

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



PROYECTO DE GRADO

“ADMINISTRACIÓN Y SOLICITUD DE PASAPORTES”

PARA LA CANCELLERÍA DE BOLIVIA
MINISTERIO DE RELACIONES EXTERIORES Y CULTO

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

POSTULANTE : CAMACHO CENTELLAS SANDRO SAÚL

TUTOR : MsC. Lic. MARIO LOAYSA MOLINA

REVISOR: JUN GONZALO CONTRERAS

LA PAZ BOLIVIA
2009

RESUMEN

Toda persona de nacionalidad boliviana, que viaje de Bolivia a otro país ya sea por un periodo corto o largo, la Cancillería le emite un pasaporte para su identificación en ese país. Si esta persona por alguna razón en cuanto a su pasaporte (pérdida, robo, expiración, etc.) no la posee, debe apersonarse al Consulado de Bolivia y realizar la Solicitud de pasaporte.

Los Consulados de Bolivia una vez que reciben las solicitudes, estas son enviadas a la Cancillería de Bolivia para su comprobación de la información recibida de cada solicitud coadyuvando con Emigración para la aceptación o rechazo, una vez aceptado los Consulados proceden a la emisión de dichos pasaportes mediante un inventario de pasaportes obtenidos de La Cancillería.


Internet es una herramienta de negocio y comunicación a grandes distancias, por lo cual este sistema está orientado a la Web, teniendo este marco de referencia se optó por las metodologías para la Web y se eligió a OOHDM como metodología de desarrollo centrado en el diseño navegacional y en las vistas abstractas para el nuevo sistema llamado **Administración y Solicitud de Pasaportes**

Este **sistema** será para la Cancillería, Consulados y se implementara en área de Sistemas de la Cancillería, brinda un servicio de registro, seguimiento y rastreo de las solicitudes realizadas por los Consulados y aceptadas por la Cancillería.

1. CAPAITULO I	6
1.1 INTRODUCCIÓN	6
1.2 ANTECEDENTES	7
1.2.1 Reseña histórica de la cancillería.....	7
1.3 PLANTEAMIENTO DE LA INVESTIGACIÓN.....	9
1.3.1 Planteamiento del problema.....	9
1.3.2 Formulación del problema.....	10
1.4 OBJETIVOS DE LA INVESTIGACION.....	10
1.4.1 Objetivo general.....	10
1.4.2 Objetivos específicos.....	11
1.4.3 Limites y alcances.....	11
1.5. JUSTIFICACIÓN DE LA INVESTIGACION	14
1.5.1 Justificación del proyecto de grado conjunto	14
1.5.2 JUSTIFICACION PRÁCTICA.....	14
2. CAPITULO II.....	17
2.1 Marco teórico	17
2.1.1 Introducción	17
2.1 DEFINICIONES.....	17
2.1.1 INTERNET.....	17
2.1.2 INTRANET/EXTRANET	17
2.1.3 WORLD WIDE WEB	18
2.1.4 EL PROTOCOLO HTTP	18
2.1.5 SITIO WEB	19
2.1.6 PAGINA WEB	19
2.1.7 APLICACIONES WEB	19
2.1.8 Atributos.....	19
2.8 METODOLOGÍA.....	22
2.8.1 OOHDM.....	22
2.8.2 Comparativa	25
2.9 Herramienta de desarrollo	31
2.9.1 El UML.....	31
2.9.2 Diagramas del UML	31
2.9.3 Casos de uso.....	34

2.9.4 ACTORES.....	34
2.9.4.3 Casos de uso con formato expandido.....	35
2.9.6 Modelado de requisitos.....	39
2.9.7 Diagrama de clases.....	39
2.9.8 Diagrama de despliegue.....	43
2.10. Fase de construcción.....	44
2.10.1 Etapa de análisis.....	44
2.10.2. Modelo conceptual.....	47
10.6 Herramientas de implementación.....	51
2.11. Métricas.....	52
2.11.1 Confiabilidad.....	52
2.11.2 Funcionalidad.....	52
2.11.3 Mantenibilidad.....	53
2.11.4 Portabilidad.....	54
3. CAPITULO III.....	56
3.1 ANÁLISIS Y DISEÑO DEL SISTEMA.....	56
3.1.1 ANÁLISIS DE LA SITUACIÓN ACTUAL.....	56
3.2.1 FASE 1: DISEÑO CONCEPTUAL.....	60
3.2.2 Modelo casos de uso.....	62
3.2.1 Curso Tipo de Evento.....	68
3.2.2 Diagramas de Secuencia.....	73
4. CAPITULO IV.....	77
4.1 CALIDAD DEL SOFTWARE.....	77
4.1 CONFIABILIDAD.....	77
4.2 FUNCIONALIDAD.....	79
CAPITULO V.....	82
5.1 CONCLUSIONES.....	82
5.2 RECOMENDACIONES.....	82
BIBLIOGRAFÍA.....	84
ANEXOS.....	85
SITUACIÓN DEL SISTEMA ACTUAL.....	86
Glosario de Términos.....	88

INTRODUCCIÓN



En este Capítulo se realiza una Introducción, desarrollando los siguientes puntos: Planteamiento del problema, objetivos, justificación y alcances del proyecto.

1.2 ANTECEDENTES

El tema surge al advertir las grandes necesidades y falencias del sistema actual en la Cancillería, ya que los procesos que se llevan a cabo, se efectúan manualmente con la subsecuente aparición de errores en las solicitudes de los pasaportes, y un ineficiente control de las mismas, dichos problemas convierten a estos procesos lentos en la obtención de información correcta y precisa.

1.2.1 Reseña histórica de la cancillería

A partir de la creación de la República y durante casi 70 años, el área de relaciones exteriores formó parte del Ministerio del Interior que contaba con un Departamento de Relaciones Exteriores. Se considera al Cnl. Facundo Infante, Ministro del Interior en 1826 como el primer "Encargado de Relaciones Exteriores" del país. Posteriormente, adoptó la denominación de Ministerio de Instrucción y Relaciones Exteriores. En esa época se dictaron algunas disposiciones legales sobre el Servicio Exterior, como la Ley de Sueldos y la Lista Diplomática de 1844. Bolivia mantuvo misiones especiales en países vecinos y excepcionalmente en naciones de Europa como Francia, Inglaterra, España y El Vaticano.

En 1884 fue creado el Ministerio de Relaciones Exteriores y Culto durante la presidencia de Gregorio Pacheco, siendo su primer Ministro titular el Dr. Jorge Oblitas. La sede de sus funciones fue la capital de la República, Sucre. El 15 de enero de 1885, mediante Decreto Supremo, se precisaron las funciones del nuevo Ministerio que inicialmente contaba con seis funcionarios, el Ministro, el Oficial Mayor, un Oficial Primero, dos auxiliares y un portero.

El 17 de diciembre del mismo año se designó como segundo Ministro titular a Juan Crisóstomo Carrillo, en cuya gestión se elaboró el proyecto de la Ley Orgánica del Servicio Diplomático promulgada el 20 de noviembre de 1886. El 27 de enero de 1887 se aprobó el Reglamento de la Organización del Servicio Diplomático que dio

origen al primer Escalafón Diplomático, por el cual se dispuso el ingreso del personal mediante exámenes y se reguló el uniforme diplomático. Ese mismo año se aprobó el Reglamento Consular.

En 1900 luego de la Guerra Federal, el Ministerio de Relaciones Exteriores, junto al Gobierno Central, trasladó la sede de sus funciones a la ciudad de La Paz. En 1910, se dictó un nuevo Reglamento Diplomático, que modernizó las prácticas de la Cancillería y dos años más tarde entró en vigencia el Reglamento de Ceremonial del Estado.

El 2 de mayo de 1922 se conformó el Consejo Consultivo del Ministerio de Relaciones Exteriores y Culto, integrado por ex Ministros Plenipotenciarios y destacadas personalidades nacionales. En 1925 se dictaron nuevas disposiciones para el Reglamento Diplomático y el Reglamento de Ceremonial del Estado. En ese mismo año, mediante Decreto Supremo, se creó la Condecoración del Cóndor de los Andes.

En 1954 se promulgó el Decreto Supremo 3710 que puso en vigencia al nuevo Estatuto Orgánico del Servicio de Relaciones Exteriores. El mismo año se iniciaron las actividades del "Instituto de Estudios Internacionales Antonio Quijarro". En 1975 se creó la Academia Diplomática Boliviana "Rafael Bustillo" para la formación del personal diplomático boliviano, con el fin de establecer una carrera diplomática permanente. El reglamento de esta Academia fue aprobado mediante Resolución Suprema 505166 de 17 de octubre de 1988.

El 11 de julio de 1989 fue aprobado el Estatuto del Servicio de Relaciones Exteriores mediante Decreto Supremo 22242. Asimismo, mediante Decreto Supremo 22243 se aprobaron los Reglamentos General, del Escalafón Diplomático Nacional, Consular y de Ceremonial Diplomático. El 15 de febrero de 1993 el Congreso Nacional sanciona la Ley N°1444 del Servicio de Relaciones Exteriores promulgada como Ley de la República el 9 de marzo de 1993.

1.3 PLANTEAMIENTO DE LA INVESTIGACIÓN

1.3.1 Planteamiento del problema

1.3.1.1 Problema central

P1: La comunicación y la mala administración de las Solicitudes de Pasaportes de El Ministerio de Relaciones y Cultos de la Cancillería y de los Consulados de Bolivia es deficiente, debido a la no existencia de un Sistema Informático que integre la información de las solicitudes de pasaportes de los Consulados de Bolivia con la Cancillería, esto afecta diariamente a los ciudadanos de Bolivia que residen en otros Países, al solicitar la tramitación del expendio de pasaporte.

1.3.1.2 Problemas secundarios

P2: La información llega diariamente entre 300 a 400 solicitudes por día en documentos de Excel, que estos documentos presentan varios formatos, debido a que no existe una coordinación y manejo de un solo formato estándar de los archivos que manejan los Consulados de Bolivia.

P3: Se reciben llamadas de las personas que solicitan estos trámites, pidiendo la información del estado de su trámite, respecto al papeleo entre otros, alcanzando un 20% del total de llamadas por día.

P4: Al manejar la información en archivos Excel se tiene las limitaciones en la capacidad de almacenamiento de la información, se dividen en varios sub archivos de cada consulado, los mismos son enviados vía correo electrónico, estos datos son

vulnerables a ser vistos por otras personas en un mayor porcentaje, se observo que en más de un caso se encontró la DUPLICIDAD de números de Cédulas de identidad y de solicitudes.

P5: La verificaron y Observación de la información de cada persona que solicita el Pasaporte, lo realizan la Policía y/o Migración, dando a conocer si el trámite es correcto o es observado si de ser así esta persona tiene que realizar otros trámites hasta que no tenga observaciones como ser: Los datos no corresponde, la información es de otra persona, etc. De esta manera no pueden corregir los errores en ese mismo trámite, la persona es afectada una o dos semanas dependiendo del tipo de trámite.

1.3.2 Formulación del problema

¿El Sistema de Administración de Solicitud de Pasaportes para la Cancillería coadyuvará en los procesos de registro envío recepción y verificación de las Solicitudes de Pasaportes juntamente a la emisión de las mismas proporcionando información eficiente y oportuna?

1.4 OBJETIVOS DE LA INVESTIGACION

1.4.1 Objetivo general

Diseñar e implementar un Sistema de Información computarizado vía Web para las Solicitudes de pasaportes y control de emisión de las mismas tanto en la Cancillería y en los consulados de Bolivia.

1.4.2 Objetivos específicos

- Gestionar en forma automática los procesos de Solicitud de Pasaportes permita agilizar las tareas realizadas en Los Consulados.
- Obtener información detallada sobre el tratamiento que se realizó dentro las Solicitudes, Pasaportes enviados a los Consulados y los Pasaportes emitidos a Personas.
- Realizar el registro de Personas que solicitan la emisión del pasaporte
- Realizar la emisión del Pasaporte de las personas verificadas y aceptadas.
- Implementar el módulo Registro y Control de Solicitudes que permita realizar la verificación de los datos de Personas al Régimen Consular de manera rápida y confiable.
- Implementar el módulo de Consulta de estado de la Solicitud de Persona para que realicen su seguimiento
- Automatizar el proceso de las emisiones de Pasaportes para facilitar el trabajo en los Consulados de Bolivia y así mejorar el tiempo de respuesta al emitir un pasaporte.
- Realizar un seguimiento detallado de las Solicitudes y Pasaportes emitidos.

1.4.3 Límites y alcances

El alcance del proyecto estará delimitado en la Cancillería, con un buen análisis y diseño de la situación actual y la posterior implementación de un sistema computarizado vía Web, hará que los servicios que ofrezca el sistema para la administración, envío, revisión y emisión de Pasaportes sea de satisfacción dentro La Cancillería, Consulados de Bolivia y de los ciudadanos Bolivianos que residen en otros países.

Se tomará en cuenta los registros, listados, búsquedas y reportes los cuales brindarán un mejor control y seguimiento de las Solicitudes realizadas, Pasaportes

enviados y Pasaportes emitidos, pudiendo realizar informes completos respecto de las mismas, cada vez que lo requieran.

Los servicios que ofrecerá el sistema están circunscritos por:

- ***Módulo Administrar Solicitudes (Lado de los Cónsules)***

Crea un grupo (CITE) de Solicitudes

Registra Las solicitudes por grupo "CITE" que realizan los Consulados de Bolivia a la base de datos.

Realiza una edición (Modifica, Elimina) cada solicitud y del grupo en si.

Envío del grupo de solicitud por un "Cite" para su revisión.

- ***Módulo Administrar Solicitudes (Lado de Régimen Consular)***

Realiza la consulta de las solicitudes enviadas por los cónsules.

Realiza la Impresión de las Solicitudes en formato de Excel para el envío a Migración para su revisión.

Carga un archivo en Excel proporcionada por Migración y procede a la verificación de los datos de las Solicitudes.

- ***Módulo Administrar Pasaportes (Lado de los Cónsules)***

Recibe Pasaportes enviado por el Régimen Consular

Lista las Todas las Solicitudes Aceptadas

Lista Pasaportes enviadas por Régimen Consular

Emite Pasaporte de dichas Solicitudes

- **Módulo Administrar Pasaportes (Lado del Régimen Consular)**

Ingresa pasaportes

Crea un grupo de pasaportes

Envía grupo de pasaportes a los Consulados

- **Módulo de Búsquedas**

Realiza búsquedas de:

Solicitudes

Pasaportes

Personas

- **Módulo de Reportes**

Seguimiento de Solicitudes

Seguimiento de Personas

Seguimiento de Pasaportes

1.5. JUSTIFICACIÓN DE LA INVESTIGACION

1.5.1 Justificación del proyecto de grado conjunto

Tomando en cuenta el estudio del sistema actual que tiene como procesos principales de Solicitud Pasaportes, verificación de Solicitudes, los cuales están ligados estrechamente con los Consulados de Bolivia mediante el proceso de envío y emisión de pasaportes, se pudo advertir la amplitud y complejidad del trabajo que comprende el análisis, diseño, desarrollo e implementación de un sistema de información que permita la emisión de un Pasaporte en el menor tiempo posible y tener un control óptimo de las mismas.

1.5.2 JUSTIFICACION PRÁCTICA

1.5.2.1 Justificación técnica

El desarrollo del sistema brindará beneficios al Ministerio de Relaciones Exteriores y Cultos de la Cancillería, aportando con el uso de la tecnología informática, el cual permitirá una comunicación en línea, además de un adecuado manejo de la información de manera fácil y amigable para los Consulados de Bolivia, anulando la duplicidad de trabajo, reduciendo el tiempo y el esfuerzo para la obtención de información.

La Oficina de Sistemas de la Cancillería en sus procesos operativos de registro, envío y entrega de información, cuenta con varios Servidores conectados a Internet para la implementación del sistema, se lo implementara en el Portal del

Ministerio.

