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

PROYECTO DE GRADO

"SISTEMA DE INFORMACION DE GESTIÓN Y CONTROL DE CORRESPONDENCIA"

CASO: TECNOLOGIA SISTEMAS Y APLICACIONES - TSA BOLIVIA

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

Postulante: Edwin Teodocio Suxo Arroba

Tutor Metodológico: Lic. Javier Reyes Pacheco

Asesor: M.Sc. Carlos Mullisaca Choque

La Paz – Bolivia

2015

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

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

DEDICATORIA

El presente Proyecto de Grado lo dedico…

A mis Padres por guiar mi camino con paciencia y dedicación a lo largo de mi vida y ante todo por su apoyo, confianza, y cariño incondicional.

Sr. Tiburcio Suxo Nina (┼)

Sra. Exaltacion Huanca (┼)

A mi Esposa Lilian quien es la fuente de mi inspiración y por la cual no podría haber llegado al final de este proyecto.

AGRADECIMIENTOS

En primer lugar a Dios, por concederme su santa bendición, dándome la sabiduría y guía perfecta para la realización y conclusión satisfactoria del desarrollo e implementación del presente Proyecto de Grado.

A mi Tutor Metodológico Lic. Javier Reyes Pacheco, por las sugerencias, control y seguimiento a la estructura de la documentación del presente Proyecto de Grado, sus sugerencias y comentarios técnicos añadidos a su experiencia profesional, ayudaron a lograr mis objetivos planteados, por su tiempo para la revisión del documento, gracias por el apoyo incondicional.

A mi docente Asesor M.Sc. Carlos Mullisaca Choque, por su constante e incondicional colaboración, por sus observaciones, sugerencias, comentarios y por su tiempo dedicado a la revisión del documento y software, durante todo el desarrollo del presente Proyecto de Grado.

Muchas gracias a toda mi familia y amigos por todo su apoyo moral en la realización del presente proyecto.

RESUMEN

El propósito del proyecto es brindar un correcto manejo de la información haciendo de esta una herramienta que ayude con las tareas de registro, control y seguimiento de la correspondencia interna dentro la empresa.

La metodología usada en este proyecto, es el Proceso Unificado Ágil (AUP), para el análisis del sistema, orientado a la Ingeniería Web (UWE) y el análisis desde el punto de vista Workflow (o flujo de trabajo automatizado), y para la realización de modelos se aplicó UML (Lenguaje de Modelo Unificado) y su extensión orientada a la web UWE (UML-Based Web Engineering).

En los diferentes capítulos descritos en el presente proyecto identificamos la problemática y objetivos del sistema, luego mostramos información teórica para comprender la base teórica del sistema. Después llegamos a la fase del análisis con ayuda del AUP, en su fase de inicio, elaboración, construcción y transición. También tomamos en cuenta la parte de seguridad de riesgos de acceso y a nivel de la base de datos. Llevamos a cabo el proceso de calidad de software y los cálculos de costo y beneficio. Por ultimo brindamos las conclusiones y las recomendaciones respectivas.

Para la implementación del sistema se utilizó el lenguaje de programación PHP y como gestor de bases de datos MySQL.

ÍNDICE

CONTENIDO **Página**

CAPITULO III: MARCO APLICATIVO

CAPITULO IV: CALIDAD DE SOFTWARE

ÍNDICE DE FIGURAS

CONTENIDO **Página**

ÍNDICE DE TABLAS

CONTENIDO **Página**

CAPITULO I
MARCO INTRODUCTORIO

1.1 INTRODUCCION

La sociedad actual exige un constante trabajo liviano, preciso y automatizado, una época donde las personas dediquen todo el tiempo posible a las actividades necesarias e indispensables para la institución y dejar de lado los procesos manuales que implican esas actividades.

La cantidad de información que es generada en una institución como ser TSA BOLIVIA (Tecnología Sistemas y Aplicaciones) provoca contratiempos en el flujo del trabajo, el cual retrasa las actividades que podrían realizarse en un menor y más efectivo tiempo.

Las diferencias tecnológicas como ser los Workflows o flujos de trabajos ayudan a modelar y diseñar estos problemas o proponer una solución efectiva. Mediante las aplicaciones Web se ha mejorado los contratiempos generados en el proceso de solución de una hoja de ruta.

Habiendo mencionado esto es por lo cual el Sistema de registro y seguimiento de correspondencia, es una herramienta que ayuda a controlar las tareas que los usuarios requieran. [Elaboración: Propia (Argumentos de la empresa)]

1.2 ANTECEDENTES

a) Antecedentes de la Institución

Tecnología Sistemas y Aplicaciones Bolivia S.A. (TSA Bolivia S.A.), es una empresa conformada por un grupo de profesionales de alto nivel, que unifican esfuerzo y trabajo para ofrecer servicios integrales de alta calidad en el área de las Tecnologías de la Información, logrando obtener y desarrollar una metodología propia de trabajo que pone de manifiesto el éxito de la empresa.

TSA Bolivia S.A., empresa creada como Sociedad Anónima mediante Testimonio Público No. 767/97 en Octubre de 1997, se constituye en la alternativa a los requerimientos de servicios de TI y gestión administrativa en Bolivia, proponiendo soluciones basadas en tecnología de punta, avaladas por el prestigio, experiencia y calidad de servicios ofrecidos por el equipo de profesionales que conforman actualmente la empresa. [Elaboración: Propia (Argumentos de la empresa)]

b) Antecedentes Temáticos

A continuación se describen trabajos similares al presente proyecto:

- "Chasqui Digital E-Correspondencia", de Aleida Raquel Ibañez Apaza. Caso: Facultad de Ciencia Puras y Naturales, de la Universidad Mayor de San Andes. (IBAÑEZ. A. 2009). El proyecto escribe el uso de metodología RUP y UML. Se ha realizado un sistema para el control de correspondencia, de este proyecto se especifica el uso de la metodología RUP el mismo que es parecido a la metodología AUP, que se usa en el presente proyecto.
- "Sistema de gestión, control y monitoreo de procesos utilizando tecnologías, Workflow, centro de multiservicios educativos CEMSE, de Laura Quisbert Bustamante. (QUISBERT L., 2009). En este proyecto se ha usado la aplicación de las tecnologías Workflow, el mismo que se está investigando en el presente proyecto, mediante los procesos de negocio automatizados.
- "Sistema de información de seguimiento de Tramites via web Gobierno Municipal de El Alto" de Renan Cancari Guarachi (CANCARI R. 2009). El proyecto menciona el uso de los parámetros de calidad software, mismo del cual se plantea usar en el presente proyecto, tuvo como referencia la funcionalidad, fiabilidad y eficacia. [Elaboración: Propia (Argumentos de la empresa)]

1.3 ANÁLISIS DE LA SITUACIÓN ACTUAL

La empresa cuenta con un sistema de correspondencia, pero dicho sistema tiene funcionalidades básicas manuales con la excepción de que se emplea el programa Excel para contabilizar datos, por lo tanto debido al análisis de la situación actual se vio la factibilidad de desarrollar un sistema de correspondencia desde el enfoque de la Ingeniería de sistemas

para aportar en la estructura organizacional de la empresa T.S.A Bolivia. Dicho sistema cumplirá características como:

- Un control de la correspondencia donde se verificara en qué estado se encuentra la correspondencia de cada cliente.
- Poder ayudar al cliente ante cualquier reclamo o solitud de información que requiera con respecto a la correspondencia remitida.
- Descripción del proceso de la correspondencia determinada donde se describirá en qué departamento se encuentra y cuáles fueron las observaciones de la misma. [Elaboración: Propia (Argumentos de la empresa)]

1.4 DEFINICIÓN DEL PROBLEMA

La Correspondencia dentro de Tecnología Sistemas y Aplicaciones (TSA BOLIVIA), sigue el siguiente flujo, en primera instancia se tiene que registrar la solicitud o carta, luego se genera una hoja de ruta para dar continuidad a la referencia de dicha carta o solicitud, el flujo regular de las hojas de ruta son mediante recepciones y derivaciones hasta concluir la hoja de ruta, para luego realizar los descargos necesarios de cada funcionario de la empresa, las transacciones de derivación son realizados en un cuaderno de Correspondencia en cada Área, donde se detalla la fecha de recepción, procedencia y proveído, en algunos casos se tiene un archivo de formato Excel para el control de las recepciones. La consulta y búsqueda de correspondencia se tiene que revisar el histórico de recepciones y así obtener la información necesaria del estado del registro o donde fue enviado.

Cada área de TSA hace la recepción y deriva una gran cantidad de información, para el cumplimiento de funciones o actividades a realizarse.

Por tanto dicho lo anterior se observa que: Los mecanismos de manejo de la información, un elevado crecimiento de la información y una estructura organizacional relativamente compleja en TSA, ocasionaran retardos, contratiempos y hasta pérdida de información. [Elaboración: Propia (Argumentos de la empresa)]

1.4.1 Problema Principal

El manejo de la correspondencia en Tecnología Sistemas y Aplicaciones (TSA BOLIVIA), se realiza de manera manual lo que genera retardos, contratiempos, perdida de información, escasa información, en la respuesta de solitudes y procesos de información, al momento de dar continuidad a las hojas de ruta. [Elaboración: Propia (Argumentos de la empresa)]

1.4.2 Problemas específicos

- El registro de la correspondencia se realiza de manera manual
- Heterogeneidad en los archivos de registro de las recepciones y las derivaciones en las diferentes unidades.
- Inconsistencia de datos, debido a que se comete errores de transcripción, durante el registro las hojas de ruta.
- Burocratización en todo el proceso y recorrido de la correspondencia dentro de la institución.
- Difícil acceso a la información en las diferentes áreas y documentación de los registros que generan las hojas de ruta.
- Perdida del recorrido de la hoja de ruta tanto del estado y la ubicación actual al momento de saber su estado actual.

1.4.3 Formulación del Problema

¿El desarrollo de un Sistema de Información de Gestión y Control, permitirá el manejo eficiente de la documentación de la empresa T.S.A. BOLIVIA (Tecnología Sistemas y Aplicaciones)?

1.5 OBJETIVO

1.5.1 Objetivo General

Desarrollar un Sistema de Información de Gestión y Control, para permitir optimizar el manejo de la Correspondencia de la empresa T.S.A. BOLIVIA (Tecnología Sistemas y Aplicaciones). [Elaboración: Propia (Argumentos de la empresa)]

1.5.2 Objetivos Específicos

- Desarrollar el diseño del manejo de la correspondencia de la institución con la tecnología Workflow.
- Elaborar la estructura de almacenamiento de información en una base de datos relacional, de acuerdo a las necesidades de correspondencia.
- Elaborar la interfaz gráfica del sistema orientado a la Web con mecanismos de validación de datos.
- Desarrollar el software de registro, recepción, derivación, búsqueda para el seguimiento de correspondencia en un gestor de base de datos MySQL y en el lenguaje de programación PHP.
- Vista de reportes especializados con información de la correspondencia.
- Control eficiente en el recorrido de las hojas de ruta en la institución.
- Desarrollar un módulo de seguridad para el ingreso no autorizado de usuarios y administrador.

1.6 JUSTIFICACIÓN

1.6.1 Justificación Económica

Disminuir el uso de papel en los trámites, para poder contribuir con la naturaleza y en el ahorro de recursos al tener un trámite electrónico se puede prescindir en la mayor parte del flujo del trámite el uso de papel que se maneja usualmente contribuyendo de esta forma en un manejo más coherente de los recursos naturales y económicos, del cual tendríamos como resultado una optimización de estos recursos que benefician en todo sentido a los usuarios, a

la empresa, la naturaleza y la reducción de documentación intermedia que es desechada como basura. [Elaboración: Propia (Argumentos de la empresa)]

1.6.2 Justificación Social

La relevancia de la importancia social del presente trabajo ira en dirección del personal encargado del trabajo respecto al proceso de correspondencia de la empresa T.S.A Bolivia. Así también el beneficio hacia los usuarios finales es un eficaz control de la documentación, un procesamiento rápido de la información, obteniendo así respuestas más eficientes sobre la documentación que integra la institución.

El funcionario de la empresa se dedicara más a procesar la información del documento que vigilar el movimiento del mismo. La información generada puede ser revisada sin perjudicar ni retrasar a los demás usuarios, y permitirá analizar el flujo de las hojas de ruta y mejorar su procesamiento. [Elaboración: Propia (Argumentos de la empresa)]

1.6.3 Justificación Operacional

Dentro del parámetro operacional el mismo será desarrollado por personal especializado en el área de sistemas, ya que la arquitectura del sistema estará basada en una metodología AUP (Agile UP o proceso Unificado Ágil), y como herramientas de apoyo se tiene a los diagramas UML (Lenguaje Unificado de Modelado) y su extensión orientada a la web denominada UWE (UML-Based Web Engineering), para el análisis del flujo de la información se utilizó la tecnología Workflow, mediante rutas Ad Hoc, con diagramas BPMN (Business Process Modeling Notation). [Elaboración: Propia (Argumentos de la empresa)]

1.6.4 Justificación Técnica

La justificación técnica del diseño del presente proyecto estará enfocado al hecho de emplear una metodología técnica que sea factible dentro el diseño de un sistema de correspondencia, ya que las características técnicas del mismo son a exigencias y necesidades que tiene la empresa T.S.A. Bolivia. [Elaboración: Propia (Argumentos de la empresa)]

1.7 METODOLOGÍA

1.7.1 Metodología Archivista

La disciplina archivística es de gran importancia para las entidades productoras de documentos, ya que esta se encarga de establecer todos los procesos para un adecuado tratamiento del documento, desde que se concibe hasta que pasa al archivo histórico. De acá lo valioso de establecer métodos óptimos en pro de garantizar el acceso a la información. [Fuente: Web [1]]

La metodología empleada será la archivística, que es la metodología que mejor expresa las cualidades de la documentación como ser origen, transferencia y expurgo. [Elaboración: Propia (Argumentos Propios)]

1.7.2 Metodología Ágil

Las metodologías ágiles son una serie de técnicas para la gestión de proyectos que han surgido como contraposición a los métodos clásicos de gestión como CMMI (Modelo de Capacidad y Madurez). Aunque surgieron en el ámbito del desarrollo de software, también han sido exportadas a otro tipo de proyectos.

Todas las metodologías que se consideran ágiles cumplen con el manifiesto ágil que no es más que una serie de principios que se agrupan en 4 valores:

1. Los individuos y su interacción, por encima de los procesos y las herramientas.
2. El software que funciona, frente a la documentación exhaustiva.
3. La colaboración con el cliente, por encima de la negociación contractual.
4. La respuesta al cambio, por encima del seguimiento de un plan. [Fuente: Web [2]]

Así también la metodología se basa en un Proceso Unificado Ágil, orientado a la Ingeniería Web (UWE) y el análisis desde el punto de vista Workflow (o flujos de trabajo automatizados), permiten diseñar una arquitectura concreta, con procesos de negocios dinámicos para un sistema software orientado al movimiento de información o documentación de una institución, garantizando la integridad del sistema, con una adecuada

combinación de metodologías, herramientas y tecnologías de desarrollo.). [Elaboración: Propia (Argumentos Propios)]

1.8 LIMITES Y ALCANCES

1.8.1 Límites

Dentro de los límites del presente proyecto se toman los siguientes puntos lo cuales deben ser tomados en cuenta al momento de la aplicabilidad del sistema de correspondencia:

- El sistema no decide si la correspondencia es improcedente.
- El sistema no se responsabiliza por una mala aplicación del mismo.
- El sistema brinda ayuda de acuerdo a las observaciones que se indiquen y no así de la redacción de la misma. [Elaboración: Propia (Argumentos Propios)]

1.8.2 Alcances

Los alcances que se tienen con el presente proyecto son:

- El sistema se desarrollar en la ciudad de La Paz-Bolivia, esto con la factibilidad de ser replicable en otras empresas o ciudades.
- El sistema permitirá realizar la codificación de la correspondencia que ingresa.
- El sistema permitirá realizar el seguimiento de un documento.
- El sistema permitirá indicar el estado de la correspondencia. [Elaboración: Propia (Argumentos Propios)]

1.9 TECNICAS Y HERRAMIENTAS

Las técnicas empleadas sobre el tratamiento de la correspondencia se basan en la archivística.

Las herramientas empleadas serán de distribución libre, en la mayoría de los casos como son:

- Enterprise Arquitect, para la realización de los diagramas UML (Lenguaje Unificado de Modelado) y su extensión orientada a la web denominada UWE (UML-Based Web Engineering)

- Base de datos, MYSQL por su facilidad y el costo gratuito
- El lenguaje de programación PHP ya que tiene gran aceptación y aun es gratuito
- Otras herramientas CASE y de distribución libre para el modelado de datos.[Elaboración: Propia (Argumentos Propios)]

1.10 APORTE

El aporte global que dará el presente proyecto, será en beneficio directo de la empresa T.S.A. Bolivia, así también con la aplicabilidad del mismo desde el enfoque de la Ingeniería de Sistemas, será un aporte técnico metodológico ya que se detallara y describirá tanto la: arquitectura, funciones, método y manera del uso del sistema. [Elaboración: Propia (Argumentos de la empresa)]

CAPÍTULO II

MARCO TEÓRICO

2.1 CORRESPONDENCIA

La correspondencia son medios de comunicación utilizados para comunicarse con personas o individuos que están a la larga o corta distancia con un motivo muy variado. A lo largo del tiempo se han ido mejorando que patrones y sus estilos hasta llegar al e-mail que es la forma más rápida de enviar y asegurarse que la información llegue al destinatario. [Fuente: Web [4]]

2.1.1 Concepto de Correspondencia

La correspondencia es el trato reciproco entre dos personas mediante el intercambio de cartas, informes, telegramas, etc. Es considerada el motor del flujo de la información. [Fuente: Web [4]]

2.1.2 Tipos de Correspondencia

De acuerdo al destino de la correspondencia se tiene 2 tipos:

a) Internos.- son aquellos que se generan de la misma institución
b) Externos.- son aquellos que se generan de parte una persona particular o establecimiento, ajena a la institución.

Según su contenido:

a) Primera clase: Se refiere a toda correspondencia de carácter personal y que solo contiene información escrita, por ejemplo cartas, postales, documentos de negocio y comerciales, periódicos, etc.
b) Segunda clase: Hace referencia a toda la correspondencia relacionada con bultos, paquetes, es decir, objetos de tamaño considerable, que son tramitados a través de una oficina de correos determinada. [Fuente: Web [4]]

2.2 HOJAS DE RUTA

Una hoja de ruta es un plan que establece a grandes rasgos la secuencia de pasos para alcanzar un objetivo.

Se especifican tiempo y recursos necesarios. Se debe entender como recursos al envío y recepción de correspondencia a la totalidad de área hasta transformarlo en otro y que debe pasar en un determinado tiempo establecido por el requerimiento de alcance de objetivo. [Elaboración: Propia (Argumento Propio)]

2.2.1 Descripción de las Hojas de Ruta

Una "hoja de ruta" es un documento en el que se especifican las operaciones que se realizan sobre un mismo artículo hasta el momento de transformarlo en otro. En cada etapa productiva, en dicha hoja de ruta se debe incorporar información relevante como funcionarios involucrados, tipo de acción a tomar, fecha y hora de entrega o paso a siguiente etapa de cumplimiento de deberes, etc. [Elaboración: Propia (Argumento Propio)]

2.2.2 Proceso de las Hojas de Ruta

La hoja de ruta comienza con el registro de una nota o carta de la unidad de correspondencia, se asigna una numeración correlativa de control en la misma unidad. Una vez tomada la correspondencia es derivada al funcionario de destino para su proceso.

Después de una serie de ciclos y flujos de recepción y derivación, llega a una etapa de conclusión o finalización de la hoja de ruta, donde la correspondencia llega a su correcto proceso, realiza su respectivo descargo hacia el funcionario que derivo dicha hoja de ruta, los descargos son una forma de indicar que su trabajo ha sido realizado y concluido.

Terminado los descargos correspondientes el último funcionario con la hoja de ruta puede entregar una respuesta y concluir el ciclo de ruta o archivar la misma para tener un histórico del trabajo realizado. [Elaboración: Propia (Conocimiento de recorrido de la empresa)]

2.3 METODOLOGÍA

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de las denominadas metodologías ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba una mayor difusión y reconocimiento.

Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es y fue el Manifiesto Ágil, un documento que resume la filosofía "ágil". [Fuente: Web [5]]

2.3.1 El Manifiesto Ágil

El Manifiesto comienza enumerando los principales valores del desarrollo ágil. Se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. No se necesitan desarrolladores brillantes, sino desarrolladores que se adapten bien al trabajo en equipo. Así mismo, las herramientas (compiladores, depuradores, control de versiones, etc.) son importantes para mejorar el rendimiento del equipo, pero el disponer más recursos que los estrictamente necesarios también puede afectar negativamente. En resumen, es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente.

Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

- Desarrollar software que funciona más que conseguir una buena documentación. Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es "no producir documentos a menos que sean necesarios de forma inmediata para tomar un decisión importante". Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.
- La colaboración con el cliente más que la negociación de un contrato. Las características particulares del desarrollo de software hace que muchos proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a los largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses. [Fuente: Web [5]]

2.3.2 Proceso Unificado Ágil - AUP

El Proceso Unificado Agil 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 (test driven development - TDD),

Modelado Agil, Gestión de Cambios Agil, y Refactorización de Base de Datos para mejorar la productividad.

El proceso unificado (*Unified Process* o UP) es un marco de desarrollo software iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto software. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (*Rational Unified Process*) de IBM.

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. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iníciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP. [Fuente: Web [6]]

.

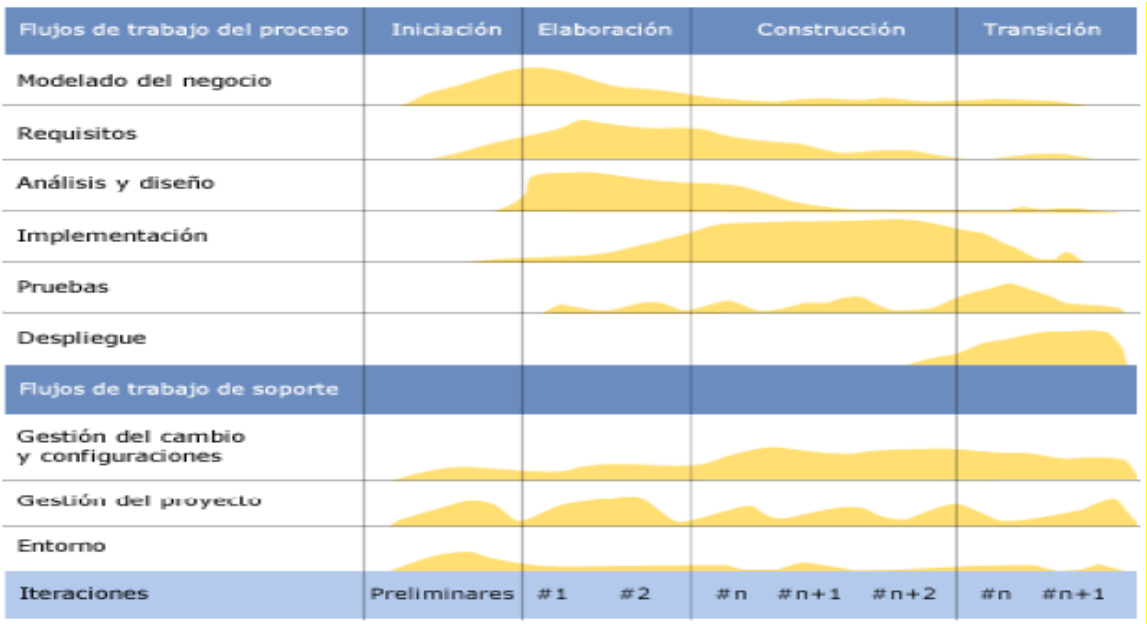

Figura 2.1 Fases de interacciones de AUP (Ciclo de Vida)

[Fuente: Web [6]]

2.3.3 Disciplinas de AUP

Las disciplinas son ejecutas de una forma iterativa, definiendo las actividades, las cuales el equipo de desarrollo ejecuta para construir, validar y liberar software funcional, el cual cumple con las necesidades de los involucrados. Las disciplinas son:

a. **Modelado.** El objetivo de esta disciplina es entender del 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.
b. **Implementación.** El objetivo de esta disciplina es transformar su modelo (s) en código ejecutable y llevar a cabo un nivel básico de las pruebas, en particular, la unidad de prueba.

c. **Pruebas.** El objetivo de esta disciplina es ejecutar una objetiva evaluación para asegurar la calidad. Esto incluye la detección de defectos, validaciones de que el sistema funciona como fue diseñado, y verificar que se cumplan los requerimientos.
d. **Despliegue.** El objetivo de esta disciplina es planificar la entrega del proyecto de desarrollo y ejecutar el plan, para dejar disponible el sistema al usuario final.
e. **Administración de la Configuración.** La meta de esta disciplina es manejar el acceso a sus productos de trabajo de proyecto. Esta no solo incluye el retraso de versiones del trabajo del producto en el tiempo, sino que también el control y administración de los cambios estos productos.
f. **Administración de Proyecto.** El objetivo de esta disciplina es manejar al acceso a sus productos de trabajo de proyecto. Esta incluye la administración del riesgo, administración del personal (asignación de tareas, rastreo del progreso, etc.), y coordinación con personas y sistemas fuera del alcance del proyecto para asegurar su liberación a tiempo y dentro del presupuesto.
g. **Ambiente.** El objetivo de esta disciplina es soportar el resto del esfuerzo asegurando que el proceso apropiado, las guías (normas y directrices), y herramientas (hardware y software) estén disponibles para cuando el equipo las necesite. [Fuente: Web [6]]

2.3.4 Fases de AUP

Las fases son implementadas de una forma serial a lo largo de un proyecto de AUP. Estas fases son:

a. Inicio

El objetivo principal de la fase de iniciación es archivar el consenso de los interesados del proyecto en relación a los objetivos del proyecto para obtener el financiamiento. Las principales actividades d esta fase incluyen:

1. **Definir el alcance del proyectó.** Esto incluye la definición, aun alto nivel, de que es lo que hará el sistema. Es igualmente importante también definir qué es lo que el sistema no va hacer. Aquí se establecen los límites desde donde el equipo

operara. Esto suele tomar la forma de una lista de características de alto nivel y/o el punto de casos de uso.

2. **Estimación de costos y calendario.** En un nivel alto, el calendario y el costo del proyecto son estimados. Estimaciones generales son realizadas en iteraciones de fases posteriores de fases posteriores, más específicamente es implementado en las fases tempranas de la Elaboración. Esto no debe interpretarse en el sentido de que todo el proyecto es planeado en este punto. Como en todas las planificaciones, estas tareas que van a ser completadas en un futuro cercano y son detalladas con más precisión y con una gran confianza mientras que las tareas bajo la línea son entidades para ser estimadas con uno que no es posible programar todo un proyecto, en su pistoletazo de salida con cualquier grado aceptable de confianza con un margen de error más grande. Esto ha sido (finalmente) reconocido por la mayoría de las industrias de que no es posible programar un proyecto completo de un solo con algún grado de aceptable desacuerdo. Lo mejor que se puede hacer es planificar para el corto plazo y precisar a largo plazo lo mejor que se pueda.

