muéstralos pasos que debe seguir con el registro de la correspondencia en un flujo BPM.

Parametros de Institucion		
<div> <div>30</div> <div> <div><=<</div> <div><<</div> <div>(1 of 1)</div> <div>>></div> <div>>=></div> </div> </div>		
Id ↕	Sigla ↕	Descripcion ↕
<input type="text"/>	<input type="text"/>	<input type="text"/>
1	UMSA	UNIVERSIDAD MAYOR DE SAN ANDRES
2	FCPN	FACULTAD DE CIENCIAS PURAS Y NATURALES
4	DIRECCION	Av. Villazon Nro. 1995, Monoblock Central, segundo patio, Edificio Carrera de Informatica
5	ZONA	CENTRAL
7	NOMISIS	Sistema de Seguimiento y Control de Correspondencia
3	C.INF.	CARRERA DE INFORMATICA

Figura 3.34. Parámetros de Institución

Fuente: [Elaboración propia]

En la figura 3.34. Se muestra los datos de la institución donde se instaló el sistema.

CAPÍTULO IV SEGURIDAD Y CALIDAD

4. INTRODUCCIÓN

En este capítulo se detallara la seguridad, como se tomaran en cuenta varios aspectos importantes, los riesgos, seguridad de acceso, seguridad en la base de datos y en la calidad de software se harán las pruebas de la tesis doctoral de OLSINA

4.1. SEGURIDAD

4.1.1. INTRODUCCIÓN

En la seguridad Informática se tienen los problemas principales que se solucionaran en gran medida manteniendo los principios de la seguridad informática.

- No dar nunca nada por hecho ni en cuestiones de seguridad ni en cuestiones del flujo normal de la aplicación. Todo el riesgo que se corra debe ser por parte del usuario no se puede tomar previsiones al respecto. (Juan Díez-Yanguas Barber, 2011), se supone que el usuario puede incurrir en fallas al momento insertar datos.
- Siempre que usemos servicios externos estamos asumiendo riesgos añadidos. Se puede crear un sistema muy seguro pero si este viene con contenido externo nos puede traer sus inseguridades, esto también aplica librerías.
- La oscuridad no es seguridad. No poner un botón acceso a la administración no impide que se pueda acceder a ella. Ocultar código no garantiza nuestra seguridad.
- Principio del mínimo privilegio: El usuario del sistema debe tener únicamente los privilegios que necesita para llevar a cabo su actividad.
- Fallar de manera segura: Hay que ser cuidadoso con fallos en la aplicación. el usuario no debe saber del fallo, si es que el sistema falla es la responsabilidad del administrador, al usuario solo hay que mostrarles errores clásicos de sistema en mantenimiento. si por cualquier motivo sucede el fallo bloquear la sesión, la información es lo más valioso y los errores son futuros *backdoor*.

4.1.2. RIESGOS

Los riesgos que se encuentran en un sistema web son varios y cada día van aumentados, en el presente documento se redactaran los problemas comunes que se presentan los sistemas web, cualquier información que puede ser valiosa por el atacante.

- **Inyección SQL.-** Consiste en intentar engañar al sistema para que realice peticiones contra la base de datos que no son las que han sido programadas y que podrían comprometer gravemente la base de datos o incluso mostrar al atacante toda la información del usuario como también la del sistema.
- **Cross Site Scripting.-** El atacante intentará enviar información a nuestro servidor por medio de nuestros formularios u otros medios con la intención de que dicha información sea almacenada en nuestra base de datos y posteriormente sea mostrada a los demás usuarios del sistema. con código JavaScript que puede ser utilizado de varias maneras como borrar contenido de formulario, ir a otras páginas, capturar cookies, variables de sesión, etc.
- **Robo de sesión.-** Como sabemos HTTP es un protocolo sin estados, lo que significa que las credenciales o información de sesión deberá ir en cada petición, debido a esto dichos datos resultan muy expuestos. Un robo de estos datos podría tener como resultado que alguien se estuviera haciendo pasar por nosotros y realizando acciones con unos privilegios que nos pertenecen.
- **Acceso a URLs restringidas.-** Consiste en la observación de una *URL* e intentar cambiarla para intentar acceder a otras zonas. Estas es una de las razones por las que la seguridad a través de la ocultación no es efectiva.

4.1.3. SEGURIDAD DE ACCESO

4.1.3.1. LA AUTENTIFICACIÓN

La pregunta del usuario es, el sistema web al que ingresa, es el verdadero o será otra página con una variación del nombre, lo importante de un sistema web es la certificación del mismo, por lo que se detallaran soluciones a los riesgos anteriormente mencionados.

- Usar SSL sobre HTTP (HTTPS) para transferir los datos y asegurarse de que el cifrado cubre los credenciales y el ID de sesión en todas las comunicaciones. El servidor Glash Fish 4 provee incorporación del certificado SSL por lo que sería política de la institución la adquisición de certificados de seguridad.
- Posibilitar el bloqueo de autenticación después de un número de intentos fallidos. Esto podría evitar ataques de diccionario o *bruteforce*.

- Implementar métodos seguros de recuperación de contraseñas: Es común que se intenten usar estos métodos para intentar ganar acceso a una cuenta del usuario, podemos ver una serie de consejos para implementar estos métodos. Pedir al usuario al menos tres datos o más, obligar a que responda preguntas de seguridad. La contraseña que recuperada deberá generarse de manera aleatoria y enviada al usuario por un canal diferente. de esta forma si el atacante consiguió sortear los primeros pasos es difícil que logre sortear el canal usado para transmitir.

4.1.3.2. SEGURIDAD POR SESIÓN

La seguridad en la sesión se basara en 2 aspectos importantes que se detallaran a continuación:

- Tiempo de vida de una sesión.- La sesión tendrá un tiempo de vida de 30 minutos si no hay actividad presente.

4.1.3.3. INICIO, IP Y LOCALIZACIÓN

Para acceder al sistema será necesario cumplir 3 requisitos como se detallaran a continuación.

- Acceso por localización.- se podrá obtener datos necesarios como la *IP*, *MAC*, *ISP*, por consiguiente se alerta cuando se ingrese al sistema fuera de la zona permitida.
- Numero de Sesiones activas.- más que una medida de seguridad se utilizara por recomendación, el sistema automáticamente destruye sesión si no son utilizadas por un tiempo determinado, pero en el caso de que estén sesiones activas se dará un mensaje de alerta sobre el número límite y si se pasó.
- Acceso por formulario.- es el más sencillo solo es el nombre del usuario y su contraseña el cual estará encriptado.

4.1.4. SEGURIDAD A NIVEL DE BASE DE DATOS

Lo más importante para una institución es la información y por eso se recomendara y se ejecutara en los servidores de la carrera de informática. Se ejecutaran pasos principales para detección de vulnerabilidades y su pronta corrección que se detallaran a continuación:

- Identificación de sensibilidad.- Creación de manuales de tablas más sensibles a modificaciones, y diseñando medidas contra ataques *SQL Injection*.

- Evaluación de la vulnerabilidad y la configuración.- Se realizara configuraciones de las S. O., para asegurarse que no tiene *backdoor*, la verificación incluye sistemas operativos, discos duros, *etc.*
- Endurecimiento.- El endurecimiento es un paso muy importante en el sentido que se realizaran correcciones a las vulnerabilidades detectadas. Y eliminación de componentes innecesarios.
- Auditar.- después de realizar el endurecimiento se realizara controles de auditoría que se detecte cada cambio que quiera realizar el usuario.
- Monitoreo.- es importante monitorear en tiempo real la actividad de la base de datos para limitar su exposición, esto evitara intrusiones, uso indebido de la base de datos y evitar un *zombi server*, aplicando las regulaciones de *SOX*.
- Autenticación, control de acceso, y Gestión de derechos.- se debe clasificar a los usuarios para determinar y restringir el acceso a los datos más sensibles.

4.2. CALIDAD DE SOFTWARE

4.2.1. INTRODUCCIÓN

El propósito fundamental de la ingeniería de software es construir un software de calidad, por consiguiente este capítulo se detalla la calidad del sistema de correspondencia.

La medición de control de calidad de software se realizara a través de métricas de control de calidad basados en la tesis Doctoral "Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web" de Luis Antonio Olsina (OLSINA, 1999), La aplicación de la tesis web trabajara en 4 aspectos como ser usabilidad, funcionalidad, confiabilidad y eficiencia

4.2.2. DEFINICIÓN Y ESPECIFICACIÓN DE REQUERIMIENTOS DE CALIDAD

Esta fase trata con actividades y procedimientos, modelado y especificación de los requerimientos de calidad. A partir de un proceso de evaluación realizado en una mezcla de estrategias prescriptivas y descriptivas (el enfoque de modelo mixto de calidad), y con el fin de analizar, comparar, comprender y potencialmente mejorar características y atributos

de artefactos Web, los requerimientos deben responder a necesidades y deseos de un perfil (de usuario y dominio establecidos. Podemos definir al dominio de la aplicación, desde el punto de vista de la evaluación, como a un sistema real o abstracto del universo que existe independientemente del sistema de evaluación. Consiste de un conjunto de entes a los que se le atribuyen propiedades atributos, características), manifiestan un comportamiento y se relacionan. (OLSINA, 1999). A continuación se mostrara las preguntas planteadas en la evaluación de la calidad de software con una evaluación con el siguiente rango de calificación:

Rango	Descripción
0	No existe la característica
1-50	Malo
51-60	Regular
61-70	Optimo
70-100	Excelente

La encuesta se realizó en las primeras fases de la implementación del sistema para la usabilidad del sistema 4.1.