1.5.2.2 Justificación económica

Dentro del área de sistemas se cuenta ya con un Portal del Ministerio de la Cancillería lo que conlleva que la Inversión para la implementación del sistema sea mínima comparada con los gastos que se estuvo realizando, al no contar con un sistema de información que los integre, ya que actualmente se gasta en el Fax, viáticos del personal, línea telefónica fija, celulares e impresiones. Para la implementación del sistema la mayoría de estos servicios se dejara de utilizar del modo como se lo iban utilizando.

1.5.2.3 Justificación social

Establecerá una nueva forma de trabajo beneficiando al personal de la Cancillería y a los consulados de Bolivia mejorando la calidad de servicio y como consecuencia la mejora de la imagen de todos los Consulados y de la Cancillería, pero sobre todo a los bolivianos que residen en otros países.

En esta Capitulo se realiza una descripción de la metodología, método y herramientas que se uso para el desarrollo del proyecto.

2. CAPITULO II

2.1 Marco teórico

2.1.1 Introducción

El proyecto de grado es una aplicación Web, el mismo que esta asociado a la plataforma .Net que interactúa con varios componentes como la Internet, Intranet, Metodologías, etc.

El mundo de la informática no para de hablar de procesos de desarrollo, el *modo* de trabajar eficientemente para evitar catástrofes que llevan a que un gran porcentaje de proyectos se terminen sin éxito.

El proceso de desarrollo de software requiere, por un lado, un conjunto de conceptos, una metodología y un lenguaje de desarrollo.

2.1 DEFINICIONES

2.1.1 INTERNET

La Internet, algunas veces llamada simplemente "la Red", es un sistema mundial de computadoras interconectadas; una red de redes en la que usuarios en cualquier computador pueden, debidamente autorizados, obtener información de cualquier otro computador (o conversar directamente con usuarios en otros computadores).

2.1.2 INTRANET/EXTRANET

Una Intranet es una red privada contenida dentro de una empresa. Puede consistir de diversas redes de área local enlazadas y también pueden hacer uso de líneas arrendadas en redes de área extensa. Típicamente, una intranet incluye conexiones a través de una o más compuertas de enlace a la Internet.

El propósito principal de una Intranet es el de compartir información de la compañía y recursos de cómputo entre sus empleados. Una intranet también puede ser utilizada para facilitar el trabajo en grupo y para videoconferencias. Una Intranet también hace uso de TCP/IP, HTTP, y otros protocolos de Internet.

Cuando parte de una intranet se hace accesible a empleados, socios, proveedores, u otros agentes externos a la empresa, esa parte es denominada parte de una extranet.

2.1.3 WORLD WIDE WEB

Según la organización W3C ("About the World Wide Web", 2001) el servicio conocido como "WWW" es un universo de información accesible mediante red. la materialización del conocimiento humano

2.1.4 EL PROTOCOLO HTTP

Pressman (2002, p 499) define el protocolo HTTP como uno de los protocolos más importantes que se utilizan dentro de Internet, es el protocolo que dirige la comunicación entre un cliente que utiliza un navegador Web tal como Internet Explorer y un servidor Web.

La función principal de un servidor Web es poner páginas Web a disposición de clientes. Estos clientes utilizarán un navegador que les conectará con el puerto 80 en el servidor Web. este es el puerto estándar que se utiliza para tales servidores. El navegador que está utilizando el cliente enviará mensajes definidos por el protocolo HTTP y serán interpretados por el servidor que llevará a cabo operaciones tales como devolver una página Web o procesar algún formulario que este insertado dentro de una página.

2.1.5 SITIO WEB

Un sitio Web es una colección de archivos Web con un tema particular que incluyen un archivo de inicio llamado página principal. Desde la página principal, se pueden acceder todas las otras páginas en el sitio. Un sitio Web no debe confundirse con un servidor Web.

2.1.6 PAGINA WEB

Para entender cómo funcionan las aplicaciones Web es necesario comprender cómo se realiza una página Web, ya que la forma dinámica de éstas son el medio de presentación hacia el usuario que utilizarán las aplicaciones Web.

2.1.7 APLICACIONES WEB

Los sistemas y aplicaciones basados en Web hacen posible que una población extensa de usuarios (males dispongan de una gran variedad de contenido y funcionalidad) se ejecutan a través de la World Wide Web, y entre sus objetivos se encuentran el de consultar fuentes de datos, procesar los formularios de entrada de datos, entre otros, pero primordialmente generan información capaz de ser visualizada por un navegador Web. Es conveniente conocer la naturaleza de estas aplicaciones Web ya que permite obtener una visión de lo que este proyecto debe hacer posible.

2.1.8 Atributos

Las aplicaciones Web presentan los siguientes atributos:

A) Generales

Intensivas de red Reside en una red y debe dar servicio a las necesidades de una comunidad diversa de clientes Una aplicación Web puede residir en Internet (haciendo posible así una comunicación abierta por todo el mundo) De forma alternativa, una aplicación se puede ubicar en una Intranet (implementando la comunicación a través de redes de una organización) o una extracción (comunicación entre redes)

Controlada por el contenido En muchas casas, la función primaria de una aplicación Web es utilizar hipermedia para presentar al usuario el contenido de textos, gráficos sonido y video

Evolución continua A diferencia del software de aplicaciones convencional que evoluciona con una serie de versiones planificadas y cronológicamente espaciadas, las aplicaciones Web están en constante evolución No es inusual que algunas aplicaciones Web (específicamente su contenido) se actualicen cada hora

Inmediatez Las aplicaciones basadas en Web tienen una inmediatez que no se encuentran en otros tipos de software Es decir el tiempo que se tarda en comercializar un sitio Web completo puede ser cuestión de días o semanas

Seguridad Dado que las aplicaciones Web están disponibles a través del acceso por red. es difícil, si no imposible, limitar la población de usuarios finales que pueden acceder a la aplicación Con objeto de proteger el contenido confidencial y de proporcionar formas seguras de transmisión de datos deberán implementarse fuertes medidas de seguridad en toda la infraestructura que apoya una aplicación Web y dentro de la misma aplicación

Estética Una parte innegable del atractivo de una aplicación Web es su apariencia e interacción Cuando se ha diseñado una aplicación con el fin de comercializarse o vender productos o ideas, la estética puede tener mucho que ver con el éxito del diserto técnico

B) De calidad

Todas las personas que hayan navegado alguna vez por la Web o hayan utilizado una Intranet de una compañía pueden opinar sobre lo que hace una buena aplicación Web. Los puntos de vista individuales varían enormemente. Algunos usuarios disfrutan con gráficos llamativos, en cambio otros solo quieren un texto sencillo. Algunos exigen motivación copiosa, otros desean una presentación abreviada. En efecto, la percepción de 'lo bueno' por parte del usuario podría ser más importante que cualquier discusión técnica sobre la calidad de la aplicación. A continuación se identifica un conjunto de atributos que conduce a aplicaciones Web de alta calidad.

C) Usabilidad

- Capacidad de comprensión del sitio global
- Servicios de ayuda y realimentación en línea
- Capacidades estéticas y de interfaz
- Servicios especiales

D) Funcionalidad

- Capacidad de recuperación y de búsqueda
- Servicios de búsqueda y navegación.
- Servicios relacionados con el dominio de aplicación

E) Fiabilidad

- Proceso correcto de enlace
- Recuperación de errores

- Validación y recuperación de la entrada del usuario.

F) Eficiencia

- Rendimiento del tiempo de respuesta
- Velocidad de generación de páginas.
- Velocidad de generación de gráficos.

2.7.2 Tipos

Se clasifican las aplicaciones que se presentan a continuación como las más frecuentes en el trabajo de la Web.

- **Informativa** se proporciona un contenido solo de lectura con navegación y enlaces simples.
- **Descarga:** un usuario descarga la información desde el servidor apropiado.
- **Personalizada** el usuario personaliza el contenido a sus necesidades específicas.

2.8 METODOLOGÍA

2.8.1 OOHDM

La metodología OOHDM (*Object Oriented Hypermedia Design Method*), propuesta por D Schwabe y G. Rossi, establece que el desarrollo de un Hiperdocumento es un proceso de cuatro fases en el que se combinan diferentes estilos de desarrollo como el incremental, iterativo y prototipado. Las tres primeras fases son de diseño, en las que se obtiene un conjunto de modelos orientados a objeto *que* describen el documento que será construido en la última fase.

Primera fase, denominada Diseño Conceptual o Análisis de Dominio, se realiza, como también ocurría en la metodología EORM. el modelado del dominio del hiperdocumento utilizando algún método análisis orientado a objetos de Sistemas de Información, por ejemplo OMT. obteniendo un *esquema conceptual de clases* en el que, además de clases abstractas y objetos, se representan las relaciones entre ellas, incluidas las de herencia y agregación, y los correspondientes atributos (que pueden ser de cualquier tipo, desde simples cadenas de caracteres a gráficos, imágenes, texto, sonido, etc.) y métodos asociados a las clases. Una vez obtenido el esquema conceptual la metodología OOHDM establece una:

Segunda fase de Diseño Navegacional en la que se ha de definir la estructura de navegación a través del hiperdocumento mediante la realización de modelos navegacionales que representen diferentes vistas del esquema conceptual de la fase anterior. Se trata, en definitiva, de reorganizar la información para adaptarla a las necesidades de los usuarios del sistema. El Diseño Navegacional se expresa, también con un enfoque orientado a objetos, a través de dos tipos de esquemas o modelos: el denominado *esquema de clases navegacionales*, con las posibles vistas del hiperdocumento a través de unos tipos predefinidos de clases, llamadas navegacionales, como son los "nodos", los "enlaces", y otras clases que representan estructuras o formas alternativas de acceso a los nodos, como los "índices" y los "recorridos guiados"; y el *esquema de contexto navegacional*, que permite la estructuración del hiperespacio de navegación en subespacios para los que se indica la información que será mostrada al usuario y los enlaces que estarán disponibles cuando se acceda a un objeto (nodo) en un contexto determinado. La metodología OOHDM contempla una:

Tercera fase de diseño, denominada Diseño de la Interface Abstracta, en la que se realiza un modelo, también orientado a objetos, para especificar la estructura y el comportamiento de la interface del sistema hipermedia con el usuario. Este modelo es abstracto y. por tanto, independiente de la implementación final del sistema. Sin

embargo, se basa en las ideas que actualmente se aplican en las Interfaces Gráficas de Usuario (IGUs), por lo que como la mayor parte de entornos hipertexto comerciales trabajan con IGUs. su implantación en un entorno de este tipo debe ser una tarea sencilla.

El modelo de la interface abstracta se expresa a través de tres tipos de diagramas que se complementan entre si En primer lugar se deben crear los denominados diagramas de *Vistas de Datos Abstractos (ADVs)* [12 que incluyen una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue establecida durante la fase de Diseño Navegacional. Un diagrama de este tipo se compone de una serie de cajas (una caja es un ADV) que representan las diferentes clases de objetos que aparecerán ante el usuario Un segundo tipo de diagramas que componen el modelo de interface es el *Diagrama de Configuración*, donde se representan principalmente los eventos externos (provocados por el usuario, como *ClicDeRatón* o *DobleClic*) que maneja un ADV, los servicios que ofrece el ADV (como "visualización") y las relaciones estáticas entre las ADVs Por último, el modelo se completa con los denominados *Diagramas de Estado* que. como ocurre en los modelos orientados a objetos de ingeniería del software [13, representan el comportamiento dinámico del sistema hipertexto mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles estados por los que puede pasar cada objeto de la interface (*oculto, desactivado, ampliado, reducido, normal, etc*) y los eventos que originan los cambios de estado.

Cuarta fase es la última fase de la metodología OOHDM es la Implementación del hipertexto o sistema hipertexto diseñado, es decir, la concreción de los modelos navegación y de interface en objetos particulares con sus correspondientes contenidos (textuales, visuales sonoros, etc.) y sus posibilidades de navegación. Aunque, al utilizar un enfoque de orientación a objetos podría parecer conveniente que la implementación se hiciera en un entorno de construcción de hipertextos también orientado a objetos, como MacWeb. debido al carácter abstracto del diseño, ésta puede hacerse fácilmente en otros entornos hipertextuales

como Hypercard, DIRECTOR o cualquiera que permita trabajar con el lenguaje HTML, como HTML Write o HTML Assistant.

2.8.2 Comparativa

Una vez enunciadas las propuestas, se van a presentar una serie de estudios comparativos de las mismas. Los estudios que se presentan se basan en tres aspectos.

El primero analiza qué requisitos son cubiertos en cada metodología.

El segundo estudio presenta las fases dentro del proceso de tratamiento de requisitos que cada una afronta y las técnicas que para ello proponen.

El último estudio, evalúa el detalle de cada propuesta a la hora de definir su proceso de desarrollo, las técnicas que aplica y los resultados que propone obtener. Por último, se presentan algunos otros aspectos que han surgido durante la realización de las comparativas.

2.8.2.1 Requisitos tratados

En base a la clasificación de requisitos:

La primera comparativa que se va a realizar de las propuestas estudiadas consiste en ver qué tipos de requisitos contempla cada propuesta. En la tabla se presentan los diferentes requisitos y se indica cuáles de ellos son tratados en cada metodología.

Analizando los resultados, y teniendo en cuenta que las propuestas están ordenadas por orden cronológico, resulta interesante observar el avance que han tenido los

requisitos en el entorno de la Web. Las primeras propuestas estaban más centradas en los requisitos de datos y los de interfaz de usuario.

Observando la tabla puede verse que las propuestas más actuales resaltan la necesidad de tratar los requisitos de personalización y navegación, así como los transaccionales de forma independiente. Esta idea de separación de conceptos se observó desde el principio en las propuestas para la Web (HDM, EORM, OOHDM, etc.) pero solo se planteaba en fases avanzadas del proceso de desarrollo, principalmente en diseño. Se puede observar la tendencia actual de partir de esta separación de concepto ya en la fase de especificación de requisitos.

Requerimiento	De datos	Interfaz de Usuario	Navegación	Personalización	Transaccionales	No Funcionales
WSDM	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
SOHDM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	
RNA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
HFPM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
OOHDM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
UWD	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
W2000			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
UWA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
NDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DDDP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

2.8.2.2 Técnicas y actividades contempladas

Otro estudio interesante surge si se observa en qué fases de las que se presentaron en la tabla 2.1 pone énfasis cada propuesta. Para realizar esta comparativa, se

presentan en la tabla 2.2 las técnicas para capturar, definir y validar requisitos que fueron enumeradas. Para cada propuesta se indica qué técnicas son descriptas o enumeradas de forma explícita en su proceso de desarrollo. En algunos puntos de la tabla se han indicado las técnicas concretas que proponen, si las mismas no son estándares o son variantes de técnicas conocidas.

		WDSM	SOHDM	RNA	HFPM	OOHDM	UWE	W2000	UWA	NDT	DDDP
CAPTURA	ENTREVISTAS	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	JAD									<input checked="" type="checkbox"/>	
	BRAINSTORMING									<input checked="" type="checkbox"/>	
	MAPPING	<input checked="" type="checkbox"/>									
	CASOS DE USO					<input checked="" type="checkbox"/>					
	CUESTIONARIO CHECK LIST						<input checked="" type="checkbox"/>				
	PROTOTIPOS										<input checked="" type="checkbox"/>
	OTRAS TECNICAS		DFD								
DEFINICION	LENGUAJE NATURAL	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>							
	GLOSARIOS			<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	
	PATRONES PLANTILLAS								<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	ECENARIOS		SAO			<input checked="" type="checkbox"/>					
	CASOS DE USO			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	LENGUAJE FORMAL							XML			
	SKETCHES DE INTERFAZ			<input checked="" type="checkbox"/>							
	PROTOTIPOS										<input checked="" type="checkbox"/>
	OTRAS TECNICAS		LIS EVT			UIDS		GRAFO	BNL		
VALIDACION	REVIEWS WALK TROUGHTS						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	AUDITORIAS						<input checked="" type="checkbox"/>				
	MATRIZ TRASABILIDAD								<input checked="" type="checkbox"/>		
	PROTOTIPOS				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>
	OTRAS TECNICAS							GRAFO			

TABLA TECNICAS CONTEPLADAS

PARA CADA TECNICA

De esta tabla comparativa se pueden sacar varias conclusiones. Por un lado, es necesario indicar que dentro de la fase de captura de requisitos, la técnica más enunciada es la de las entrevistas. Por otro, se puede observar que hay muchas propuestas que no recomiendan ninguna técnica concreta para la realización de la captura. Normalmente estas propuestas tratan el proceso de determinación de requisitos como parte de la definición de requisitos.