3. **Definición de riesgos.** Los riesgos del proyecto son primeramente definidos. La administración del riesgo es importante en proyectos de AUP. La lista de riesgos es una complicación en vivo que cambiara en el tiempo cuando los riesgos serán identificados, mitigados, evitados y/o materializados o exterminados. El control de riesgos del proyecto, como los riesgos de más alta prioridad, maneja la programación de las iteraciones. Los riesgos más altos, por ejemplo, son dirigidos en iteraciones más tempranas que los riesgos de menor prioridad.

4. **Determinar la factibilidad del proyecto.** Su proyecto debe tener sentido desde la perspectiva técnica, operacional y del negocio. En otras palabras, se debe ser capaz de crearlo, una vez desplegado debe ser capaz de correrlo, y debe tener un sentido económico para hacer estos aspectos. Si su proyecto no es viable, este debe ser cancelado.

5. **Preparar el entorno del proyecto.** Esto incluye la reserva de las áreas de trabajo para el equipo. Solicitar del personal que se necesitara, obtenido hardware y software que será necesitado después. Además, deberá ajustar AUP para completar las necesidades de su equipo.

Para salir de la etapa de Iniciación su equipo debe terminar el hito de Objetivos del Ciclo de Vida (LCO). El principal aspecto es hacer que el equipo entienda el alcance del proyecto y el esfuerzo requerido y como los usuarios patrocinaran el proyecto. Si el equipo pasa es hito, el proyecto sigue a las fase de Elaboración, en otra forma el proyecto deberá ser redirigido o cancelado. [Fuente: Web [6]]

Disciplina	**Actividades principales**
Modelado	✓ Inicial, modelado de requerimientos de alto nivel ✓ Inicial, modelado de la arquitectura de alto nivel
Implementación	✓ Prototipo técnico ✓ prototipo de interfaces de usuario
Pruebas	✓ Plan de prueba inicial ✓ Revisión inicial de producto de proyecto ✓ Revisión inicial de modelos
Despliegue	✓ Identificar la ventana potencial de liberación ✓ Iniciar el plan de despliegue de alto nivel
Administración de la configuración	✓ Establecer la configuración del entorno ✓ Colocar todos los productos bajo el Control de la Configuración
Administración del proyecto	✓ Inicia la creación del equipo ✓ Crear relaciones con los interesados del proyecto ✓ Determinar la factibilidad del proyecto ✓ Determinar un cronograma de alto nivel para el proyecto

	✓ Desarrollar un plan de iteraciones para las siguientes iteraciones ✓ Administrar el riesgo ✓ Obtenga el apoyo y financiamiento de los interesados ✓ Cerrar la fase
Entorno	✓ Establecer el entorno de trabajo ✓ Identificar la categoría del proyecto

Tabla 2.1. Disciplinas de la Fase de inicio AUP

[Fuente: Web [6]]

b. Elaboración

El principal objetivo de la fase de elaboración es probar la arquitectura del sistema a desarrollar. El punto es asegurar que el equipo puede desarrollar un sistema que satisfaga los requisitos, y la menor manera de hacerlo que es la construcción de extremo o extremo del esqueleto de trabajo del sistema conocido como "prototipo de la arquitectura". Esto es en realidad un concepto pobre porque mucha gente piensa es deshacerse de los prototipos. En cambio, su significado es software funcional de alto nivel, el cual incluye varios casos de uso de alto de riesgos (a partir de un punto de vista técnico) para demostrar que el sistema es técnicamente factible. [Fuente: Web [6]]

Disciplina	**Actividades principales**
Modelado	✓ Identificar los riesgos técnicos ✓ Modelado de la Arquitectura ✓ Prototipo de Interfaces de Usuario
Implementación	✓ Probar la arquitectura
Pruebas	✓ Validar la arquitectura ✓ Evolucionar el modelo de pruebas

Despliegue	✓ Actualizar su plan de desarrollo
Administración de la configuración	✓ Poner todos los productos bajo el Control de Administración de la Configuración (CM control).
Administración del proyecto	✓ Construir el equipo ✓ Proteja el equipo ✓ Obtenga los recursos ✓ Maneje el riesgo ✓ Actualice su plan de proyecto ✓ Cierre la fase
Entorno	✓ Evolucione el entorno de trabajo ✓ Ajuste los materiales de procesos

Tabla 2.2. Disciplinas de la Elaboración

[Fuente: Web [6]]

c. Construcción

El objetivo de la fase de Construcción consiste en desarrollar el sistema hasta el punto en que está listo para la pre-producción de pruebas. En las etapas anteriores, la mayoría de los requisitos han sido identificados y la arquitectura del sistema se ha establecido. El énfasis es priorizar y comprender los requerimientos, modelado que ataca una solución y, a luego, la codificación y las pruebas del software. Si es necesario, las primerias versiones del sistema se desarrollan, ya sea interna o externamente, para obtener los comentarios de los usuarios.

Para salir de la fase de Construcción su equipo debe pasar el hito de la Capacidad Operativa Inicial (IOC). Si el equipo pasa esta etapa el proyecto pasa de la fase de Transición, de lo contrario puede ser re-dirigido o cancelado. [Fuente: Web [6]]

Disciplina	**Actividades principales**
Modelado	✓ Abordaje del análisis el Modelado ✓ Diseño por modelado de lluvia de ideas ✓ Documente
Implementación	✓ Primeras pruebas ✓ Crear constantemente ✓ Evolución de la lógica de dominio ✓ Evolucionar las interfaces de usuario ✓ Evolucionar el esquema de datos ✓ Desarrollo de interfaces de activos legados ✓ Generar el script de conversión de datos
Pruebas	✓ Prueba de software ✓ Evolucionar su modelo de pruebas
Despliegue	✓ Desplegar el script de instalación ✓ Desplegar Notas publicadas ✓ Desplegar documentación inicial ✓ Actualizar su plan ✓ Desplegar el sistema en un ambiente de pre-producción
Administración de la configuración	✓ Poner todos los productos bajo el Control CM control
Administración del proyecto	✓ Administré el equipo del proyecto ✓ Manejo del riesgo ✓ Actualizar su plan de proyecto ✓ Cerrar esta fase
Entorno	✓ Apoyar al equipo ✓ Evolucionar el entorno de trabajo ✓ Establecer el ambiente de capacitación

Tabla 2.3. Disciplinas de la Fase de Construcción

[Fuente: Web [6]]

d. Transición

La fase de Transición se enfoca en liberar el sistema a producción. Deben hacerse pruebas extensivas a los largo de esta fase, incluyendo las pruebas beta.

Para finalizar la fase de Transición su equipo debe pasar el hito de Liberación del Producto (PR). El principal problema aquí es el sistema pude ser desplegado segura y eficientemente en producción. Si el equipo pasa ese hito el proyecto se mueve a producción. Si el proyecto fracasa en alguna de las áreas de arriba, el proyecto podría ser redirigido o cancelado (algunos proyectos son tan desastrosos que no querrá ni siquiera instalarlos). [Fuente: Web [6]]

Disciplina	**Actividades principales**
Modelado	✓ Modelado de lluvia de ideas ✓ Finalice la documentación general del sistema
Implementación	✓ Corrija defectos
Pruebas	✓ Valide el sistema ✓ Valide la documentación ✓ Finalice su modelo de pruebas
Despliegue	✓ Finalice el paquete de entrega o liberación ✓ Finalice la documentación ✓ Anuncie el despliegue o liberación ✓ Capacite al personal ✓ Libere el sistema en producción
Administración de la configuración	✓ Poner todos los productos bajo el CM control
Administración del proyecto	✓ Administre el equipo de proyecto ✓ Cerrar la fase ✓ Inicie el próximo ciclo del proyecto
Entorno	✓ Establezca las operaciones y/o el ambiente soporte ✓ Recupere las licencias del software

Tabla 2.4. Disciplinas de la Fase de Transición

[Fuente: Web [6]]

2.3.5 Los Hitos del AUP

Como puede ver en la Figura 2.1, Hay cuatro hitos en AUP. En cada uno de estos hitos, los cuales señalan el final de la fase, usted debería considerar tener una "revisión de hitos" que verifique que su equipo de trabajo está cumpliendo satisfactoriamente con los criterios de hitos. Los cuatro hitos son: [Fuente: Web [6.1]]

- **Hito de la Fase de Inicio: Objetivos del Ciclo de Vida (LCO, por sus siglas en inglés)**

  En éste hito, los involucrados evalúan el estado del proyecto. Debe estar de acuerdo en lo siguiente:

  Acuerdo del Alcance. Los interesados llegan a un acuerdo sobre el alcance del proyecto.

  Definición Inicial de Requerimientos. Existe un acuerdo en que el conjunto correcto de requisitos han sido capturados, en un nivel alto, y hay un entendimiento común de esos requisitos.

  Acuerdo del Plan. Los involucrados llegan a un acuerdo con el costo inicial y la estimación del cronograma.

  Aceptación del Riesgo. El riesgo ha sido identificado, evaluado y se han abordado estrategias aceptables para controlarlos.

  Aceptación de Proceso. La Metodología de Proceso Unificado Ágil (AUP, por sus siglas en inglés) ha sido inicialmente adoptada y aceptada por todas las partes.

  Viabilidad. El proyecto tiene sentido desde la perspectiva técnica, operacional y del negocio.

  Plan del Proyecto. Existen adecuados planes para la siguiente fase (Elaboración).

  Cumplimiento del Portafolio. ¿El alcance del proyecto encaja bien en su organización general del portafolio de proyectos?

- **Hito de la Fase de Elaboración: Arquitectura del Ciclo de Vida (LCA, por sus siglas en inglés)**

  En este hito, los involucrados evalúan el estado del proyecto. Ellos deben estar de acuerdo en la siguiente:

  Estabilidad de la visión. La visión del proyecto ha sido estabilizada y es realista.

  Estabilidad de la arquitectura. Esté de acuerdo en que la arquitectura está estable y es suficiente para satisfacer los requerimientos. La arquitectura ha sido prototipada apropiadamente para ser direccionada con los riesgos de la arquitectura principales.

  Aceptación del riesgo. El riesgo ha sido evaluado para asegurar que ha sido apropiadamente entendido, documentado y que se han desarrollado estrategias para manejarlo como aceptable.

  Viabilidad. El proyecto aún tiene sentido desde la perspectiva técnica, operacional y del negocio.

  Plan del Proyecto. Plan de iteración detallado para las próximas iteraciones de la etapa de Construcción, así como un plan de proyecto de alto nivel ya elaborado.

  Cumplimiento de la organización. ¿La arquitectura del sistema refleja las realidades de la arquitectura de la empresa?

- **Hito de la fase de Construcción: Capacidad Operacional Inicial (IOC, por sus siglas en inglés)**

  En este hito, los involucrados del proyecto deben estar de acuerdo en:

  Estabilidad del Sistema. El software y la documentación de soporte son aceptables (estable y madura) para implementar el sistema a los usuarios.

  Involucrados preparados. Los involucrados (y el negocio) están listos para que el sistema sea implementado (aunque aún necesiten entrenamiento).

  Aceptación del riesgo. El riesgo ha sido evaluado para asegurar que ha sido apropiadamente entendido, documentado y que se han desarrollado estrategias para manejarlo como aceptable.

  Aceptación y estimación del costo. Los gastos son aceptables y las estimaciones razonables han sido calculadas y programadas para los costos futuros.

  Plan del proyecto. Plan de iteración detallado para las próximas iteraciones de la etapa de Transición, así como un plan de proyecto de alto nivel ya elaborado.

Cumplimiento de la organización. ¿El producto elaborado por el equipo cumple con los estándares apropiados de la organización?

- **Hito de la Fase de Transición: Liberación del Producto (PR, por sus siglas en inglés)**

  En este hito, los involucrados del proyecto deben estar de acuerdo en:

  Aceptación de los involucrados del negocio. Los involucrados del negocio están satisfechos con el sistema y lo aceptan.

  Operaciones de aceptación. Las personas se responsabilizan de operar el sistema una vez que este está en producción y están satisfechos con los procedimientos y documentación relevantes.

  Aceptación del soporte. Las personas se responsabilizan del soporte del sistema una vez que este está en producción y están satisfechos con los procedimientos y documentación relevantes.

  Aceptación del costo estimado. Los gastos actuales son aceptados, y las estimaciones razonables han sido hechas parar los costos futuros de producción.

2.4 HERRAMIENTAS DE DISEÑO

2.4.1 UWE (UML-Base Web Engineering)

UWE es un enfoque de ingeniería de software para el dominio web con el objetivo de cubrir todo el ciclo de vida de desarrollo de aplicaciones Web. El aspecto clave que distinguen a UWE es la confianza en los estándares. Presentado por la Dra. Nora Koch, para el desarrollo de aplicaciones Web, está fundada en un entorno Orientado a Objetivos utilizando para esto la notación "ligera" de UML UWE proporciona guías para la construcción de modelos de forma sistemática, enfocándose en la personalización y el estudio de casos de uso [Fuente: Web [7]]

2.4.2 Acerca de UWE

UWE (UML- Base Web Engineering, Ingeniería Web Basada en UML) es un método de ingeniería del software para el desarrollo de aplicaciones web en UML. Cualquier tipo de

diagrama UML puede ser usado en el desarrollo de UWE, ya es extensión de UML. Las actividades de modelado principales son el análisis de requerimientos, el diseño conceptual, el diseño de navegación y el diseño de presentación, y producen los siguientes modelos [Fuente: Web [7]]

1. Modelo de casos de uso: para capturar los requisitos del sistema.
2. Modelo conceptual: para el contenido (modelo del dominio).
3. Modelo de usuario: modelo de navegación que incluye modelos estáticos y dinámicos.
4. Modelo de estructura de presentación, modelo de flujo de presentación.
5. Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.
6. Modelo de adaptación.

El lenguaje de Modelo Unificado UML (Unified Modeling Language) es una herramienta que cubre todos los requerimientos que surgen cuando se modela una aplicación Web. UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

2.4.3 Objetivos de UWE

El objetivo principal del enfoque UWE es proporcionar un UML de dominio, basado en el lenguaje de modelado especifico, metodología basada en modelos, herramienta de apoyo para el diseño sistemático y herramienta de apoyo para la generación semiautomática de aplicaciones Web.

La notación UWE se define como “ligero”, extensión del Lenguaje Unificado de Modelado (UML) proveer lo que se llama perfil UML para el dominio Web

Por “ligero” se entiende que puede ser fácilmente adaptada a otras herramientas de modelado y que no significa un gran impacto en el intercambio de formatos. Los estereotipos son un nuevo tipo de elementos de modelado definidos dentro del modelo basado en un tipo de elementos de un modelo existente. Un valor etiquetado es un par (etiqueta), valor (que permite adjuntar información arbitraria a cualquier elemento de modelado). Una restricción es una condición que permite especificar nuevas semánticas para un elemento de modelado. [Fuente: Web [7]]

UWE proporciona soporte de herramientas para el diseño de modelos, comprobaciones en el modelo de consistencia y semiautomático de generación de sistemas Web. ArgoUWE (ArgoUML) y MagicUWE (MagicDraw) son complementos que apoyan a la anotación del perfil de UWE. [Fuente: Web [7]]

2.4.4 Modelo de Casos de Uso

Para describir los requerimientos funcionales de una aplicación se puede usar un modelo de casos de uso. Este modelo describe un trozo de comportamiento de la aplicación sin revelar su estructura interna.

El modelo de casos uso está conformado por dos elementos de modelado principales, llamados casos de uso y actores. Un caso de uso es una unidad coherente de funcionalidad provista de aplicaciones que interactúan como uno o más actores eternos de la aplicación. Un actor, es el rol que un usuario puede desempeñar con respecto a un sistema o una entidad, tales como otro sistema o una base de datos. Además, existen relaciones entre estos dos elementos, tales como asociaciones entre actores y casos de uso y las dependencias “includes” y “extendes” entre casos de uso. [Fuente: Web [7.1]]

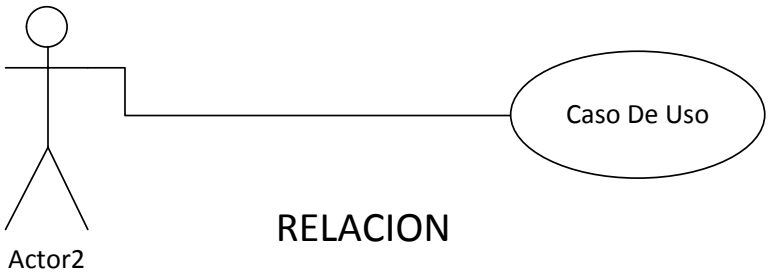

Figura 2.2. Elementos en Caso de Uso

[Fuente: Web [7.1]]

2.4.5 Representación del Modelo Conceptual

Se trata de obtener el esquema conceptual de la base de datos a partir de la lista descriptiva de objetos y asociaciones identificadas en la organización durante el análisis.

El Modelador debe asegurar la representación formal de los fenómenos; es decir, realizar su Modelización. Esta Modelización debe conservar la semántica de lo real expresado en la lista y descripción de los objetos y asociaciones y traducirla en forma no redundante.

Los principales elementos usados para el modelo conceptual son las clases y asociaciones. Sin embargo el poder del diagrama de clases es dado por una variedad de características adicionales que pueden ser usadas. Entre estas característica están los nombres de asociación y los nombres de roles de asociación, la cardinalidad, diferentes formas de asociaciones soportadas por UML como agregación, herencia composición y la clase asociación, toda estas representaciones gráficamente utilizando notación de UML.

Una clase es descrita por un nombre, atributos y operaciones que actúan sobre los atributos. [Fuente: Web [7.1]]

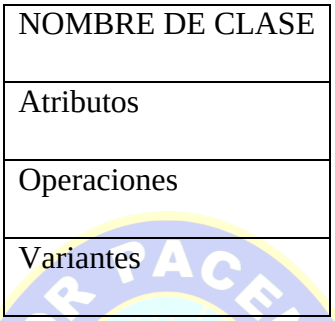

Figura 2.3. Clase de un Modelo Conceptual

[Fuente: Web [7.1]]

2.4.6 Modelo de Navegación

El diseño de navegación es un paso crítico en el diseño de la aplicación Web. El modelo de navegación se comprime en el modelo de espacio de navegación y el modelo de estructura de navegación. El primero especifica que objetos son alcanzados. [Fuente: Web [7.1]]

a. Modelo de espacio de Navegación

El modelo de espacio de navegación es construido con las clases de navegación y las asociaciones de navegación y están representadas gráficamente por un diagrama de clases de UML [Fuente: Web [7.1]]

La clase de navegación modela una clase cuyas instancias son visitadas por usuarios durante la navegación, se les asigna el nombre que diera a las correspondientes clases conceptuales. Sin embargo se diferencia de esta por el estereotipo <<navigation class>>. Además, una clase de navegación puede contener atributos de otras clases del modelo conceptual, siempre que la clase de navegación tenga alguna asociación con la clase de la que se presta el o los atributos. Para diferenciar dichos atributos se coloca una barra inclinada a la derecha (/) antes del nombre.

La navegación directa es representada por asociaciones en el modelo de espacio de navegación que provienen de la clase de navegación de origen. Por lo tanto, sus semánticas son diferentes de las asociaciones usadas en el modelo conceptual. Estas

asociaciones son interpretadas como el enlace o vínculo entre la clase de navegación inicial (página Web de inicio) y la clase de navegación final (página Web destino). Es estereotipo utilizado para identificar a esta asociación es <<direct navigability>>

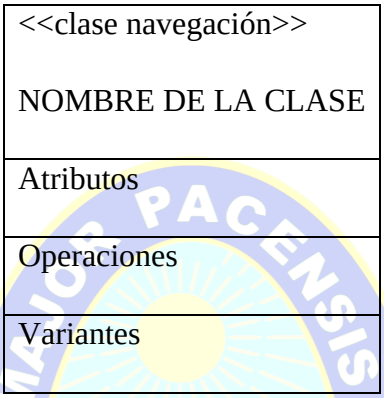

Figura 2.4. Clase de Navegación

[Fuente: Web [7.1]]

b. Modelo de Estructura de Navegación

El modelo de estructura de navegación describe como la navegación es soportada por elementos de acceso tales como índices, visitas guiadas, preguntas y menús. El resultado es un diagrama de clases UML construido con estereotipos, los cuales están definidos según mecanismos de extensión UML. Las primitivas de acceso son nodos de navegación adicionales requerida para acceder a objetos de navegación. Las siguientes primitivas de acceso son definidas como estereotipos UML: índices, visitas guiadas, consultas y menús [Fuente: Web [7]]

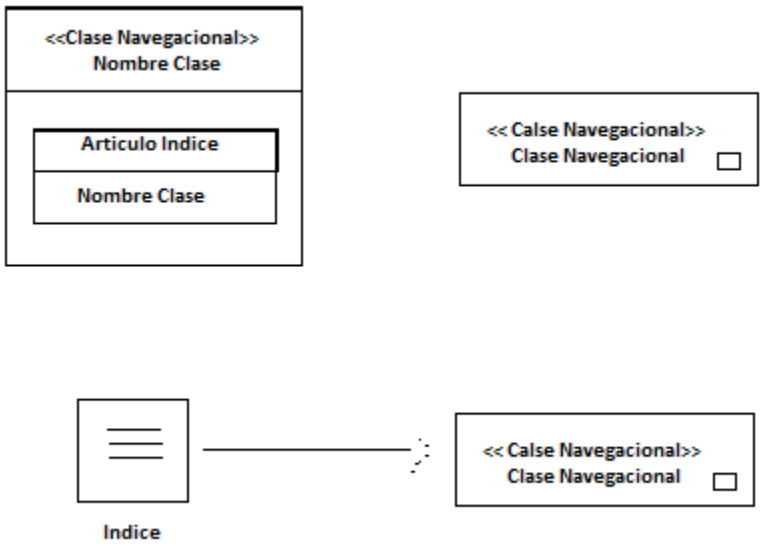

Figura 2.5. Clase de Navegación, Notación largo y corta

[Fuente: Web [7]]

Los índices permiten el acceso directo a las instancias de la clase de navegación. Cualquier índice miembro de la clase índice y utiliza el estereotipo <<index>> con su icono correspondiente.

Las visitas guiadas proveen acceso secuencial a las instancias de una clase navegación. Para clases contienen objetos de visita guiada se usa el estereotipo <<guiedTour>> y su icono correspondiente. Las visitas guiadas deben ser controladas por el usuario o por el sistema.

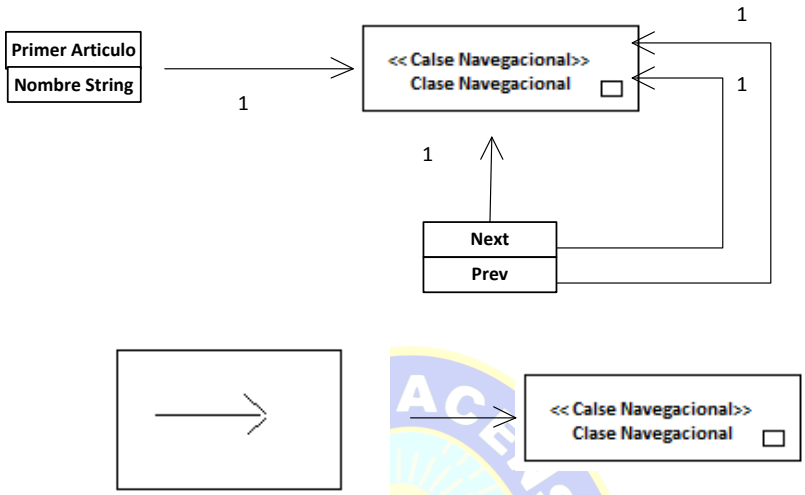

Figura 2.6. Clase Visita Guiada y Visita Guiada en Notificación Corta

[Fuente: Web [7.1]]

Una consulta es modelada por una clase que tiene una serie de preguntas como atributo. Para la clase consulta se utiliza el estereotipo <<query>> y su icono corresponde.

Un menú es una primitiva de acceso adicional, es un índice de un conjunto de elementos heterogéneos, tales como índices, visitas guiadas, consultas, una instancia de una clase navegación u otro menú. Este es modelado por un objeto compuesto que contiene un número fijado de ítems de menú. Cada ítem de menú tiene un nombre constante y posee un enlace, ya sea a una instancia de una clase de navegación o a un elemento de acceso. Cualquier menú es una instancia de alguna clase que es estereotipada por <<menú>> con su icono correspondiente.

2.4.7 Modelo de Presentación

El diseño de presentación soporta la construcción de un modelo de presentación basado en el modelo de estructura de navegación e información adicional que se recolecta durante el análisis de requerimientos. El modelo de presentación consiste en un conjunto de vistas que muestran el contenido y la estructura de los nodos simples, es decir como cada nodo es presentado al usuario y como el usuario puede interactuar con ellos.

Las vistas de interfaz de usuario especifican que cada instancia de esta clase es un nodo de todos los elementos abstractos de interfaces de usuario los cuales están presentados simultáneamente al usuario.

Para las vistas de interfaz de usuario el estereotipo <<UIview>> y su respectivo icono

La clase presentación es una unidad estructural que permite particionar una vista de interfaz de usuario dentro de grupos de elementos de interfaz de usuario. Para la clase presentación se utiliza el estereotipo <<presentation class>>.

El elemento de interfaz de usuario des una clase de abstracción que tiene varias especializaciones describiendo elementos de interfaz particulares. [Fuente: Web [7.1]]

2.5 TECNOLOGÍA DE IMPLEMENTACIÓN

2.5.1. Workflow

El workflow se define como un sistema informático que organiza y controla tareas, recursos y reglas, necesarias para completar el proceso de negocio.