1. Usabilidad	Descripción	Nota %	Personas
1.1 Comprensibilidad Global del Sitio	1.1 Comprensibilidad Global del Sitio		
	1.1.1 Esquema de Organización Global		
	1.1.1.1 Mapa del Sitio	0	5
	1.1.1.2 Tabla de Contenidos	50	5
	1.1.1.3 Índice Alfabético	0	5
	1.1.2 Calidad en el Sistema de Etiquetado	50	5
	1.1.3 Visita Guiada Orientada	60	5
	1.1.4 Mapa de Imagen	30	5
1.2 Mecanismos de Ayuda y Retroalimentación en línea	1.2.1 Calidad de la Ayuda		
	1.2.1.1 Ayuda Explicatoria Orientada	90	5
	1.2.1.2 Ayuda de la Búsqueda	95	5
	1.2.2 Indicador de Ultima Actualización		
	1.2.2.1 Global (de todo el sitio Web)	40	5
	1.2.2.2 Restringido (por subsitio o página)	15	5
	1.2.3 Directorio de Direcciones		
	1.2.3.1 Directorio E-mail	40	5
	1.2.3.2 Directorio TE-Fax	80	5
	1.2.3.3 Directorio Correo Postal	0	5
	1.2.4 Facilidad FAQ	80	5

1.3 Aspectos de Interfaces y Estéticos	1.3.1 Cohesividad al Agrupar los Objetos de Control Principales	100	5
	1.3.2 Permanencia y Estabilidad en la Presentación de los Controles Principales		
	1.3.2.1 Permanencia de Controles Directos	80	5
	1.3.2.2 Permanencia de Controles Indirectos	60	5
	1.3.2.3 Estabilidad	60	5
	1.3.3 Aspectos de Estilo	80	5
	1.3.4 Preferencia Estética	75	5
1.4 Misceláneas	1.4.1 Soporte a Lenguaje Extranjero	0	5
	1.4.2 Atributo “Qué es lo Nuevo”	0	5
	1.4.3 Indicador de Resolución de Pantalla	30	5

Tabla 4.1. Sub- Características de Usabilidad

Fuente: [Elaboración propia]

2. Funcionalidad	Descripción	Nota %	Personas
2.1 Aspectos de Búsqueda y Recuperación	2.1.1 Mecanismo de Búsqueda en el Sitio Web		
	2.1.1.1 Búsqueda Restringida	100	5
	2.1.1.2 Búsqueda Global	50	5
2.2 Aspectos de Navegación y Exploración	2.2.1 Navegabilidad	100	5
	2.2.2 Objetos de Control Navegacional	30	5
	2.2.3 Predicción Navegacional	50	5
2.3 Aspectos del Dominio	2.3.1 Relevancia de Contenido	100	5
	2.3.2 Relevancia de Enlaces	75	5
	2.3.3 Servicio de Grupo de Noticias	0	5
	2.3.4 Aspectos Varios	60	5

Tabla 4.2. Sub- Características de Funcionalidad

Fuente: [Elaboración propia]

En la tabla 4.2. Mostraran las sub características tomadas para la parte de control de calidad de la funcionalidad

3. Confiabilidad		Nota %	Personas
3.1 No Deficiencia	3.1.1 Errores de Enlaces		
	3.1.1.1 Enlaces Rotos	70	5
	3.1.1.2 Enlaces Inválidos	80	5
	3.1.1.3 Enlaces no Implementados	90	5
	3.1.2 Errores o Deficiencias Varias		
	3.1.2.1 Deficiencias o cualidades ausentes	60	5
	3.1.2.2 Deficiencias o resultados	60	5
	3.1.2.3 Nodos Destinos (inesperadamente)	90	5
	3.1.2.4 Nodos Web Muertos (sin enlaces de	90	5

retorno)

Tabla 4.3. Sub- Características de Confiabilidad

Fuente: [Elaboración propia]

En la tabla 4.3. Mostraran las características principales para el control de calidad de la Confiabilidad

4. Eficiencia		Nota %	Personas
4.1 Performancia	4.1.1 Páginas de Acceso Rápido	80	5
4.2 Accesibilidad	4.2.1 Accesibilidad de Información		
	4.2.1.1 Soporte a Versión sólo Texto	60	5
	4.2.1.2 Legibilidad al desactivar la Propiedad Imagen del Browser		
	4.2.1.2.1 Imagen con Título	100	5
	4.2.1.2.2 Legibilidad Global	60	5
	4.2.2 Accesibilidad de Ventanas	70	5

Tabla 4.4. Sub- Características de Eficiencia

Fuente: [Elaboración propia]

En la tabla 4.4. Mostraran las características principales para el control de calidad de la Eficiencia.

4.2.3. DEFINICIÓN E IMPLEMENTACIÓN DE LA EVALUACIÓN ELEMENTAL

Con respecto a la fase de *Definición e Implementación de la Evaluación Elemental* la misma trata con actividades, modelos, técnicas, heurísticas y herramientas para determinar criterios de evaluación para cada atributo cuantificable y realizar el proceso de medición. Particularmente, nos interesa la calidad de artefactos Web como característica de estudio. A continuación se mencionaran los distintos criterios que se consideran para la recolección de datos:

Título: Mapa del Sitio	Código: 1.1.1.1	Tipo: Atributo
Característica de más Alto Nivel	Usabilidad	

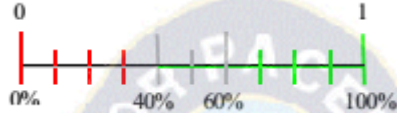
Super-característica	Esquema de Organización Global
Definición / Comentarios	Un mapa del sitio es una representación con componentes gráficos, que muestra la estructura o arquitectura global (a menudo jerárquica) del sitio Web como un todo
Modelo para determinar el Cómputo Global	modelo LSP (en nuestro caso), o modelo meramente Aditivo
Tipo de Criterio Elemental	Es un criterio binario, discreto y absoluto: sólo se pregunta si está disponible (1) o si no está disponible (0).
Escala de Preferencia	
Tipo de Recolección de Datos	Manual, Observacional

Tabla 4.5. Criterios de recolección de datos de más alto nivel para Usabilidad

Fuente: (OLSINA, 1999)

En la tabla 4.5. Se muestra los criterios de recolección de datos de alto nivel para la usabilidad como ejemplo.

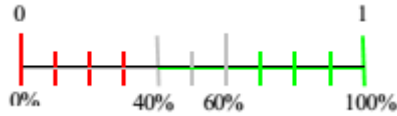
Título: Búsqueda Restringida	Código: 2.1.1.1.1	Tipo: Atributo
Característica de más Alto Nivel	Funcionalidad	
Super-característica	Búsqueda Restringida	
Definición / Comentarios	Algunas veces, tiene sentido brindar al usuario con la facilidad de búsqueda restringida a un subsitio o parte de un sitio, debido a que el mismo es altamente cohesivo o distintivo del resto de la información del sitio Web global	
Tipo de Criterio Elemental	Es un criterio multi-nivel, discreto y absoluto, definido como subconjunto. Podemos decir que: 0 = no disponible mecanismos de búsqueda restringida; 1 = búsqueda básica: mecanismo de búsqueda por nombre/apellido; 2 = 1 + búsqueda extendida o avanzada	
Escala de Preferencia		
Tipo de Recolección de Datos	Manual	

Tabla 4.6. Criterios de recolección de datos de más alto nivel para Funcionalidad

Fuente: (OLSINA, 1999)

En la tabla 4.6. Se muestra los criterios de recolección de datos de alto nivel para la Funcionalidad como ejemplo

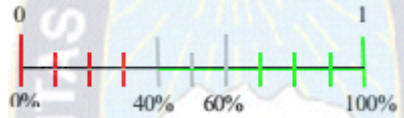
Título: Enlaces Rotos	Código: : 3.1.1	Tipo: Atributo
Característica de más Alto Nivel	Confiabilidad	
Super-característica	Errores de Enlaces	
Definición / Comentarios	Este atributo representa básicamente a los enlaces encontrados que conducen a nodos destino ausentes (también llamados enlaces ausentes o pendientes)	
Tipo de Criterio Elemental	es un criterio de variable normalizada, continuo y absoluto; En donde si BL=Número de enlaces rotos encontrados. TL = Número total de enlaces del sitio. La fórmula para computar la variable es: $X = 100 - (BL * 100/TL) * 10$, donde, si $X < 0$ entonces $X = 0$.	
Escala de Preferencia	 <p>The scale shows a horizontal line from 0 to 100%. There are red tick marks at 0%, 20%, 40%, and 60%. There are green tick marks at 80%, 100%, and a point between 60% and 80%.</p>	
Tipo de Recolección de Datos	Automatizado	

Tabla 4.7. Criterios de recolección de datos de más alto nivel para Confiabilidad

Fuente: (OLSINA, 1999)

En la tabla 4.7. Se muestra los criterios de recolección de datos de alto nivel para la confiabilidad como ejemplo

Título: Páginas de Acceso Rápido	Código: 4.1.1	Tipo: Atributo
Característica de más Alto Nivel	Eficiencia	
Super-característica	Performancia	
Definición / Comentarios	<p>Para este atributo, se mide el tamaño de todas las páginas (Estáticas) del sitio Web considerando todos sus componentes gráficos, tabulares y textuales. El tamaño de cada página se especifica como una función del tiempo de espera y de la velocidad mínima establecida para una línea de comunicación dada.</p> <p>Se especifica un tamaño umbral aceptable, para el tamaño total de cada página, por ejemplo, el de 35,2 Kb. Una página de este tamaño requiere 20 segundos para ser bajada a una tasa de 14,400 bps. Ese es el tiempo aceptable que un usuario debe</p>	

	esperar, sin que se ponga impaciente.
Tipo de Criterio Elemental	es un criterio multi-variable, continuo y absoluto
Escala de Preferencia	
Tipo de Recolección de Datos	Automatizado

Tabla 4.8. Criterios de recolección de datos de más alto nivel para Eficiencia

Fuente: (OLSINA, 1999)

En la tabla 4.8. Se muestra los criterios de recolección de datos de alto nivel para la eficiencia como ejemplo

4.2.4. DEFINICIÓN E IMPLEMENTACIÓN DE LA DEFINICIÓN GLOBAL

En la fase de *Definición e Implementación de la Evaluación Global* se trata con actividades, modelos, procedimientos y herramientas para determinar los criterios de agregación de las preferencias de calidad elemental (obtenidas en la fase anterior, a partir del árbol de requerimientos), para producir la preferencia global para cada sistema de información interviniente.