Con respecto a la definición de requisitos, se puede observar que es el aspecto central del tratamiento de requisitos para todas las propuestas. Se puede concluir en que existe una tendencia a usar la técnica de casos de uso como base Sin embargo, ante esto conviene decir que hay dos opiniones encontradas. Algunas propuestas consideran los casos de uso una técnica óptima para representar los requisitos, como es el caso de UWE o HF'PM. Sin embargo, hay otras como OOHDM o NDT que aunque indican que es una buena técnica, resaltan que es ambigua y que es necesario obtener modelos más concretos para sistematizar más el resto del proceso de ciclo de vida.

Otra conclusión muy relevante que se obtiene de este estudio es el hecho de la poca importancia que las propuestas han prestado a la validación de requisitos. Las técnicas de validación que proponen se basan principalmente en la revisión de los modelos y resultados de la definición de requisitos. La gran mayoría de las propuestas ni siquiera contemplan esta fase en su proceso de ingeniería de requisitos

2.8.2.3 Grado de detalle en las descripciones

El siguiente estudio que se presenta se basa en la forma en que definen las metodologías el tratamiento de requisitos, para ello fueron comparadas las descripciones de las propuestas. La definición de algunas propuestas se basan principalmente en describir de forma concreta el proceso de desarrollo, sin llegar a concretar qué técnicas aplicar o la estructura de los resultados que se deben obtener.

De esta forma, se van a evaluar las propuestas analizando:

- ☑ Si son *orientadas al proceso* o lo que es lo mismo, es si describen un proceso claro y concreto a seguir para realizar la definición, captura y validación de requisitos.
- ☑ Si son *orientadas a la técnica* o dicho de otro modo si describen técnicas concretas a aplicar durante el proceso.
- ☑ Si son *orientadas al resultado*, es decir, si da una descripción concreta del producto que se debe obtener tras el proceso de desarrollo.

Evaluando la definición de las propuestas, se han catalogado cada una de ellas en base a lo detalladas que son a la hora de describir el proceso de definición de requisitos, las técnicas a aplicar o los resultados a obtener. Los resultados de esta evaluación se pueden ver en la tabla 2.3. Hay que tener en cuenta que la valoración se hace solo con respecto a la fase de ingeniería de los requisitos.

El peso dado a cada técnica corresponde con la siguiente evaluación:

- Orientación al proceso: describe claramente los pasos a seguir (+), describe el proceso sin detallarlo (o), no describe ningún proceso (-)
- Orientación a la técnica: describe claramente las técnicas y la forma de aplicarlas (+), enumera las técnicas a aplicar (o), no propone ninguna técnica concreta o referencia a técnicas generales (-).
- Orientación al producto: describe claramente la estructura del producto a obtener (+), describe el contenido del producto sin entrar en detalle de su estructura (o), no comenta nada sobre el producto resultante (-).

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

TABLA ORIENTACION DE LAS PROPUESTAS

2.8.2.4 Resultados

Se ha comenzado planteando la estructura básica de dicho proceso y las técnicas más comunes aplicadas de forma clásica en la ingeniería de requisitos. Tras esto, se han enumerado las propuestas para la Web que cubren esta fase de ingeniería del software y se han comparado desde diferentes puntos de vista.

Como resultado de este estudio podemos afirmar que la metodología y técnica es la más óptima para el diseño de nuestro sistema Web

2.9 Herramienta de desarrollo

2.9.1 El UML

El UML (Lenguaje Unificado de Modelado) es una herramienta para el desarrollo de sistemas. Esto se debe a que esta herramienta nos permitirá que el sistema propuesto genere diseños que capturen las ideas en una forma convencional y fácil de comprender.

El UML es un sistema de notación estándar en el mundo del desarrollo de sistemas. Es el resultado del trabajo hecho por Grady Booch, James Rumbaugh e Ivar Jacobson. El UML está constituido por un conjunto de diagramas, y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Es necesario contar con todos esos diagramas dado que cada uno se dirige a cada tipo de persona implicada en el sistema.

Un modelo UML indica *qué* es lo que supuestamente hará el sistema, mas no *cómo* lo hará.

2.9.2 Diagramas del UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

La explicación de la arquitectura se basará en los diagramas, en lugar de en vistas o anotación, ya que son estos la esencia de UML. Cada diagrama usa la anotación pertinente y la suma de estos diagramas crean las diferentes vistas.

Las vistas existentes en UML son:

- Vista casos de uso: Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- Vista de diseño: Se forma con los diagramas de clases, objetos, colaboración, estados y actividades
- Vista de procesos Se forma con los diagramas de la vista de diseño Recalcando las clases y objetos referentes a procesos
- Vista de implementación Se forma con los diagramas de componentes colaboración estados y actividades
- Vista de despliegue Se forma con los diagramas de despliegue, interacción, estados y actividades

Se dispone de dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica.

Los diagramas estáticos son

- Diagrama de clases: Muestra las clases, interfaces, colaboraciones y sus relaciones son los más comunes y dan una vista estática del proyecto.
- Diagrama de objetos: Es un diagrama de instancias de las clases mostradas en el diagrama de clases Muestra las instancias y como se relacionan entre ellas se da una visión de casos reales

- Diagrama de componentes: Muestran la organización de los componentes del sistema, un componente se corresponde con una o varias clases, interfaces o colaboraciones.
- Diagrama de despliegue Muestra los nodos y sus relaciones Un nodo es un conjunto de componentes se utiliza para reducir la complejidad de los diagramas de Casos y componentes de un gran sistema sirve como resumen e índice
- Diagrama de casos de uso Muestran los casos de uso, actores y sus relaciones muestra quien puede hacer que y relaciones existen entre acciones (casos de uso) Son muy importantes para modelar y organizar el comportamiento del sistema

Los diagramas dinámicos son

- Diagrama de secuencia, Diagrama de colaboración Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.
- Diagrama de estados: muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.
- Diagrama de actividades: Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

Se puede apreciar que el número de diagramas es muy alto, en la mayoría de los casos excesivos, y UML permite utilizar solo los necesarios.

2.9.3 Casos de uso

El caso de uso es una estructura para describir la forma en que un sistema lucirá, como deberá comportarse para los usuarios. Es una colección de escenarios iniciados por una entidad llamada actor.

Un caso de uso debería dar por resultado algo de valor ya sea para el actor que lo inició o para otro de los requerimientos desde el punto de vista del usuario. (Ver Fig.



Figura 4. Icono del UML que representa un Caso de Uso [LARMAN, 1999]

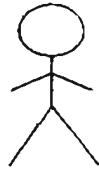
Existen pasos para la identificación de los casos de uso esto requiere una lluvia de ideas y revisar los documentos actuales sobre la especificación de los requerimientos.

Un método con que se identifican los casos de uso se basa en los actores:

- Se identifican los actores relacionados con un sistema.
- En cada actor, se identifican los procesos que inician o en que participan

2.9.4 ACTORES

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes (Ver Fig. 2.9). Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).



Cónsul

Figura 5. Representación de un Actor [LARMAN, 1999]

2.9.4.1 Formato de los casos de uso

Los casos de uso pueden ser expresados en dos formatos:

2.9.4.2. Casos de uso de alto nivel

Permite entender los principales procesos globales, Larman sugiere la estructura de la tabla 2.3.

Casos de uso:	Nombre del caso de uso
Actores:	Lista de actores (Agentes externos), en la cual se indica quien inicia el caso de uso.
Tipo:	Primario, secundario u opcional (a explicar) Esencial o real (a explicar)
Descripción:	Resumen del proceso concerniente al Caso de Uso.

Tabla 2.3. Estructura de casos de alto nivel [LARMAN, 1999]

2.9.4.3 Casos de uso con formato expandido

Muestran más detalle que uno en formato de alto nivel, este tipo de casos permite alcanzar un conocimiento mas profundo del proceso y requerimientos que se

esta tratando, Craig Larman propone el formato que se advierte en la tabla 2.4.

Casos de uso:	Nombre del caso de uso
Actores:	Lista de actores (Agentes externos), en la cual se indica quien inicia el caso de uso
Propósito:	Intención del caso de uso
Resumen:	Repetición del caso de uso de alto nivel o alguna síntesis similar
Tipo:	1. Primario, secundario u opcional 2. Esencial o Real
Referencia cruzada:	Casos relacionados de uso y funciones también relacionadas del sistema.

Curso normal de los eventos

Acción del actor	Respuesta del sistema
Acciones numeradas de los actores.	Descripciones numeradas de las respuestas del sistema.

Cursos alternos

Alternativas que pueden ocurrir en e número de línea. excepciones	Descripción de
--	----------------

Tabla 2.4. Estructura de casos de uso expandido [LARMAN, 1999]

Los casos de uso pueden ser:

- *Casos de uso Primarios.* Representan los procesos más importantes.
- *Casos de uso Secundarios.* Corresponden a los procesos menores o raros.
- *Casos de uso Opcionales.* Estos casos de uso corresponden a los procesos que pueden no abordarse.

- **Casos de uso Esenciales.** Se expresan en forma teórica y son muy abstractas.
- **Casos de uso Reales.** Describen concretamente el proceso a partir de su diseño actual.

2.9.4.5 Diagramas de los casos de uso

Se emplean para visualizar el comportamiento del sistema, una parte de él o de una sola clase De forma que se pueda conocer cómo responde esa parte del sistema. El diagrama de uso es muy útil para definir como debería ser el comportamiento de una parte del sistema, ya que solo especifica cómo deben comportarse y no como están implementadas las partes que define Por ello es un buen sistema de documentar partes del código que deban ser reutilizables por otros desarrolladores El diagrama también puede ser utilizado para que los expertos de dominio se comuniquen con los informáticos sin llegar a niveles de complejidad Un caso de uso especifica un requerimiento funcional, es decir indica lo que una parte debe hacer esto cuando algo suceda.

En el diagrama se dispone de diferentes figuras que pueden mantener diversas relaciones entre ellas:

En el diagrama se dispone de diferentes figuras que pueden mantener diversas relaciones entre ellas:

Casos de uso: representado por una elipse, cada caso de uso contiene un nombre, que indique su funcionalidad. Los casos de uso pueden tener relaciones con otros casos de uso Sus relaciones son:

- **Include:** Representado por una flecha, en el diagrama de ejemplo podemos ver como un caso de uso, el de totalizar el coste incluye a dos casos de uso.

- Extends: Una relación de un caso de uso A hacia un caso de uso B indica que el caso de uso B implementa la funcionalidad del caso de uso A.
- Generalization: Es la típica relación de herencia.

Actores: se representan por un muñeco. Sus relaciones son:

- Communicates: Comunica un actor con un caso de uso, o con otro actor.

Parte del sistema (System boundary): Representado por un cuadro, identifica las diferentes partes del sistema y contiene los casos de uso que la forman.

Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso de sistema, los actores y la relación entre estos y los casos de uso (Ver Fig. 2.10). Hay líneas de comunicaciones entre los casos y los actores; las flechas indican el flujo de la información o el estímulo.

El diagrama tiene por objeto ofrecer una clase de diagrama contextual que *nos* permite *conocer* rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan.

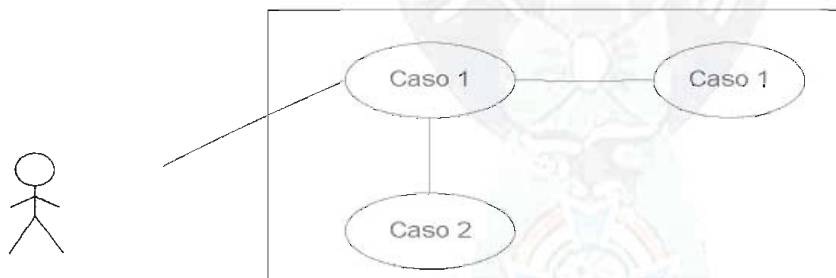


Figura 6. Diagrama de casos de uso [LARMAN, 1999]

2.9.5 Modelado del contexto

Se debe modelar la relación del sistema con los elementos externos, ya que son estos elementos los que forman el contexto del sistema. Los pasos a seguir son:

- Identificar los actores que interactúan con el sistema.
- Organizar a los actores.

- Especificar sus vías de comunicación con el sistema.

2.9.6 Modelado de requisitos

La función principal, o la más conocida del diagrama de casos de uso es documentar los requisitos del sistema, o de una parte de los requisitos establecen un contrato entre el sistema y su exterior, definen lo que se espera que realice el sistema, sin definir su funcionamiento interno. Es el paso siguiente al modelado del contexto, no indica relaciones entre actores, tan solo indica cuales deben ser las funcionalidades (requisitos) del sistema. Se incorporan los casos de uso necesarios que no son visibles desde los usuarios del sistema.

Para modelar los requisitos es recomendable:

- Establecer su contexto, para lo que también podemos usar un diagrama de casos de uso.
- Identificar las necesidades de los elementos del contexto (Actores).
- Nombrar esas necesidades, y darles forma de caso de uso.
- Identificar que casos de uso pueden ser especializaciones de otros, o buscar especializaciones comunes *para* los casos de uso ya encontrados.

En la Figura se observan nuevos casos de uso que no son visibles por ninguno de los actores del sistema, pero que son necesarios para su correcto funcionamiento.

2.9.7 Diagrama de clases

Forma parte de la vista estática del sistema. En el diagrama de clases se definen las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización. Es decir, es donde se da rienda suelta a los conocimientos de diseño orientado a objetos, definiendo las clases e implementando

las relaciones de herencia y agregación En el diagrama de clases se deben definir las clases y a sus relaciones.

La clase

Una clase está representada por un rectángulo que dispone de tres apartados, el primero para indicar el nombre, el segundo para los atributos y el tercero para los métodos Cada clase debe tener un nombre único, que las diferencie de las otras.

Un atributo representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase. Los atributos pueden representarse solo mostrando su nombre, mostrando su nombre y su tipo, e incluso su valor por defecto. Un método u operación es la implementación de un servicio de la clase, que muestra un comportamiento común a todos los objetos. En resumen es una función que le indica a las instancias de la clase que hagan algo.

Para separar las grandes listas de atributos y de métodos se pueden utilizar estereotipos. En la FIGURA 2. 4 se presenta un ejemplo. La clase usuario contiene tres atributos. Nombre, que es public; dirección, que es protected; y situación, que es private Situación empieza con el valor 3. También dispone de tres métodos Entrar. Salir y Trabajar.

Relaciones entre clases

Existen tres relaciones diferentes entre clases. Dependencias, Generalización y Asociación. En las relaciones se habla de una clase destino y de una clase origen. El origen es desde la que se realiza la acción de relacionar Es decir desde la que parte la flecha, la destino es la que recibe la flecha. Las relaciones se pueden modificar con estereotipos o con restricciones.

Dependencias

Es una relación de uso, es decir una clase usa a otra, que la necesita para su cometido. Se representa con una flecha discontinua va desde la clase utilizadora a la clase utilizada. Con la dependencia mostramos que un cambio en la clase utilizada puede afectar al funcionamiento de la clase utilizadora, pero no al contrario. Aunque las dependencias se pueden crear tal cual, es decir sin ningún estereotipo (palabra que aparece al lado de la línea que representa la dependencia) UML permite dar más significado a las dependencias, es decir concretar más. mediante el uso de estereotipos

Estereotipos de relación Clase-objeto.