Su importancia en el proceso de reingeniería del negocio en las empresas, lo convierte en una herramienta básica de la agilización y descentralización de actividades administrativas y comerciales, impuestas por las nuevas tendencias que regulan las organizaciones. [Fuente: Web [8]]

- Gestión de riesgos.
- Atención al cliente.
- Ciclo de producción.
- Provisión de servicios.
- Control de calidad (ISO).
- Tratamiento de expedientes.

a. Beneficios

La implantación de un sistema de Workflow aporta numerosos beneficios dependiendo de los procesos de negocio involucrados:

- Ahorro de tiempo y mejora de la productividad.
- Mejora del control de procesos.
- Mejor atención y servicio al cliente.
- Establecimiento de mecanismos de continua mejora en los procesos.
- Optimizar la circulación de información interna con clientes y proveedores.
- Integración total de los procesos empresariales. [Fuente: Web [8]]

b. Integración

Una característica fundamental del sistema Workflow es la correcta integración con los sistemas de información actuales, como Bases de Datos, Gestión Documental, Mensajería, ERP, GroupWare, Call Centers, Mainframe, etc ... Existen firmas especializadas de sistemas que llevan años integrando soluciones basadas en esta tecnología, garantizando la correcta ejecución de cualquier proyecto.

c. Productos

Existen en el mercado gran cantidad de herramientas de Workflow

- Workflow Corporativo
- Workflow de Aplicación
- Workflow Documental
- Workflow de Producción

Workflow se refiere al flujo de trabajo a seguir para la consecución de una tarea o trabajo predeterminado. Se defina como un sistema de secuencia de tareas de un proceso de negocio. [Elaboración: Propia (Argumento Propio)]

2.6 RECURSOS TÉCNICOS

2.6.1 XAMPP

XAMPP es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. De hecho su nombre viene de

hay, X (para cualquier sistema operativo), A (Apache), M (MySQL), P (PHP) y P (Perl). XAMPP es independiente de plataforma y tiene licencia GNU GPL. Existen versiones para Linux (testeado para SuSE, RedHat, Mandrake y Debian), Windows (Windows 98, NT, 2000, XP y Vista), MacOS X y Solaris (desarrollada y probada con Solaris 8, probada con Solaris 9).

Una de las ventajas de XAMPP es que de una forma muy sencilla y rápida (no más de 5 minutos) te puedes montar en tu máquina un entorno de desarrollo de cualquier aplicación web que use PHP y base de datos. La configuración por defecto de XAMPP tiene algunas deficiencias de seguridad por lo que no es recomendable usarla como una herramienta para producción, sin embargo con algunas modificaciones es lo suficientemente seguro para ser usada como servidor de sitios web en internet. Desde LAMPP (Linux AMPP) sí que podrá hacer una instalación segura haciendo "/opt/lampp/lampp security".[Fuente: Web [9]]

2.6.1.1 CARACTERÍSTICAS Y REQUISITOS

XAMPP solamente requiere descargar y ejecutar un archivo ZIP, tar, exe o fkl, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado. Puede encontrarse tanto en versión completa, así como en una versión más ligera que es portátil.

Oficialmente, los diseñadores de XAMPP, fueron los Baiker y Anthony Coorporation los cuales solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica, sin embargo, XAMPP es utilizado actualmente como servidor de sitios Web, ya que, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes. [Fuente: Web [10]]

2.6.2 MYSQL

El sistema de base de datos operacional MySQL es hoy en día uno de los más importantes en lo que hace al diseño y programación de base de datos de tipo relacional. Cuenta con millones de aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo.

La historia del MySQL (cuya sigla en inglés se traslada a My Structured Query Language o Lenguaje de Consulta Estructurado) se remite a principios de la década de 1980. Programadores de IBM lo desarrollaron para contar con un código de programación que permitiera generar múltiples y extendidas bases de datos para empresas y organizaciones de diferente tipo. Desde esta época numerosas versiones han surgido y muchas de ellas fueron de gran importancia. Hoy en día MySQL es desarrollado por la empresa Sun Mycrosystems.

Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. Por otro lado, MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores. Las plataformas que utiliza son de variado tipo y entre ellas podemos mencionar LAMP, MAMP, SAMP, BAMP y WAMP (aplicables a Mac, Windows, Linux, BSD, Open Solaris, Perl y Phyton entre otras).

Se están estudiando y desarrollando nuevas versiones de MySQL que buscan presentar mejoras y avances para permitir un mejor desempeño en toda aquella actividad que requiera el uso de bases de datos relacionales. Entre estas mejoras podemos mencionar un nuevo dispositivo de depósito y almacenamiento, backup para todos los tipos de almacenamientos, replicación segura, planificación de eventos y otras más. [Fuente: Web [11]]

2.6.3 Lenguaje de Programación PHP

El lenguaje de programación PHP Hypertext Pre-processor, fue desarrollado puntualmente para diseñar páginas web dinámicas programando scripts del lado del servidor.

El lenguaje PHP siempre va incrustado dentro del HTML y generalmente se le relaciona con el uso de servidores linux.

Originalmente diseñado por el programador danés-canadiense Rasmus Lerdorf, en el año 1994 en base a la escritura de un grupo de CGI binarios escritos en el lenguaje C. En un comienzo, PHP sólo estaba compuesto por algunas macros que permitían trabajar más fácilmente en la creación de páginas web. En el año de 1995 Rasmus Lerdorf le añadió el analizador sintáctico y se llamó PHP/F1 Versión 2, sólo reconocía texto HTML y algunas directivas de MySQL. Después de esta fecha la contribución al código fue pública.

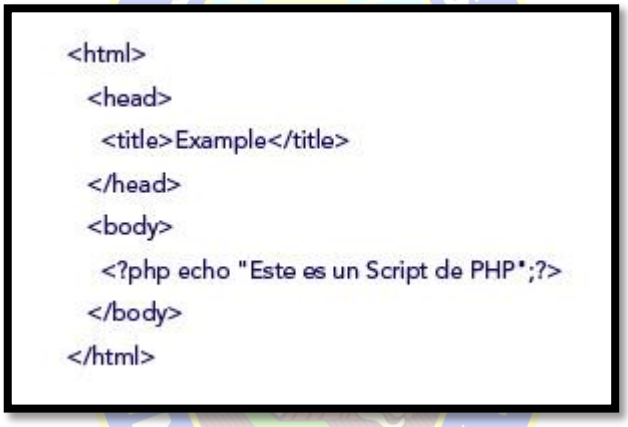

Figura 2.7 Script de Código

[Fuente: Web [12]]

PHP se caracteriza por ser un lenguaje gratuito y multiplataforma. Además de su posibilidad de acceso a muchos tipos de bases de datos, también es importante destacar su capacidad de crear páginas dinámicas, así como la posibilidad de separar el diseño del contenido de una web.

PHP es la solución para la construcción de Webs con independencia de la Base de Datos y del servidor Web, válido para cualquier plataforma.

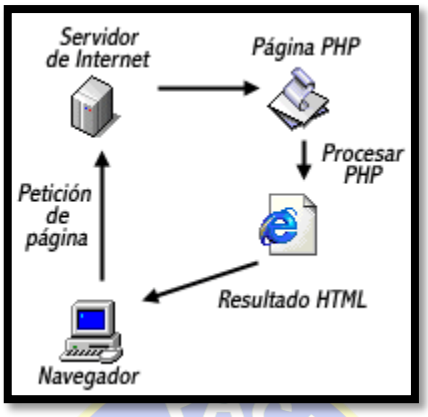

Figura 2.8 Proceso de Integración

[Fuente: Web [12]]

El objetivo final es conseguir la integración de las paginas HTML con aplicaciones que corran en el servidor como procesos integrados en el mismo, y no como un proceso separado, como ocurría con los CGIs (Common Gateway Interface).

El lenguaje php presenta cuatro grandes características:

1.- Velocidad: PHP no solo es rápido al ser ejecutado sino que no genera retrasos en la máquina, por esto no requiere grandes recursos del sistema. PHP se integra muy bien junto a otras aplicaciones, especialmente bajo ambientes Unix.

2.- Estabilidad: PHP utiliza su propio sistema de administración de recursos y posee de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

3.- Seguridad: PHP maneja distintos niveles de seguridad, estos pueden ser configurados desde el archivo **.ini**

4.- Simplicidad: Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente. Además PHP dispone de una amplia gama de librerías, y permite la posibilidad de agregarle extensiones. Esto le permite su aplicación en múltiples áreas, tales como encriptado, gráficos, XML y otras.

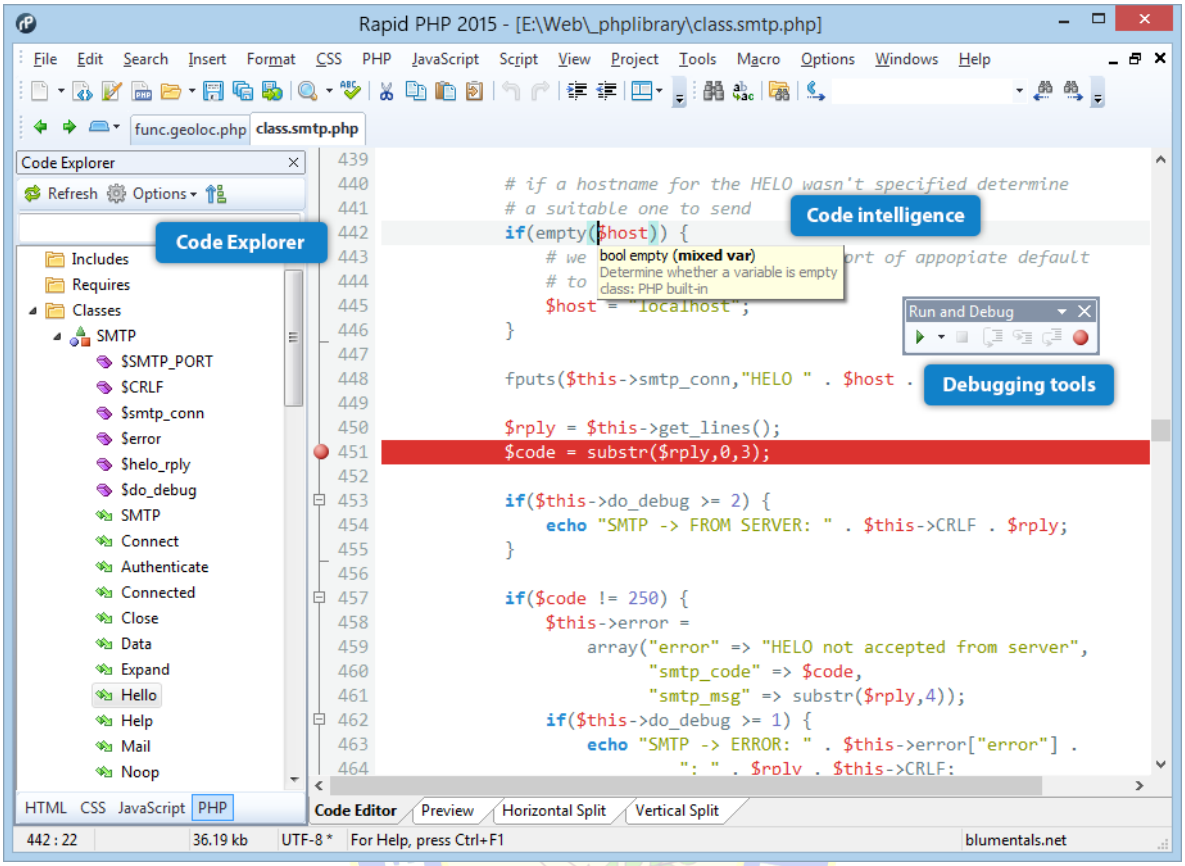

Figura 2.9 Script de Código y Captura

[Fuente: Web [12]]

Ventajas adicionales de PHP

1.- PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente,

2.- La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP.

3.- PHP es completamente expandible y modificable. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código.

4.- Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen yTHTTPD. Otra alternativa es configurarlo como módulo CGI.

5.- Permite la interacción con gran cantidad de motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, etc.

6.- PHP es Open Source, (código abierto) esto significa que no depende de ninguna compañía comercial y que no requiere de licencias. [Fuente: Web [12]]

2.6.4 Lenguaje de Programación JAVASCRIPT

JavaScript es un lenguaje compacto, y basado en objetos, diseñado para el desarrollo de aplicaciones cliente-servidor a través de Internet. Netscape Navigator 2.0 es capaz de interpretar sentencias JavaScript embebidas en programas CGI (Script del lado de servidor).

En una aplicación cliente para un browser, las sentencias JavaScript **embebidas** en un documento HTML pueden **reconocer y responder a eventos** generados por el usuario, como clicks del mouse, información en formularios y navegación de documento a documento.

Por ejemplo, se puede escribir una función JavaScript que verifique que la información ingresada por el usuario sea correcta. Sin que haya transmisión de datos por la red.

Un documento HTML con JavaScript embebido es capaz de interpretar la información ingresada por el usuario, verificar que sea correcta y alertar al usuario en caso que no lo sea. Se puede también ejecutar archivos de audio, applet's o comunicar con una extensión de Netscape (plug-in's) en respuesta a la apertura o cierre de una página.

JavaScript y Java. El lenguaje JavaScript imita a Java, pero sin verificación estática de tipos ni restricciones fuertes en tiempo de ejecución. JavaScript soporta la mayoría de las expresiones y sintaxis válidas en Java, así como las sentencias de control de flujo de Java. En contraste con el sistema de clases definidas en tiempo de compilación en Java, Javascript trabaja en base a un simple sistema en tiempo de ejecución, el cual permite el uso de tipos de datos básicos como enteros, números de punto flotante, booleanos y strings. JavaScript soporta un sistema simple de objetos basado en instancias (no en clases), que provee capacidades significativas.

Además, JavaScript **soporta funciones**, pero de nuevo sin ningún requerimiento especial al declararlas. Estas funciones pueden ser *propiedades de objetos* (aunque el usuario no puede definir clases), y se ejecutan como métodos bastante flexibles debido a que no se verifican los tipos.

JavaScript **complementa a Java** al exponer muchas características de las applets en Java a los autores de HTML. JavaScript puede además obtener y establecer propiedades de las applets, esto es, sólo aquellas que alteran el estado de la applet o plug-in.

Java es un lenguaje de extensión diseñado, en particular, para ejecución rápida y seguridad de tipos, lo que se ve reflejado en la imposibilidad de transformar un entero en una referencia, por ejemplo.

Los programas en Java consisten exclusívamente de clases y sus métodos, lo que, junto a los requerimientos de declaraciones de clases, métodos y tipos hacen que la programación sea más compleja que en JavaScript. Otro punto imortante es que la herencia que provee Java crea jerarquías muy acopladas de objetos.

En contraste, JavaScript posee el espíritu de un lenguaje pequeño y verificado dinámicamente. Estos lenguajes ofrecen herramientas de programación a una audiencia mayor, ya que son más fáciles de utilizar, y proveen de una interface más especializada. Como consecuecia, necesitan sólo de un mínimo de requerimientos para creación de objetos. [Fuente: Web [13]]

La siguiente tabla compara y contrasta JavaScript con Java:

JavaScript	Java
• Interpretado en el Cliente.	• Compilado en el Servidor antes de ejecución en el Cliente.
• Basado en Objetos. El código usa objetos incorporados del sistema, pero no provee creación de clases o herencia.	• Orientado a Objetos. El código consiste de clases con herencia, permitiendo crear objetos y crear jerarquías.
• Código embedido en HTML.	• Applets referenciadas desde HTML.
• Variables y tipos de datos no se declaran.	• Variables y tipos de datos se declaran.
• Enlazado dinámico. Referencias a objetos se validan en tiempo de ejecución.	• Enlazado estático. Referencias a objetos se validan en tiempo de compilación.

Tabla 2.5 Tabla de Comparaciones

[Fuente: Web [13]]

2.7 SEGURIDAD

En la actualidad la seguridad es uno de los aspectos más relevantes del desarrollo de sistemas, sobre todo en ambiente web, miles de personas por hora pueden visitar tu sitio dependiendo de su popularidad, y todas ellas pueden tener acceso a información que quizás tu no quieras compartir.

Está claro que si tienes información de carácter confidencial no es buena idea mantenerla en el mismo servidor web que el de tu aplicación, pero si se trata de alguna información que debemos compartir con nuestros usuarios pero de forma limitada debemos tomar medidas.

Existen muchas personas que se plantean y preguntan si realmente es necesaria la seguridad en un sitio web, muchos de nosotros en algunos casos tomamos este aspecto como algo irrelevante, y lo dejamos a lo último, o incluso no lo tomamos en cuenta.

Es claro que no todos los datos son blancos de protección, la seguridad es algo que se debe aplicar a la medida, para la información que lo amerite y de la forma adecuada, no es necesario invertir una gran cantidad de horas frente al monitor planeando y codificando formas de proteger nuestro sitio, simplemente basta con saber qué información proteger y de qué forma, y que mejor que esto sea automatizado. [Fuente: Web [14]]

Para ayudar a proteger esta información que desplegamos, o para restringir el uso de alguno de nuestros sistemas de inserción o eliminación de datos, podemos utilizar lo que se conoce como "manejo de sesiones", que en PHP resulta ser una tarea bastante sencilla de realizar.

- Seguridad en el cliente. Código móvil
- Seguridad en el servidor. Servidor web, servidor de bases de datos, lenguajes de servidor
- Seguridad en la aplicación. Control de acceso, Validación de datos de entrada, Programación segura
- Seguridad en la comunicación. Certificados digitales, SSL

Primera recomendación: Disponer siempre de versiones actualizadas de Apache y PHP, Aspectos de PHP que pueden dar lugar a vulnerabilidades: Variables globales – Nombres de ficheros – Subida de ficheros – Bibliotecas – Datos enviados desde formularios.

Otra recomendación: Manejo de sesiones seguras, existen varios aspectos complejos en la gestión de sesiones, pero todo esto se empieza a simplificar si centralizamos en un único módulo todo el código encargado de esta gestión. Una vez definido el fichero que hará de repositorio central de las funciones de gestión de sesiones, es necesario incluir dicho módulo en todas nuestras páginas web.

Dependiendo de nuestro framework de desarrollo, la inclusión de dichas funciones las realizaremos a través de métodos “autoload”, a través de require() / include(), o de otros mecanismos habilitados para la carga de módulos.

Nuestro código debe garantizar que se inicialice una sesión en cada petición HTTP . Vamos a usar para este ejemplo el fichero security.php que será cargado por todos los módulos de la aplicación web a través de los métodos mencionados en el parrafo anterior.

```
/* Inicializamos la sesion*/
session_start();
```

Si queremos asegurarnos de la privacidad de nuestras páginas, y que estas indiquen que no deben ser cacheadas por ningun proxy intermedio, podemos incluir las siguientes directivas.

```
/* Establecemos que las paginas no pueden ser cacheadas */
header("Expires: Tue, 01 Jul 2001 06:00:00 GMT");
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");
header("Cache-Control: no-store, no-cache, must-revalidate");
header("Cache-Control: post-check=0, pre-check=0", false);
header("Pragma: no-cache");
```

En este punto, en el que la sesión del usuario ha sido inicializada en el servidor, y que la información de la sesión ha sido transmitida al usuario a través de una cookie, debemos definir en algún lado de nuestra aplicación una función que se encargue de destruir la sesión del usuario. Esta función deberá ser llamada desde varios sitios, siendo el fundamental la página logout.php de nuestra aplicación (o el método utilizado para desconectar al usuario), así como la página de login.php, ya que consideraremos que cualquier usuario que acceda a esta página desea iniciar sesión en el software, y debemos invalidar sus credenciales anteriores.

```
function logOut() {
session_unset();
session_destroy();
session_start();
session_regenerate_id(true);
}
```

Para destruir una sesión en php debemos borrar todas las variables del array global $_SESSION mediante una llamada a session_unset(). La llamada a session_destroy() eliminará toda la información asociada con la sesión en el servidor.Trás inutilizar la sesion anterior, vamos a crear una nueva sesión con un identificador nuevo con session_regenerate_id(). Estos pasos nos garantizarán que la sesión anterior ha quedado completamente invalidada.

Un ataque de session fixation es aquel en el que un usuario es capaz de robar la sesión de otro usuario y suplantar su identidad. Para protegernos de dichos ataques un primer enfoque que debemos realizar nada mas inicializar la sesión del usuario es guardar ciertas variables de sesión con información del cliente:

```
$_SESSION['userAgent'] = $_SERVER['HTTP_USER_AGENT'];
$_SESSION['SKey'] = uniqid(mt_rand(), true);
$_SESSION['IPaddress'] = ExtractUserIpAddress();
$_SESSION['LastActivity'] = $_SERVER['REQUEST_TIME'];
```

En nuestro caso, los dos campos más importantes son la IP del usuario que realiza la petición HTTP, y la versión de su navegador. Por temas de control de actividad también guardaremos la fecha en la que se realiza la petición al servidor web.

Para protegernos de dichos ataques lo que debemos comprobar en cada petición es que no existan modificaciones en los parámetros del navegador (IP, navegador), ya que esto significaría que alguien desde una ubicación distinta a la anterior del usuario está intentando hacerse pasar por él. La fecha de la última actividad no debe superar un umbral definido por nosotros, como por ejemplo 120 minutos. En caso de que suceda un cambio en alguno de los

mencionados parámetros, o que la sesión del usuario haya caducado en el acceso al servidor web, debemos llamar inmediatamente al método logOut() y salir con un exit();

El acceso desde dispositivos móviles puede trastocar los planes de un control exhaustivo de la dirección IP. Un planteamiento alternativo puede ser ignorar el cambio de ip si se ha producido desde un dispositivo móvil, identificable por el User Agent del navegador. En el futuro hablaremos de un control adicional basado en geolocalización y de reglas heurísticas adicionales para determinar si el usuario que se encuentra al otro lado de la conexión es realmente quien se suponía que era al principio.

Ya nos hemos quitado un gran número de problemas para evitar molestos ataques contra la sesión del usuario así que vamos con una última recomendación adicional para proteger una aplicación web frente a ataques de usuarios no autenticados. .

Imaginemos que nuestra página web tiene varias páginas que deben ser accedidas de forma anónima, como por ejemplo login.php , register.php y logout.php, y que tras autenticarse al usuario se establece una variable de sesión $_SESSION[‘registered’] = 1 y empieza a acceder a una serie de recursos protegidos;

Al final de nuestro script security.php podemos comprobar la información de la sesión, de modo que si el usuario no está registrado en la aplicación ($_SESSION['registered']) y la página accedida no es ninguna de las mencionadas en el punto anterior, se debe realizar una llamada a exit(). De este modo, un usuario no autenticado no podrá ejecutar ningún script del sistema.

2.8 COSTO BENEFICIO

2.8.1 COCOMO II

El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del ingles COnstructive COst MOdel) es un modelo matemático de base empírica utilizado para estimulación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Este modelo fue desarrollado por Barry W. Boehm a finales de los 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981)

2.8.2 Modelo de Estimación

Las ecuaciones que se utilizan en los modelos son:

- $E = a(Kl)^b * m(X)$, en persona-mes
- $Tdev = c(E)^d$, en meses
- $P = E/Tdev$, en personas

Dónde:

- E es el esfuerzo requerido por el proyecto, en persona-mes.
- Tdev es el tiempo requerido por el proyecto, en meses.
- P es el número de personas requerido por el proyecto.
- a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo.
- Kl es la cantidad de líneas de código, en miles.
- m(X) Es un multiplicador que depende de 15 atributos.

A la vez, cada sub-modelo también se divide en modos que representa el tipo de proyecto, y puede ser:

1. Modo orgánico: un pequeño grupo de programadores esperimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).

2. Modo semilibre o semiencajado: corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede inclinar una mescla de personas experimentadas y no experimentadas.

3. Modo rígido o empotrado: el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

Para el modelo básico se utiliza una aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

MODO	A	B	C	D
Orgánico	2,4	1,05	2,5	0,38
Semilibre	3,0	1,12	2,5	0,35
Rígido	3,6	1,2	2,5	0,32

Tabla 2.6 Modo Orgánico de COCOMO

Fuente: Ingeniería de software, (IAN SOMMERVILLE, 2005)

2.8.3 Cálculo de Beneficios con VAN y TIR

Para evaluar los beneficios del proyecto se utiliza el método de análisis Valor Actual Neto (VAN) y tasa interna de Retorno (TIR).

2.8.3.1 VAN

El valor actual neto, cuyo acrónimo es VAN, es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual(es decir, actualizar

mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta le inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto.

Consiste en determinar la equivalencia en el tiempo 0 de los flujos de afectivo futuros que generan un proyecto y comparar esta equivalencia con el desembolso inicial. Cuando dicha equivalencia es mayor que el desembolso inicial. Entonces, es recomendable que el proyecto sea aceptado.

La fórmula que dos permite calcular el valor actual neto es:

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

Vt Representa los flujos de caja en cada periodo t.

Io Es el valor del desembolso inicial de la inversión.

n Es el número de periodos considerado.

k, ***d*** *o* ***TIR*** es el tipo de interés

El tipo de interés es K. si el proyecto no tiene riesgo, se tomara como referencia el tipo de la renta fija, de tal manera que con el VAN se estimara si la inversión es mejor que invertir en algo seguro, sin riesgo especifico. En otros casos, se utilizara el coste de oportunidad.

Valor	Significado	Decisión a tomar
VAN > 0	La inversión produciría ganancias por encima de la rentabilidad exigida (r)	El proyecto puede aceptarse
VAN < 0	La inversión produciría pérdidas por debajo de la rentabilidad exigida (r)	El proyecto debería rechazarse
VAN = 0	La inversión no produciría ni ganancias ni pérdidas	Dado que el proyecto no agrega valor monetario por encima de la rentabilidad exigida (r), la decisión debería basarse en otros criterios, como la obtención de un mejor posicionamiento en el mercado u otros

Tabla 2.7 Interpretación del VAN

Fuente: Ingeniería de software, (IAN SOMMERVILLE, 2005)

2.8.3.2 TIR

La tasa interna de retorno o tas interna de rentabilidad (TIR) de una inversión es el promedio geométrico de los rendimientos futuros esperados de dicha inversión, y que implica por cierto el supuesto de una oportunidad para ¡"reinvertir!". Se conceptualiza como la tasa de descuento con la que el valor actual neto (VAN) es igual a cero. La TIR puede utilizarse como indicador de la rentabilidad de un proyecto: a mayor TIR, mayor rentabilidad. Se utiliza como uno de los criterios para decidir sobre la aceptación o rechazo de un proyecto de inversión. Para ello, la TIR se compara con una tasa mínima o tasa de corte, el coste de oportunidad de la inversión (si la inversión no tiene riesgo, el coste de oportunidad utilizado para comparar la TIR será la tasa de rentabilidad libre de riesgo). Si la tasa de rendimiento del proyecto esta expresado por la TIR- supera la tasa de corte, de acepta la inversión; en caso contrario, se rechaza.

$$VAN = \sum_{t=1}^{n} \frac{F_t}{(1+TIR)^t} - I = 0$$

Ft Es el flujo de caja en el periodo t.

n Es el número de periodos

I es el valor de la inversión inicial.

El criterio general para saber si es conveniente realizar un proyecto es el siguiente:

- Si TIR ≥ r → Se aceptara el proyecto. La razón es que el proyecto da una rentabilidad mayor que la rentabilidad minima requerida (el coste de oportunidad).
- Si TIR < r →Se rechasara el proyecto, La razón es que el proyecto da una rentabilidad menor que la rentabilidad minima requerida.