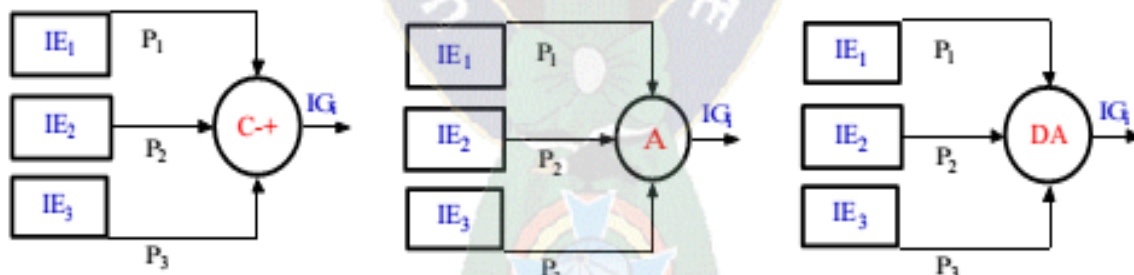


Figura 4.1. Tres funciones simples de agregación de preferencias

Fuente: (OLSINA, 1999)

En la siguiente Figura 4.2 se muestra los indicadores que se deben considerar para el cálculo de calidad de software.

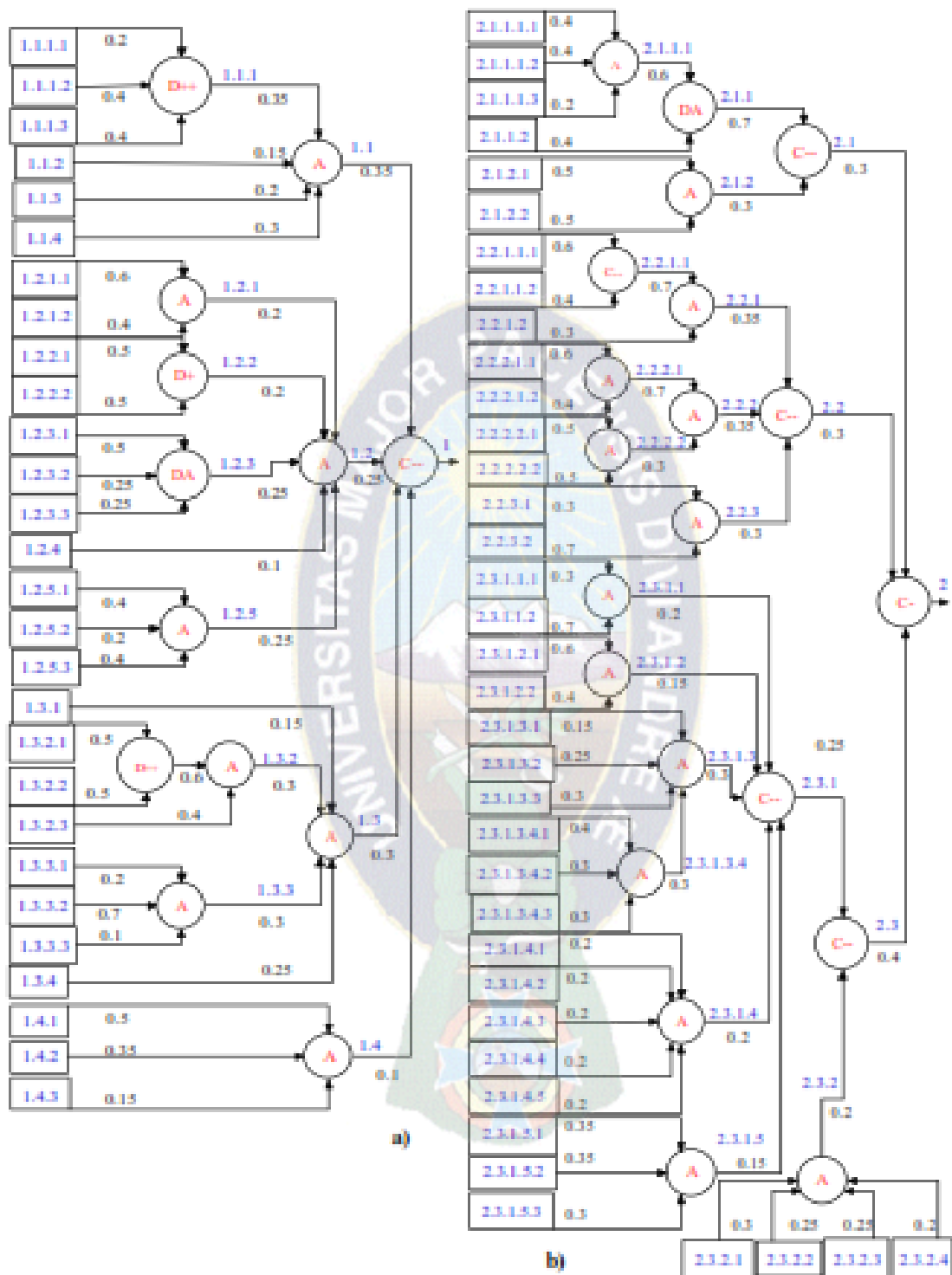


Figura 4.2. Ejemplo de Estructura de agregación de preferencias parciales para la usabilidad y funcionalidad

Fuente: (OLSINA, 1999)

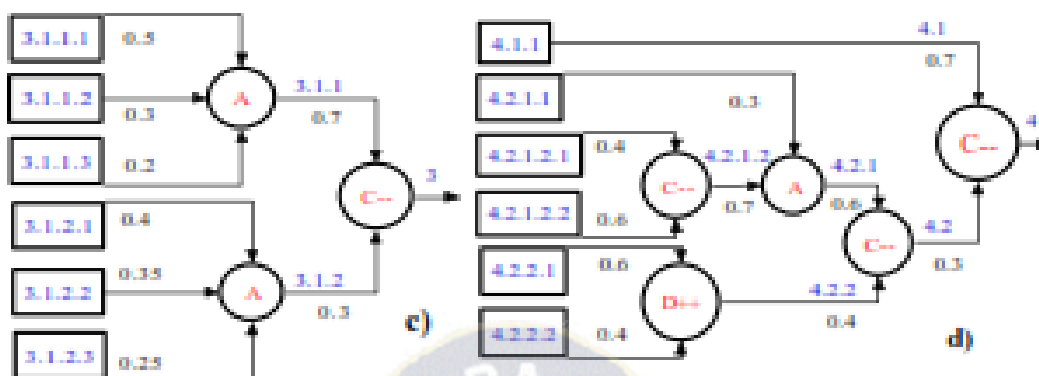


Figura 4.3. Ejemplo de Estructura de agregación de preferencias parciales para la Confiabilidad y Eficiencia

Fuente: (OLSINA, 1999)

En la figura 4.3 se muestra los indicadores necesarios para calcular la eficiencia y confiabilidad. Para obtener el indicador de calidad global e indicadores parciales (IG/P), se puede emplear la siguiente estructura de agregación, tomando en cuenta el modelo aditivo

$$\frac{IG}{P} = (P_1 * IE_1 + \dots + P_n * IE_n)(0)$$

Donde:

IE_i Son los indicadores elementales o parciales, conforme al nivel del árbol que se calcule.

P_i Son los Pesos que modelan la importancia relativa de cada indicador elemental o parcial dentro de un grupo

Consideraciones:

Cada P_i debe ser mayor a cero y la sumatoria en un grupo o subgrupo debe dar uno, Realizando los cálculos necesarios con la ecuación (0) se obtendrán los siguientes resultados.

Preferencias de calidad para atributos de usabilidad			
1.1.1 Esquema de Organización Global			
	IEi	Pi	(IEi*Pi)
1.1.1.1 Mapa del Sitio	0	0,3	0

1.1.1.2 Tabla de Contenidos	50	0,4	20
1.1.1.3 Índice Alfabético	0	0,3	0
		Total	20
1.2.1 Calidad de la Ayuda			
	IEi	Pi	(IEi*Pi)
1.2.1.1 Ayuda Explicatoria Orientada	90	0,6	54
1.2.1.2 Ayuda de la Búsqueda	95	0,4	38
		Total	92
1.2.2 Indicador de Última Actualización			
	IEi	Pi	(IEi*Pi)
1.2.2.1 Global (de todo el sitio Web)	40	0,6	24
1.2.2.2 Restringido (por subsitio o página)	15	0,4	6
		Total	30
1.2.3 Directorio de Direcciones			
	IEi	Pi	(IEi*Pi)
1.2.3.1 Directorio E-mail	40	0,35	14
1.2.3.2 Directorio TEL -FAX	80	0,3	24
1.2.3.3 Directorio Correo Postal	0	0,35	0
		Total	38
1.3.2 Permanencia y Estabilidad en la Presentación de los Controles Principales			
	IEi	Pi	(IEi*Pi)
1.3.2.1 Permanencia de Controles Directos	80	0,4	32
1.3.2.2 Permanencia de Controles Indirectos	60	0,4	24
1.3.2.3 Estabilidad	60	0,2	12
		Total	68

Tabla 4.9. Resultado de las preferencias de calidad para atributos de usabilidad

Fuente: [Elaboración propia]

En la tabla 4.9. Se muestra los resultados obtenidos para el cálculo de atributos usabilidad, para el control de calidad.