- Bind: La clase utilizada es una plantilla, y necesita de parámetros para ser utilizada, con Bind se indica que la clase se instancia con los parámetros pasándole datos reales para sus parámetros.
- Derive: Se utiliza al indicar relaciones entre dos atributos, indica que el valor de un atributo depende directamente del valor de otro. Es decir el atributo edad depende directamente del atributo Fecha nacimiento.
- Friend: Especifica una visibilidad especial sobre la clase relacionada. Es decir podrá ver las interioridades de la clase destino.
- InstanceOF; Indica que el objeto origen es una instancia del destino.
- Instantiate: indica que el origen crea instancias del destino.
- Powertype: indica que el destino es un contenedor de objetos del origen, o de sus hijos.

- Refine: se utiliza para indicar que una clase es la misma que otra, pero mas refinada, es decir dos vistas de la misma clase, la destino con mayor detalle

Generalización

Pues es la herencia, donde se llene una o varias clases padre, también llamada superclase o madre, y una clase hija o subclase. UML soporta tanto herencia simple como herencia múltiple. Aunque la representación común es suficiente en el 99.73% de los casos UML nos permite modificar la relación de Generalización con un estereotipo y dos restricciones.

Estereotipo de generalización

- Implementation: El hijo hereda la implementación del padre, sin publicar ni soportar sus mterfaces

Restricciones de generalización

- Complete: La generalización ya *no* permite mas hijos.
- Incomplete: Podemos incorporar mas hijos a la generalización. Disjoint solo puede tener un tipo en tiempo de ejecución, una instancia del padre solo podrá ser de un tipo de hijo.
- Overlapping. puede cambiar de Upo durante su vida, una instancia del padre puede ir cambiando de tipo entre los de sus hijos.

Asociación

Especifica que los objetos de una clase están relacionados con los elementos de otra clase. Se representa mediante una línea continua, que une las dos clases. Se puede indicar el nombre, multiplicidad en los extremos, su rol, y agregación.

En la figura 2.5 se han creado cuatro clases. La clase principal es Usuario, que tiene dos clases hijas UsuarioADM y UsuarioINF. El usuario mantiene una relación de asociación con la clase Clave, se indica que es propietario de una clave, o de un número indeterminado de ellas. Se le crea también una relación de dependencia con la clase Perfil, es decir las instancias de usuario contendrán como miembro una instancia de Perfil.

2.9.8 Diagrama de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo. Un nodo se representa como un cubo, un nodo es un elemento donde se ejecutan los componentes, representan el despliegue físico de estos componentes. En la figura 2.6 se tienen dos nodos, el cliente y el servidor, cada uno de ellos contiene componentes. El componente del cliente utiliza un interface de uno de los componentes del servidor. Se muestra la relación existente entre los dos Nodos. Esta relación podría tener asociado un estereotipo para indicar qué tipo de conexión se utiliza entre el cliente y el servidor, así como también modificar su cardinalidad para indicar que el servidor soporta diversos clientes.

Como los componentes pueden residir en más de un nodo, se puede situar el componente de forma independiente, sin que pertenezca a ningún nodo, y relacionarlo con los nodos en los que se sitúa.

2.10. Fase de construcción

La fase de construcción requiere de varios ciclos de desarrollo (probablemente con plazos fijos) a lo largo de los cuales se extiende el sistema. Un ciclo individual de desarrollo esta conformado por dos etapas: Análisis y Diseño. (Ver Fig. 2.11), para mejorar el prototipo cuando llegue la hora de la entrega del producto. El objetivo es llegar a obtener un sistema de software que llene las expectativas del usuario. El análisis y el diseño consisten en definir el dominio de un problema y su solución lógica.



Figura 7. Ciclo de Desarrollo [LARMAN, 1999]

Durante la fase de Análisis lo que se desea ante todo es identificar y describir los objetos (conceptos) dentro del dominio del problema.

En la fase de Diseño, el objetivo es definir los objetos lógicos del software que finalmente serán implementados en un lenguaje de programación. A continuación se presenta una explicación etapa por etapa que corresponden a esta fase.

2.10.1 Etapa de análisis

Una vez concluida la fase de planeación y elaboración, donde se han identificado ya los requerimientos y los casos de uso, comienza la etapa de

análisis de la fase de construcción donde se siguen actividades, en la cual se cumplen los ciclos del desarrollo iterativo e incremental.

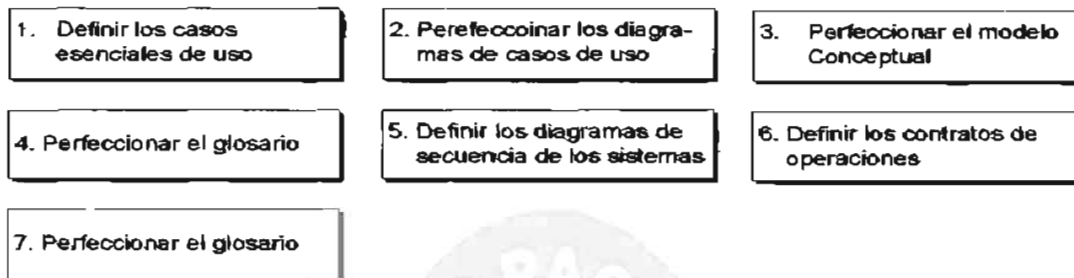


Figura 2.12. Actividades de la etapa de Análisis dentro un ciclo de desarrollo [LARMAN, 1999]

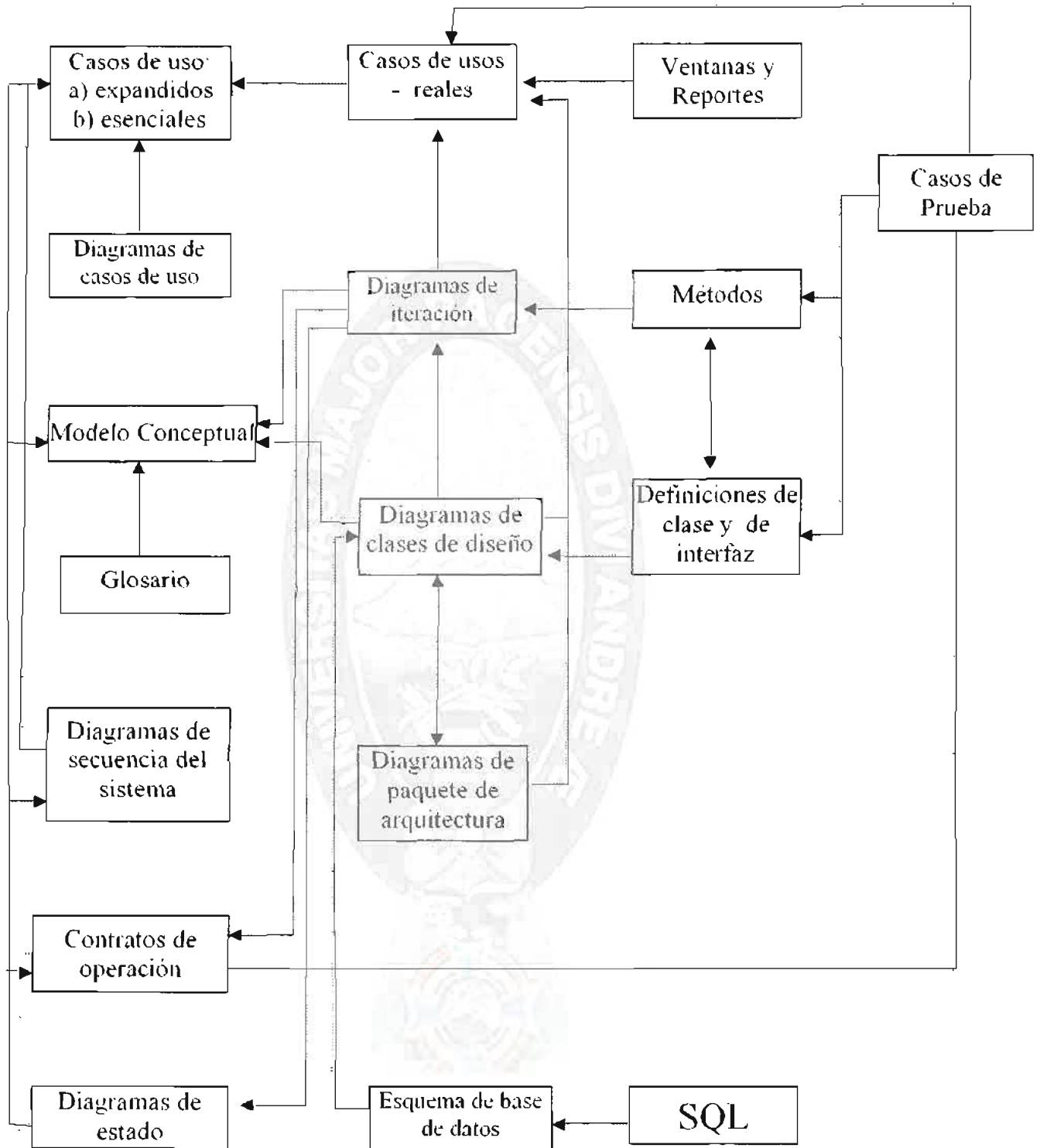


Figura 8. Dependencias de los artefactos durante la Fase de Construcción
[LARMAN, 1999]

2.10.2. Modelo conceptual

Un modelo conceptual explica (a sus creadores) los conceptos significativos en un dominio del problema; es el artefacto más importante a crear durante el análisis orientado a objetos, representa esencialmente cosas del mundo real - los casos de uso son un importante artefacto del análisis de requerimientos.

En lo referente a este modelo, no se debe tomar en cuenta lo que es el software, como las interfaces de usuario (ventana), ni la base de datos, los procesos.

El paso esencial de un análisis o investigación orientados a objetos es descomponer el problema en conceptos y objetos individuales. Un modelo conceptual es una representación de conceptos en un dominio del problema. En el UML, los ilustramos con un grupo de diagramas de estructura estática donde no se define ninguna operación.

Cuando nos referimos al modelo conceptual, este nos puede mostrar:

- o Conceptos
- o Asociaciones entre conceptos
- o Atributos de conceptos

Craig Larman, para desarrollar el modelo conceptual recomienda los siguientes pasos:

- 1) Liste los conceptos idóneos usando la Lista de categorías de conceptos y la identificación de la frase nominal relacionada con los requerimientos en cuestión.
- 2) Dibújelos en un modelo conceptual.
- 3) Incorpore las asociaciones necesarias para registrar las relaciones para las cuales debe reservar un espacio en la memoria.
- 4) Agregue los atributos necesarios para cumplir con las necesidades de

información.

Es necesario identificar las asociaciones de los conceptos, lo que contribuirá a entender el modelo conceptual. Una asociación es una relación entre dos conceptos, lo que indica alguna conexión entre ellos.

Al generarse el modelo conceptual, se realiza la asociación luego es necesario identificar los atributos de los conceptos que se necesitan para satisfacer los requerimientos de información de los casos de uso.

Un atributo es un valor lógico de un dato de un objeto.

Los extremos de una asociación pueden contener una expresión de multiplicidad que indique la relación numérica entre las instancias de los conceptos.

La multiplicidad define cuantas instancias de un tipo A pueden asociarse a una instancia del tipo B en determinado momento

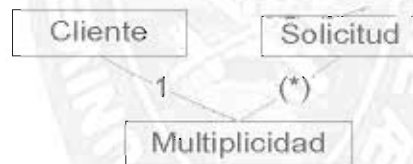


Figura 10 Multiplicidad en una asociación [LARMAN 1999]

Por ejemplo, una instancia individual de una *Tienda* puede asociarse a "muchas" instancias (cero o mas marcadas con *) de *Producto*.

En la figura 11 se ofrecen algunas expresiones de multiplicidad.

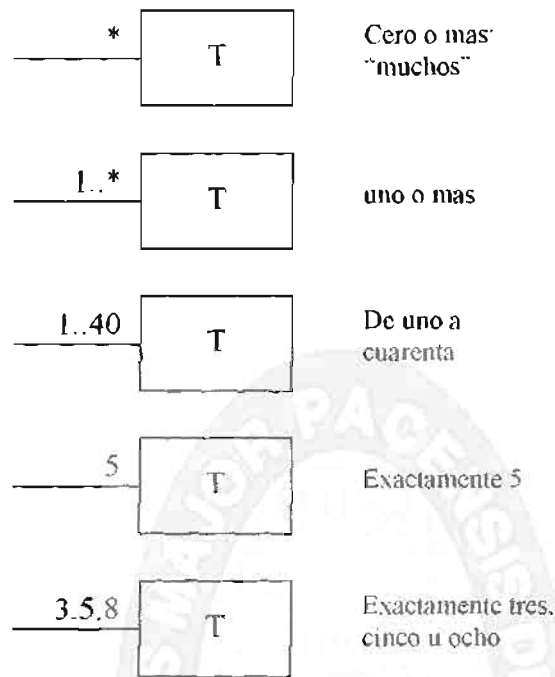


Figura 11 Valores de multiplicidad [LARMAN 1999]

2.10.3. Etapa de diseño

En la fase de análisis del desarrollo se da prioridad al conocimiento de los requerimientos, los conceptos y las operaciones relacionadas con el sistema. A menudo la investigación y el análisis se caracterizan por estar centradas en cuestiones concernientes al qué, es decir: cuáles son los procesos, que son los conceptos, etc.

Durante el ciclo de desarrollo iterativo es posible pasar a la fase de diseño, una vez terminados estos documentos del análisis. En la etapa de diseño se logra una solución lógica, su esencia es la elaboración de diagramas de interacción, que muestran gráficamente cómo los objetos se comunicaran entre ellos a fin de cumplir con los requerimientos.

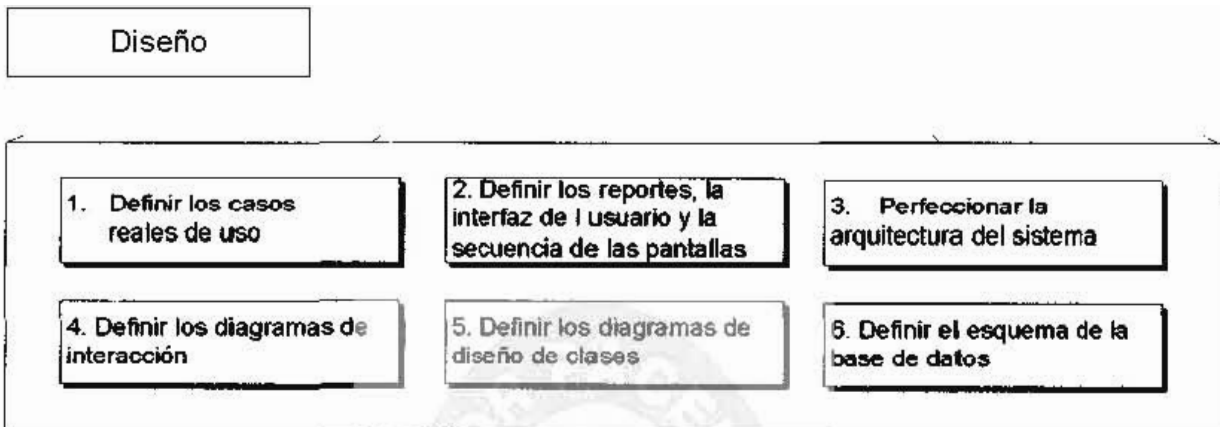


Figura 12. Actividades de la Etapa de Diseño dentro un ciclo de desarrollo
[LARMAN, 1999]

2.10.4. Fase de aplicación

El ciclo iterativo de la segunda fase ocurre hasta que el sistema responda eficientemente a los requerimientos del usuario.