En donde **r** representa el costo de oportunidad.

2.9 CALIDAD DEL SOFTWARE

2.9.1 Definición de Calidad del Software

La calidad del software es el conjunto de calidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad, integridad. (PRESSMAN, 2010)

La calidad del software es medible y varia de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable a nivel de CERO FALLAS; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo periodo (10 años o más), necesita ser confiable, mantenerle y flexible para disminuir los costó de mantenimiento y perfeccionamiento durante el tiempo de explotación.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del cielo de vida del software. (PRESSMAN, 2010)

Calidad de software es hacer el uso adecuado de procedimientos, técnicos e instrumentos aplicados por antes capacitados para garantizar que un producto cumpla o supere los estándares predefinidos durante el ciclo de desarrollo de un producto. El control

de calidad implica que un producto cumpla o supere el nivel mínimo aceptable para su comercialización.

2.9.2 Como Obtener un Software de Calidad

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permita uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico.

- El principio tecnológico define las técnicas e utilizar en el proceso de desarrollo del software.
- El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.
- El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado.

La adopción de una buena política contribuye en gran medida a lograr la calidad del software pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación. (PRESSMAN, 2010)

2.9.3 Como Controlar la Calidad del Software

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como plantea TOM De Marco, USTED NO PUEDE CONTROLAR LO QUE NO PUEDE MEDIR.

Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan t agrupan de formas diferentes. Por ejemplo, Jhon Wiley define

métricas de calidad y criterios, donde cada métrica se obtiene a partir de combinaciones de los diferentes criterios. Todos los autores coinciden en que el software posee determinados índices medibles que son las bases para la calidad, el control y el perfeccionamiento de la productividad.

Una vez seleccionados los índices de calidad, se debe establecer el control, que requiere los siguientes pasos:

1. Definir el software que va a ser modelado; clasificación por tipo, esfera de aplicación, complejidad, etc., de acuerdo con los estándares establecidos para el desarrollo del software.

2. Selecciona una medida que pueda ser aplicada al objeto de control. Para cada clase de software es necesario definir los indicadores y sus magnitudes.

3. Crear o determinar los métodos de valoración para la medición de criterio periciales y herramientas automatizadas para medir los criterios de cálculo.

4. Definir las regulaciones organizativas para realizar el control: quienes participan en el control de la calidad cuando se realiza, que documentos deben ser revisados y elaborados, etc.

El libro de ingeniería del software de Roger Pressman nos dice en du capítulo 19 sobre métricas técnicas de software que la calidad de software en una compleja mezcla de factores que varían a través de diferentes aplicaciones y según los clientes que las pidan (PRESSMAN, 2010).

2.9.4 ISO 9126

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

• Funcionalidad, evalúa el grado en que el software satisface las necesidades indicadas por los siguientes sub-atributos: corrección, inter-operatividad y seguridad. (PRESSMAN, 2010)

• Confiabilidad, mide la cantidad de tiempo que el software está disponible para su uso. Para medir la confiabilidad del sistema se tiene que conocer la madurez, tolerancia a fallos y facilidad de recuperación. (PRESSMAN, 2010)

• Facilidad de uso, es el grado en que el software es fácil de usar, de acuerdo a la valoración individual por parte de un conjunto de usuarios. (PRESSMAN,2010)

• Eficiencia, establece la relación entre desempeño y cantidad de recursos utilizados bajo condiciones establecidas. (PESSMAN, 2010)

• Facilidad de mantenimiento, se centra en el cambio que va asociado a la corrección de errores (mantenimiento correctivo) durante la creación del sistema; a las adaptaciones requeridas a medida que evoluciona el entorno del software(mantenimiento adaptativo), como cambios en las reglas o políticas de la empresa y a la adaptación de nuevas versiones del sistema operativo, así también a descubrir funciones adicionales que van a producir beneficios, más allá de sus requisitos funcionales originales (mantenimiento de mejora o perfectivo). Y finalmente prevé los cambios en programas de computadora a fin de que se puedan corregir, adaptar y mejorar más fácilmente (mantenimiento preventivo o reingeniería del software). (PRESSMAN, 2010)

• Portabilidad, esfuerzo necesario para transferir un programa de un entorno de sistemas harware y/o software otro. (PRESSMAN, 2010)

2.10 MÉTRICAS DE CALIDAD

Para el desarrollo del sistema informático resulta importante utilizar métricas de calidad ya que las métricas son medidas cuantitativas del grado en que un sistema, posee un atributo dado, de alguna forma permite medir algunos aspectos individuales del proyecto, es decir la métrica proporciona una visión más profunda. Estas mediciones pueden servir

también para identificar los problemas que tiene un sistema con el objetivo de solucionarlo. (PRESSMAN, 2010)

2.10.1 Métrica de Punto de Función (PF)

La métrica de punto de función (PF) se utiliza para medir el tamaño de un sistema que se obtiene de un modelo de análisis, en la cual se define las siguientes características:

- Número de entradas de usuarios.
- Número de salidas de usuario.
- Número de peticiones de usuario.
- Numero de archivos.
- Numero de interfaces externas.

PF = Cuenta Total *{R(t) /0,01 + }∑ Fi}

Dónde:

Cuenta total: Es la suma de todas las entradas obtenidas.

R (t): Es el porcentaje de confiabilidad.

Fi (i=1-14): Son valores de ajuste de la complejidad.

CAPITULO 3

MARCO APLICATIVO

En este capítulo se presenta el análisis, diseño y desarrollo del proyecto en base a la metodología mencionada en el anterior capitulo, los diagramas UML y el enfoque UWE. Par la lógica del negocio se utiliza los Workflows con la notación BPMN.

Para iniciar el marco aplicativo primeramente se muestra el flujo regular de la información de la correspondencia de la institución, se observa en la siguiente figura 3.1 Posteriormente se realiza el desarrollo el proyecto bajo la metodología AUP (Proceso Unificado Ágil).

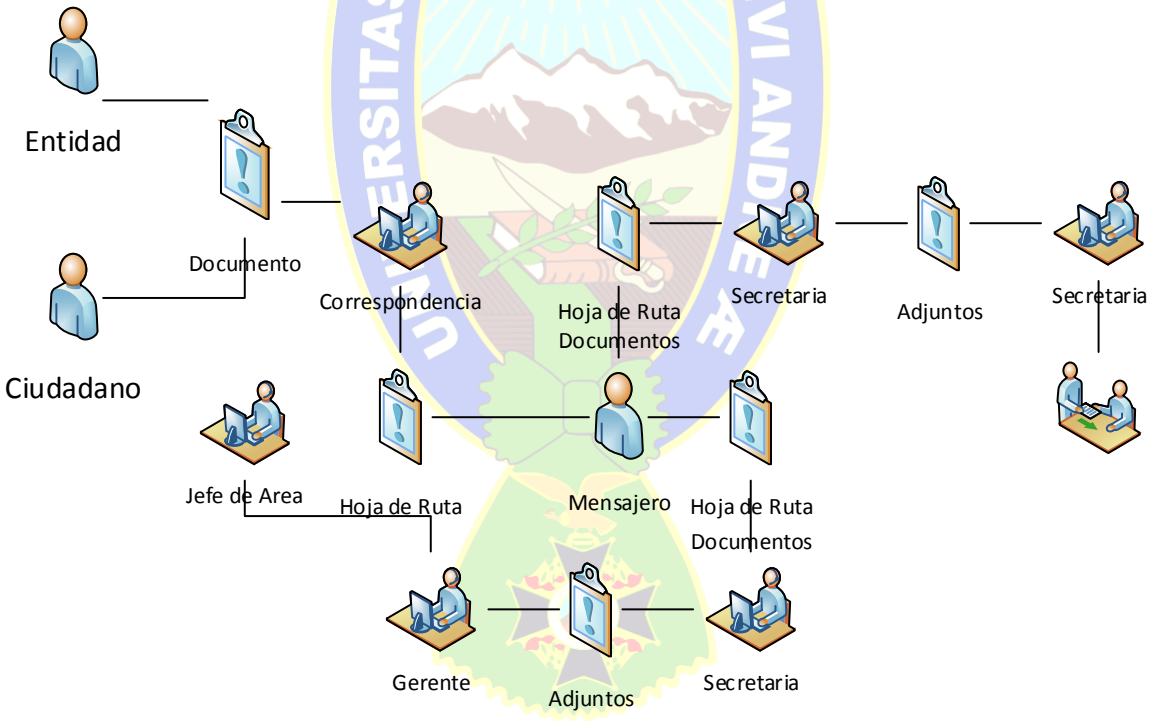

Figura 3.1: Flujo regular de la información de la correspondencia del TSA

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.1 ANALISIS SITUACIONAL

Tecnología Sistemas y Aplicaciones Bolivia S.A. (TSA Bolivia S.A.), tiene conformada un estándar rudimentario el cual se ha quedado obsoleto a las exigencias de la empresa que unifica esfuerzos y trabajos para ofrecer servicios integrales de alta calidad en el área de las Tecnologías de la Información, para luego trabajar continuamente logrando obtener y desarrollar una metodología acorde y propia de trabajo que pone de manifiesto el éxito de la empresa para logros superiores en su ámbito laboral. [Fuente: Elaboración Propia (Argumentos de la empresa)]

3.1.1 Estado Actual

Tecnología Sistemas y Aplicaciones Bolivia S.A. (TSA Bolivia S.A.), es una empresa conformada por un grupo de profesionales de alto nivel, que unifican esfuerzo y trabajo para ofrecer servicios integrales de alta calidad en el área de las Tecnologías de la Información, logrando obtener y desarrollar una metodología propia de trabajo que pone de manifiesto el éxito de la empresa.

TSA Bolivia S.A., empresa creada como Sociedad Anónima mediante Testimonio Público No. 767/97 en Octubre de 1997, se constituye en la alternativa a los requerimientos de servicios de TI y gestión administrativa en Bolivia, proponiendo soluciones basadas en tecnología de punta, avaladas por el prestigio, experiencia y calidad de servicios ofrecidos por el equipo de profesionales que conforman actualmente la empresa. [Fuente: Elaboración Propia (Argumentos de la empresa)]

3.2 PROPUESTA DE FLUJO DE CORRESPONDENCIA

Una propuesta puede darse de mil formas ya que esta es sucesiva en el transcurso de las actividades a realizar, ya enmarcándonos en el transcurso de las actividades de las correspondencias de la empresa Tecnología Sistemas y Aplicaciones Bolivia S.A. (TSA Bolivia S.A.), nos encontramos con preguntas del problema establecido en el primer capítulo y con ello se podría llegar a tener el siguiente flujo de correspondencia. [Fuente: Elaboración Propia (Argumentos de la empresa)]

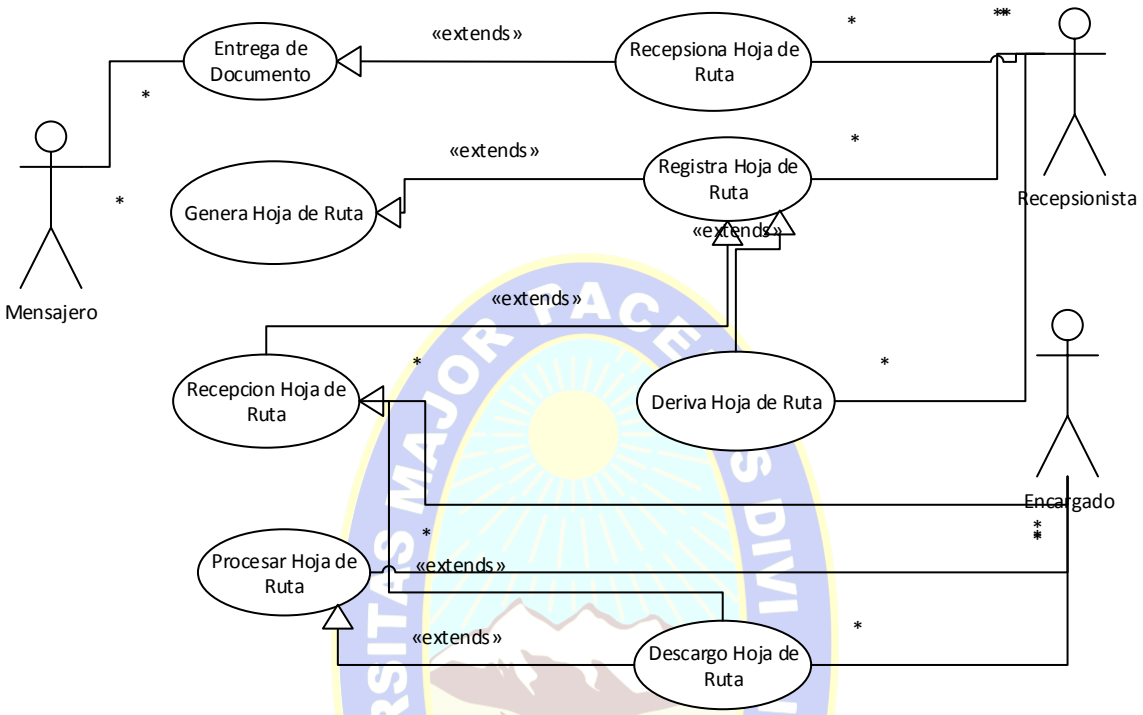

Figura 3.2: Diagrama del Flujo mejorado de la Correspondencia en el TSA

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Se observa cómo se inicia la correspondencia a partir de un documento ingresado a la institución y se lo transforma en hoja de ruta para continuar con las derivaciones correspondientes.

El proceso de finalización de la hoja de ruta puede ser obstruido en el transcurso de su derivación, ya que se depende de un mensajero para realizar el correcto flujo de los documentos.

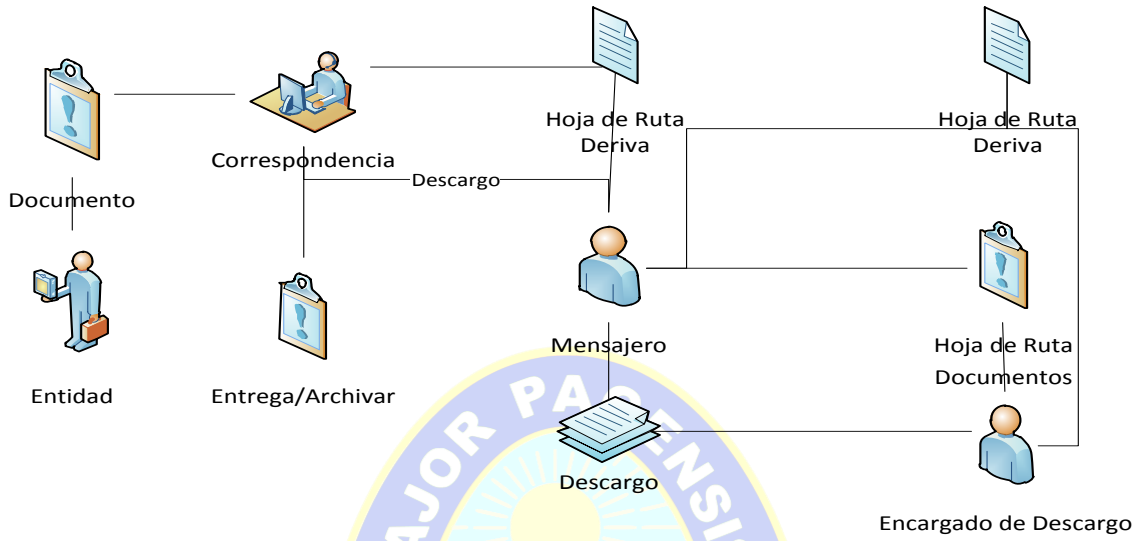

Figura 3.3: Flujo mejorado de la Correspondencia en el TSA

[Fuente: Elaboración Propia (Argumentos de la empresa)]

En la figura 3.3 se observa el camino que recorre el documento (carta, nota, etc.) enviado por la entidad solicitante (Externa o Interna), para convertirse en hoja de ruta y ser procesado por los funcionarios públicos, hasta llegar a una solución, en el proceso, los funcionarios pueden derivarse la hoja de ruta de acuerdo al cargo que ocupa, para luego poder realizar su respectivo descargo de la tarea realizada.

Para el diseño físico y formal del sistema, se utiliza el enfoque de análisis y diseño de la metodología AUP, mediante la estructura UWE, en el cual se tiene un conjunto de actividades que transforma los requisitos de usuario en el Sistema Software.

Considerando que se desarrolló el software bajo un ciclo iterativo en lo pequeño y serial en lo grande, con incrementos en la mejora del software de manera ágil y confiable. Esto nos permite obtener un sistema solido que:

- Disminuye el riesgo de un mal producto.
- Disminuye el riesgo de no obtener un producto en el tiempo previsto.
- Elimina problemas con requisitos incompletos.
- Permite soluciones rápidas.

3.3 FASE DE INICIACION

En esta fase se establece la planificación y alcance del proyecto, riesgo y prepara el entorno del proyecto.

El objetivo es tomar en cuenta todos los problemas que conlleva la llegada de correspondencia y hacia donde está dirigida; es decir, problemas enmarcados en el árbol de problemas para una mejor solución.

3.3.1 Definición del Sistema

El sistema de seguimiento de correspondencia para el TSA, se encarga del envió y despacho de las diferentes derivaciones de la correspondencia.

El proceso de trabajo que realiza en TSA, es anual, desde el registro de las Hojas de ruta, asignación de las mismas y todo el flujo de trabajo que con lleva la derivación, recepción y descargo de las correspondencias de parte de cada funcionario de la empresa, con el objetivo de satisfacer o concluir la Hoja de ruta.

3.3.2 Proceso de Correspondencia

a. **Recepción de correspondencia.** Se registra la hoja de ruta de acuerdo al siguiente cuadro:

Nro.	Registro	Descripción
1	Documento	Nota, carta, informe, circular, memorándum, etc.
2	Fuente	Externo, interno, reingreso.
3	Procedencia	Persona natural, jurídica, del interior del país
4	Remitente	Persona que envió el documento
5	A quien va dirigido	Funcionario de la institución.
6	Fojas	Documentos adjuntos

Tabla 3.1: Datos de Registro

[Fuente: Elaboración Propia (Argumentos de la empresa)]

En la siguiente figura se observa el flujo de registro de hoja de ruta- Recepción de correspondencia.

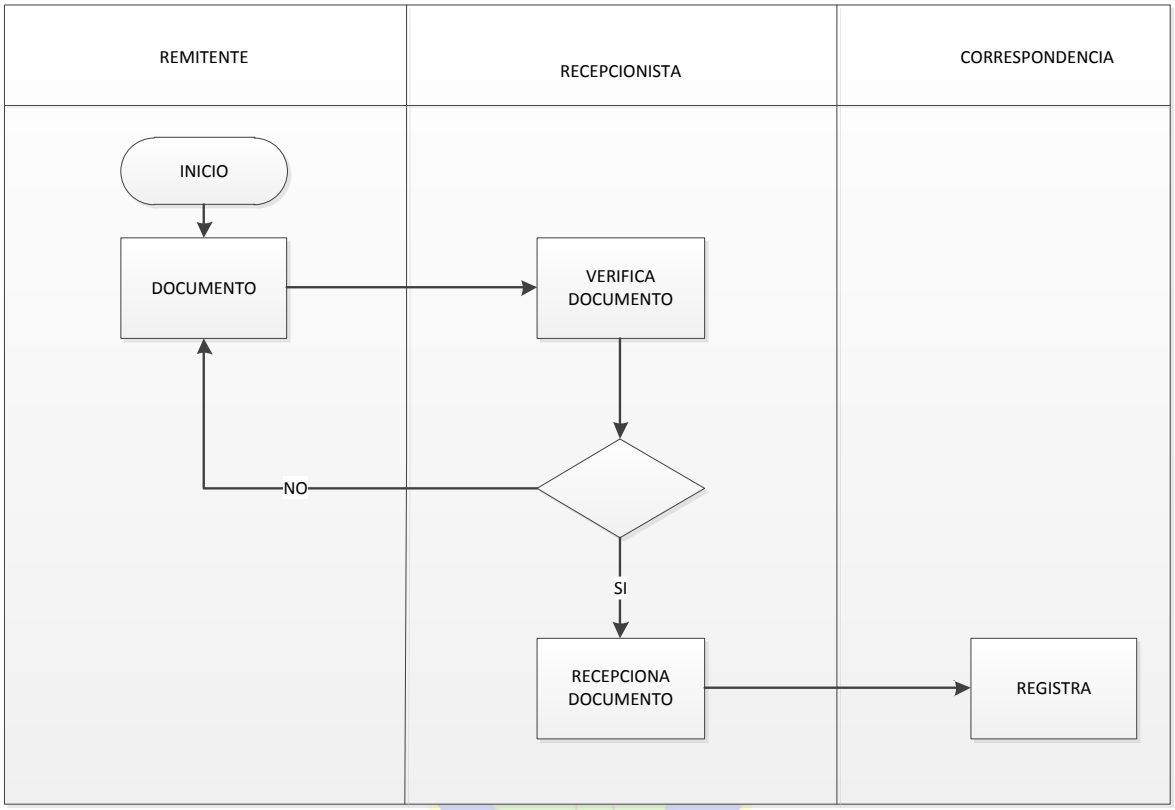

Figura 3.4: Recepción de Correspondencia

[Fuente: Elaboración Propia (Argumentos de la empresa)]

b. **Registro de código de hoja de ruta.** El registro del código de la hoja de ruta se realiza de forma manual, de acuerdo al correlativo de las Hojas de ruta que ya fueron realizadas desde el inicio de la gestión.

   En la siguiente figura se muestra el proceso:

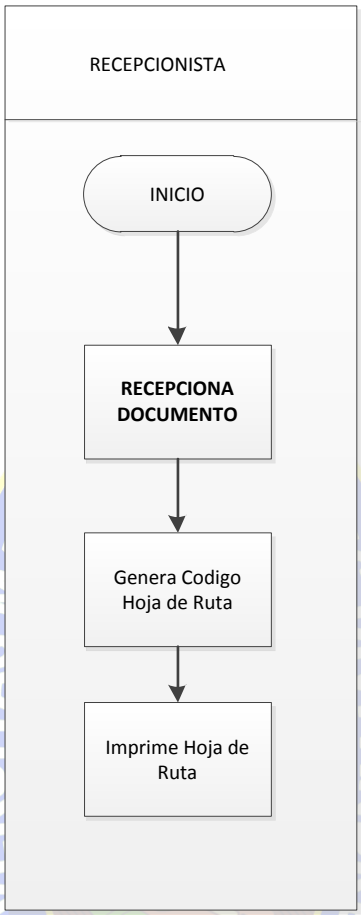

Figura 3.5: Registro de Código de Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

c. **Asignación de la hoja de ruta.** La asignación de la Hoja de Ruta depende a quien va dirigido el documento registrado, el funcionario debe elaborar o derivar la Hoja de ruta a quien corresponda, de acuerdo a la referencia y datos adjuntos en la Hoja de ruta. En la siguiente figura se muestra el flujo de asignación de Hoja de ruta:

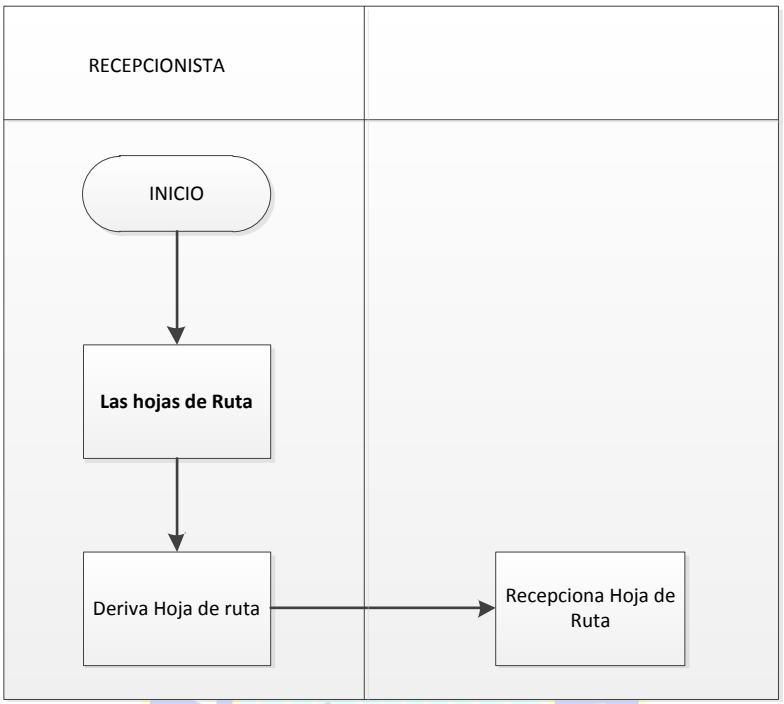

Figura 3.6: Asignación de Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

d. **Guardar libro o cuaderno de registro.** Cada área tiene por lo general una secretaria, la cual tiene un libro o cuaderno de registro, donde describe la correspondencia recepcionada, registrando la fecha y hora de recepción. Se utiliza este cuaderno para el control de las Hojas de ruta que ingresan en tal fecha.

e. **Seguimiento de correspondencia.** El seguimiento y búsqueda de correspondencia se realiza de acuerdo a las hojas de ruta o documentos adjuntos. Se usa esta característica debido a que algunas hojas de ruta llegan a perderse en el camino o simplemente no llegan a su destino, en todo caso se traban de alguna forma, así que se realiza el seguimiento de la hoja de ruta, para saber su trayecto y obtener donde se encuentra, y quien es el responsable.

f. Flujo de la correspondencia

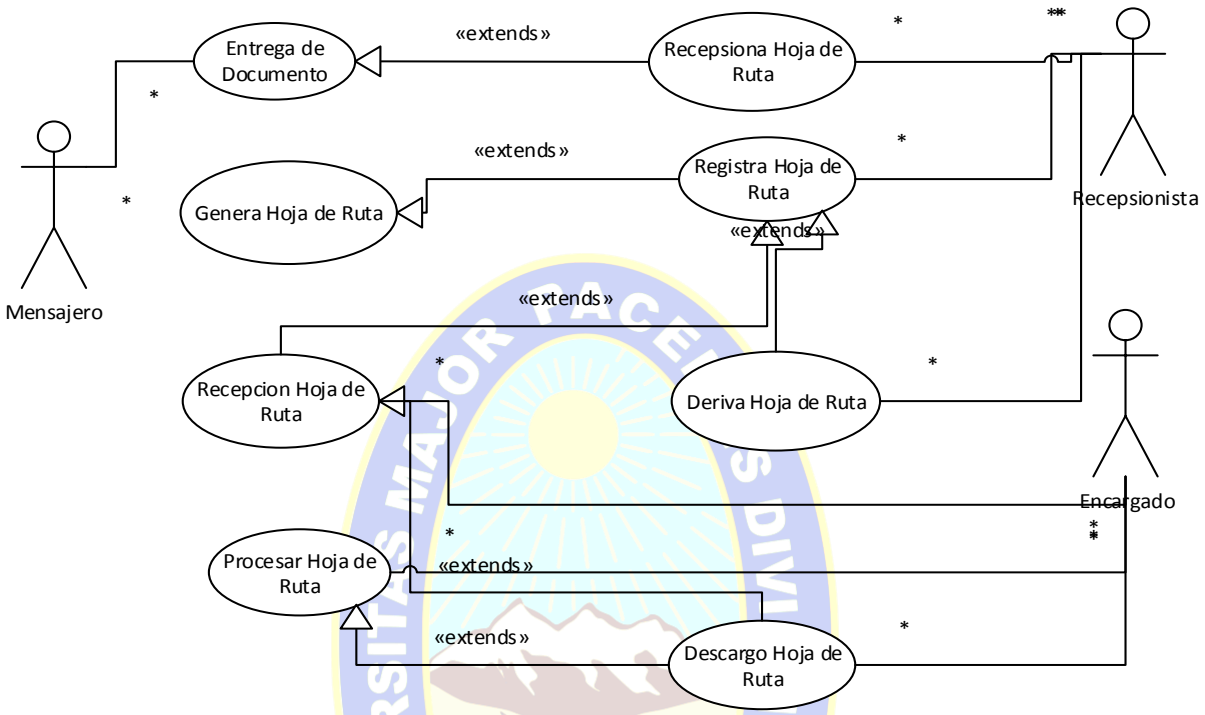

Figura 3.7: Flujo de la Correspondencia

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Por lo expuesto se identificó el dominio del problema que se aborda en el proyecto y se determina una solución para resolver el problema.