Preferencias de calidad para sub- características de usabilidad			
1.1 Comprensibilidad Global del Sitio			
	IEi	Pi	(IEi*Pi)
1.1.1 Esquema de Organización Global	20	0,35	7
1.1.2 Calidad en el Sistema de Etiquetado	50	0,2	10

1.1.3 Visita Guiada Orientada	60	0,25	15
1.1.4 Mapa de Imagen	30	0,2	6
	Total		38
1.2 Mecanismos de Ayuda y Retroalimentación en línea			
	IEi	Pi	(IEi*Pi)
1.2.1 Calidad de la Ayuda	92	0,25	23
1.2.2 Indicador de Última Actualización	30	0,25	7,5
1.2.3 Directorio de Direcciones	38	0,25	9,5
1.2.4 Facilidad FAQ	80	0,25	20
	Total		60
1.3 Aspectos de Interfaces y Estéticos			
	IEi	Pi	(IEi*Pi)
1.3.1 Cohesividad al Agrupar los Objetos de Control Principales	100	0,2	20
1.3.2 Permanencia y Estabilidad en la Presentación de los Controles Principales	68	0,3	20,4
1.3.3 Aspectos de Estilo	80	0,25	20
1.3.4 Preferencia Estética	75	0,25	18,75
	Total		79,15
1.4 Misceláneas			
	IEi	Pi	(IEi*Pi)
1.4.1 Soporte a Lenguaje Extranjero	0	0,8	0
1.4.2 Atributo “Qué es lo Nuevo”	0	0,1	0
1.4.3 Indicador de Resolución de Pantalla	30	0,1	3
	Total		3

Tabla 4.10. Resultado de las preferencias de calidad para sub- características de usabilidad

Fuente: [Elaboración propia]

En la tabla 4.10. Se muestra los resultados obtenidos para el cálculo de los sub atributos de usabilidad, para el control de calidad.

1. Usabilidad			
	IEi	Pi	(IEi*Pi)
1.1 Comprensibilidad Global del Sitio	38	0,35	13,3
1.2 Mecanismos de Ayuda y Retroalimentación en línea	60	0,3	18
1.3 Aspectos de Interfaces y Estéticos	79,15	0,25	19,7875
1.4 Misceláneas	3	0,1	0,3

Total	51,3875
-------	---------

Tabla 4.11. Resultado de las preferencias de calidad para características de usabilidad

Fuente: [Elaboración propia]

En la tabla 4.11. Se muestra los resultados obtenidos de la usabilidad del sistema de correspondencia dando un total de 51.38 lo que significa que el sistema es usable, las características que afectaron la usabilidad del sistema y tomar en cuenta para futuras mejoras estas descritas en las tablas anteriores.

Preferencias de calidad para atributos de Funcionalidad			
2.1.1 Mecanismo de Búsqueda en el Sitio Web			
	IEi	Pi	(IEi*Pi)
2.1.1.1 Búsqueda Restringida	100	0,8	80
2.1.1.2 Búsqueda Global	50	0,2	10
		Total	90

Tabla 4.12. Resultado de las preferencias de calidad para atributos de funcionalidad

Fuente: [Elaboración propia]

En la tabla 4.12. Se muestra los resultados obtenidos para el cálculo de los atributos de funcionalidad, para el control de calidad.

Preferencias de calidad para sub- características de usabilidad			
2.2 Aspectos de Navegación y Exploración			
	IEi	Pi	(IEi*Pi)
2.2.1 Navegabilidad	100	0,35	35
2.2.2 Objetos de Control Navegacional	30	0,3	9
2.2.3 Predicción Navegacional	50	0,35	17,5
		Total	61,5
2.3 Aspectos del Dominio			
	IEi	Pi	(IEi*Pi)
2.3.1 Relevancia de Contenido	100	0,35	35
2.3.2 Relevancia de Enlaces	75	0,25	18,75
2.3.3 Servicio de Grupo de Noticias	0	0,25	0
2.3.4 Aspectos Varios	60	0,15	9
		Total	62,75

Tabla 4.13. Resultado de las preferencias de calidad para sub- características de funcionalidad

Fuente: [Elaboración propia]

En la tabla 4.13. Se muestra los resultados obtenidos para el cálculo de los sub atributos de funcionalidad, para el control de calidad.

2. Funcionalidad			
	IEi	Pi	(IEi*Pi)
2.1 Aspectos de Búsqueda y Recuperación	90	0,25	22,5
2.2 Aspectos de Navegación y Exploración	61,5	0,35	21,525
2.3 Aspectos del Dominio	62,75	0,4	25,1
		Total	69,125

Tabla 4.14. Resultado de las preferencias de calidad para características de funcionalidad

Fuente: [Elaboración propia]

En la tabla 4.14. Se muestra los resultados obtenidos de la funcionalidad del sistema de correspondencia dando un total de 69.125 lo que significa que el sistema es funcional, las características que afectaron la funcionalidad del sistema y tomar en cuenta para futuras mejoras estas descritas en las tablas anteriores.

Preferencias de calidad para atributos de confiabilidad			
3.1.1 Errores de Enlaces			
	IEi	Pi	(IEi*Pi)
3.1.1.1 Enlaces Rotos	70	0,4	28
3.1.1.2 Enlaces Inválidos	80	0,4	32
3.1.1.3 Enlaces no Implementados	90	0,2	18
		Total	78
3.1.2 Errores o Deficiencias Varias			
	IEi	Pi	(IEi*Pi)
3.1.2.1 Deficiencias o cualidades ausentes	60	0,25	15
3.1.2.2 Deficiencias o resultados	60	0,25	15
3.1.2.3 Nodos Destinos (inesperadamente)	90	0,25	22,5
3.1.2.4 Nodos Web Muertos (sin enlaces de retorno)	90	0,25	22,5

Total	75
-------	----

Tabla 4.15. Resultado de las preferencias de calidad para atributos de confiabilidad

Fuente: [Elaboración propia]

En la tabla 4.15. Se muestra los resultados obtenidos para el cálculo de los atributos de confiabilidad, para el control de calidad.

Preferencias de calidad para sub- características de confiabilidad			
3.1 No Deficiencia			
	IEi	Pi	(IEi*Pi)
3.1.1 Errores de Enlaces	78	0,6	46,8
3.1.2 Errores o Deficiencias Varias	75	0,4	30
	Total		76,8

Tabla 4.16. Resultado de las preferencias de calidad para sub - características de confiabilidad

Fuente: [Elaboración propia]

En la tabla 4.16. Se muestra los resultados obtenidos para el cálculo de los sub-características de confiabilidad, para el control de calidad.

3. Confiabilidad			
	IEi	Pi	(IEi*Pi)
3.1 No Deficiencia	76,8	1	76,8
	Total		76,8

Tabla 4.17. Resultado de las preferencias de calidad para características de confiabilidad

Fuente: [Elaboración propia]

En la tabla 4.17. Se muestra los resultados obtenidos de la confiabilidad del sistema de correspondencia dando un total de 76.8 lo que significa que el sistema es confiable, las características que afectaron la confiabilidad del sistema y tomar en cuenta para futuras mejoras estas descritas en las tablas anteriores.

Preferencias de calidad para atributos de eficiencia
4.2.1.2 Legibilidad al desactivar la Propiedad Imagen del Browser

	IEi	Pi	(IEi*Pi)
4.2.1.2.1 Imagen con Título	100	0,5	50
4.2.1.2.2 Legibilidad Global	60	0,5	30
	Total		80
4.2.1 Accesibilidad de Información			
	IEi	Pi	(IEi*Pi)
4.2.1.1 Soporte a Versión sólo Texto	60	0,8	48
4.2.1.2 Legibilidad al desactivar la Propiedad Imagen del Browser	80	0,2	16
	Total		64

Tabla 4.18. Resultado de las preferencias de calidad para atributos de eficiencia

Fuente: [Elaboración propia]

En la tabla 4.18. Se muestra los resultados obtenidos para el cálculo de los atributos de eficiencia, para el control de calidad.

Preferencias de calidad para sub - características de eficiencia			
4.1 Performancia			
	IEi	Pi	(IEi*Pi)
4.1.1 Páginas de Acceso Rápido	80	1	80
	Total		80
4.2 Accesibilidad			
	IEi	Pi	(IEi*Pi)
4.2.1 Accesibilidad de Información	64	0,8	51,2
4.2.2 Accesibilidad de Ventanas	70	0,2	14
	Total		65,2

Tabla 4.19. Resultado de las preferencias de calidad para sub - características de eficiencia

Fuente: [Elaboración propia]

En la tabla 4.19. Se muestra los resultados obtenidos para el cálculo de los sub-atributos de eficiencia, para el control de calidad.

4. Eficiencia			
	IEi	Pi	(IEi*Pi)
4.1 Performancia	80	0,8	64
4.2 Accesibilidad	65	0,2	13

Tabla 4.20. Resultado de las preferencias de calidad para características de eficiencia

Fuente: [Elaboración propia]

En la tabla 4.20. Se muestra los resultados obtenidos de la eficiencia del sistema de correspondencia dando un total de 77 lo que significa que el sistema es eficiente, las características que afectaron la eficiencia del sistema y tomar en cuenta para futuras mejoras estas descritas en las tablas anteriores.

4.2.5. RESULTADOS DE LA CALIDAD DEL SISTEMA

La tesis Doctoral WEBQEM de Olsina plantea 3 niveles de barras de calidad esto es, Insatisfactorio (de 0 a 40 %), Marginal (desde 40 al 60 %) y Satisfactorio (desde 60 a 100%) se basa en el esquema de aceptabilidad de modelo ISO como IEEE. Como se muestra en la figura 4.4.