Una vez terminado las anteriores fases lo único que queda es la implementación del sistema. La fase de aplicación, es la tercera etapa y final de esta metodología. El proceso o actividades que le siguen son (Ver Fig. 2.28):

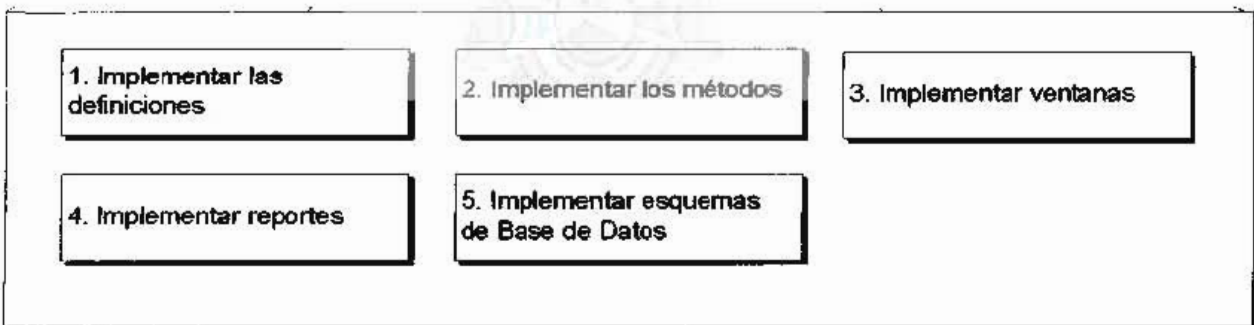


Figura 13. Actividades de la Fase de Aplicación [LARMAN, 1999]

2.10.5.1. Pruebas

Se diseñan e implementan las pruebas creando los casos de prueba. Un caso de prueba específica *una forma* de probar el sistema, incluyendo *ja entrada o resultado* con la *que se* ha de probar y las condiciones bajo las que ha de probarse. [Jacobson, 2000] Se pueden especificar otros casos de prueba para probar el sistema como un todo:

- Las *pruebas de instalación* verifican que el sistema puede ser instalado en la plataforma del cliente y que el sistema funcionará correctamente cuando sea instalado.
- Las *pruebas de configuración* verifican que el sistema funciona correctamente en diferentes configuraciones.
- Las *pruebas negativas* intentan *provocar* que el sistema falle para poder así revelar sus debilidades.
- Las *pruebas de tensión o de estrés* identifican problemas con el sistema cuando hay *recursos* insuficientes o cuando hay competencia por los recursos.
- Los casos de prueba de *integración del sistema* se utilizan para verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados en una construcción.

10.6 Herramientas de implementación

(IIS WXP .NET SQL2000)

2.11. Métricas

Podríamos decir, basados libros y aplicaciones Web. que un sistema Web tiene tres componentes fundamentales el confiabilidad, mantenibilidad, portabilidad y la funcionalidad (o código ejecutable). Y es importante hacer esta diferenciación, pues el perfil de la persona encargada de implementar cada componente, es distinto.

A continuación se exponen las características de estos tipos de funcionalidad y se presentan algunas métricas para medirlos.

2.11.1 Confiabilidad

La Confiabilidad implica un comportamiento aceptable frente a situaciones inesperadas, en un caso ideal diremos que un software es confiable si cumple con los requerimientos especificados y no ocasiona graves problemas frente a situaciones imprevistas. La idea detrás de la confiabilidad es que quizás aunque en ciertas ocasiones el programa reaccione en forma inesperada, el usuario se encuentra cómodo usándolo ya que estos fallos no ocasionan problemas graves, además es relativa es decir un software puede aun ser confiable si la consecuencia de un error no es seria o si la cantidad de errores por unidad de tiempo no es alta.

En otras palabras la confiabilidad es la probabilidad de ocurrencia de fallas en el sistema.

2.11.2 Funcionalidad

La medición del tamaño funcional es un aparte importante del proceso de desarrollo de software ya que es uno de los factores clave que potencialmente afecta el coste y el tiempo de los proyectos software. El tamaño de un sistema de software es derivado de la cuantificación de los requisitos funcionales del usuario

Un método para medir el tamaño funcional de artefactos Web a partir de esquemas conceptuales orientados a objeto se basa en el método estándar Análisis de Puntos de Función para el paradigma de orientación a objetos.

2.11.3 Mantenibilidad

El mantenimiento del software consiste en modificar el producto o un componente del mismo una vez que ya se ha entregado, ya sea para corregir errores, mejorar el funcionamiento o alguna otra característica o para adaptarlo a cambios en el entorno. Se estima que en sistemas de información una gran parte del esfuerzo del desarrollo se invierte en mantenimiento, cosa que en el caso de los sistemas hipermedia puede variar, especialmente cuando se trata de productos cerrados

En algunas ocasiones la aplicación hipermedia se concibe como un CD que una vez generado no va evolucionar Pero salvo en esos casos específicos. las aplicaciones hipermedia deben mantenerse como cualquier otro tipo de productos de software, un mantenimiento especialmente acusado en el caso de los sitios Web

Existen cuatro tipos de mantenimiento, los cuales son

- *Mantenimiento Correctivo* consiste en realizar cambios al sistema de forma que corrija deficiencias y así cumpla con sus requerimientos. *Mantenimiento Preventivo* este típico mantenimiento es realizado con el objeto de evitar posibles fallas en el sistema sobre todo para llevar a cabo una posible reutilización.
- *Mantenimiento Perfectivo:* consiste en realizar cambios al sistema que permitan que los requerimientos sean mas efectivos o simplemente que amplíe las funciones del mismo.

- *Mantenimiento Adaptativo* este tipo de mantenimiento es frecuentemente mas utilizado, debido a que su objetivo es cambiar al sistema para que cumpla con nuevos requerimientos.

2.11.4 Portabilidad

La portabilidad es la capacidad de llevar exitosamente el sistema a otro entorno, también es un atributo que puede ser poseído por una unidad del software a un grado específico de ambientes. También la portabilidad es cuando un sistema es ejecutado e implementado en un nuevo ambiente, basado sobre una versión existente tanto del software como del hardware, pero un concepto adecuado describe (Mccall, Richards y Walter. 1997) que dice: "Es el esfuerzo necesario para transferir el programa de un entorno de sistema de hardware y/o software a otro".

El proceso de portabilidad tiene dos componentes que son;

- *Transportación.* Se refiere a la fase del programa, esto quiere decir que es movido el programa físicamente a un sistema compatible
- *Adaptación.* Es cualquier modificación que debe realizarse a la versión original, la misma puede ser realizada directamente por el sistema a través de algún proceso definido con anticipación o también a través del lenguaje en el se están desarrollando.



ESQUEMA



En esta Capitulo se realiza el análisis y diseño del sistema, aplicando la metodología, método y las herramientas para lograr un sistema Web.

3. CAPITULO III

3.1 ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se hace huso de las metodologías y herramientas mencionadas en el capítulo II, pero primeramente describamos el entorno de trabajo

3.1.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

Dentro la CANCELLERÍA se cuenta con la siguiente estructura.

1. MINISTERIO DE RELACIONES EXTERIORES Y DE CULTO
2. UNIDAD DE REGIMEN CONSULAR
3. UNIDAD DE SISTEMAS

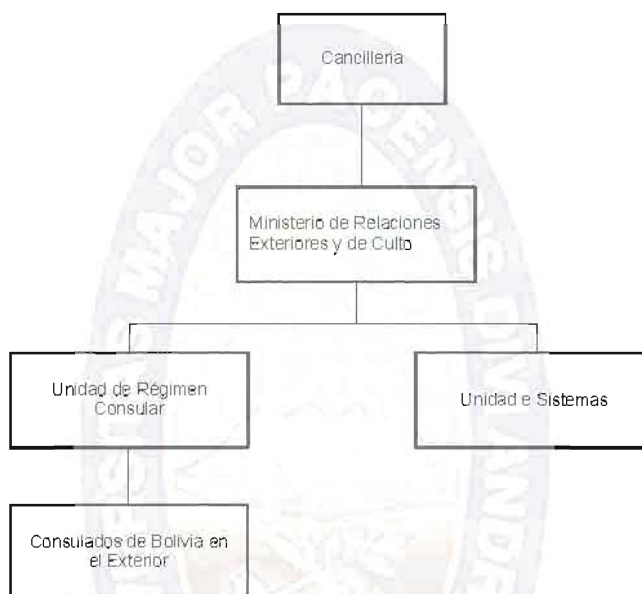
LA MISIÓN EN EL EXTERIOR DE BOLIVIA

1. CONSULADOS DE BOLIVIA
2. MINISTERIOS

La UNIDAD DE REGIMEN CONSULAR esta encargada de la administración de las solicitudes para la emisión de pasaportes hacia los ciudadanos bolivianos que residen en exterior del país, dichas solicitudes son enviadas por parte de los CONSULADOS DE BOLIVIA, que de acuerdo a la respuesta proceden a la emisión del pasaporte. (ver anexo para mejor detalle)

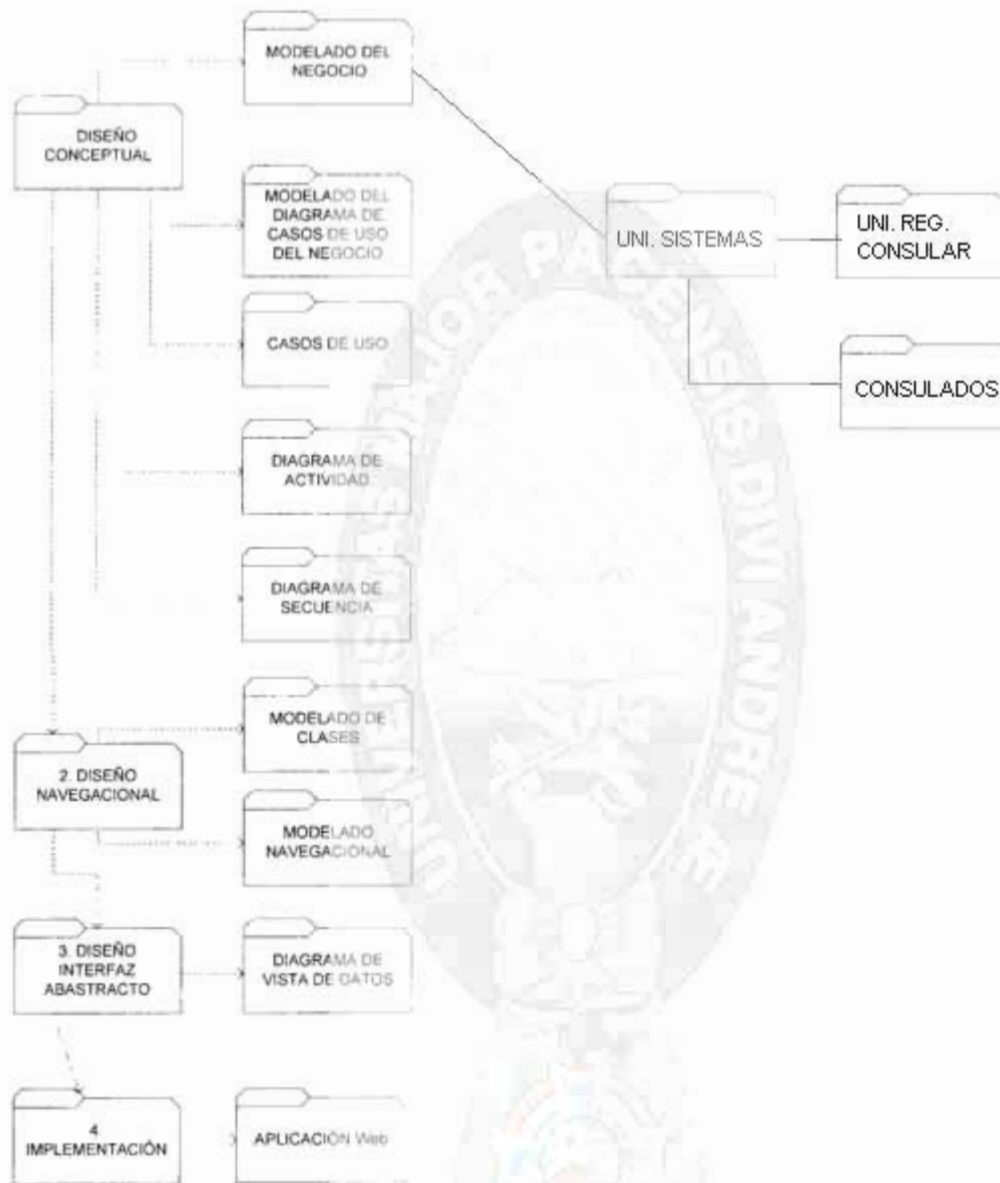
El punto 2 UNIDAD DE REGIMEN CONSULAR y el punto 1 CONSULADOS DE BOLIVIA son objeto de estudio del presente proyecto dentro del punto 3 UNIDAD DE SISTEMAS.

Estructura orgánica de la Cancillería (Delimitada)



Unidad	OBJETIVO
Cancillería	Dentro al Cancillería se encuentra el Canciller de la República de Bolivia Como Máxima Autoridad
Ministerio de Relaciones Exteriores y de Culto	Es la parte regidora de todo el ministerio en cuanto a la Administración de las demás unidades
Unidad de Régimen consular	Para el caso de Nuestro sistema es La Parte de Administrar los asuntos de Los consulados de Bolivia denominados Misión en el Exterior
Consulados de Bolivia en el Exterior	Son los responsables directos de los tramites que realizan los ciudadanos bolivianos

Desarrollo del Proyecto



Las fases de desarrollo de la metodología OOHDM es la siguiente:

Fase 1, Diseño Conceptual:

Esta etapa de desarrollo consiste en establecer un esquema conceptual en términos de clases, relaciones y subsistemas, para ello utilizaremos específicamente el lenguaje unificado de modelos (UML)

Fase 2, Diseño Navegacional:

uno de los aspectos más importantes en los diseños de información en la Web es el de la navegación. La navegación lleva asociada la idea de moverse por un espacio de información donde es crucial el concepto de posición actual o contexto, ya que muchas veces el siguiente paso a dar mientras se navega dependerá de que pasos se hay dado antes.

Fase 3, Diseño de la Interfaz

Abstracta: Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de internas. Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer como los objetos de internas activaran la navegación y el resto de la funcionalidad de la aplicación, que transformación de la internas son pertinentes y cuando es necesario realizarlas.

Fase 4, Implementación:

Terminando el diseño de internas abstracta y habiendo añadido el diseño navegacional tal y como se ha descrito en los apartados anteriores, el camino a la fase de implementación resulta mas sencillo.

3.2.1 FASE 1: DISEÑO CONCEPTUAL.

Para el desarrollo del proyecto se han utilizado vistas, que son las siguientes:

Modelo conceptual de casos de uso.

Aplicación de diagramas de actividad.

Aplicación de diagramas de secuencia.

Diagrama conceptual.

3.2.1.1 Modelado del negocio

El Modelado del negocio tiene como finalidad, comprender el entorno en el que se va a funcionar el sistema concretamente, el estudio se centra en las actividades desarrolladas en la organizacional, la información que se maneja y los actores participantes (personas o sistemas externos) en dichas actividades.



Figura 3.3 MODELADO DEL NEGOCIO

La información recopilada para modelar el negocio, se obtuvo en base a entrevistas directas con los implicados del sistema, además del propio conocimiento del funcionamiento, en este modelado (Fig. 3.3) se describe como intervienen en el proceso.

3.2.1.2. Modelado del contexto

Se modela la relación del sistema con los elementos externos, ya que son estos elementos los que forman el contexto del sistema, observamos en la figura 3.4.1 y 3.4.2 como el sistema interactúa.