3.3.3 Identificación de Riesgos

De acuerdo a la naturaleza del proyecto se ha podido establecer los riesgos en el desarrollo del proyecto a lo largo de las fases de la metodología, las cuales se muestran en el siguiente cuadro:

Riesgo	**Probabilidad que suceda**	**Impacto**
Las actividades definidas en el cronograma de anexo no hayan sido distribuidas de manera equitativa.	Media	Alto
Descuido del desarrollo del proyecto	Media	Alto
Poco conocimiento en las herramientas y plataforma de desarrollo, lo cual ocasionara retrasos y alteración en el cronograma de actividades.	Media	Alto
Que las actividades planificadas no se realizan en el tiempo establecido para la entrega	Media	Media
Los requerimientos son demasiado ambiciosos para el tiempo disponible.	Baja	Media
Haber dejado de lado los artefactos de software necesarios para un mejor diseño de la solución	Baja	Baja
Cambios en los requerimientos del proyecto	Media	Baja

Tabla 3.2. Identificación de Riesgos

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.3.4 Plan de Riesgos:

De acuerdo a los riesgos identificados y su impacto sobre el desarrollo del proyecto, se tiene el siguiente plan de riesgos, para evitar casos:

- ✓ En riesgos relacionados con la planificación, es conveniente dedicar más tiempo en definir los procesos necesarios para el desarrollo del proyecto.
- ✓ Invertir todo el tiempo posible, en reanudar el desarrollo del proyecto, y re investigar nuevas herramientas o actualizaciones en las herramientas utilizadas.
- ✓ Reorganizar el cronograma de acuerdo al límite de entrega del proyecto, para obtener más información sobre nuevas herramientas y plataformas de desarrollo.
- ✓ Dedicar más tiempo al desarrollo del proyecto, y fusionar tareas incompletas, un avance en paralelo.
- ✓ Especificar un tiempo de desarrollo para los nuevos requerimientos y establecer su factibilidad de aplicación.
- ✓ Buscar una forma de implementar esos artefactos sin alterar demasiado la estructura funcional del sistema.

Para nuevos requerimiento se los analizara de acuerdo al impacto del requerimiento, se continuara con el desarrollo de manera ágil.

3.4 FASE DE ELABORACIÓN

Se tomara en cuenta las herramientas necesarias para la elaboración del sistema de control de registro de correspondencia.

3.4.1 Modelado del Negocio

a. Requerimientos Funcionales

Ref.	Requerimientos
R1	Autentificación de usuario y privilegios
R2	Registrar correspondencia, clasificar, generar número único de hoja de ruta, tipo de correspondencia, remitente, referencia, proveído (instrucción), fechas de registro y envió, destinatario, documentos adjuntos, numero de fojas (una vez)…
R3	Generación de hoja de ruta y caratula de correspondencia (grabar)
R4	Derivar la correspondencia ingresada al siguiente (primer o siguiente destinatario)destinatario incluyendo instructiva si corresponde (una o varias veces)
R5	Procesar la instrucción de acuerdo a correspondencia e informar a superior(una o varias veces)
R6	Descargo de correspondencia por instrucción anterior (una o varias veces)
R7	Seguimiento de correspondencia, por Número de Hoja de Ruta, por remitente, por fechas de registro y otros campos de forma general.
R8	Búsqueda de correspondencia, por Numero de Hoja de Ruta y fecha y otros campos
R9	Reportes de Registros de Correspondencia por fecha, área, hoja de ruta
R10	Recepción del flujo de la Hoja de Ruta

Tabla 3.3. Requisitos Básicos del Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

b. Requerimientos No Funcionales

- ✓ Sistema operativo Windows 7.
- ✓ Procesador mínimamente core 2 Duo.
- ✓ Memoria RAM mínimamente 2 gb.

3.4.2 Identificación De Actores

Nos permite conocer a las personas involucradas en los distintos procesos de la correspondencia, se muestra en la siguiente tabla:

Actor	**Descripción**
Encargado de Correspondencia	Tiene la función de registrar la correspondencia, generar hojas de ruta, derivar, recepcionar, realizar su descargo, hacer su seguimiento, búsqueda y obtener reportes de los mismos.
Personal Administrativo Interno	Tiene la función de recepcionar las hojas de ruta dentro de la institución, derivar, realizar su descargo, hacer su seguimiento, búsquedas y obtener reportes.
Administrador del Sistema	Tiene la función de controlar el acceso al sistema, y obtener reportes.

Tabla 3.4. Identificación de Actores

[Fuente: Elaboración Propia (Argumentos de la empresa)]

La tabal muestra la disciplina de modelado de actores, en el cual se muestran las interacciones que hay de un actor con caso de uso, cada actor con su receptiva función representa los flujos de trabajo (WORKFLOWS) identificados, este modelo no tiene una base automatizada, sin embargo presenta una idea principal de los procesos de negocio.

3.4.3 Diagrama De Casos De Uso

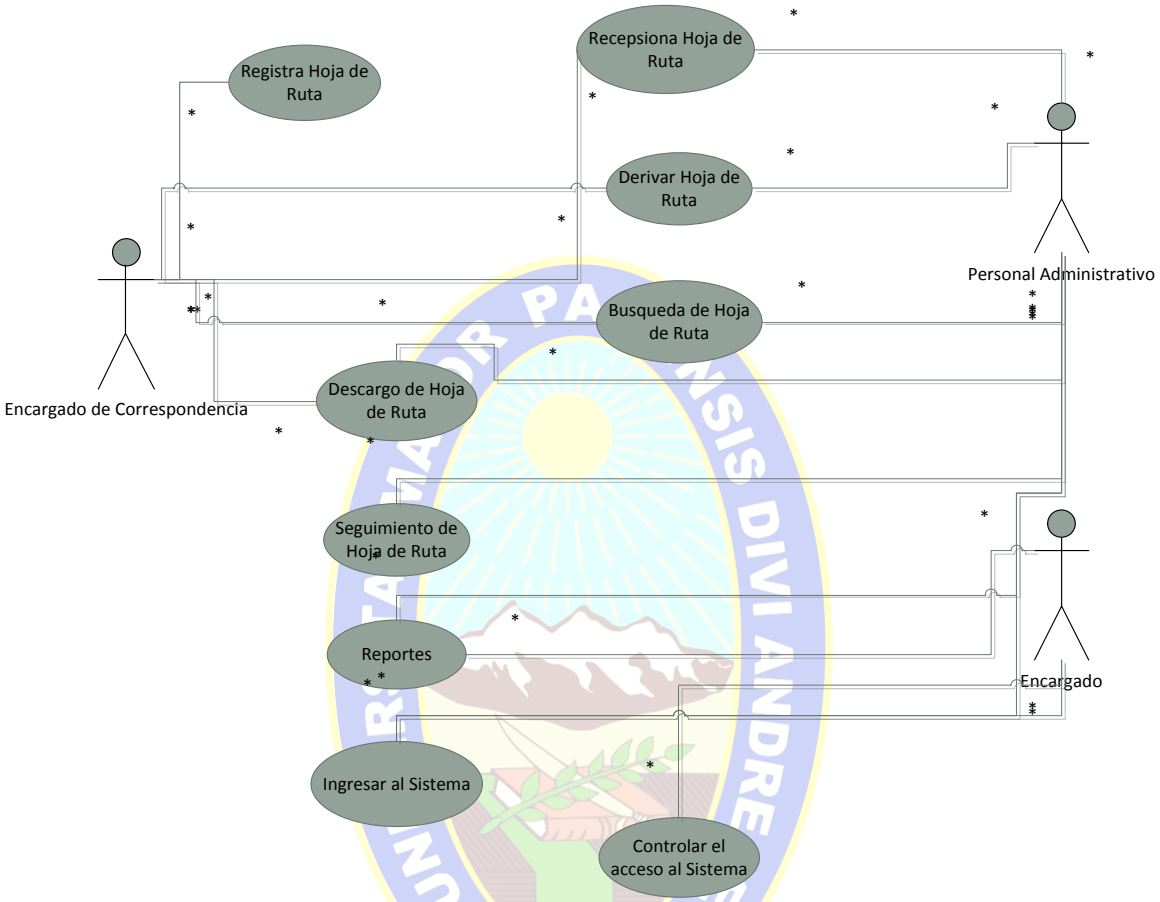

Figura 3.8: Diagrama de casos de uso

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Los diagramas de casos de uso nos muestran la interacción de los actores con los procesos del sistema. El caso de uso de alto nivel propuesto se ilustra en la figura 3.8.

3.4.4 Relación Entre Actores

En la figura 3.9 se observa que los actores “Responsable de Correspondencia” y “Personal Administrativo interno” tienes los mismos casos de uso, de lo cual se desprende que existe una herencia.

3.4.5 Casos De Uso Por Actor

Se especifica los casos de uso extraídos del diagrama de casos de uso de alto nivel del sistema (Figura 3.8)

a. Personal administrativo Interno

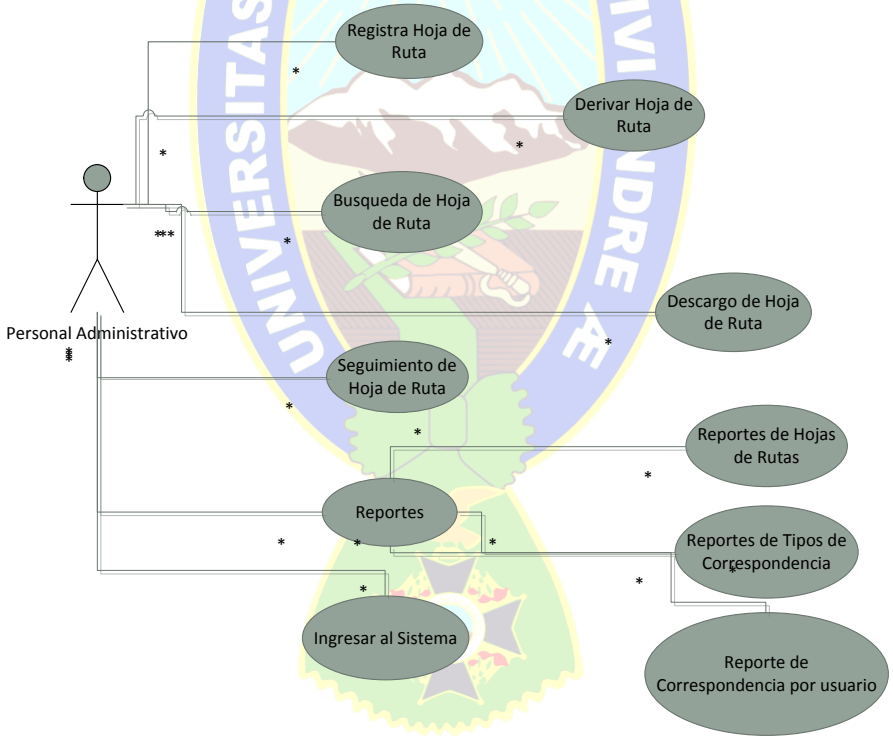

Figura 3.9: Diagrama de Caso de Uso Personal Administrativo Interno

[Fuente: Elaboración Propia (Argumentos de la empresa)]

b. Encargado de correspondencia

Figura 3.10: Diagrama de Caso de Uso Encargado de correspondencia

[Fuente: Elaboración Propia (Argumentos de la empresa)]

c. Administrador de sistema

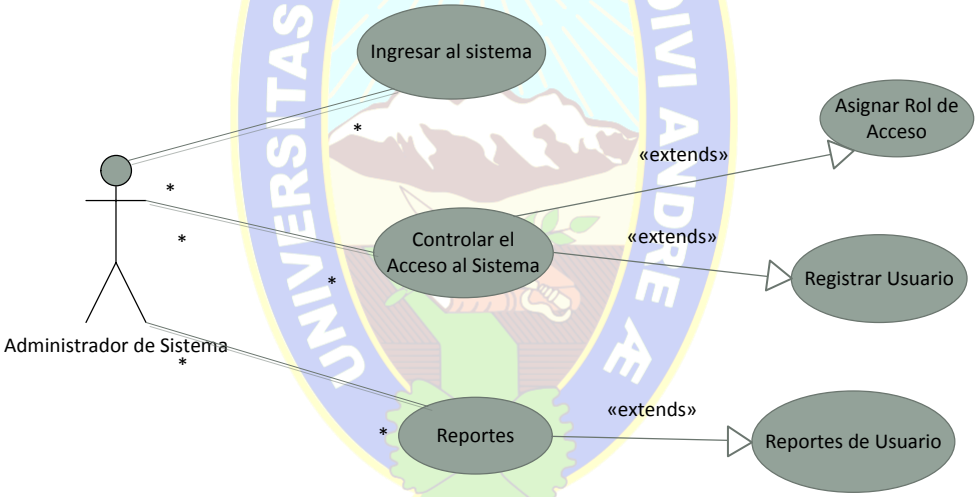

Figura 3.11: Diagrama de Caso de Uso Administrador de Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.4.6 Descripción de Casos de Uso

Las siguientes tablas muestran el detalle de los casos de uso de los procesos del sistema y la interacción con los actores.

<table>
<tr><td>Caso de uso:</td><td colspan="4">Ingresar al Sistema</td></tr>
<tr><td>Actores:</td><td colspan="4">Encargado de Correspondencia, Personal Administrativo Interno, Administrador</td></tr>
<tr><td>Propósito:</td><td colspan="4">Verificar el acceso al sistema a los usuarios</td></tr>
<tr><td>Referencia cruzada</td><td colspan="4">R1</td></tr>
<tr><td rowspan="4">Flujo normal de eventos</td><td colspan="2">Actores</td><td colspan="2">Sistema</td></tr>
<tr><td>1</td><td>Ingresar Login y PIN</td><td>2</td><td>Verifica Datos</td></tr>
<tr><td></td><td></td><td>3</td><td>Carga roles de usuario</td></tr>
<tr><td></td><td></td><td>4</td><td>Permite ingresar al sistema</td></tr>
<tr><td>Flujo alterativo</td><td colspan="4">En 2 si no se encuentra el usuario ,despliega mensaje de usuario no valido</td></tr>
<tr><td></td><td colspan="4">En 2 si existe usuario pero no PIN, permite registrar PIN</td></tr>
</table>

Tabla 3.5. Caso de Uso Ingresar al Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Caso de uso:	**Registrar Hojas de Ruta**			
Actores:	Encargado de Correspondencia			
Propósito:	Registrar datos del documento para generar hoja de ruta			
Referencia cruzada	R2,R3			
Flujo normal de eventos	Actores		Sistema	
	1	Recepcionar el documento	2	Verifica Datos
			3	Carga el código de Hoja de ruta
			4	Almacena en la base de datos
	5	Imprime hoja de Ruta		
Flujo alternativo	En 1 Buscar y Modificar Hoja de ruta			

Tabla 3.6. Caso de Uso Registrar Hojas de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Caso de uso:	**Derivar Hoja de Ruta**			
Actores:	Encargado de Correspondencia, Personal Administrativo Interno			
Propósito:	Derivar hoja de ruta a otro funcionario			
Referencia cruzada	R4			
Flujo normal de eventos	Actores		Sistema	
	1	Buscar hoja de ruta	2	Despliega datos de Hoja de ruta
	3	Llena datos de derivación		
	4	Guardar derivación	5	Verifica datos de derivación
			6	Registra la derivación
			7	Muestra datos de derivación
Flujo alterativo	En 3 Cancelación de derivación de todos los actores			
	En 5 Mensaje de error al grabar derivación			

Tabla 3.7. Caso de Uso Derivar Hojas de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Caso de uso:	**Recepción Hoja de Ruta**			
Actores:	Encargado de Correspondencia, Personal Administrativo Interno			
Propósito:	Decepcionar Hoja de Ruta de una derivación o descargo			
Referencia cruzada	R10			
Flujo normal de eventos	Actores		Sistema	
	1	Selecciona la hoja de ruta para recepción	2	Despliega datos de Hoja de ruta
	3	Llena datos de recepción		
	4	Guarda recepción	5	Verifica datos de recepción
			6	Registra recepción
			7	Muestra datos de recepción
Flujo alterativo	En 3 Cancelación de recepción de todos los actores			
	En 5 Mensaje de error al grabar recepción			

Tabla 3.8. Caso de Uso Recepción Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Caso de uso:	**Descargo de Hoja de Ruta**			
Actores:	Encargado de Correspondencia, Personal Administrativo Interno			
Propósito:	Informe de conclusión de Hoja de Ruta del funcionario asignado			
Referencia cruzada	R5,R6			
Flujo normal de eventos	Actores		Sistema	
	1	Selecciona la hoja de ruta	2	Despliega datos de Hoja de ruta
			3	Despliega opciones de proceso
	4	Llena datos de descargo de Hoja de ruta		
	5	Escoge opción descargo y concluir de hoja de ruta		
	6	Guarda descargo	7	Verifica datos de descargo
			8	Registra descargo
			9	Muestra datos de descargo
Flujo alterativo	En 5 Cancelación de descargo			
	En 5 Se realiza descargo hacia el funcionario derivante			
	En 5 entrega de documentación de Hoja de Ruta			
	En 5 Archivo de la documentación de Hoja de Ruta			
	En 6 Mensaje de error al grabar descargo			

Tabla 3.9. Caso de Uso Descargo Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Caso de uso:	**Recepción Hoja de Ruta**			
Actores:	Encargado de Correspondencia, Personal Administrativo Interno			
Propósito:	Se realiza la búsqueda mediante el Numero de Hoja de Ruta, y por rango de fechas de ingreso y de registro			
Referencia cruzada	R8			
Flujo normal de eventos	Actores		Sistema	
	1	Selecciona el tipo de búsqueda		
	2	Ingresa de búsqueda		
	3	Presiona la opción buscar	4	Despliega Lista de Hojas de Ruta
	5	Presiona ver detalle	6	Muestra detalle
Flujo alterativo	En 2 Cancelar Búsqueda			

Tabla 3.10. Caso de Uso Búsqueda de Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Caso de uso:	**Seguimiento de Hoja de Ruta**			
Actores:	Encargado de Correspondencia, Personal Administrativo Interno			
Propósito:	Obtener el detalle del recorrido de la Hoja de Ruta			
Referencia cruzada	R7			
Flujo normal de eventos	Actores		Sistema	
	1	Ingresa número de Hoja de Ruta	2	Despliega recorrido de Hoja de Ruta
			3	Indica donde se encuentra la Hoja de Ruta
Flujo alterativo	En 1 cancela seguimiento de Hoja de Ruta			

Tabla 3.11. Caso de Uso Seguimiento de Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Caso de uso:	**Reportes**			
Actores:	Encargado de Correspondencia, Personal Administrativo Interno, Administrador del Sistema			
Propósito:	Vista de reportes de Hoja de Ruta			
Referencia cruzada	R9			
Flujo normal de eventos	Actores		Sistema	
	1	Ingresa en reportes	2	Carga los tipos de reportes
	3	Selecciona un reporte	4	vista en pantalla los datos del reporte
Flujo alterativo	En 4 no imprime en pantalla, sin datos de reporte.			

Tabla 3.12. Caso de Uso de Reportes

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.4.7 Glosario De Términos

A continuación se describen algunos elementos del modelo que podrían llegar a tener doble significado, en la siguiente tabla el detalle de ellos:

Términos	**Categoría**	**Descripción**
Hoja de Ruta	Tipo	Número de registro de un documento en el recorrido por la institución
Correspondencia	Tipo	Área de recepción de documentos a la institución
Administrador del sistema	Actor	Responsable de dar permisos y configurar el sistema
Derivar	Caso de uso	Pasa de un responsable a otro una hoja de ruta

Recepcionar	Caso de uso	Registrar fecha y hora en que la hoja de ruta llega a manos del responsable asignado
Descargo	Caso de uso	Enviar respuesta a remitente de Hoja de ruta sobre la acción realizada sobre la Hoja de Ruta
Personal administrativo Interno	Actor	Funcionario de la Institución, ya sea responsable o secretaria, quienes procesan la información contenida en las hojas de ruta
Documento	Tipo	Son las cartas, notas, memorándums, etc., que se procesan en la institución
PIN	Tipo	Código de seguridad que cada usuario al sistema tiene para acceder al sistema

Tabla 3.13 Glosario de Términos

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.5 Diseño de Workflow

Se observa la interacción de los actores del sistema con cada proceso de la correspondencia, desde la generación de la Hoja de Ruta y las derivaciones que se procesan, se muestra en el proceso la conclusión de la hoja de ruta por medio de la entrega de la misma y su archivo en la institución.

3.5.1 Diagrama de Actividades

Los diagramas de actividades muestran el comportamiento de las actividades entre los elementos del dominio, como se pueden ver en la figura siguiente:

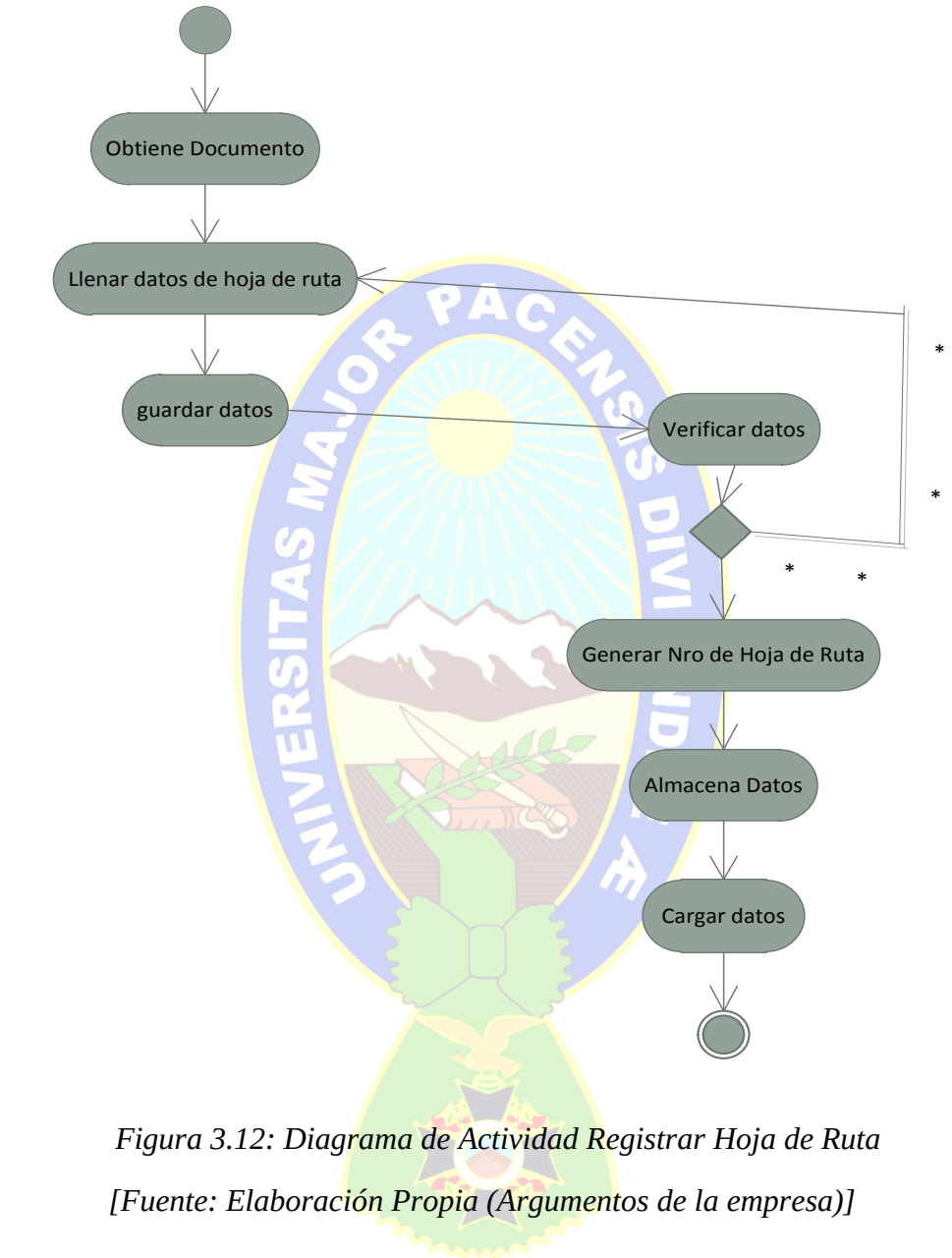

Figura 3.12: Diagrama de Actividad Registrar Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