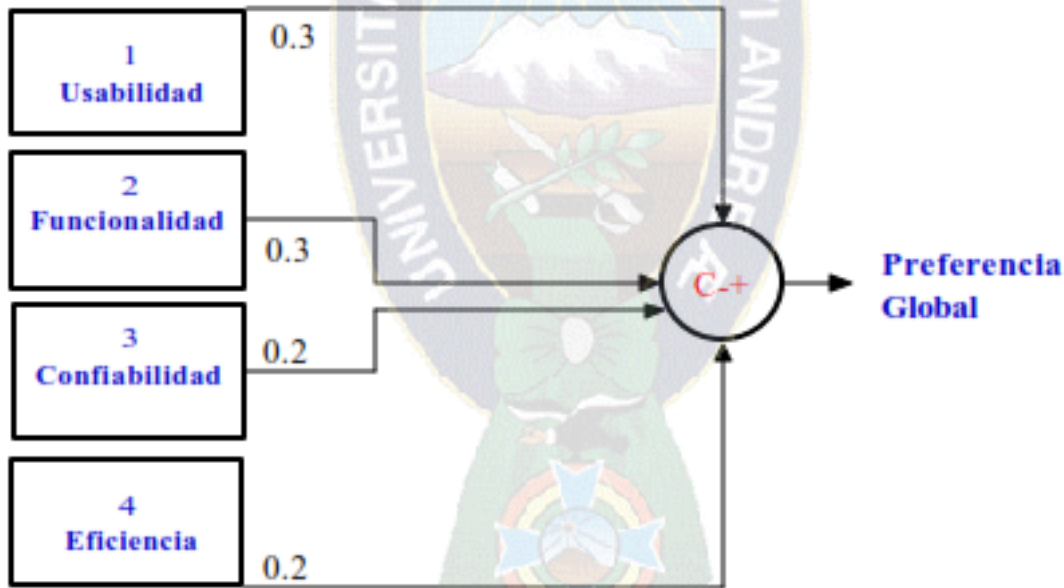


Figura 4.4. Preferencias Parciales para las características de más alto nivel

Fuente: (OLSINA, 1999)

Característica de calidad			
	IEi	Pi	(IEi*Pi)
1. Usabilidad	51,39	0,3	15,42
2. Funcionalidad	69,13	0,3	20,74

3. Confiabilidad	76,8	0,2	15,36
4. Eficiencia	77	0,2	15,40
	Total		66,92

Tabla 4.21. Resultado de las preferencias de calidad para características de eficiencia

Fuente: [Elaboración propia]

De acuerdo a los resultados de la tabla 4.21 que se recopiló durante todo este proceso de calidad de software se puede observar los resultados más mínimos como superiores, haciendo un análisis se puede observar que hay que mejorar la usabilidad del sistema por su resultado bajo y lo más sobresaliente es la eficiencia del sistema, De acuerdo a la valoración y evaluación de calidad de software aplicando el modelo WEBQEM se puede observar que el usuario tendrá la satisfacción del 66.92%, el valor obtenido se encuentra dentro de los márgenes de satisfacción definidos en WEBQEM descritos en la figura 4.4.



CAPÍTULO V ANÁLISIS DE COSTOS Y BENEFICIOS

5. INTRODUCCIÓN

Para calcular el esfuerzo y el costo de desarrollo de software se utilizara el modelo COCOMO II, que en función del programa expresado en las líneas de código estimadas podemos calcular los valores del esfuerzo y costo.

Lenguaje	Nivel	Factor LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
Visual C++	9.50	34
PHP	11.00	29
ASP	9.00	36
Visual Basic	7.00	46
Ansi Cocol	74.3	107
PL/I	4	80

Tabla 5.1 Conversión de puntos de función

Fuente: Elaboración propia

En la tabla muestra los valores del punto función respecto a las líneas de código empleadas para el sistema. Para el desarrollo del sistema se usó el lenguaje JAVA, cual su nivel es 6 y el factor.

$$\text{LDC/PF} = 53$$

Los valores monetarios se expresaran en \$. Dólares Americanos, para todas las fases de desarrollo a priori y pos arquitectura.

5.1. ANÁLISIS DEL DISEÑO TEMPRANO DE COSTO

El análisis de costo a priori detalla un costo antes de ejecutar el proyecto, en los cuales se detallaran costos estimados antes de realizar el proyecto, para esto se utilizara el programa USC Cocomo II, para hacer el cálculo de costo a priori.

Primeramente se definirán los módulos principales que tendrá el sistema, como ser registro, digitalización, seguimiento, búsquedas, reportes y estados. Para el análisis de costos priori se determinó cuantificar los módulos más importantes del sistema. Que se detallan en la tabla 5.2.

Nombre de Modulo	Tamaño en líneas de Código
Registro	2500
digitalización	2100
seguimiento	2000
búsquedas	2500
reportes	2100
estados	2100

Tabla 5.2 Módulos y tamaño

Fuente: Elaboración propia

Introduciendo los datos al USC Cocomo II se obtendrán después de las iteraciones los siguientes datos que se detallaran en la tabla 5.3

Nombre del modulo	Registro	digitalización	seguimiento	búsqueda	reporte	estado
Líneas de código fuente	2500	2100	2000	2500	2100	2100
Factor de ajuste del esfuerzo	1	1	1	1	1	1
Nominal meses-persona	9	8	7	9	8	8
Personas-mes estimados	9	8	7	9	8	8
productividad	257	257	257	257	257	257
Cambio del Trabajo	150	150	150	150	150	150
costo	1458	1225	1166	1458	1225	1225
El costo por la Instrucción	0.58	0.58	0.58	0.58	0.58	0.58
Tiempo completo Software de Personal	0.73	0.62	0.59	0.73	0.62	0.62
riesgo	0	0	0	0	0	0
rotura del módulo	0	0	0	0	0	0

Dimensionamiento Método	SLOC	SLOC	SLOC	SLOC	SLOC	SLOC
----------------------------	------	------	------	------	------	------

Tabla 5.3 Costos por modulo

Fuente: Elaboración propia

Para un cálculo total del costo del sistema se obtendrá la siguiente tabla, que es la suma de cada costo de cada módulo planteado.

Sistema de Correspondencia	Optimista	Tiempo Normal	Pesimista
SLOC total		13300	
Semanas	12	13	15
Productividad Total estimado	385.626	257.084	172.246
Costo total estimado	2199	4760	8640
Costo total estimado por la Instrucción	0.39	0.58	0.87
Total estimado Tiempo completo Software de Personal	3	4	5

Tabla 5.4 Costos total del proyecto a priori

Fuente: Elaboración propia

En la tabla 5.4 se obtienen los costos del proyecto a priori que se detallan en un tiempo óptimo, tiempo normal y el tiempo pesimista.

Lo que nos quiere decir que en un tiempo normal se desarrollara en 13 semanas, con un costo estimado de 4.760 \$us. Y con 4 personas que trabajaran tiempo completo al desarrollo del sistema.

5.2. ANÁLISIS DE COSTO POST ARQUITECTURA.

El costo total de desarrollo con el cual se terminó el sistema de correspondencia se detallara a continuación.

Primeramente se desarrolló en java lo cual su PF es igual a 53 al que se detalló en las tabla 5.1., Los puntos de función trabajan con la siguiente formula.

$$FP = UFP \times TCF$$

FP.-Punto Función

UFP.- Puntos de función no ajustados.

TCF.- Factor de complejidad técnica

Donde se consideran los siguientes factores

- Entradas Externas
- Salidas Externas
- Archivos lógicos internos, por ejemplo archivos a base de datos, etc.
- Archivos Externos de Interfaces, pantallas, etc.
- Solicitudes Externas, por ejemplo consultas a las base de datos.

Para obtener el *UFP* (total ajuste) se utilizara la siguiente formula:

$$UFP = \sum_{i=1}^n (CantidadItemsTipo_i) * Peso_i$$

Luego de usar la ecuación de *UFP* hay que convertirlo a líneas de código. Para lo cual se utilizara la siguiente ecuación.

$$SLOC = UFP * Puntos_Java$$

La ecuación de *SLOC* calculara las líneas de código para cada sub modulo que se detalla en la tabla 5.5.

Nombre del módulo	Punto-Función	Entrada	Salida	Archivos	Interfaces	consultas		Equivalente líneas de código fuente
	Lenguaje	Prom.	Prom.	Prom.	Prom.	Prom.	Total Ajuste	ESLOC
Registro_cor	Java	7	4	3	3	5	119	6307
despacho_cor	Java	2	2	5	9	3	143	7579
seguimiento	Java	2	1	0	9	1	80	4240
busquedas	Java	1	1	5	9	2	130	6890
reportes_ges	Java	1	1	3	1	20	126	6678
estados	Java	1	1	1	1	3	38	2014
registro_var	Java	1	5	3	9	1	126	6678

Tabla 5.5 Peso de Factor de complejidad por cada sub modulo

Fuente: Elaboración propia

Para calcular el factor de escala B

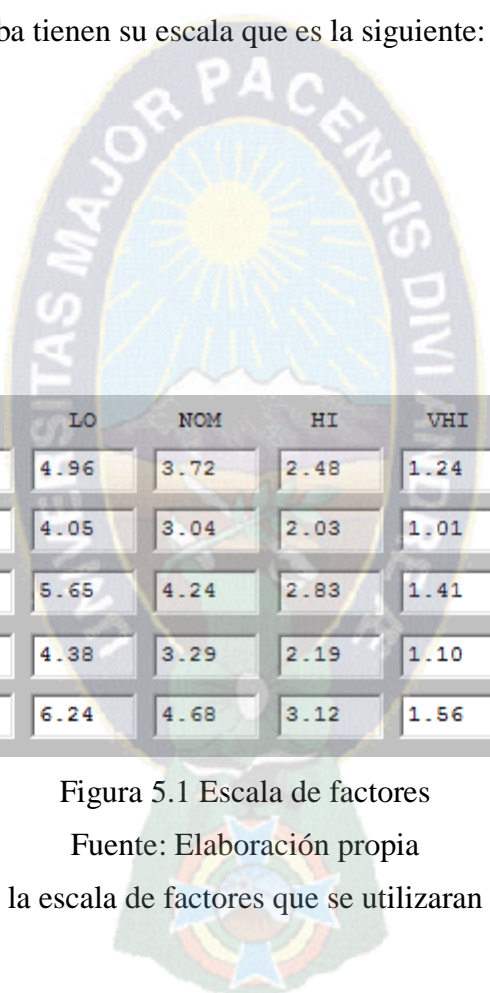
$$B = 1.01 + 0.01 * \sum_{j=1}^n w_j$$

Donde el factor de escala w_j contiene los siguientes:

- PREC.- Precedencia
- FLEX.- Flexibilidad en el desarrollo
- RESL.-Arquitectura, resolución de riesgo
- TEAM.- Cohesión de equipo
- PMAT.- Madurez del proceso

Los valores detallados arriba tienen su escala que es la siguiente:

- VLO.- Muy bajo
- LO.- Bajo
- NOM.- Normal
- HI.- Alto
- VHI.- Muy Alto
- XHI.- Extra



	VLO	LO	NOM	HI	VHI	XHI
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

Figura 5.1 Escala de factores

Fuente: Elaboración propia

En la figura 5.1 se detallan la escala de factores que se utilizaran para el análisis de costo de software.