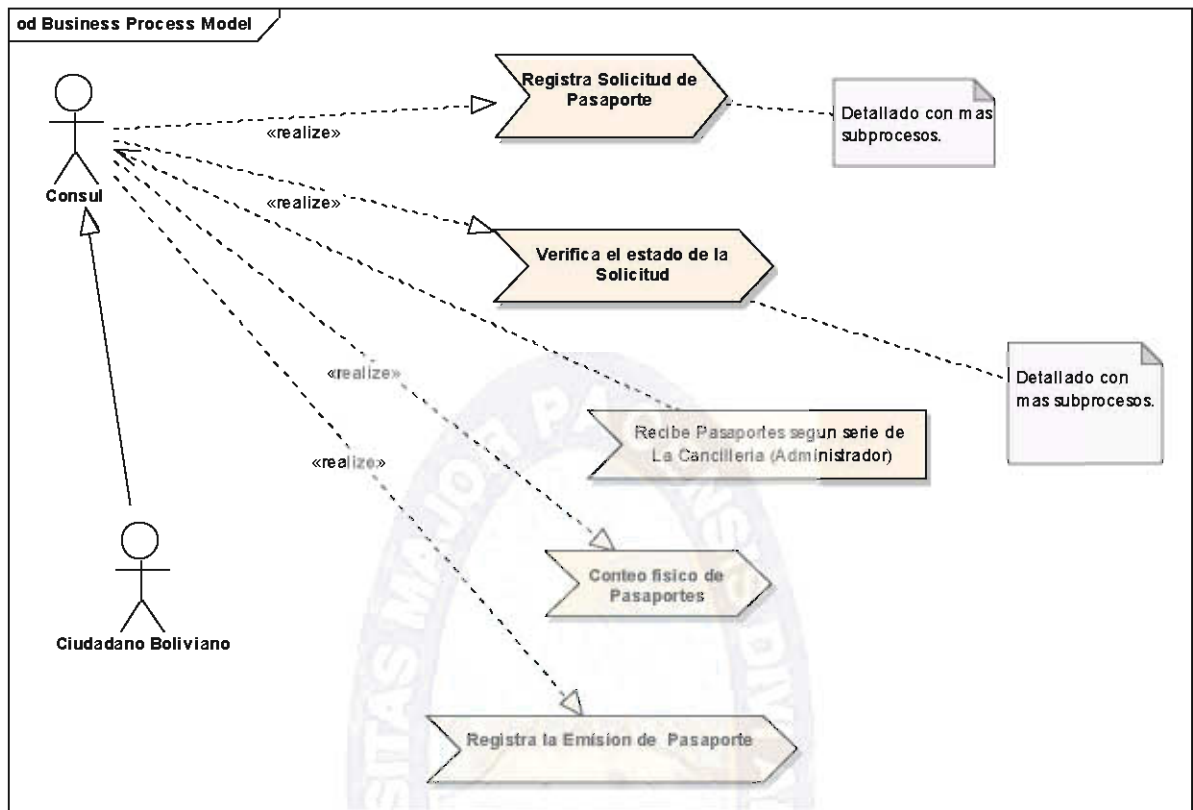


Figura 3.4.1 Modelado del negocio (caso Cónsul)

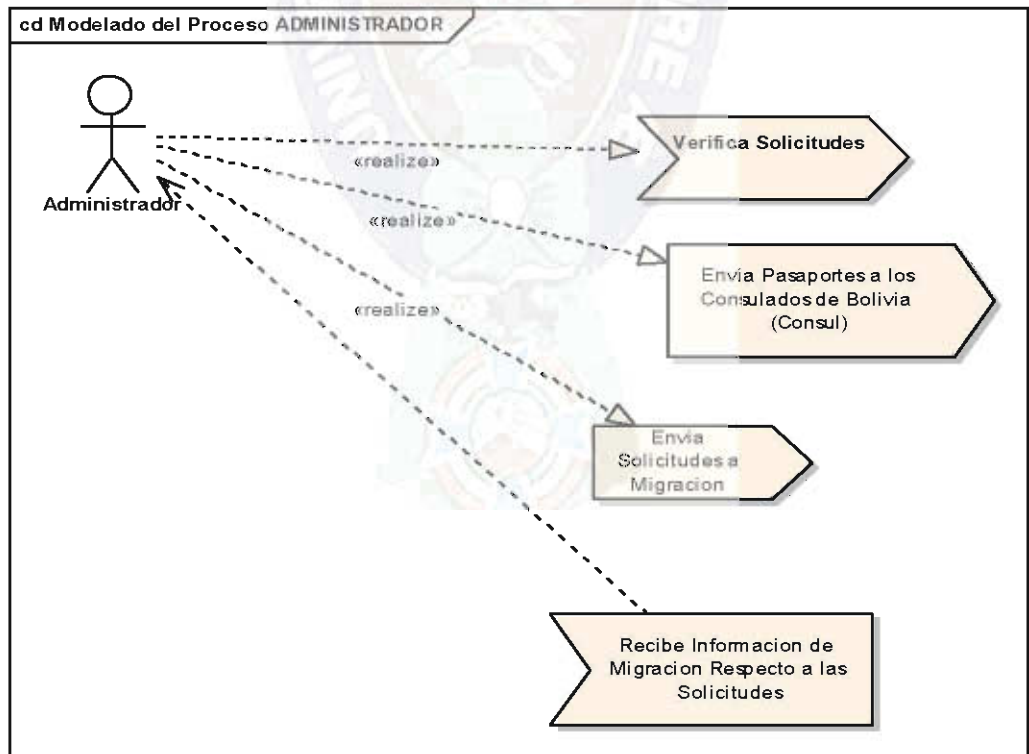


Figura 3.4.1 Modelado del negocio (caso Administrador)

3.2.2 Modelo casos de uso

Los principales actores que interactúan con el sistema son: Ciudadano Boliviano ,
Cónsul, Administrador

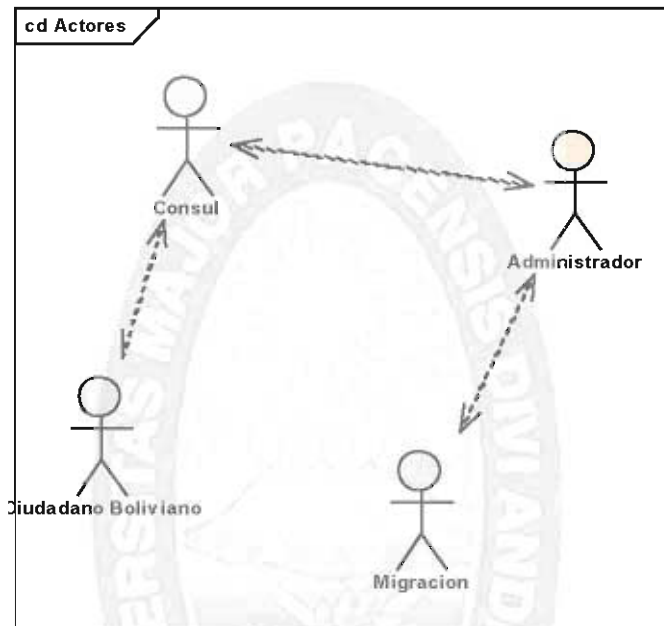


Figura 3.5 DIAGRAMA DE CLASES DE ACTORES

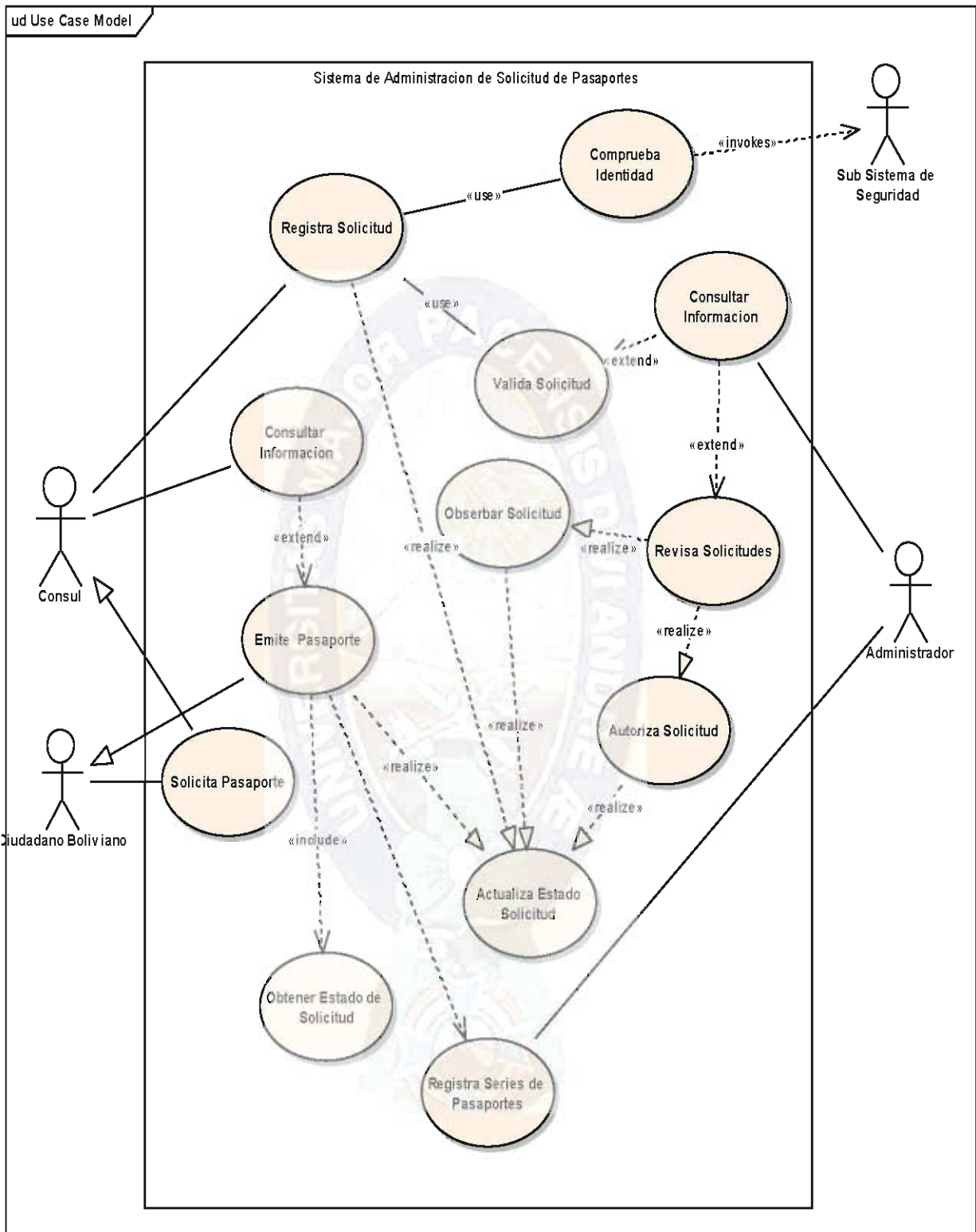


Figura 3.6 Diagramas de Casos de Uso

En la figura 3.6 se observan las actividades que realizan y los estados de la solicitud para la emisión del pasaporte.

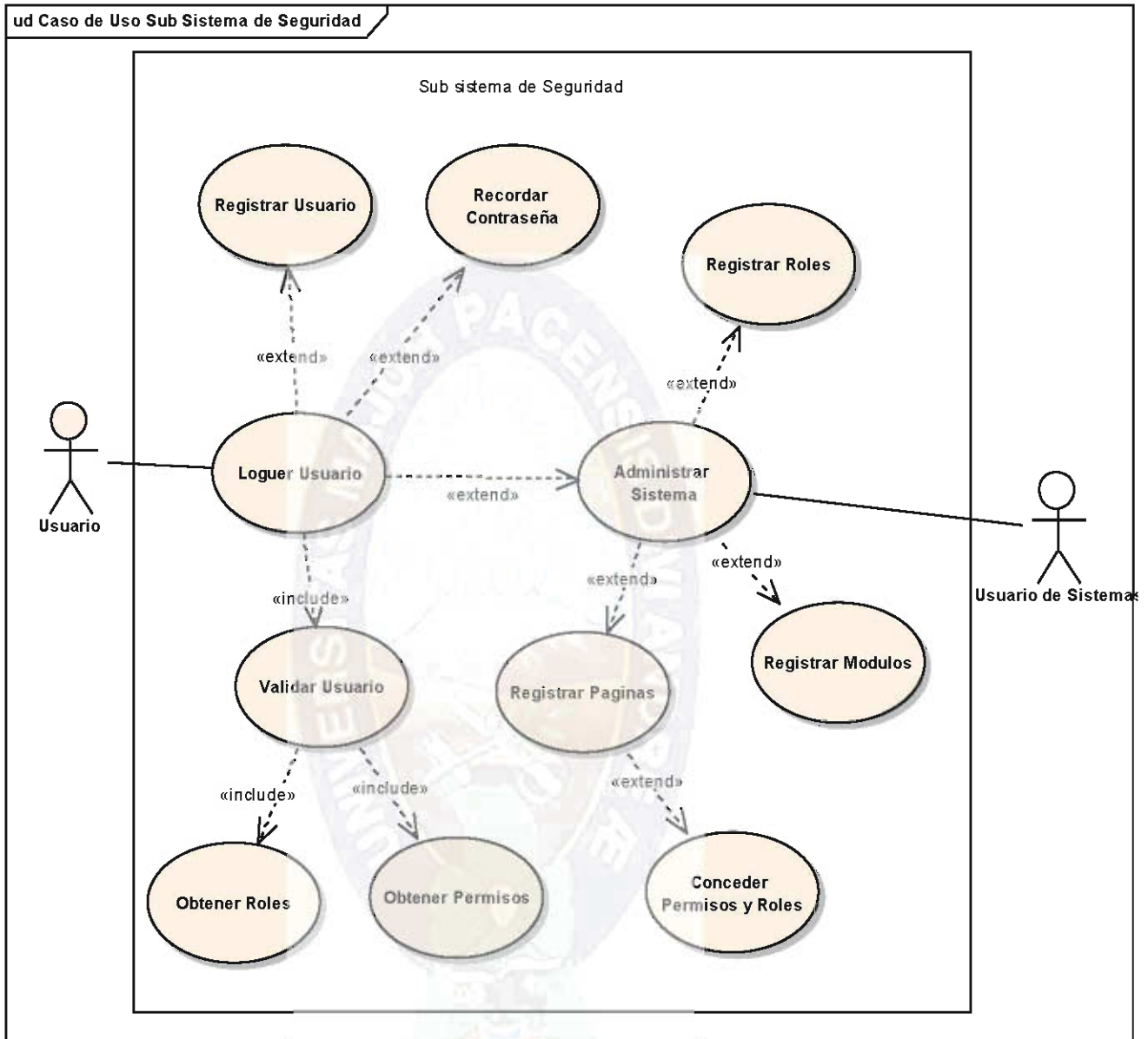


Figura 3.6.1

En la figura 3.6.1 se observa la parte de seguridad juntamente a la participación del mismo entorno al sistema.