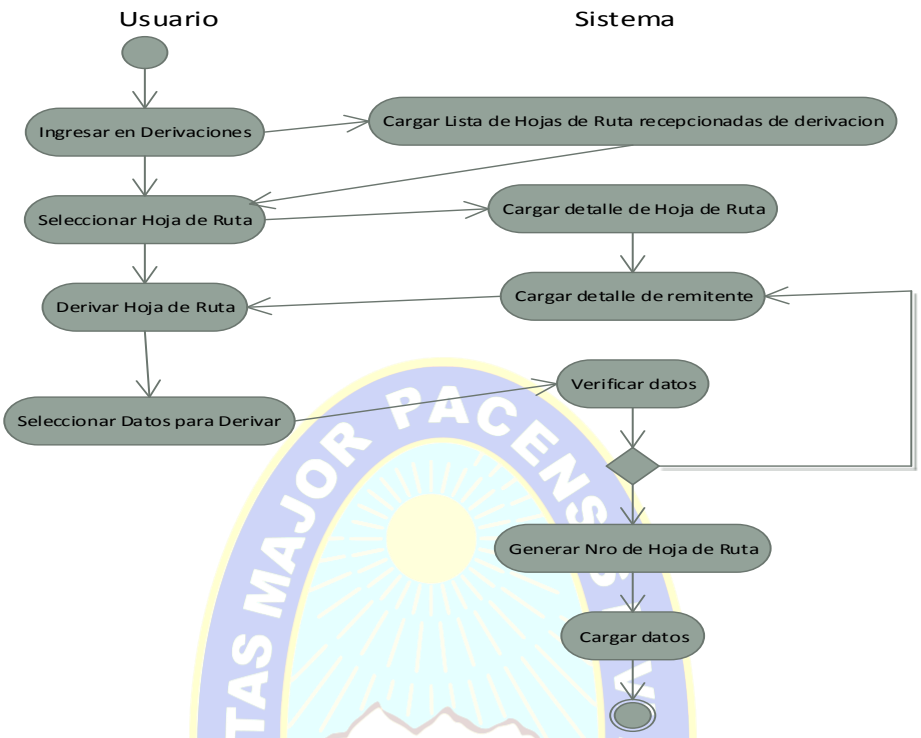

Figura 3.13: Diagrama de Actividad Derivar Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

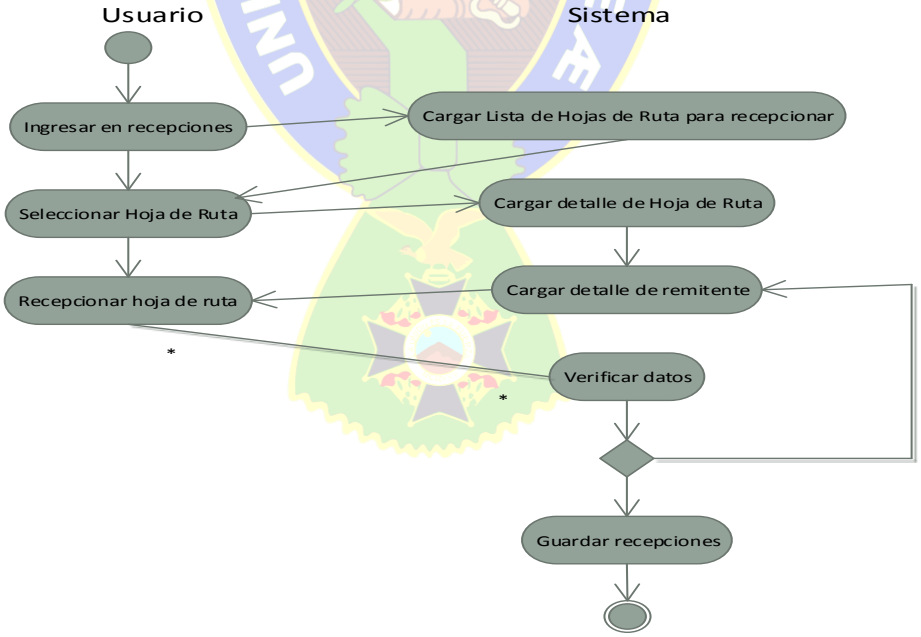

Figura 3.14: Diagrama de Actividad Recepcionar Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

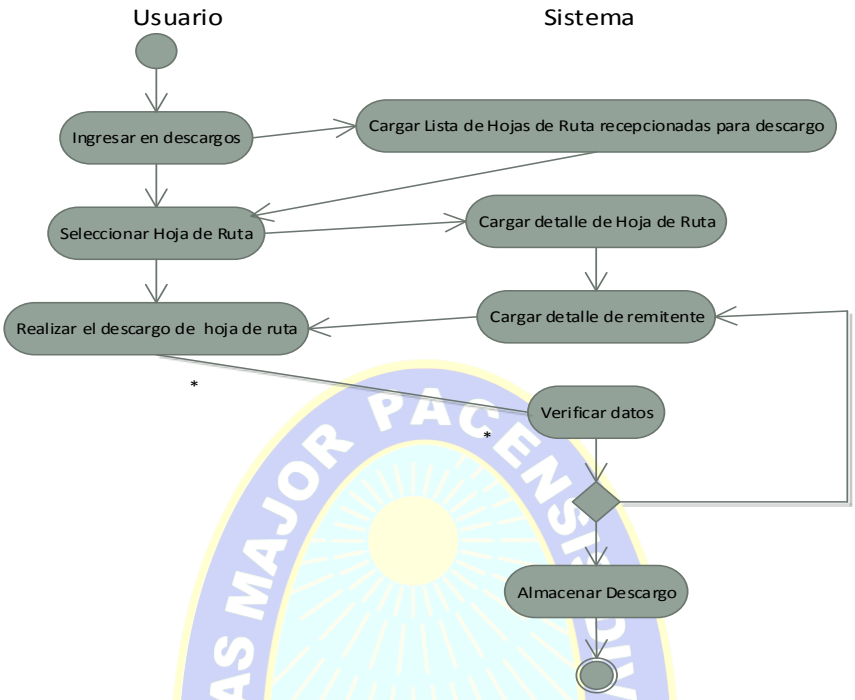

Figura 3.15: Diagrama de Actividad Descargo de Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

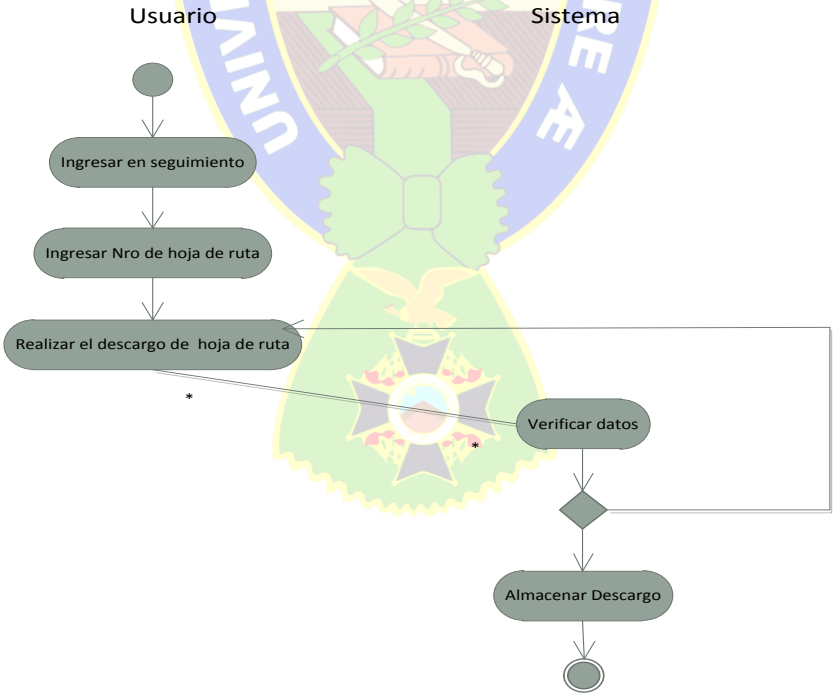

Figura 3.16: Diagrama de Actividad Seguimiento de Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

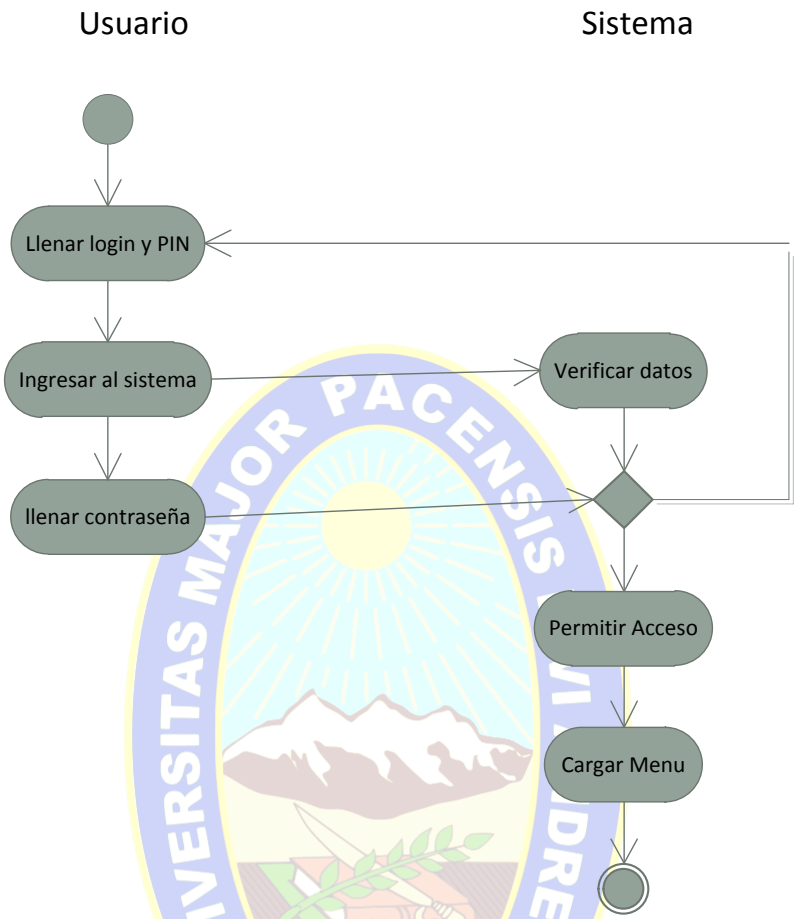

Figura 3.17: Diagrama de Actividad Ingresar al Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.5.2 Diagrama de Comunicación

En esta parte hemos de ver cómo va comunicándose todas las entidades que encontramos de este tipo, desde la llegada de la correspondencia hasta la llegada final de la correspondencia de a quién debe ser designado para su mejor funcionamiento.

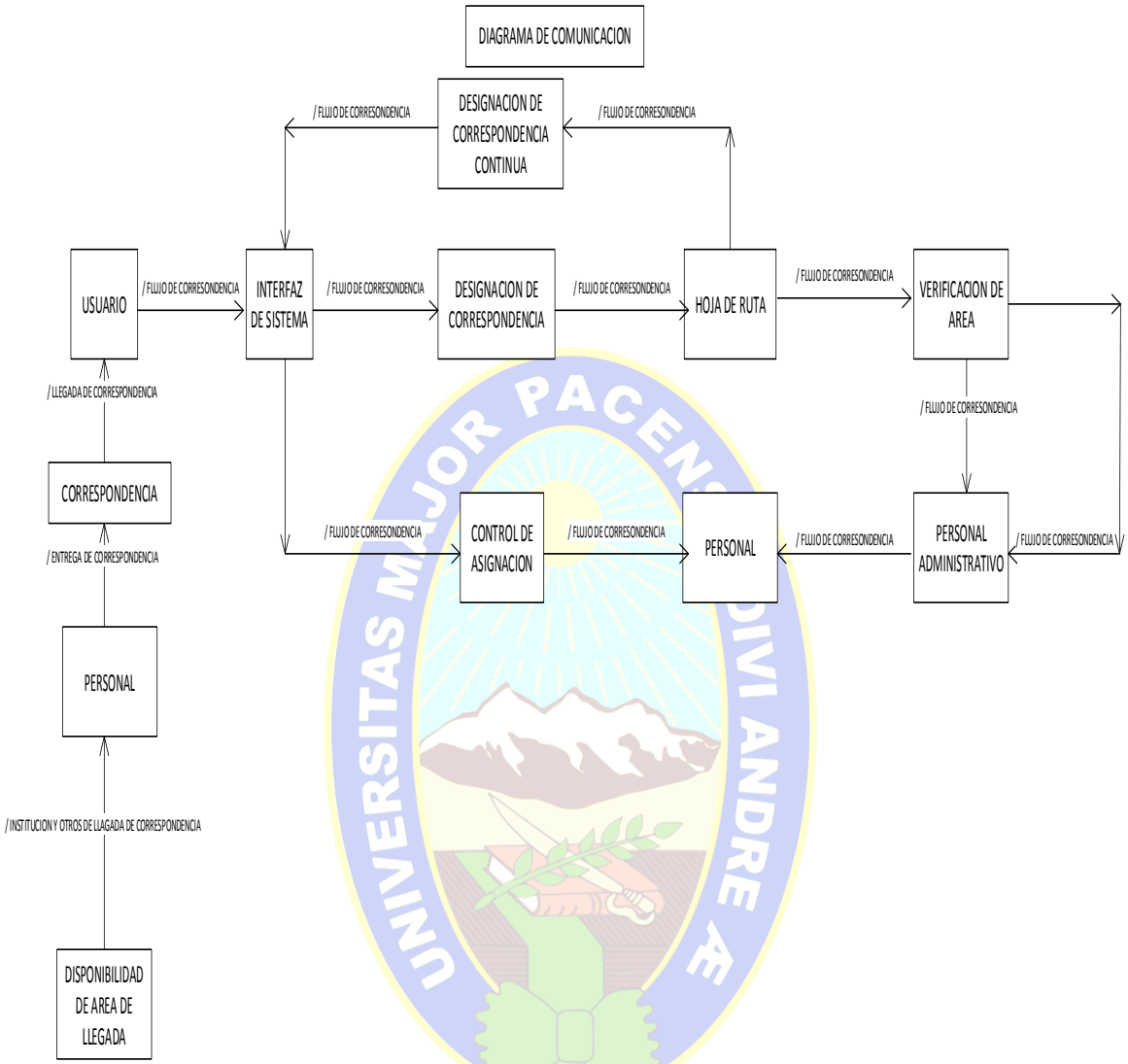

Figura 3.18: Diagrama de Comunicación

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.5.3 Diagrama de Estados

Estos son los estados del sistema a su ingreso y su mejor uso para el llenado de los datos conseguir un enclaustramiento de funciones necesarias.

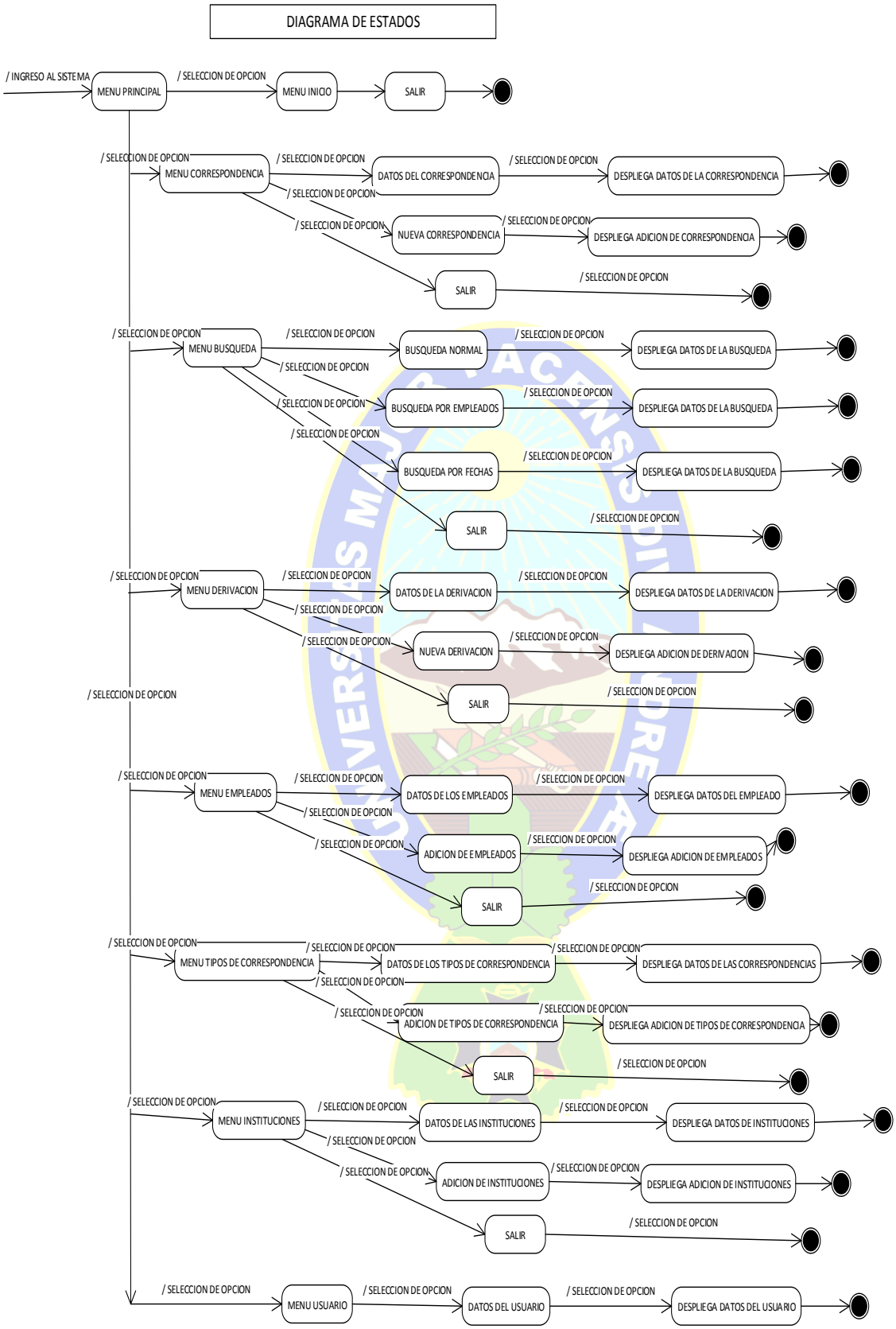

Figura 3.19: Diagrama de Estados

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.5.4 Modelo Conceptual

El modelo conceptual o de contenidos muestra las actividades que debe realizar la correspondencia hasta llegar a su objetivo, la cual es la paralela a un diagrama de clases. El diseño preliminar es como se muestra en la figura 3.18:

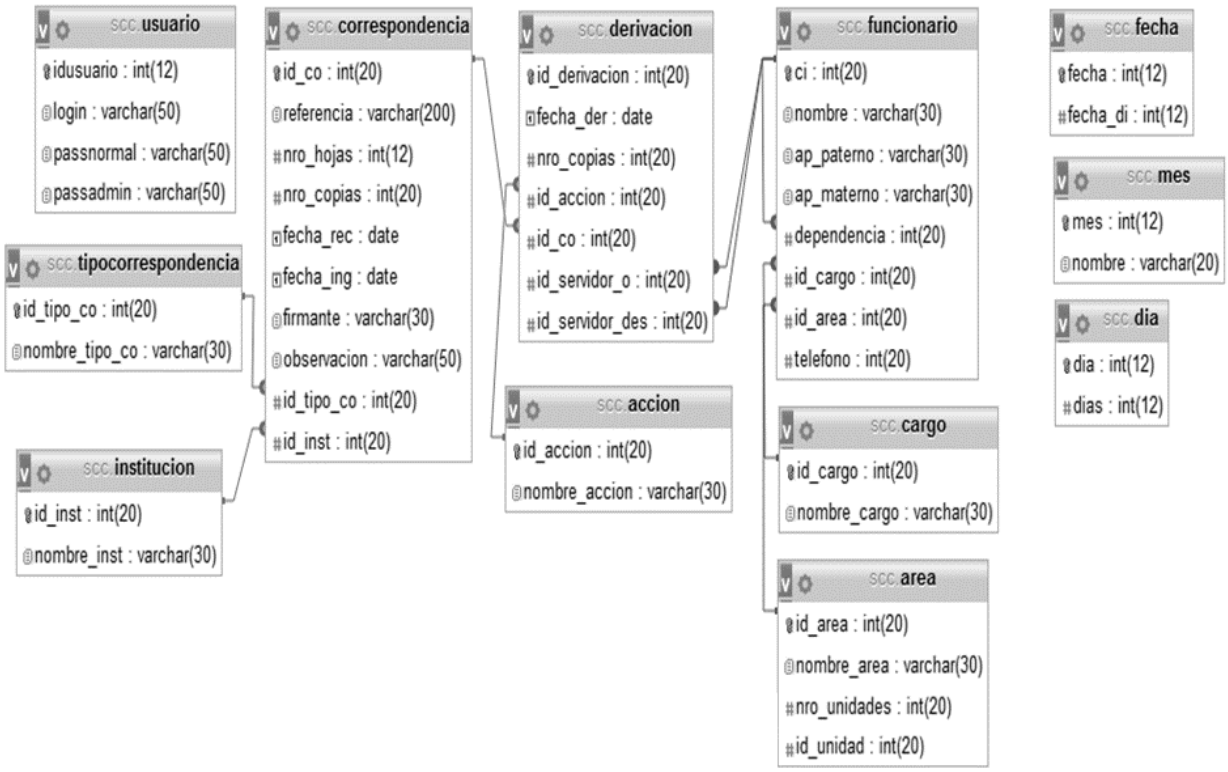

Figura 3.20: Modelo conceptual del Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.5.5 Arquitectura del Sistema

En este se punto se realizara una descripción de la arquitectura 3 capas: capaz de interfaz o presentación, capa lógica de negocios y una de base de datos. (MICROSOFT, 2011)

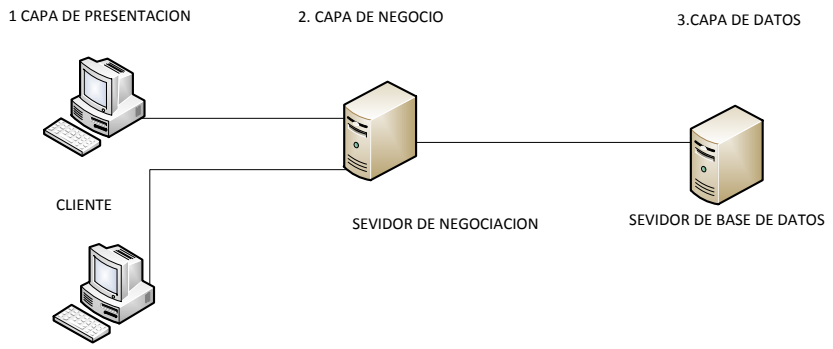

Figura 3.21: Arquitectura 3 capas

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Para la interfaz se puede usar Internet Explorer, Firefox y otros. La estructura del sistema se lo considera de la siguiente forma:

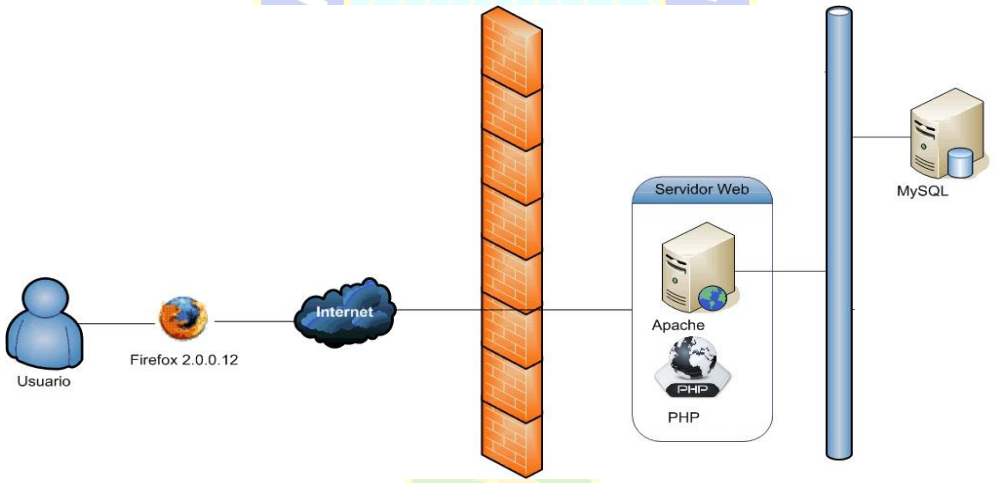

Figura 3.22: Modelo de 3 capas

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.6 FASE DE CONSTRUCCIÓN

En la fase de construcción, el objeto principal es desarrollar el prototipo de software hasta la producción para las pruebas.

3.6.1 Diagrama de Secuencia

Los diagramas de secuencia muestran de manera gráfica los eventos y operaciones que los actores ejecutan al interactuar con el sistema. La primera interacción que todo actor debe realizar es el acceso al sistema, en la siguiente figura (3.23) se detalla el acceso de un usuario al sistema.

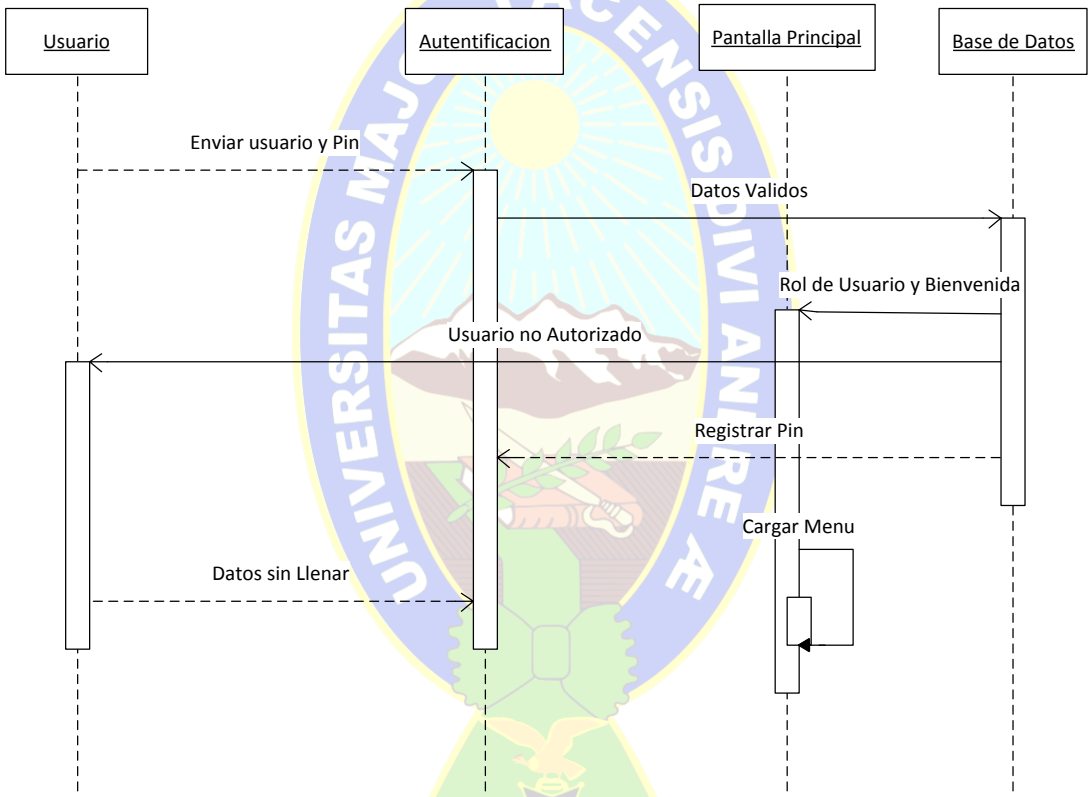

Figura 3.23: Diagrama de Secuencia Acceso al Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

El sistema solicita un usuario y un PIN de autentificación, para que a través de esto el usuario pueda tener acceso al sistema. El usuario tiene que estar registrado en el sistema, para que pueda crearse un PIN o ingresar con el mismo. La creación de PIN, es un proceso simple donde el usuario introduce dígitos alfanuméricos y con un mínimo de longitud de 8 dígitos.