Después de realizar las siguientes operaciones usando el software USC COCOMO II se obtendrá el costo post - arquitectura.

Infocor V2	Total SLOC	Tiempo	Totales nominales meses-persona	Total de meses persona estimadas	Estimado Productividad Total	Costo total	Costo total estimado por la Instrucción	Tiempo Total estimado por Software del Personal	Riesgo
Optimo		13	84	84	480	2888.9 4	0.1	6	
Normal	4038 6	14	105	105	384	3611.1 7	0.1	7	0.0 0
Pesimista		15	131	131	307	4513.9 6	0.1	8	

Tabla 5.6 Costo total del Software.

Fuente: Elaboración propia

Donde InfocorV2 es el nombre del proyecto del sistema, en tiempo normal se detalló el costo del sistema en 3611.17 \$us, que se desarrollara en un tiempo de 15 semanas, con una jornada laboral de 160 horas/mes, con una cantidad de 7 programadores y/o analistas.

El resultado obtenido en la tabla 5.6 es el resultado solo del costo del software, el método cocomo II solo permite calcular el software, a este resultado hay que añadirle los costos directos e indirectos que se detallaran a continuación en la tabla 5.7.

	Descripción	Monto (\$us)
Costo del Software	InfocorV2	3611.17
Costos Directos	Servicios Básicos (Luz)	300
	Insumos Básicos	
	Hojas 3 paquetes	50
	Material de Escritorio	150
	Capacitación(mes inicial)	500
Costos Indirectos	seguros	200
	depreciación (6 meses)	800
	Total	5611,17

Tabla 5.7 Costo total del sistema.

Fuente: Elaboración propia

5.3. ANÁLISIS DE LOS BENEFICIOS CON EL VAN Y EL TIR

Para analizar los beneficios que se obtendrá con la implementación del sistema se hará uso del método VAN y TIR. El VAN (Valor Actual Neto) es un indicador financiero que mide los flujos de los futuros ingresos y egresos que tendrá el proyecto, para determinar si luego de descontar la inversión inicial, nos quedaría alguna ganancia. Si el resultado es positivo, el resultado es viable. Para hallar el VAN del proyecto de inversión requerimos tres valores de acuerdo a la siguiente fórmula:

$$VAN = BNA - Inversion$$

Donde BNA es el beneficio neto actualizado, el cual debe ser actualizado de acuerdo a la tasa de descuento TD, que es la tasa de oportunidad, rendimiento o rentabilidad mínima que se espera ganar. Entonces para hallar el VAN se necesitan: tamaño de la inversión, flujo de caja neto proyectado y la tasa de descuento. De estos datos contamos con la inversión igual a 5611,17 \$u\$ y la tasa de descuento del 10% que se espera ganar, faltar el flujo de caja neto proyectado que se lo obtiene del siguiente análisis.

Descripción	Año 1	Año 2	Año 3	Año 4	Año 5
Producto por uso de software	16500	16500	16500	16500	16500
Total	16500	16500	16500	16500	16500

Tabla 5.8. Ingreso estimados.

Fuente: Elaboración propia

En la tabla 5.8 se espera ingresos por 16500 \$us. Que provienen de recursos del uso del producto.

Descripción	Año 1	Año 2	Año 3	Año 4	Año 5
Materiales directos					
Hojas	230	235	240	245	250
Tinta para Impresora	200	210	220	230	240
Material de escritorio (bolígrafos,	150	160	170	180	190

lápices, etc.)					
Remuneraciones al personal					
Sueldo – Salario CAT 2 (4 empleados) destinan un 25 % de su tiempo	4450	4450	4450	4450	4450
Sueldo - Salario CAT 5 (1 empleado) destina el 80 %	4480	4480	4480	4480	4480
Previsión Social, Indemnización, etc.	1000	1000	1000	1000	1000
Costos Indirectos					
Comunicaciones	100	100	100	100	100
Depreciación(*)	1500	1500	1500	1500	0
Adecuación y/o Capacitación(**)	1500	1500	1500	1500	1500
Materiales Indirectos(***)	500	500	500	500	500
Mano de obra Indirecta(***)	500	500	500	500	500
Total	14610	14635	14660	14685	13210

Tabla 5.9. Egresos estimados.

Fuente: Elaboración propia

En la tabla 5.9 de egresos estimados expresados en \$us. Se tomaran en cuenta las siguientes observaciones:

(*).-La depreciación estimada

(**).-La Adecuación y/o Capacitación, son gastos extraoficiales del sistema, puede que estos recursos estimados se gasten en la adecuación del software, como también capacitación en el sistema o uso de herramientas.

(***).- gastos extraoficiales por mal funcionamiento de hardware o software de las maquinas y/o reparación de impresoras, scanner, etc.

Tomando en cuenta las observaciones anteriores a los gastos estimados, se tiene un costo de 14360 \$us. En estos 5 años del proyecto.

Descripción	Año 1	Año 2	Año 3	Año 4	Año 5
Total Ingreso	16500	16500	16500	16500	16500
Total Egreso	14610	14635	14660	14685	13210
Flujo de Caja	1890	1865	1840	1815	3290

Tabla 5.10. Flujo de Caja Neto.

Fuente: Elaboración propia

En la tabla 5.10 se obtiene el flujo de caja neto el cual es la diferencia entre ingresos y egresos. El beneficio neto nominal sería de 10700, y la utilidad lógica sería 5088.83, pero este beneficio o ganancia no sería real (Solo nominal) por que no se estaría considerando el valor del dinero en el tiempo, por lo que cada periodo debemos actualizarlo a través de una tasa de descuento (Tasa de rentabilidad mínima que esperamos ganar).Aplicando la formula tenemos:

$$VAN = \frac{1890}{(1 + 0.1)^1} + \frac{1865}{(1 + 0.1)^2} + \frac{1840}{(1 + 0.1)^3} + \frac{1815}{(1 + 0.1)^4} + \frac{3290}{(1 + 0.1)^5} - 5611,17$$

$$VAN = 7924,42 - 5611,17$$

$$VAN = 2313,25$$

Como el resultado obtenido es mayor a cero en gran medida, se concluye que el proyecto es viable y la utilización del sistema va por mucho más tiempo que cinco años.

La TIR (Tasa Interna de Retorno) es la tasa de descuento (TD) de un proyecto de inversión que permite que el BNA sea igual a la inversión (VAN igual a 0). La TIR es la máxima TD que puede obtener un proyecto para que sea rentable, pues una mayor tasa ocasionaría que el BNA sea menor que la inversión (VAN menor a 0).

Entonces para hallar el TIR se necesita la inversión igual a 5611,17, además de los valores de ganancia esperados son descritos en la tabla 5.10.

Para hallar el TIR hacemos uso de la fórmula del VAN, solo que en vez de hallar el VAN (El cual reemplazamos por 0), estaríamos hallando la tasa de descuento.

$$0 = \frac{1890}{(1 + r)^1} + \frac{1865}{(1 + r)^2} + \frac{1840}{(1 + r)^3} + \frac{1815}{(1 + r)^4} + \frac{3290}{(1 + r)^5} - 5611,17$$

$$r = TIR = 24.00\%$$

Como el resultado obtenido es mayor a la tasa de descuento (TIR > TD), se concluye que el proyecto es rentable.

CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES

6. INTRODUCCIÓN

En este capítulo se detallaran las conclusiones sobre el desarrollo del sistema, así también se hará recomendaciones para el mejor uso y optimización.

6.1. CONCLUSIONES

Al momento del desarrollo del software se pudo identificar flujo de trabajo y procesos fundamentales que con ayuda de la herramienta *BPM* que se optimizo.

Después de realizar las pruebas pertinentes se detallaran a continuación las conclusiones:

- Se ha desarrollado un módulo de registro, control de correspondencia
- Se ha implementado el módulo de prioridad de correspondencia
- Se ha desarrollado el módulo de digitalización de correspondencia el cual avisa al usuario de correspondencias pendientes
- Se ha desarrollado e implementado un módulo de eficiencia para el personal
- se ha desarrollado el módulo de existencia de documentación
- Se ha desarrollado e implementado el motor de búsquedas para la correspondencia
- Se ha desarrollado el módulo de alerta temprana
- El sistema cuenta con las últimas herramientas de desarrollo de software y lenguaje de programación principalmente JAVA 7 y PostgreSQL 9.3, lo que garantizara el soporte al sistema.

Habiendo desarrollado y realizado todas las tareas, inferimos que se ha cumplido con el objetivo general.

6.2. RECOMENDACIONES

Por la dinámica de los sistemas de información de cada Institución siempre existirán nuevos requerimientos, dependiendo de las necesidades que se tengan, sin embargo se requiere concluir e implementar los siguientes proyectos.

- Combinación del sistema de correspondencia con los diferentes sistemas que cuenta la Carrera de Informática.
- Unificación con los diferentes sistemas que cuenta la Universidad
- Efectuar la integración de la Firma Digital para el sistema web
- Implementar políticas de adquisición de certificados de seguridad SSL.
- Hacer políticas de adquisición de protocolos de seguridad HTTPS.
- Agregar nuevos tipos de documentos y su proceso que tiene dentro de la carrera de informática.
- Agregar un control a los procesos de la carrera, por cada tipo de documento.
- Adicionar más documentos al sistema para que lo pueda autogenerar.
- Adicionar más características a los parámetros del sistema.