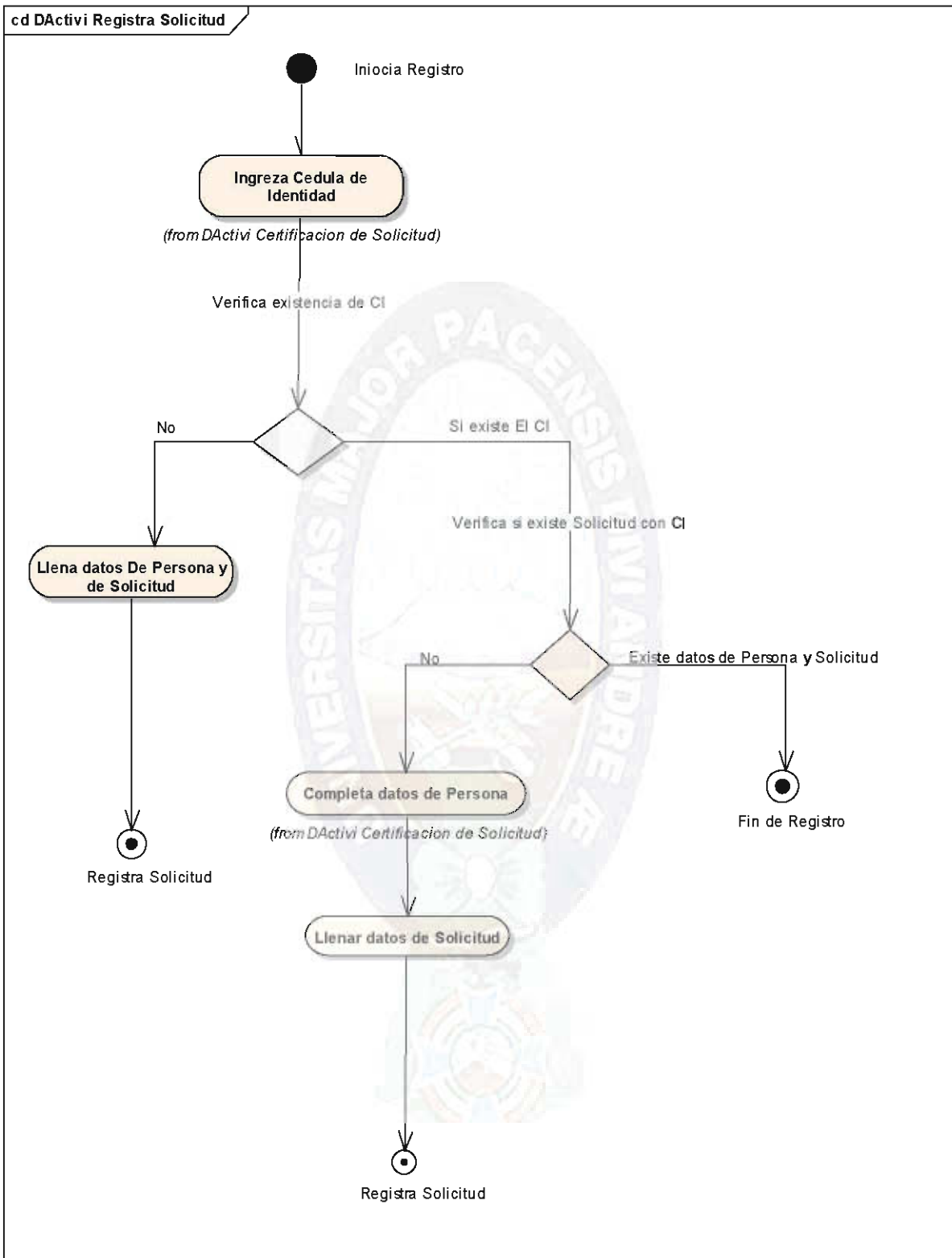


Figura 3.7 Diagrama de Actividad Registro de Solicitud

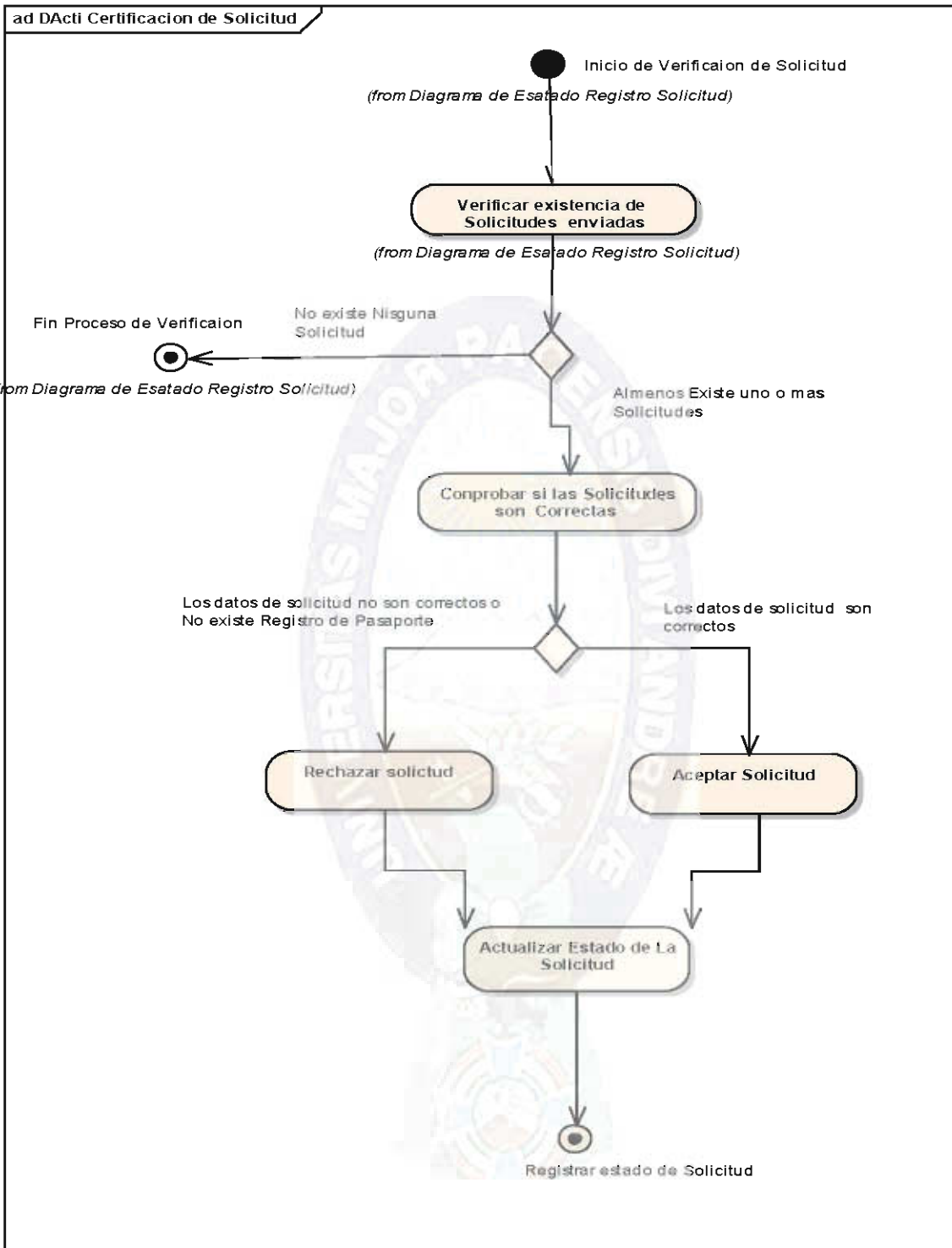


Figura 3.7.1 Diagrama de Actividad Verificación de solicitudes

En la figura 3.7.1 describe cada paso para poder verificar las solicitudes enviadas por los cónsules y la información proporcionada por migración

Actores

Actor:	<i>Usuario</i>
Casos de Uso:	<i>Validar Usuario. Registrar Usuario. Registrar Solicitud.</i>
Tipo:	<i>Primario</i>
Descripción:	Es el actor principal y representa a cualquier persona que desee utilizar del sistema de reservaciones.

Casos de Uso

Validar Usuario	
Caso de Uso	<i>Validar Usuario</i>
Actores	<i>Usuario Sistemas, Cónsul, Administrador</i>
Tipo	<i>Inclusión</i>
Propósito	Validar a un usuario ya registrado para el uso del sistema.
Resumen	Este caso de uso es iniciado por el <i>Usuario</i> . Valida al usuario mediante un <i>Usuario</i> y <i>Contraseña</i> ser validado con su respectivo registro de usuario para así poder utilizar los roles y permisos en el sistema.
Precondiciones	Si el <i>Usuario</i> aún no se ha registrado, requerirá ejecutar el caso de uso <i>Registrar Usuario</i> sub flujo <i>Crear Registro Usuario</i> .
Flujo Principal	Se presenta al usuario la Pantalla Principal (P-1). El Usuario puede seleccionar entre las siguientes opciones: "Registrarse por ". "Aceptar" y "Salir".

2. Elige 'Registrar Nueva Solicitud', introduce la Cedula de Identidad del solicitante.	3. Busca el CI de la Persona y si existe lista las personas que poseen el mismo CI.
4. Decide si registrar con los mismos datos o realizar una nueva solicitud con otros datos.	5. Si realiza con los mismos datos autocompleta los datos de la persona.
6. Concluye con el llenado del formulario de solicitud.	7. Guarda los datos ingresado por el cónsul.

Caso de Uso	Valida Solicitud
Actores	Administrador
Propósito	Validar o Observar Solicitudes
Visión general	El Administrador revisa las solicitudes enviadas por los consulados y procede a la revisión de cada solicitud según datos proporcionados por Migración. Si los datos son validos procede a la autorización, en caso contrario observa la solicitud.
Tipo	Primario y Esencial

Acción del Actor	Respuesta del sistema
1. El Administrador lista las solicitudes registradas por los cónsules según Consulado.	2. El sistema de acuerdo al estado de la solicitud habilita y deshabilita la opción para revisión de la solicitud.
3. Una vez comparada la solicitud procede a modificar el estado de la solicitud, autorizado o observando la misma.	4. Guarda los estados de las solicitudes según contacto, usuario fecha.

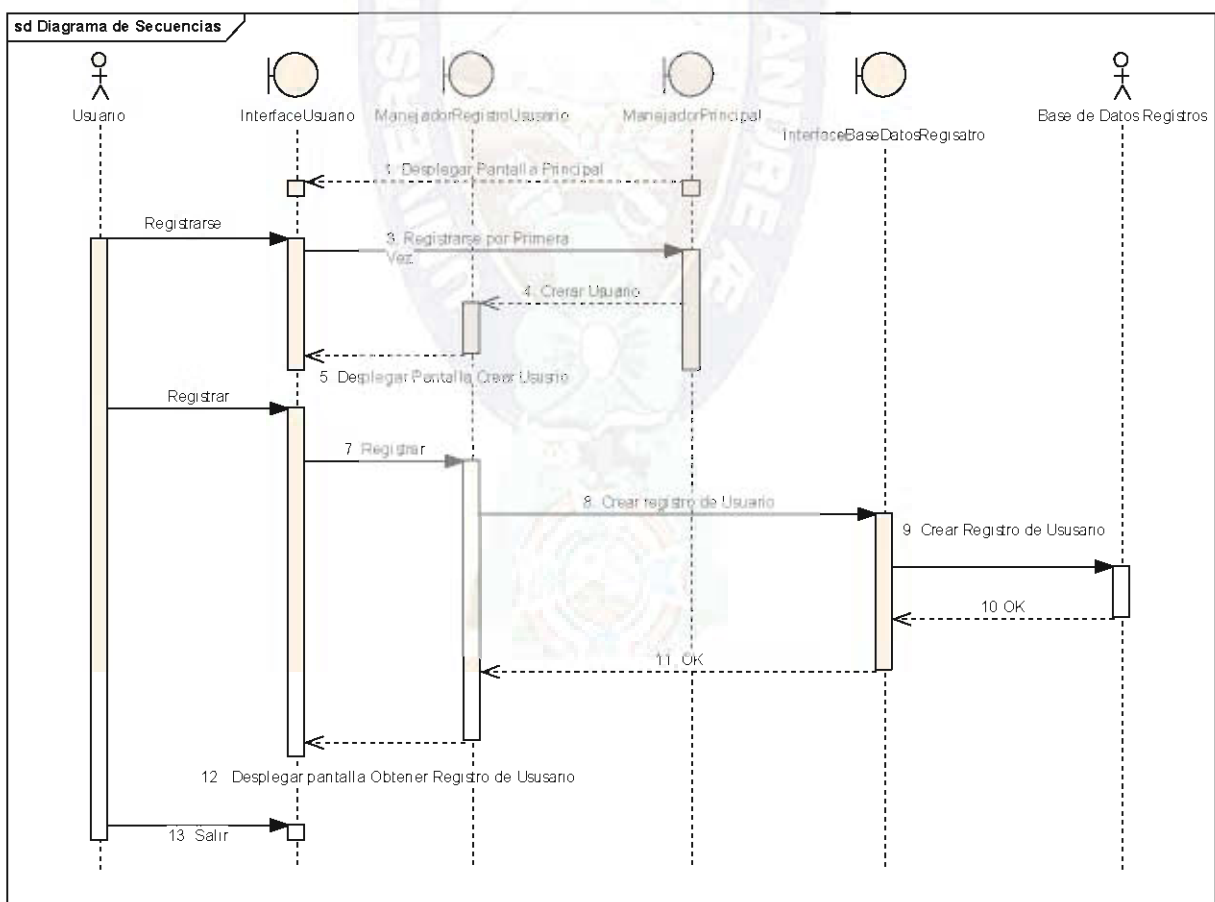
Caso de Uso	<i>Registrar Usuario</i>
Actores	<i>Usuario.</i>
Tipo	<i>Básico</i>
Propósito	Permitir a un usuario registrarse con el sistema para su uso posterior.
Resumen	Este caso de uso es iniciado por el <i>Usuario</i> Ofrece funcionalidad para crear, modificar, y eliminar el registro de usuario con el sistema
Precondiciones	Todos los sub flujos, con excepción de <i>Crear Registro Usuario (S-1)</i> , requieren ejecutar inicialmente el caso de uso <i>Validar usuario</i> .
Flujo Principal	Se ejecuta el caso de uso <i>Validar Usuario</i> . Dependiendo de las opciones seleccionadas por el usuario, se continuará con los diversos sub flujos de este caso de uso
Sub flujo:	<p><i>S-1 Crear Registro Usuario</i></p> <p>Se presenta al usuario la Pantalla Crear Registro Usuario esta pantalla contiene información de registro que debe ser llenada por el usuario, lo cual incluye nombre, apellido, país, teléfonos de la casa y oficina, <i>usuario</i> y <i>contraseña</i> y una entrada adicional de repetir contraseña para asegurarse de su corrección El <i>Loguear</i> serán utilizados por el sistema para validar al usuario.</p> <p>El usuario puede seleccionar entre las siguientes actividades: "Registrar" y "Salir".</p> <p>Si el usuario selecciona "Registrar", el sistema genera un nuevo registro de usuario</p> <p>Se continúa con el sub flujo <i>Administrar Registro Usuario (S-3)</i>.</p> <p>Si la actividad seleccionada es Cancelar se saldrá del formulario.</p>
	<p><i>S-2 Obtener Registro Usuario</i></p> <p>El sistema obtiene el registro de usuario de la base de datos de registro.</p> <p>Se continúa con el sub flujo <i>Administrar Registro Usuario (S-3)</i>.</p>

	<p>S-3 <i>Administrar Registro Usuario</i></p> <p>Se presenta al usuario la Pantalla Obtener Registro Usuario con la información de registro de usuario</p> <p>El usuario podrá seleccionar entre las siguientes actividades: "Actualizar", "Eliminar", "Registrar Servicios" y "Cerrar sesión".</p> <p>Si el usuario presiona "Actualizar" se ejecuta el sub flujo <i>Actualizar Registro Usuario (S-4)</i>.</p> <p>Si el usuario selecciona "Eliminar" se ejecuta el sub flujo <i>Eliminar Registro Usuario (S-5)</i>.</p> <p>"Actualizar", la nueva información será perdida).</p>
	<p>S-4 <i>Actualizar Registro Usuario</i></p> <p>Se actualiza el registro de usuario con la información modificada (E-1. E-3. E-4).</p> <p>Se continúa con el sub flujo <i>Administrar Registro Usuario (S-3)</i>.</p>
	<p>S-5 <i>Eliminar Registro Usuario</i></p> <p>Se elimina el registro de usuario y se continúa con el sub flujo <i>Crear Registro Usuario (5-1)</i>.</p>
Excepciones	<p>E-1 Información incompleta: Falta llenar información en el registro de usuario. Se vuelve a solicitar al usuario que complete el registro.</p> <p>E-2 registro ya existe: Si ya existe un registro bajo ese login. se solicitará al usuario que lo cambie o que termine el caso de uso.</p> <p>E-3 login incorrecto: El login no es válido. Se le solicita al usuario que corrija el registro.</p> <p>E-4 contraseña incorrecta: La contraseña escogida es muy sencilla o no se validó correctamente, se solicita al usuario que corrija el registro.</p>