Una vez que el usuario ingreso al sistema puede continuar las tareas que se muestra y detallan en los siguientes diagramas de secuencia.

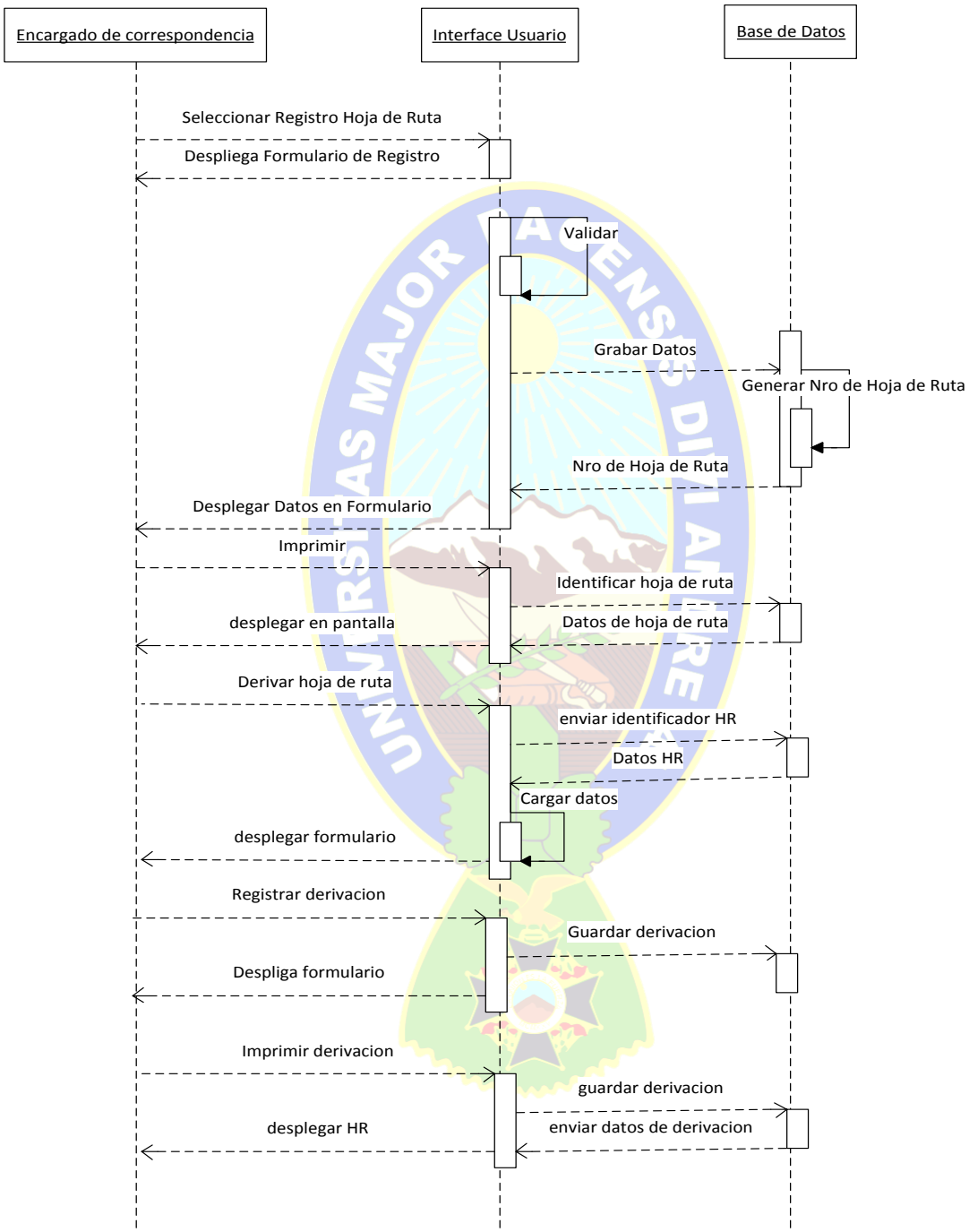

Figura 3.24: Diagrama de Secuencia Registro de Hojas de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

El registro de la hoja de ruta es un proceso donde se realiza todos los envíos y recepciones de la correspondencia hasta llegar a su objetivo, es el inicio de todo el flujo que lleva una hoja de ruta, desde la solicitud hasta el procesado de la nota o documento que ingresa a la institución. La derivación de la hoja de ruta se puede realizar en el mismo registro o por medio del siguiente diagrama de secuencia.

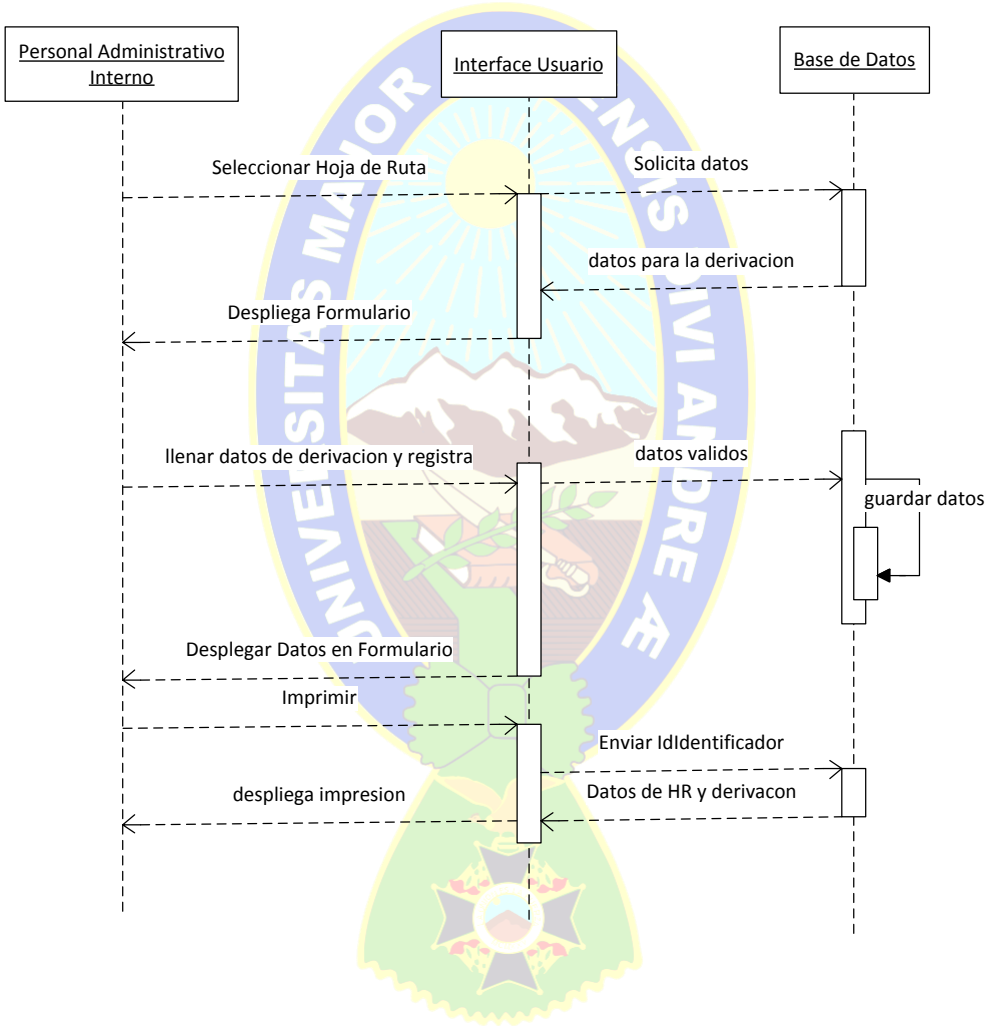

Figura 3.25: Diagrama de Secuencia Derivar Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Después del registro de la hoja de ruta, el sistema requiere que este sea derivado hacia el funcionario que es el encargado de la correspondencia deriva finalmente correspondiente de la institución.

El proceso de recepción de hojas de ruta es muy simple, ya que el funcionario lo único que debe realizar es; seleccionar la hoja de ruta de la bandeja de entrada y especificar la hora y fecha, para que después el sistema guarde la hoja de ruta como recibida por el usuario que realizo la recepción.

La recepción de las hojas de ruta se la puede realizar de las derivaciones o de los descargos que llegan al funcionario.

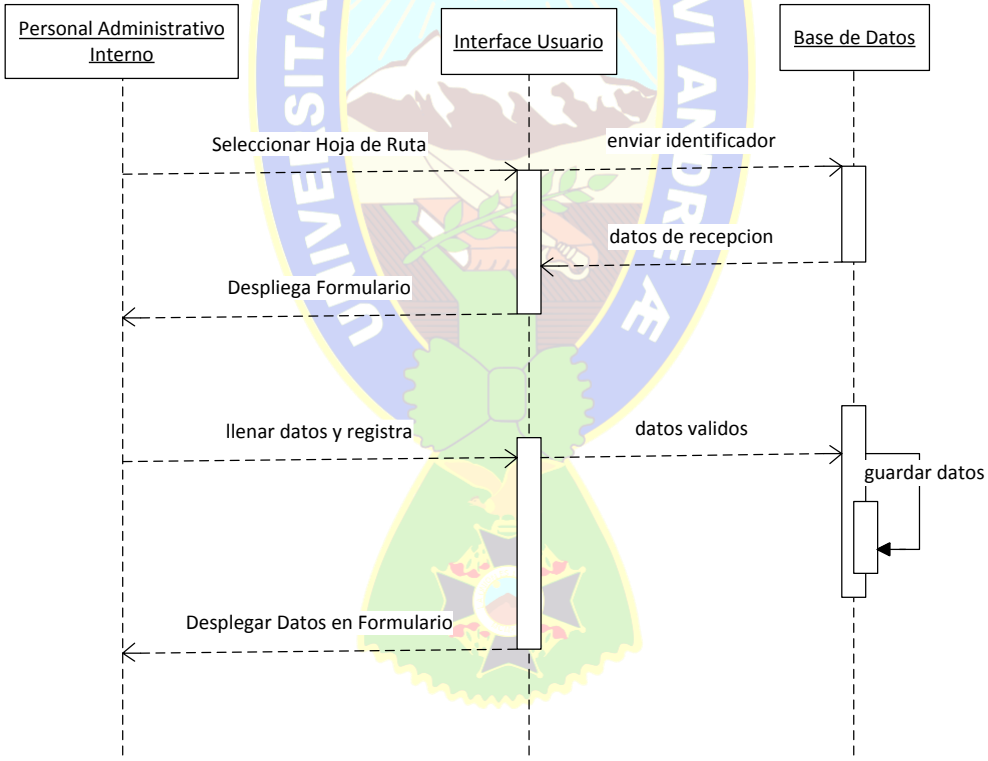

Figura 3.26: Diagrama de Secuencia Recepcionar Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

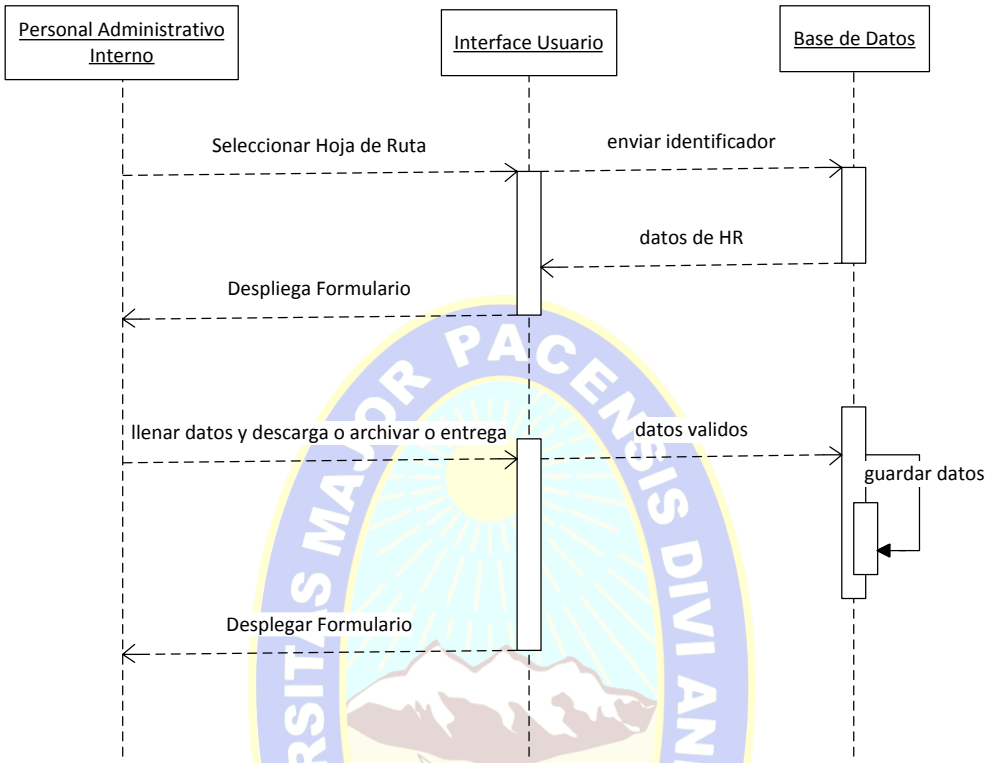

Figura 3.27: Diagrama de Secuencia Descargo de Hoja de Ruta

[Fuente: Elaboración Propia (Argumentos de la empresa)]

El descargo de las hojas de ruta, es un proceso necesario, debido a que el funcionario tiene la obligación de realizar una determinada acción y presenta una respuesta hacia el funcionario quien le encargo esa tarea.

3.6.2 Diagrama de Clases

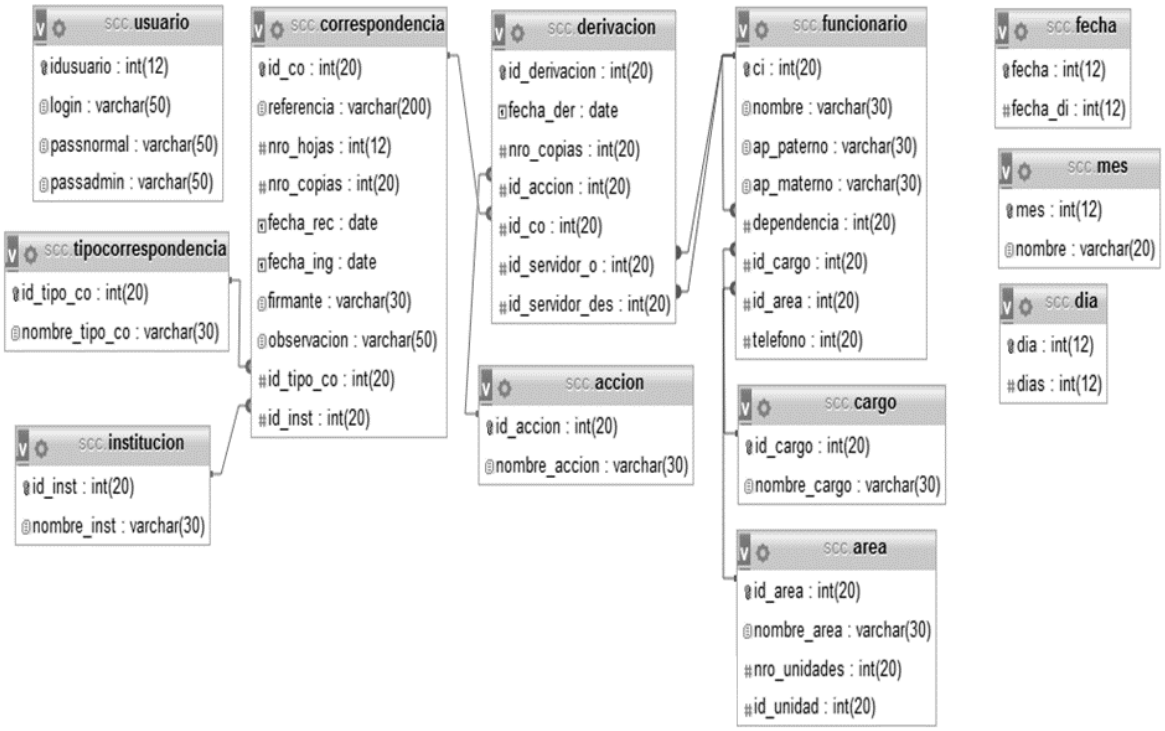

Figura 3.28: Diagrama de Clases del Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Se observó en el modelo conceptual, de acuerdo a los diagramas UWE, el uso del diagrama de clases, el cual es una abstracción del dominio del sistema, que permite identificar las clases principales de software que interactúan con el sistema.

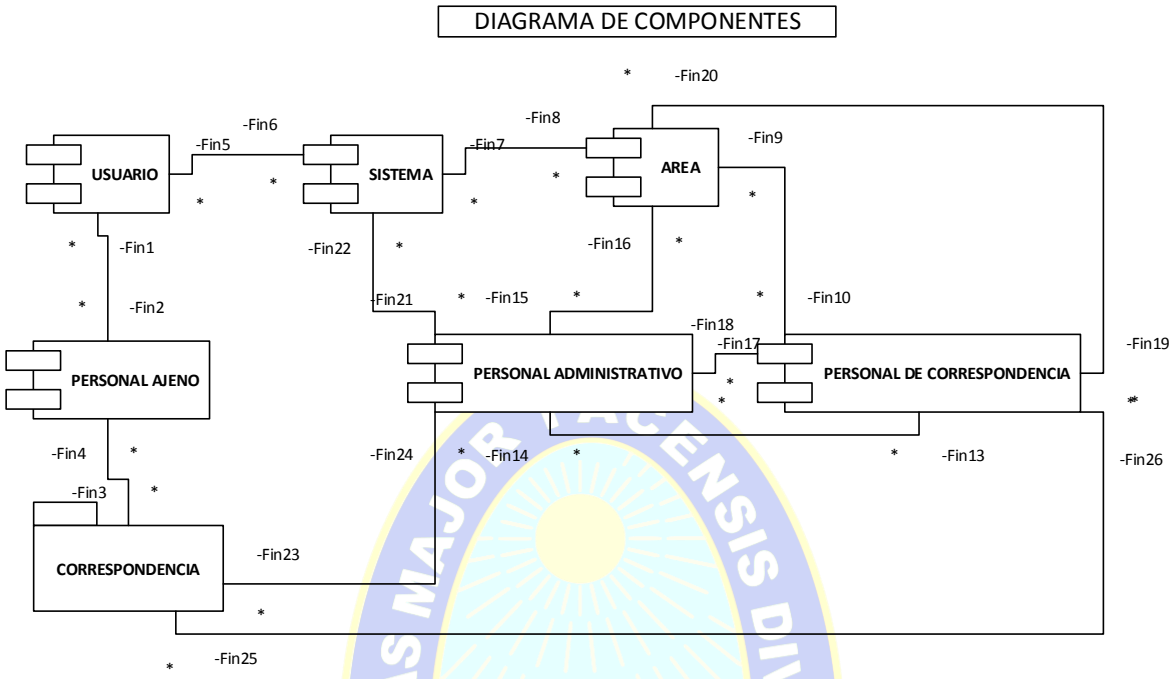

Figura 3.29: Diagrama de Componentes

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.6.3 Diagrama De Despliegue

El diagrama de despliegue muestra cómo y dónde se instalara el sistema en forma física. En la figura (3.29) se ilustra la configuración básica del despliegue del sistema.

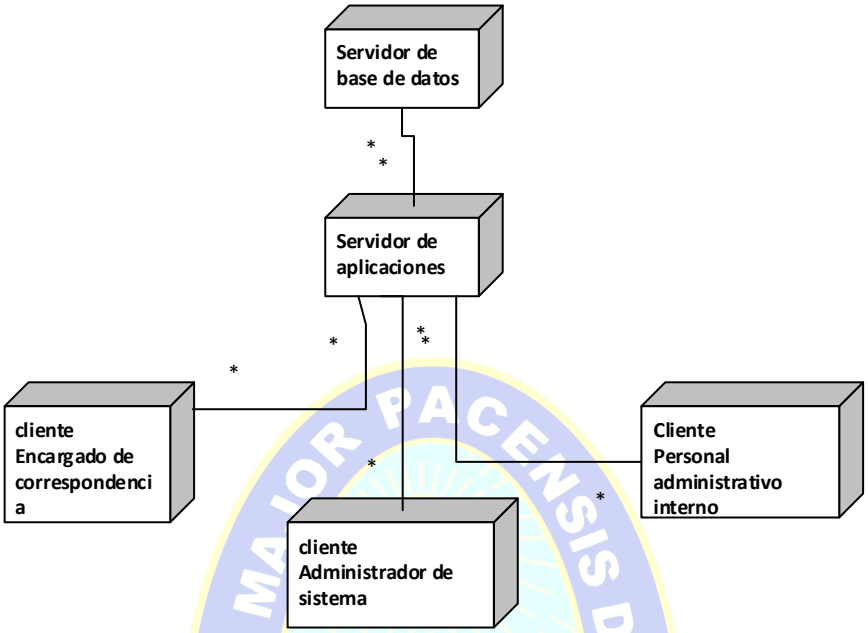

Figura 3.30: Diagrama de despliegue del Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

3.7 MÓDULO DE SEGURIDAD

3.7.1 Seguridad Mediante Validación de Usuario

Para el módulo de seguridad se utilizó el Login que será el nombre de usuario el cual debe darse el ingreso privilegiado al sistema seguido de números de Identificación Personal (Contraseña), que está compuesto por dígitos los cuales van del 1 al 9999 y letras (longitud mínima 8 dígitos), será usado por determinados funcionarios de la empresa.

Cada una de estas será determinada por los privilegios de dicha empresa que será usuario regular y usuario administrativo que gozara de determinados privilegios que ayudaran a salvaguardar todos los recursos del sistema en todos sus ámbitos.

La funcionalidad de la contraseña es la siguiente; Una vez registrado el usuario en el sistema, y quien lo hará él es usuario administrativo el cual podrá dar de alta y baja a todos los usuarios normales como administrativos para salvaguardar los datos necesarios tiene la siguiente estructura:

Figura 3.31: Registro de PIN de acceso

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Una vez registrados el usuario que es de dos tipos, se procesa el ingreso del usuario al sistema de distintas formas:

- La primera que es de usuario normal que puede dar altas y bajas a correspondencias y no así a los datos de usuarios, y con ello podremos colocar la seguridad necesaria de los datos.
- La segunda del administrador que cuenta con todos los privilegios necesarios para realizar altas y bajas a los usuarios más el manejo de la correspondencia de forma necesaria.

CAPITULO 4

CALIDAD DE SOFTAWE

En este capítulo se evalúa la calidad del sistema, que será sometido a criterios de calidad que permitan medir, analizar y comprender el grado de cumplimiento que el sistema debe cumplir.

La evaluación de los productos web o factores de calidad, proporcionan medidas indirectas y verificaciones excelentes para evaluar la calidad de un sistema. La ISO 9126 identifica seis factores de calidad software, los mismos coinciden con los definidos para la evaluación web.

FACTORES DE CALIDAD

4.1.1 Funcionalidad

La funcionalidad media se da a través del punto de función (PF) que obtiene una medida cuantitativa, objetiva o auditable del tamaño de la aplicación, en los términos de las funciones de usuario.

La métrica de punto de función se usa como medio para predecir el tamaño del sistema que se va obtener por medio de un análisis, siendo este la medida indirecta del software.

Para medir la funcionalidad del sistema se deben determinar las siguientes cinco características, dadas en el ámbito de la información.

a. Número de entradas de Usuario

Representa el número de entradas que el usuario llega a realizar desde que ingresa a un archivo del sistema que puede ser en distintas operaciones altas, bajas y modificaciones con el objetivo de probar la aplicación.

1. Pantalla de registro de Hojas de Ruta
2. Pantalla de derivaciones de Hojas de Ruta
3. Pantalla de recepciones de Hojas de Ruta
4. Pantalla de modificación de Hojas de Ruta y/o derivación
5. Pantalla de eliminación de Hoja de Ruta y/o derivación

b. Número de salidas de usuario

Representa cada salida que llega por pantalla, que despliega exactamente al exterior del sistema. Entre estas salidas son reportes y mensajes de error.

1. Reporte principal de Hoja de Ruta
2. Reporte de Seguimiento de Hoja de Ruta
3. Reporte de usuario del sistema
4. Reporte de recepciones de usuario

c. Número de peticiones usuario

Representa la entrada y salida en el cual una entrada general una salida. Entre las peticiones de usuario puede ser consultas externas desde la sesión de usuario y aquellas que entran desde otra petición.

1. Consulta de búsqueda de Hoja de Ruta
2. Generación de Hoja de Ruta
3. Búsqueda de recepciones
4. Búsqueda de derivaciones
5. Búsqueda de seguimiento de Hoja de Ruta.

d. Numero de Archivo

Es un conjunto lógico de datos que puede ser parte de una base de datos o archivos independientes.

1. Base de datos de Correspondencia.

e. Numero de Interfaces externas

Es el número de interfaces legibles por la máquina que se utilizan para transmitir información a otra máquina.

1. Exportación a PDF
2. Impresora.

f. Calculo del punto de función

	Factor de comparación		
Parámetros de medición	Cuenta	Simple	Total
Número de entradas de usuario	6	3	18
Número de salidas de usuario	5	4	20
Número de peticiones de usuario	4	3	12
Número de archivo	1	7	7
Número de Interfaces externas	2	5	10
Cuenta total			**67**

Tabla 4.1 Calculo de Punto de Función

[Fuente: Elaboración Propia (Argumentos de la empresa)]

No influencia	0
Incidencia	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Tabla 4.2 Ponderación para el ajuste

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Nro.	Factor	Valor
1	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
2	¿Se ha diseñado el código para ser reutilizable?	5
3	¿Existen funciones de procesamiento distribuido?	1
4	¿Se ejecutará el sistema en un entorno operativo existente o fuertemente utilizado?	4
5	¿Requiere el sistema entrada de datos iterativos?	4
6	¿Requiere la entrada de datos que las transacciones en entadas de datos se lleven a cabo sobre múltiples pantallas y operaciones?	4
7	¿Es crítico el rendimiento?	3
8	¿Se utilizan los archivos maestros de forma interactiva?	3
9	¿Son complejos las entradas, salidas, los archivos o las peticiones?	3
10	¿Es complejo el procesamiento interno?	1
11	¿Se requiere comunicación de datos?	4
12	¿Están incluidas en el diseño la conversión y la instalación?	4
13	¿Requiere el sistema copias de seguridad?	5
14	¿Se ha diseñado las aplicaciones para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
	Fi	51

Tabla 4.3 Cuadro de Valores de Ajuste de Complejidad

Fuente (PRESSMAN, 2010)

Donde el punto de función está dado por la fórmula:

PF=cuenta total * (0.65+0.01*SUM (Fi))

Dónde:

Cuenta total – Suma de todas las entradas obtenidas.