BIBLIOGRAFÍA

- Juan Díez-Yanguas Barber. (1 de Noviembre de 2011). *http://jdiezfoto.es/*. Recuperado el 4 de Noviembre de 2013, de blog de fotografia e informatica:
http://jdiezfoto.es/informatica/java-ee-seguridad-en-aplicaciones-web-i/
- Ambler, S. (21 de 07 de 2013). *www.ambysoft*. Obtenido de *www.ambysoft*:
http://www.ambysoft.com/unifiedprocess/agileUP.html
- Avilez, J. (s.f.). *monografias.com*. Recuperado el 13 de Septiembre de 2013, de *monografias.com*: *http://www.monografias.com/trabajos11/corres/corres.shtml*
- Bolivariana, U. U. (2013). *ingenieriadesoftware*. Recuperado el 2013 de Septiembre de 2013, de *ingenieriadesoftware*: *http://ingenieriadesoftware.mex.tl/63758_AUP.html*
- Carrera de Informatica. (21 de Mayo de 2010). *Carrera de Informatica*. Recuperado el 5 de Agosto de 2013, de *Carrera de Informatica*: *http://informatica.edu.bo/*
- ejemplosTIW. (2013). *ejemplosTIW*. Recuperado el 18 de Septiembre de 2013, de *ejemplosTIW*: *http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html*
- eumed net. (2009). *eumed net*. Recuperado el 20 de Agosto de 2013, de *eumed net*:
http://www.eumed.net/libros-gratis/2009c/587/Metodologias%20y%20Tecnologias%20Actuales%20para%20la%20construccion%20de%20Sistemas%20Multimedia.htm
- González, C. (23 de Marzo de 2009). *Adictos al trabajo*. Recuperado el 1 de Octubre de 2013, de *Adictos al trabajo*:
http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJ
ava#_Toc225422688
- Huanca, M. (2008). *SISTEMA DE REGISTRO Y SEGUIMIENTO DE CORRESPONDENCIA CASO:SERVICIO DEPARTAMENTAL DE GESTION SOCIAL (SEDEGES)*. La Paz - Bolivia: Licenciatura en informática mención Ingeniería de Sistemas Informáticos Universidad Mayor de San Andrés Carrera de Informática .
- Huanca, S. (2008). *SISTEMA DE REGISTRO Y SEGUIMIENTO DE CORRESPONDENCIA Y ASIGNACION DE AULAS PARA EL CENTRO DE MULTISERVICIOS EDUCATIVOS CEMSE-CERPI*. La Paz -Bolivia: Licenciatura

en informática mención Ingeniería de Sistemas Informáticos Universidad Mayor de San Andrés Carrera de Informática.

Ibañez, A. (2009). *"CHASQUI DIGITAL" E- CORRESPONDENCIA CASO:FACULTAD DE CIENCIAS PURAS Y NATURALES*. La Paz - Bolivia: Licenciatura en informática mención Ingeniería de Sistemas Informáticos Universidad Mayor de San Andrés Carrera de Informática.

MARCADÉ, T. (26 de Enero de 2009). *laurel.datsi.fi.upm.es*. Recuperado el 1 de Octubre de 2013, de *laurel.datsi.fi.upm.es*:
http://laurel.datsi.fi.upm.es/~ssoo/DAW/Trabajos/2008-2009/001/func_es

OLSINA, L. A. (1999). *Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web*. La Plata: Facultad de Ciencias Exactas Universidad Nacional de La Plata - Argentina.

prestashop5estrellas. (29 de Marzo de 2010). *prestashop5estrellas*. Recuperado el 01 de Octubre de 2013, de *prestashop5estrellas*:
<https://prestashop5estrellas.wordpress.com/2010/03/29/el-patron-mvc-modelo-vista-controlador/>

sanchez, L. (22 de Noviembre de 2004). *degerencia.com*. Recuperado el 1 de Octubre de 2013, de *www.degerencia.com*:
http://www.degerencia.com/articulo/business_process_management_bpm_articulan_do_estrategia_procesos_y_tecnologia

UPV. (s.f.). *Universidad del Pais Vasco*. Recuperado el 21 de Octubre de 2013, de Universidad del Pais Vasco: <http://www.sc.ehu.es/jiwdocoj/mmis/cocomo.htm>

UWE – UML-based Web Engineering. (7 de Septiembre de 2012). *UWE – UML-based Web Engineering*. Recuperado el 6 de Noviembre de 2013, de UWE – UML-based Web Engineering: <http://uwe.pst.ifi.lmu.de/>

Vargas, C. (2009). *SISTEMA DE CORRESPONDENCIA PARA LA EMPRESA SIMSA*. LA PAZ - BOLIVIA: Licenciatura en informática mención Ingeniería de Sistemas Informáticos Universidad Mayor de San Andrés Carrera de Informática.

WIKIPEDIA. (s.f.). Recuperado el 18 de Septiembre de 2013, de http://es.wikipedia.org/wiki/UWE_UML

Wikipedia. (2003). *Wikipedia, UWE UML*. Recuperado el 08 de Agosto de 2013, de Wikipedia: http://es.wikipedia.org/wiki/UWE_UML

Wikipedia. (2006). *Wikipedia UML*. Recuperado el 20 de Agosto de 2013, de Wikipedia:
http://es.wikipedia.org/wiki/UWE_UML

Wikipedia. (s.f.). *Wikipedia*. Recuperado el 18 de Septiembre de 2013, de Wikipedia:
http://es.wikipedia.org/wiki/Spring_Framework

www.soaagenda.com. (2013). <http://www.soaagenda.com/>. Recuperado el 01 de Octubre de 2013, de <http://www.soaagenda.com/>:
<http://www.soaagenda.com/journal/articulos/que-es-bpm-que-es-bpms/>



GLOSARIO

Workflow.- El flujo de trabajo en español es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas. Generalmente los problemas de flujo de trabajo se modelan con redes de Petri.

LeanOffice.- Oficina Esbelta en español es una nueva tendencia de organización que coadyuvara a la administración de procesos de negocios.

Stakeholder.- Se puede definir como cualquier persona o entidad que es afectada o concernida por las actividades o la marcha de una organización

I.I.I.- Instituto de investigaciones en informática que fue fundado en 1994

UMSA.- Fue creada por Decreto Supremo de 25 de octubre de 1830. Debido a la importancia comercial de la ciudad de La Paz, desde su creación la UMSA tuvo influencia en la vida, sobre todo, social y en la historia de Bolivia

SEDEGES.- Servicio departamental de gestión social

S.I.M.S.A...-Sociedad industrial molinera sociedad anónima.

CEMSE – CERPI. - Centro multiservicios Educativos

UWE.- UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación

OOHDM.- OOHDM es una mezcla de estilos de desarrollo basado en prototipos, en desarrollo interactivo y de desarrollo incremental. En cada fase se elabora un modelo que recoge los aspectos que se trabajan en esa fase

ISO.- Organización de Estándares Internacionales en español

Framework Spring MVC.- Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. La primera versión fue escrita por Rod Johnson.

Java.- El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems

UML.- Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Está respaldado por el OMG (Object Management Group).

Postgresql.- es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Pgplsql.- es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas.

JavaScript.- JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, 3 basado en prototipos, imperativo, débilmente tipado y dinámico.

RUP.- El Proceso Unificado de Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM.

AUP.- El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe

de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas

CM.- Configuration manager en español Administración de la Configuración

MVC.- El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones

WebML.- WebML (Web Modeling Language) es una notación visual para el diseño de aplicaciones Web complejas que usan datos intensivamente. Provee especificaciones gráficas formales para un proceso de diseño completo que puede ser asistido por herramientas de diseño visuales.

UWE.-UWE UML (UML-Based Web Engineering) es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos).

Browsing.- navegador

Processing.- procesamiento

BPM.- (Business Process Management), Se llama Gestión o administración por procesos de negocio (Business Process Management o BPM en inglés) a la metodología corporativa cuyo objetivo es mejorar el desempeño (Eficiencia y Eficacia) de la Organización a través de la gestión de los procesos de negocio, que se deben diseñar, modelar, organizar, documentar y optimizar de forma continua.

BPMS.- (BPM Suite) , “Una nueva categoría de software empresarial que permite a las empresas modelizar, implementar y ejecutar conjuntos de actividades interrelacionadas –es decir, Procesos- de cualquier naturaleza, sea dentro de un departamento o permeando la entidad en su conjunto, con extensiones para incluir los clientes, proveedores y otros agentes como participantes en las tareas de los procesos”.

Backdoor.- En la informática, una puerta trasera, en un sistema informático es una secuencia especial dentro del código de programación, mediante la cual se pueden evitar los sistemas de seguridad del algoritmo (autenticación) para acceder al sistema. Aunque estas puertas pueden ser utilizadas para fines maliciosos y espionaje no siempre son un error, pueden haber sido diseñadas con la intención de tener una entrada secreta.

HTTPS.- Hypertext Transfer Protocol Secure (en español: Protocolo seguro de transferencia de hipertexto), más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.

HTTP.- Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxis) para comunicarse.

SSL.- SSL; en español capa de conexión segura

Bruteforce.- En criptografía, se denomina ataque de fuerza bruta a la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

Inyección SQL.- Es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar consultas a una base de datos.

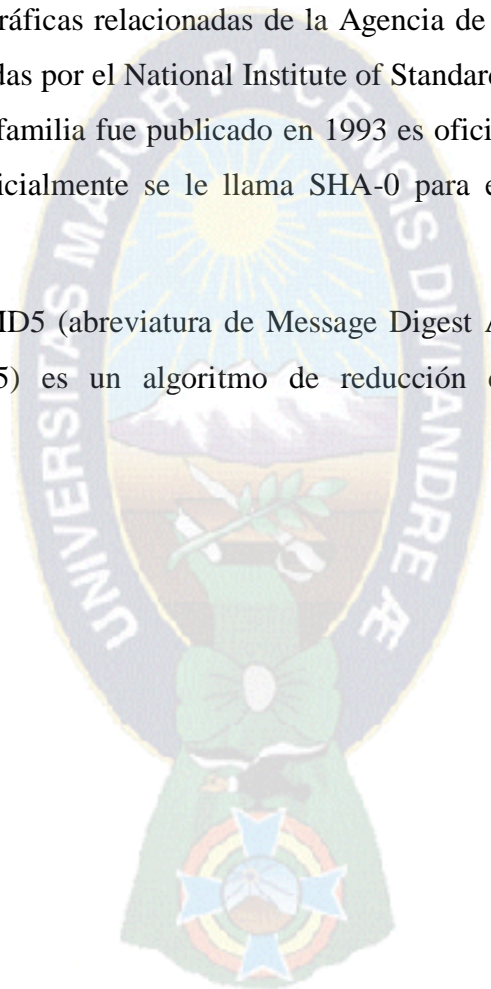
Zombi Server.- Zombi, es la denominación que se asigna a computadores personales que tras haber sido infectados por algún tipo de malware, pueden ser usados por una tercera persona para ejecutar actividades hostiles. Este uso se produce sin la autorización o el

conocimiento del usuario del equipo. El nombre procede de los zombis o muertos vivientes esclavizados, figuras legendarias surgidas de los cultos vudú.