0.01- Error de confiabilidad de Sistema.

0.02- SUM (Fi)- Suma de factores de complejidad.

Remplazando en la formula anterior de punto de Función:

67(0.65+ 0.01 * 51)= 77.72 pf

Para poder encontrar las líneas de código se aplica la equivalencia entre líneas de código y punto de función, un valor de 20 (1de/pf)

LDC = PF * equivalencia

LDC = 77.72 (pf) * 20 (lde/pf)

LDC= 1554.2

Considerando como máximo valor de ajuste de complejidad SUM (Fi) =70, tiene:

PF Máximo= 67 * (0.65 + (0.01 *70)

PF Máximo = 90.45

Por tanto la funcionalidad está dada por:

PF/PF Máximo⟹77.72 /90.45 = 0.95⟹ 0.95 * 100= **95%**

Entonces la funcionalidad y utilidad del sistema es de un 95%, lo que quiere decir que el sistema tiene un 95% sin riesgo a fallos.

4.1.2 Fiabilidad

La fiabilidad se puede definir como la probabilidad de fallas que existe en un determinado tiempo del sistema. El cálculo de la fiabilidad se plantea a través de la siguiente formula.

Fiabilidad= 1 - PFBD

Tiempo de servicio	Peticiones Realizadas	Fallas Encontradas	Probabilidad de Fallo Bajo Demanda	Tiempo Medio Entre Fallos
8 Hrs	20	1	0.05	8 Hrs
16 Hrs	40	3	0.075	5 Hrs
40 Hrs	100	4	0.04	10 Hrs
160 Hrs	400	15	0.0375	10 Hrs

Tabla 4.4 Fiabilidad del Sistema

[Fuente: Elaboración Propia (Argumentos de la empresa)]

PFBD= (0.05 + 0.075 + 0.0375)/4

PFBD = 0.05

Reemplazando en la fórmula de fiabilidad se tiene.

Fiabilidad =1- PFBD = 1-0.05=0.95

Lo cual nos indica que el sistema tiene un grado de fiabilidad de **95%.**

4.1.3 Usabilidad

La usabilidad o calidad de uso depende de la visión que tiene el usuario del sistema, los cuales dependen del esfuerzo necesario que se quiere para usar con el apoyo de una evaluación de uso del sistema. La calidad de usabilidad se los usuarios dependerá de las siguientes características:

- Comprensibilidad.-Atributo que mide el esfuerzo del usuario en reconocer el concepto lógico de software y su aplicabilidad.
- Factibilidad de aprender.-Atributos que miden el esfuerzo del usuario en aprender la aplicación (control, operación, entradas, salidas)
- Operatividad.-Atributos que miden el esfuerzo del usuario en operar y controlar el sistema.

Usuarios	Factibilidad comprensión	Factibilidad aprendizaje	Factibilidad operación
Encargado de Correspondencia	95%	90%	95%
Secretarias	90%	96%	94%
Directores	97%	98%	95%
Funcionarios	92%	94%	95%
Promedio	93.5%	94.5%	94.8%

Tabla 4.5 Parámetros de Usabilidad

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Por lo tanto el promedio de factibilidad de uso del sistema es de **94.3 %.**

4.1.4 Mantenibilidad

Durante la elaboración del sistema de información se realizó una serie de modificaciones y actualizaciones para mejorar el producto y así de esta manera cumplir con los requerimientos.

Es el esfuerzo necesario para localizar y arreglar un error en el programa, la ecuación matemática es la siguiente:

Factibilidad de mantenimiento=1-01 (número de días-horas hombre por corrección)

Factibilidad de mantenimiento =1-0.1 (3-1 persona por corrección)

Factibilidad de mantenimiento=0.8*100

Por tanto la factibilidad de mantenimiento es de un 80%.

4.1.5 Portabilidad

Es el hecho de que el sistema pueda funcionar en diferentes entornos de procesadores no menores a Core 2 Duo, Memoria RAM de 2 GB y Sistema Operativo Windows 7.

a. Servidor

Es portable en entornos Windows XP, SERVER 2008 R2 y posteriores.

- XAMPP
- MYSQL
- PHP

En cuanto a hardware

- Dual Core
- RAM de 2 GB
- Procesador de 3 GHZ
- Disco duro de 160 GB

b. Cliente

Para la ejecución en equipos cliente se requiere de un navegador de internet Explorer en su versión 8 para adelante, Firefox, etc. Para el hardware son necesarias las siguientes características:

- Pentium IV
- RAM de 1 GB
- Procesador 2 GHz
- Disco duro de 160 GB

4.1.6 Eficiencia

Para evaluar este factor de calidad se considera los atributos correspondientes a las características de eficiencia, implícitamente en la tabla de evaluación de usabilidad.

INFORMACION	VALOR EN PORCENTAJE
Comprensibilidad del sistema	95
Mecanismos de ayuda y retroalimentación	95
Aspectos de interfaz	95
Aspectos de exploración	90
Carencia de errores	90
Porcentaje de evaluación	93

Tabla 4.6 Facturas de Calidad para la Evaluación

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Este resultado indica que el Sistema de Registro y Seguimiento de Correspondencia, logra un servicio con una eficacia del 93% que significa que el sistema presenta un rendimiento satisfactorio al momento de generar notas de salida, reportes y de ser comprensible en el manejo, debido a que estas fueron diseñadas con elementos simples que reducen el tiempo de proceso.

4.1.7 Flexibilidad

La flexibilidad es costo de modificación del producto cuando cambian sus especificaciones. Las siguientes formulas son usadas.

Flexibilidad=1-0.05 (número de días-hombre por cambio)

Flexibilidad=1-0.05 (3-1)

Flexibilidad=0.9 *100

Por tanto la flexibilidad del sistema es de 90%.

CAPITULO V

EVALUACIÓN DE COSTOS Y BENEFICIOS

El análisis de costo y beneficio del sistema que el cliente valora cuando los beneficios exceden a los costos de la implementación del sistema. Por lo tanto se analiza el costo y beneficio, en concordancia con lo enmarcado en los puntos anteriores.

5.1 ANÁLISIS DE COSTOS

Para la cuantificación del costo se toma en cuenta el siguiente aspecto:

a. Costo de desarrollo del software

Para hallar el costo de construcción software utilizaremos el valor obtenido como PF obtenido con la fórmula de la ecuación de punto de función, la cual tiene un valor de 74.24 y se utilizara la conversión LDC con un valor de 1484.8.

LENGUAJE	**NIVEL**	**FACTOR LDC/PF**
XAMPP	8	128
MYSQL	5	64
JAVASCRIPT	6	53
PHP	11	36

Tabla 5.1 Conversión de Puntos de Función a KLDC
[Fuente: Elaboración Propia (Argumentos de la empresa)]

LDC=PF Obtenido * Factor LDC/PF $\Longrightarrow$ LDC=74.24 *36 $\Longrightarrow$ LDC = 2672,64

KDL= (2672,64/1000) $\Longrightarrow$ KLDC=2.67

Aplicando las fórmulas de esfuerzo, el tiempo calendario y personal requerido.

$E= a_{b\,(KLDC)}{}^{b}{}_{b}$; $D= c_{b}\,(KLDC)^{d}{}_{b}$

Dónde:

E: Esfuerzo aplicado en personas por mes.

D: Tiempo de desarrollo en meses cronológicos

KLDC: Número de líneas de código distribuido en miles.

Proyecto Software	a_b	c_b	b_b	d_b
Orgánico	2,4	1,05	2,5	0,38
Semi-acoplado	3	1,12	2,5	0,35
Empotrado	3,6	1,2	2,5	0,32

Tabla 5.2 Constantes del COCOMO

Fuente (PRESSMAN, 2010)

En este caso el sistema desarrollado estaría entre los semi-acoplados, debido a que el proyecto de tamaño y complejidad intermedia.

$E= 3 * (2,67)^{1.12} \Longrightarrow E= 9,01$ $D=2.5 * (6.01)^{0.35} \Longrightarrow D= 5,4$

Por tanto el personal requerido es:

Número de programadores=E/D= 9.01/5.4=1,67

En conclusión se requieren 1 programadores.

Si el salario que percibe un programador es de 3000, entonces se tiene:

Costo de desarrollo=número de programadores* por salario de programador

Costo de desarrollo= 1*3000=3000 Bs/mes

Ahora por los 4 meses de desarrollo se tiene

Costo total del software =3000 *4

Por lo tanto el costo del sistema podría ascender máximamente a los 12000.

5.2 CALCULO DE LOS BENEFICIOS CON EL VAN Y TIR

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

VAN

El VAN (Valor Actual Neto) es una técnica de análisis de inversión que mide los flujos de los futuros, ingresos y egresos que tendrá el proyecto, para determinar si después de la inversión inicial se obtiene ganancias. Si el resultado es (+) positivo, el proyecto es viable.

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

$$VAN = I_0 + \sum_{i=1}^{n} \frac{C-P}{(1+r)n}$$

Dónde:

I_0 = Inversión Inicial
C= P= Flujo de Caja (Cobros-Pagos)
n= Periodos
r= Tasa de Actualización

En resumen:

VAN= Beneficio Neto Actualizado-Inversión Inicial

La regla VAN, indica que decisión tomar:

- Si VAN ≥ 0, se debe aceptar.
- Si VAN = 0, se debe ser neutral.
- Si VAN ≤ 0, se debe rechazar.

Los valores para calcular el VAN del proyecto son:

- Inversión Final=12000 Bs.?
- Tasa de Actualización = o tipo de interés está dada por el Banco Central de Bolivia con la venta de bonos a plazo fijo es igual a 4%.
- Flujo de caja=está de acuerdo al siguiente detalle
- Periodos=4

GASTOS	RECUPERACION	
DETALLE	Mes	Semestre
Horas extra	800	4800
Material de trabajo	500	3000
TOTAL	1300	7800

Tabla 5.3 Costos recuperados por periodo

[Fuente: Elaboración Propia (Argumentos de la empresa)]

Los valores de ganancia para el presente proyecto se calculan para 2 años esperando ganar 7800 Bs por cada 6 meses.

Flujo de caja	1° semestre	2° semestre	3° semestre	4° semestre
	7800 Bs.	7800 Bs.	7800 Bs.	7800 Bs.

Aplicando la formula tenemos:

$$VAN= \frac{7800}{(1-0,04)^1} + \frac{7800}{(1-0,04)^2} + \frac{7800}{(1-0,04)^3} + \frac{7800}{(1-0,04)^4} - 12000$$

VAN=28313,18-12000

VAN=16313,18≥ 0, se debe Aceptar

Por tanto la ganancia esperada en 2 años de trabajo con el sistema es de 4.313,18 Bs, con lo que se concluye que el proyecto es viable, ya que genera ganancias para la institución.

TIR

El TIR (Tasa Interna Retorno) es la tasa de descuento TD de un proyecto de inversión que permite que el Beneficio Neto Actualizado sea igual a la inversión inicial (VAN=0). La TIR es la máxima Tasa de Descuento que puede tener un proyecto para que sea rentable:

La regla TIR, indica que decisión tomar:

- Si la TIR ≥ TD, se debe aceptar.
- Si la TIR = TD, se debe ser neutro.
- Si la TIR ≤ TD, se debe rechazar.

Entonces para hallar la TIR tenemos la inversión inicial de 24000Bs.

Flujo de caja	1° semestre 7800 Bs.	2° semestre 7800 Bs.	3° semestre 7800 Bs.	4° semestre 7800 Bs.

Se usa la fórmula del VAN, solo que igualado a 0:

$$0= \frac{7800}{(1-i)^1} + \frac{7800}{(1-i)^2} + \frac{7800}{(1-i)^3} + \frac{7800}{(1-i)^4} - 12000$$

i= 11%

TIR=11% ≥4%.se debe Aceptar

La cual indica que el proyecto puede retornar hasta el 11% de la inversión cada 6 meses.

Los beneficios calculados en dinero repercuten en beneficios cualitativos como ser, ahorro de tiempo y control de la información.

CAPITULO 6

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

El desarrollo del Sistema, llego a su conclusión de una manera satisfactoria, cumpliendo con todos los requisitos especificados.

La implementación del Sistema mejora el desempeño de los funcionarios con respecto al manejo de documentos, reduce la carga de trabajo a los mismos y ofrece confiabilidad en el manejo de la información correspondiente a los documentos.

El sistema registra los datos necesarios para salvaguardar los recursos de correspondencia con ello llegamos a los objetivos necesarios y hasta los alcances que se propuso en anteriores capítulos.

Por lo anteriormente indicado, se ha desarrollado e implementado un "Sistema de Información de Gestión y Control de Correspondencia " para nuestro caso de estudio, en el cual se automatizan las tareas de recepción, control, seguimiento y remisión de correspondencia, de tal manera que la información que se proporciona es más eficiente, precisa, oportuna e inmediata.

RECOMENDACIONES

El sistema se debe registrar todos los datos necesarios que son los datos ya adquiridos anteriormente que fueron salvaguardados en los registros a partir de años pasados.

Realizar controles a los accesos de los usuarios, a las plataformas de procesamiento informático y a los datos que gestionan para evitar irregularidades de confidencialidad, exactitud y disponibilidad de la información.

Realizar procesos programados de backups con el fin de salvaguardar la base de datos del sistema.

UNIVERSITAS MAJOR PACENSIS DIVI ANDREÆ

BIBLIOGRAFIA

(IBAÑEZ, 2009) IBAÑEZ APAZA, ALEIDA RAQUEL, Chasqui digital e correspondencia. Caso: Facultad de ciencias Puras y Naturales, UMSA, Edición: 2009.

(QUISBERT, 2009) QUISBERT BUSTAMANTE, LAURA, Sistema de gestión, control y monitoreo de procesos utilizando tecnologías workflow centro de multiservicios educativos CEMSE. UMSA, Edición: 2009.

(MICROSOFT, 2011) MICROSOFT, guía de Arquitectura N-Capas ADO.NET 4.0, Editorial: Krassis Press, Edición: 2011.

(ROSSI & PASTOR & SCHWABE, 2008) GUSTAVO ROSSI, OSCAR PASTOR, DANIEL SCHWABE, LUIS OLSINA, Web Engineering: Modelling and Implementing Web Applications, Edicion: 2008, ISBN 978-1-84628-922-4.

(LARMAN, 2004) LARMAN, CRAING, Uml y Patrones, Editorial: Prentice Hall, Edicion: 2004, ISBN: 8420534382.

(PRESSMAN, 2010) PRESSMAN, R. Ingenieria Del Software, Editorial: McGraw Hill, Edicion Septima 2010, ISBN: 978-607-15-03145.

(IAN SOMMERVILLE, 2005) Ingeniería del software. Séptima edición. Traducción. María Isabel Alfonso Galipienso. Antonio Botía Martínez. Francisco Mora Lizán, ISBN 84-7829-074-5

(HITPASS, 2011) HITPASS. BERNHARD. FREUND, JAKOB. RUECKER, BERND. BPMN 2.0 Manual de Referencia y guía Práctica, Edición: 2011, ISBN: 9781460903933.

(KOCH, 2011) Una propuesta orientada a objetivos para el análisis de requisitos en RIAs para Lenguajes y Sistemas Informáticos, Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos. Edición 2011.

WEBGRAFÍA

[WEB 1] (METODOLOGIA ARCHIVISTA) LINK: https://mundoarchivistico.wordpress.com/2012/09/30/metodologia-archivistica/

[WEB 2] (METODOLOGIA AGIL) LINK: http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/

[WEB 3] (COSTOS Y BENEFICIOS) LINK: https://www.educoas.org/Portal/bdigital/contenido/interamer/BkIACD/Interamer/Interamerhtml/Riverahtml/riv_zav_villa.htm

[WEB 4] (CORRESPONDENCIA) LINK: http://www.tipos.co/tipos-de-correspondencia/

[WEB 5] (METODOLOGIA AGIL) LINK: http://www.cyta.com.ar/ta0502/v5n2a1.htm

[WEB 6] (METODOLOGIA AGIL AUP) LINK: http://ingenieriadesoftware.mex.tl/63758_AUP.html

[WEB 6.1] (METODOLOGIA AGIL AUP) LINK: http://www.cc.una.ac.cr/AUP/html/milestones.html

[WEB 7] (UWE (UML-BASE WEB ENGINEERING) LINK: http://proyectogradoingenieriasistemas.blogspot.com/2015/03/metodologia-uwe-uml-uml-based-web.html

[WEB 7.1] (HERRAMIENTA UWE (UML-BASE WEB ENGINEERING) LINK: http://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html

[WEB 8] (TECNOLOGIA WORKFLOW) LINK: http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/administracion/v03_n6/tecnologias.htm

[WEB 9] (XAMPP) LINK: http://julianvelasquez7546gta.blogspot.com/2011/07/definicion-de-xampp.html

[WEB 10] (CARACTERISTICAS XAMPP) LINK: http://myu-charly.blogspot.com/

WEB [11] (MYSQL) LINK: http://www.definicionabc.com/tecnologia/mysql.php

[WEB 12] (LENGUAJE DE PROGRAMACION PHP) LINK: http://redgrafica.com/El-lenguaje-de-programacion-PHP

[WEB 13] (LENGUAJE DE PROGRAMACION JAVASCRIPT) LINK: http://www.infor.uva.es/~jmrr/tgp/tgprecurso/intro1.htm

[WEB 14] (SEGURIDAD) LINK: http://www.lawebera.es/como-hacer/ejemplos-php-mysql/seguridad-registro-usuarios-manejo-sesiones-php-i.php

[WEB 14.1] (SEGURIDAD MEDIANTE SESIONES SEGURAS) LINK: https://www.tarlogic.com/blog/como-generar-sesiones-en-php-de-forma-segura/

ANEXOS

Figura: Árbol de problemas

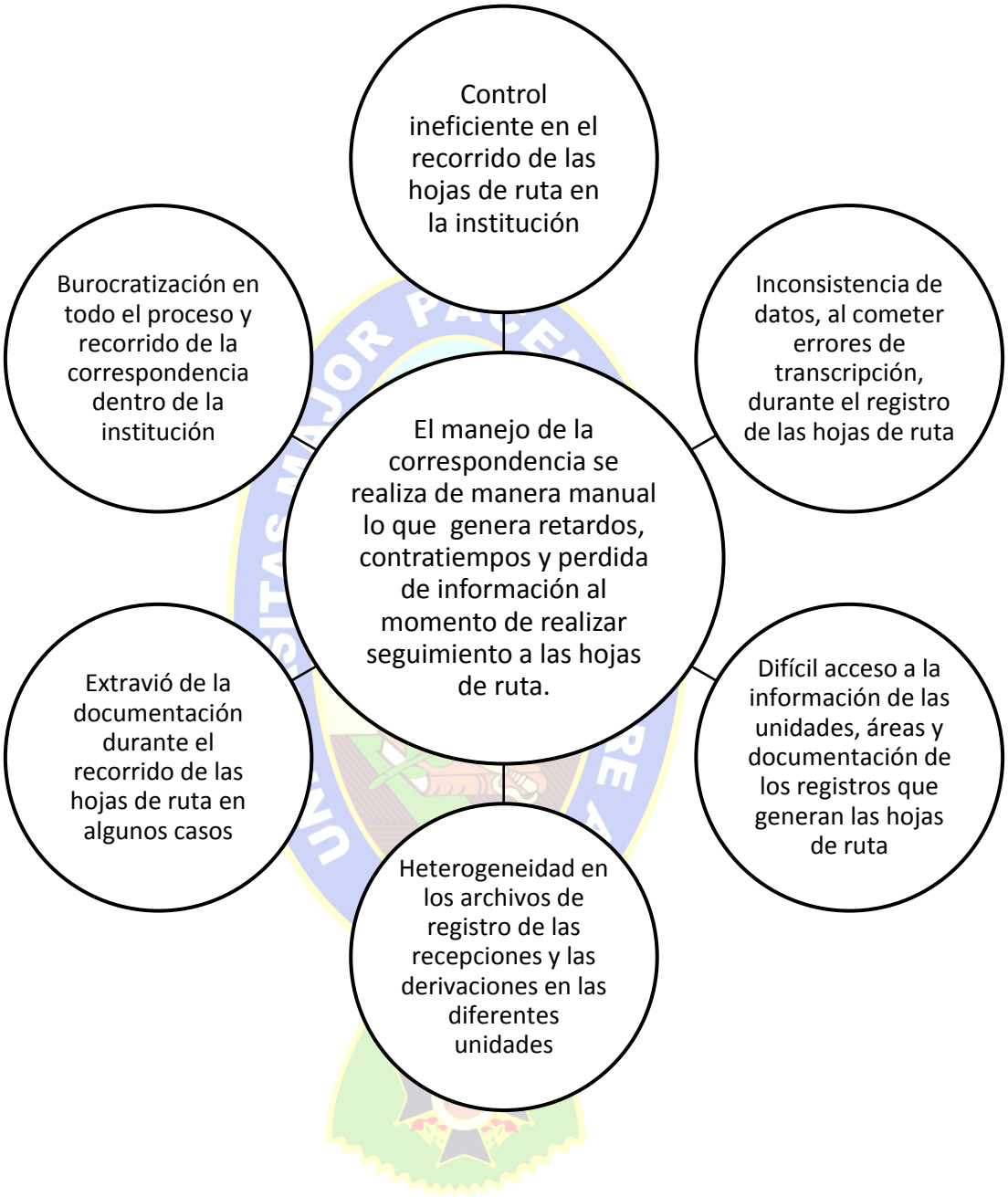

Figura: Árbol de Objetivos

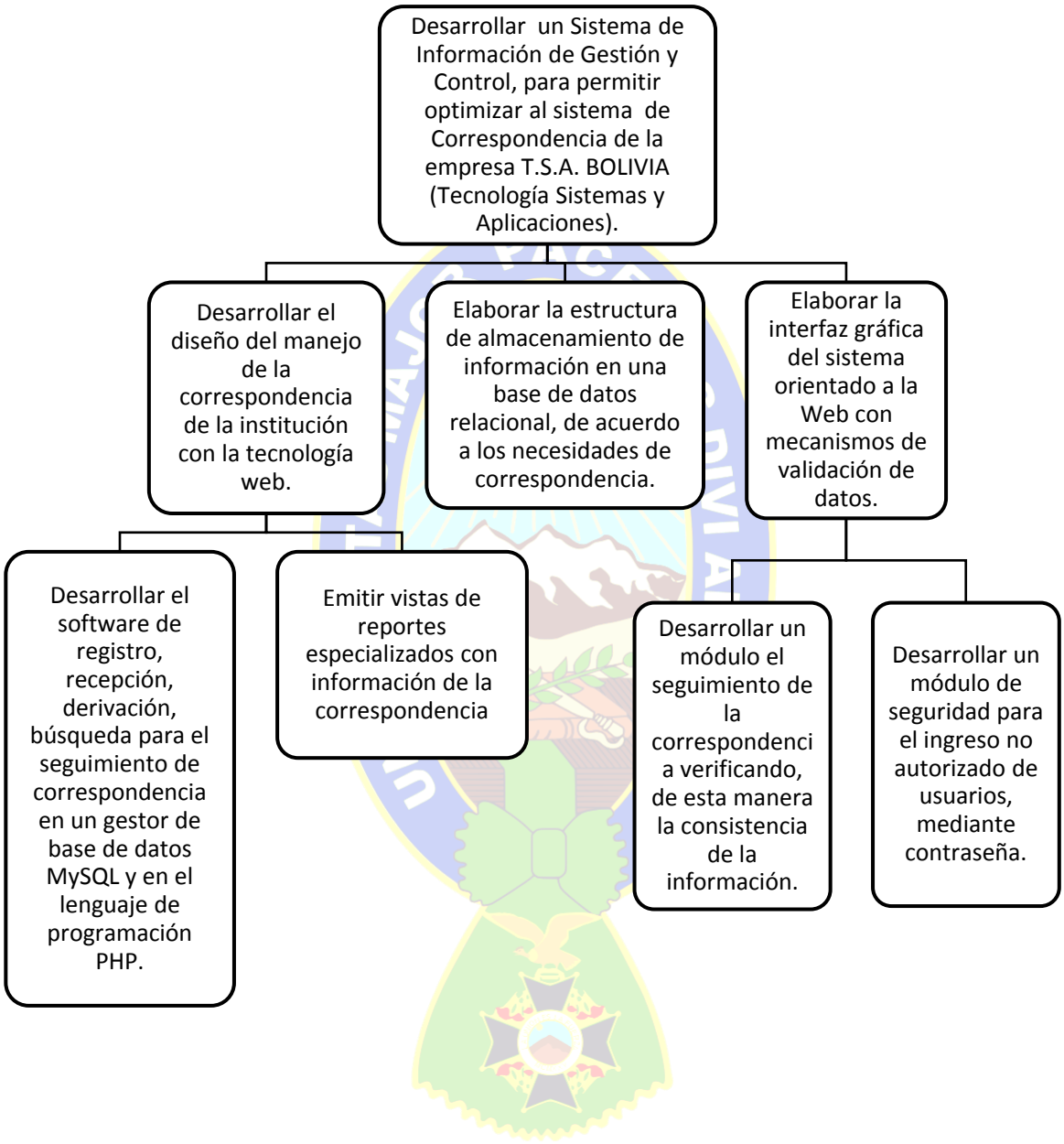

Figura: Cronograma de actividades y Diagrama de Gantt

	PLAN DE PROYECTO DE GRADO SISTEMA DE CONTROL Y CORRESPONDENCIA																								
	20-04-2015	20-04-2015	20-04-2015	20-04-2015	20-05-2015	20-05-2015	20-05-2015	20-06-2015	20-06-2015	20-06-2015	20-07-2015	20-07-2015	20-07-2015	20-07-2015	20-08-2015	20-08-2015	20-08-2015	20-09-2015	20-09-2015	20-09-2015	20-09-2015	20-10-2015	20-10-2015	20-10-2015	20-10-2015	20-10-2015
PERFIL DE TESIS																									
CORRECCIONES																									
RECABAR INFORMACIÓN																									
ELABORACIÓN																									
REVISIÓN																									
PROYECTO FINAL																									
SISTEMA																									

Figura: Organigrama de la Empresa