SOX.- SOX., lo que se esperaba encontrar, era un listado de controles y requisitos a nivel de seguridad y de todo el gobierno en las Tecnologías de la Información

SHA.- La familia SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el National Institute of Standards and Technology (NIST). El primer miembro de la familia fue publicado en 1993 es oficialmente llamado SHA. Sin embargo, hoy día, no oficialmente se le llama SHA-0 para evitar confusiones con sus sucesores.

MD5.- En criptografía, MD5 (abreviatura de Message Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

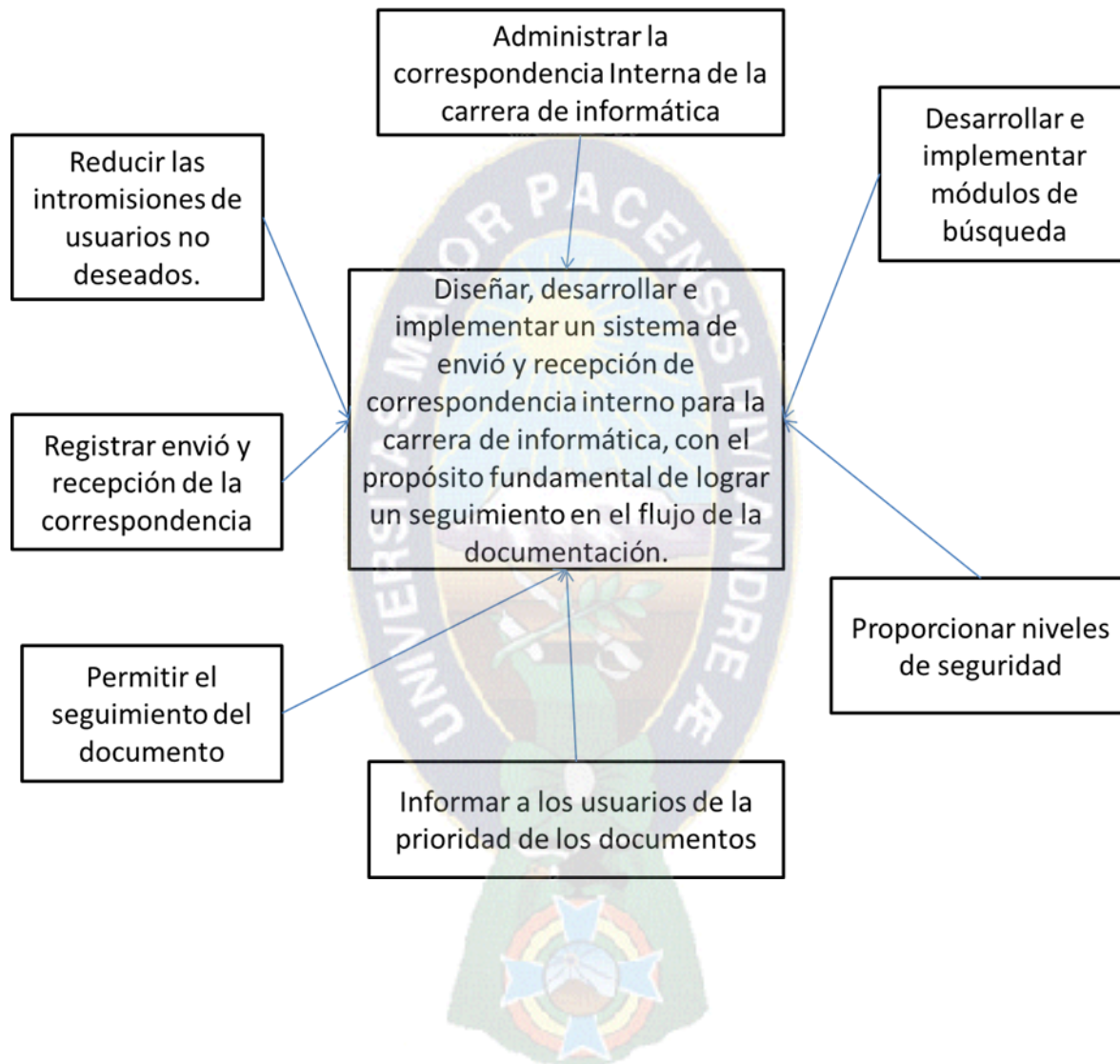


ANEXOS

ANEXO A. Árbol de Problemas



ANEXO B. Árbol de Objetivos



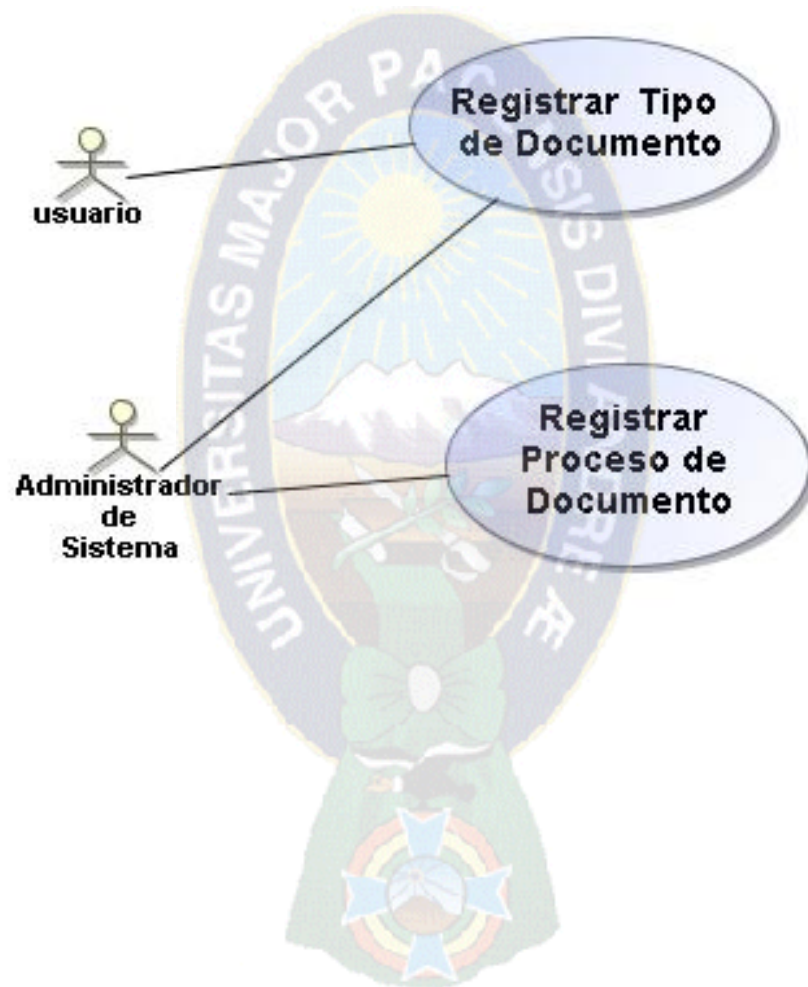
ANEXO C. MARCO LÓGICO

Resumen Narrativo	Indicadores	Medios de Verificación	Supuestos
Fin Mejorar el envío y recepción de la correspondencia dentro de la carrera de informática.	Impactos Disponibilidad de la información acerca de un manejo.	Medios de Verificación. Respaldo de reportes a través del sistema	Sostenibilidad Los usuarios del sistema
Propósito Desarrollar un sistema de envío y recepción de correspondencia para la carrera de informática con el propósito de lograr el flujo de la documentación	Resultados Los usuarios del sistema obtendrán una información completa y actualizada a partir de noviembre del 2013	Medios de Verificación Nota de certificación de evaluación por parte del personal autorizado.	Propósito a Fin Los usuarios que necesiten ayuda la requieran de acuerdo al manejo de software.
Componentes/Productos Módulos de registro Módulos de búsqueda Módulos de reportes	Productos Simplificar , clarificar, economizar la gestión de los procesos de correspondencia dentro de la carrera de	Medios de Verificación El sistema se alojara en el servidor de la carrera de Informática.	Componentes/ Productos a Propósito La carrera de informática proveerá de la computadora para la implementación del sistema

	informática		
Actividades	Costos	Medios de Verificación	Actividades a Componentes
Recopilación de información de sistemas similares Planificación Recopilación de información de la institución Modelado del sistema Análisis Diseño del sistema Implementación y Pruebas Capacitación	Los costos del sistema en si será de 3500 \$ Dólares americanos	Documentación del sistema	Colaboración del personal de Secreta de la carrera de informática

ANEXO D. ADICIONAR NUEVOS DOCUMENTOS Y SUS PROCESO

Diagrama de Caso de uso para la adición de nuevos tipos de documentos.



Caso de uso:	Registrar tipo de documento
Actores:	Usuario Administrador del sistema
Propósito:	Registra nuevo tipo de documento para la correspondencia.
Descripción:	El usuario que está recibiendo la correspondencia de entrada, recibe un tipo de documento que no está en el sistema, se tiene que registrar para luego poder registrar ese tipo de documento.
Tipo:	Primario

Caso de uso:	Registrar proceso de documento
Actores:	Administrador de sistema
Propósito:	Registrar el proceso que debe seguir la correspondencia que ingreso para su culminación.
Descripción:	<p>El administrador del sistema debe detallar detenidamente el proceso del documento que fue recibido, por ejemplo certificado de alumno regular.</p> <p>El proceso que se debe antora de la siguiente manera:</p> <ul style="list-style-type: none"> • Verificar el documento, debe contener fotocopia simple de C.I. y matricula de la presente gestión. • Entregar comprobante de registro. • Remitir a Kardex. • Kardex emite certificado de Alumno regular. • Culminación.
Tipo:	Primario y esencial

Flujo BPM del proceso de adición de nuevos tipos de documento y sus procesos.