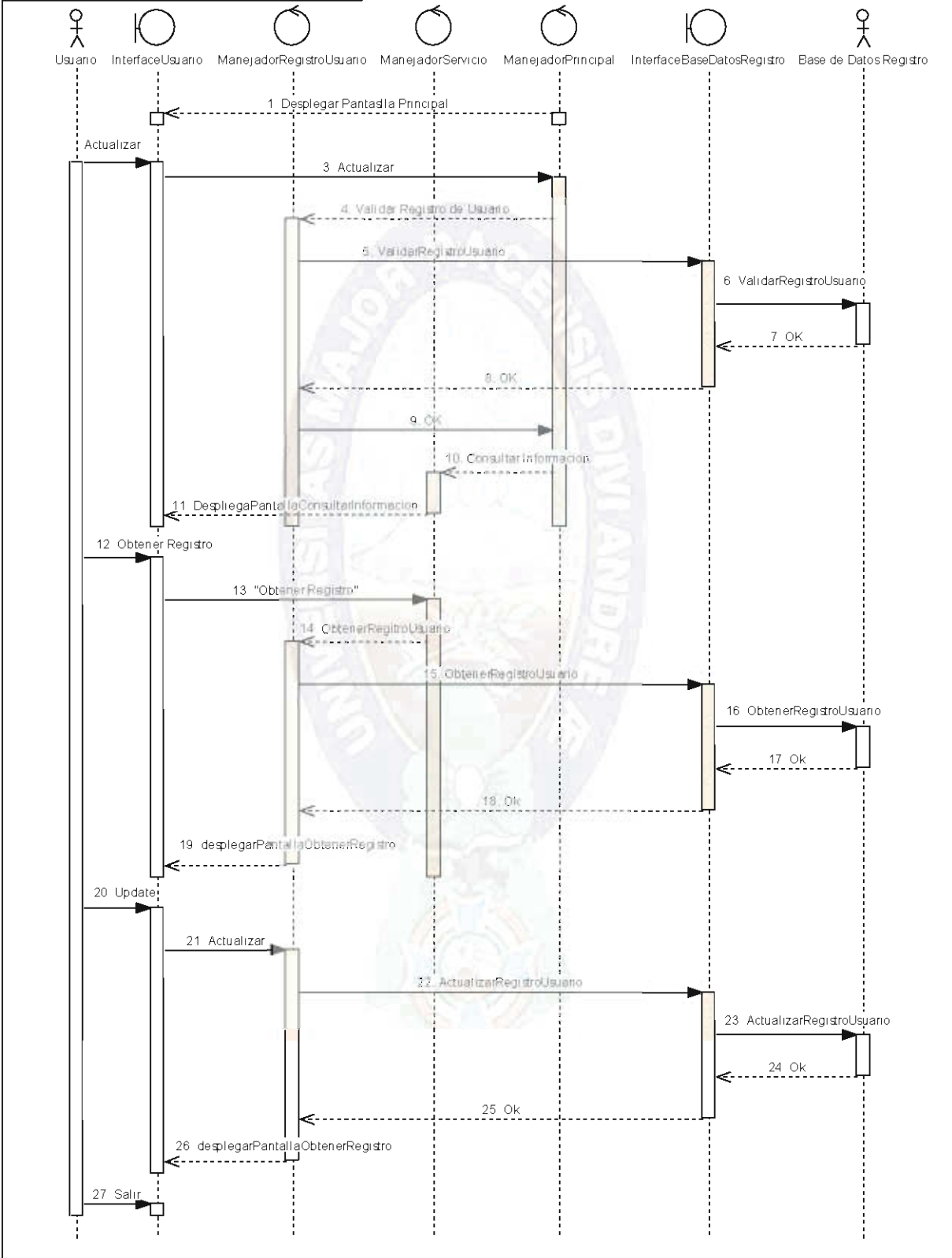
Caso de Uso	<i>Consultar Información</i>
Actores	<i>Cónsul, Administrador</i>
Tipo	<i>Básico</i>
Propósito	Permitir a un usuario consultar información con el sistema.
Resumen	Este caso de uso es iniciado por el <i>Usuario</i> . Ofrece funcionalidad para consultar información de las solicitudes según estados: Registrado, En Revisión, Observado, Autorizado y anulados, a la vez podrá especificar la fecha (año, mes) o especificar una persona.
Precondiciones	Se requieren haber ejecutado anteriormente el caso de uso <i>Validar Usuario</i> .
Flujo Principal	Se ejecuta el caso de uso <i>Validar Usuario</i> . Dependiendo de las opciones seleccionadas por el Usuario, se continuará con los diversos sub flujos de este caso de uso.
Sub flujos	<p>S-1 Ver Solicitudes</p> <p>Se despliega la Pantalla Consultas. El usuario puede seleccionar entre las siguientes Estados: "Registrado". "Revisado". "Observado". "Autorizado" y especificar la fecha o una persona.</p> <p>Si el usuario selecciona "Registrado", se desplegara todos los usuarios se activa sub flujo (S-2) Despliega opciones</p> <p>Si el usuario presiona "'Revisados'", listara todas las solicitudes registrabas y se activa el sub flujo (S-2). Despliega opciones</p> <p>Si el usuario presiona "'Observados'" listara todas las solicitudes Observadas.</p> <p>Si el usuario presiona "Autorizados" listara todas las solicitudes Autorizadas por el administrado y se activara el sub flujo (S-3) Emitir Pasaporte.</p> <p>Si el usuario presiona "'Anulados'" listara todas las solicitudes Anuladas.</p>

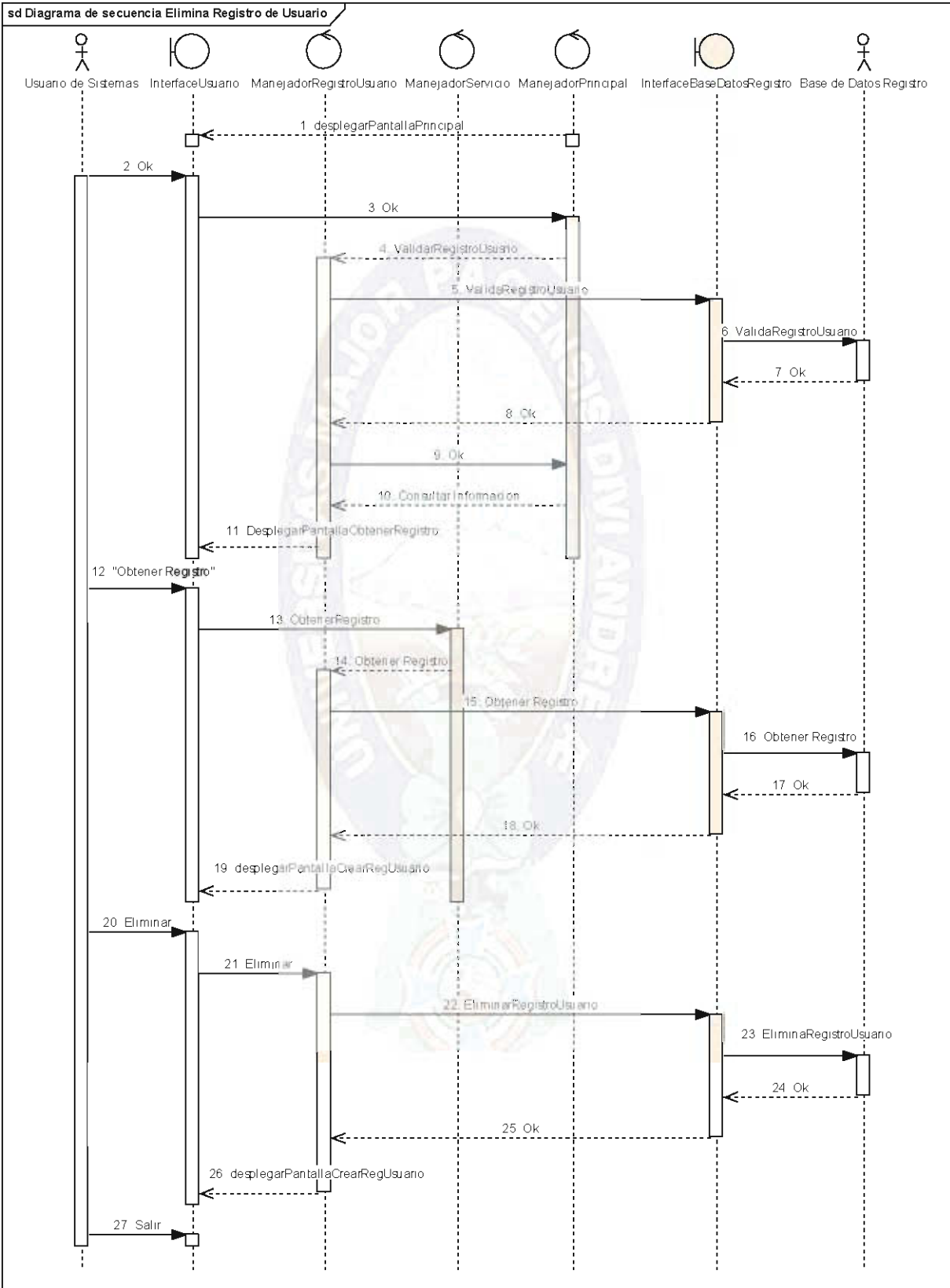
	<p>S-2 Despliega opciones</p> <p>Si el usuario es Un Cónsul y la solicitud esta en estado de registrado se le habilitara la opción de Anular y Modificar la solicitud.</p> <p>Si el usuario es Administrador y el estado de la solicitud no esta en anulado y emitido se le habilitara las opciones de Autorizar, observar, poner en revisión a la ves la casilla de observación.</p>
	<p>(S-3) Emitir pasaporte</p> <p>Si el usuario es Cónsul y el estado de la solicitud es Autorizado se activara la casilla de Emitir pasaporte.</p>

3.2.2 Diagramas de Secuencia



sd Diagrama de Secuencia Actualizar Registro de Usuario





En esta Capitulo se realiza una verificación si nuestro sistema cumple con las normas de calidad.

$$R(t) = P [T > t] = 1 - Y(t)$$

Considerando que el tiempo T para fallar es una variable aleatoria exponencial entonces se tiene:

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

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

Donde:

Landa Es la tasa constante de fallo

t: Periodo de operación en tiempo

Una vez obtenido la función de confiabilidad, se ira analizando la confiabilidad de cada uno de los módulos definidos anteriormente, habiendo realizando un periodo de pruebas se constato los siguientes datos para cada modulo del sistema.

TABLA 4.1 RESULTADO DE CONFIABILIDAD DE LOS MÓDULOS

R ₁ : λ = 0.06, t = 2	R ₁ = e ^{-λt} = e ^{-(0.06)(2)} = 0.86
R ₂ : λ = 0.03, t = 1	R ₂ = e ^{-λt} = e ^{-(0.03)(1)} = 0.97
R ₃ : λ = 0.05, t = 6	R ₃ = e ^{-λt} = e ^{-(0.05)(6)} = 0.62
R ₄ : λ = 0.03, t = 2	R ₄ = e ^{-λt} = e ^{-(0.03)(2)} = 0.94
R ₅ : λ = 0.07, t = 5	R ₅ = e ^{-λt} = e ^{-(0.07)(5)} = 0.70
R ₆ : λ = 0.06, t = 2	R ₆ = e ^{-λt} = e ^{-(0.06)(2)} = 0.87
R ₇ : λ = 0.02, t = 1	R ₇ = e ^{-λt} = e ^{-(0.02)(1)} = 0.98

Conociendo la contabilidad de cada uno de los módulos del sistema, el siguiente paso es analizar la contabilidad del sistema. Para esto consideramos dos situaciones: en un caso considera que el sistema entero falla, si falla cualquiera de sus componentes, en otro caso se considera que el sistema falla si y solo si falla todos sus componentes.

La primera situación modela una conexión de subsistemas en serie y la segunda modela una conexión en paralelo de los subsistemas o componentes.

En nuestro caso el análisis de la contabilidad del sistema se lo realiza tomando en cuenta un sistema con conexión compuesta.

Realizando los cálculos correspondientes tenemos:

$$\begin{aligned}
 R_a &= R_1 = 0.86 \\
 R_b &= 1 - (1-R_2) (1-R_3) (1-R_4) (1-R_5) (1-R_6) (1-R_7) \\
 R_b &= 1 - (1-0.97)(1-0.62)(1-0.94)(1-0.70)(1-0.87)(1-0.98) \\
 R_b &= 0.98 \qquad \text{En porcentaje } R_b = 98\%
 \end{aligned}$$

Lo que indica que la confiabilidad del sistema de Sistema de Administracion de Solicitud de Pasaportes en la Web para la Cancillería es de 98%

4.2 FUNCIONALIDAD

Se debe determinar cinco características de dominio de la información:

- Numero de entradas de osario
- Numero de salidas de usuario
- Numero de peticiones de usuario
- Numero de archivos

Conociendo la formula de Punto de Fusión (PF):

$$PF = CUENTA \text{ total} * (X + Y * \sum Fi) \quad (1)$$

PARÁMETROS DE MEDICIÓN	CUENTA	FACTOR	TOTAL
Numero de entradas de usuario	10	2	20
Numero de salidas de usuario	16	4	46
Numero de peticiones de usuario	7	3	21
Numero de archivo	8	6	48
CUENTA TOTAL			135

TABLA 4.2 SUMA DE PARÁMETROS DE PUNTO DE FUSIÓN

Los valores de ajuste de complejidad se obtienen respondiendo a las preguntas destacables que se muestra en la fig. Del sistema.

TABLA 4.3 VALORES DE AJUSTE DE COMPLEJIDAD

Métricas	Sin Influencia 0%	Incidencia 20%	Moderado 40%	Medio 60%	Significativo 80%	Esencial 100%
FACTOR	0	1	2	3	4	5
1. ¿Requiere el sistema copias de seguridad y recuperación fiables?					X	
2.- ¿Se requiere comunicación de datos?					X	
3.- ¿Es crítico el rendimiento?				x		
4.- ¿Será ejecutado el sistema en un S.O. existente?					X	
5.- ¿Requiere la entrada de datos interactiva?					X	
6.- ¿Se actualiza los archivos maestros de forma interactiva?		X				
7.- ¿Son complejas las entradas, salidas, peticiones y archivos?			x			
8.- ¿Se ha utilizado el código para ser utilizable?					X	

La sumatoria del punto de fusión:

$$F_i = 4 + 4 + 3 + 4 + 4 + 1 + 2 + 4 = 26 \quad (2)$$

Reemplazando valores (2) en la formula (1), tenemos:

Donde:

PF: Medida de función concedida a la aplicación como valor de normalización CUENTA total

: Es la suma de las entradas obtenidas en la tabla

X: Nivel de confiabilidad del sistema igual a 98%

Y: Nivel de significación de error igual a 0.01

ΣF_i : Son los valores de ajuste de la complejidad

$$PF = 135 \cdot (0.98 + (0.01 \cdot 26)) = 135 \cdot (1.24) = 167$$

Este PF supone 167 líneas de código.

Si $F_i = 52$ considerando el 100%. Reemplazando en la ecuación (1) se tiene:

$$P_f = 135 \cdot (0.98 + (0.01 \cdot 52)) = 135 \cdot (1.5) = 203$$

Por lo tanto la funcionalidad esta dado por:

$$\% \text{funcionalidad} = PF/P_f = 167/203 = 0.82$$

Esto quiere decir que la funcionalidad del sistema es del 82%.

4.4 PORTABILIDAD

En el sistema desarrollado gracias a se esta manejando el gestor de base de datos SQL Server, se puede exportar sus datos con diferentes gestores de manera confiable y 100 % compatibles con varios gestores de bases de datos.

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

En esta Capitulo se realiza las conclusiones y recomendaciones tras la culminación del presente proyecto

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

En el desarrollo del presente proyecto se pretende dar un mejor servicio a los Consulados de Bolivia juntamente con la Cancillería, una mejor accesibilidad al sistema para los Usuarios, debido a la distancia entre los otros países con relación a Bolivia se desarrollo el sistema vía Web.


A continuación se describe las consideraciones observadas durante la implementación y uso del sistema:

1. Se logro cumplir con los objetivos planteados.
2. La aplicación de la metodología OOHDM ayudo de gran manera en el desarrollo del sistema especialmente en los puntos de diseño navegacional y en el diseño de interfaz abstracta, ya que muestra mediante los diagramas la estructura que tendrá las distintas clases navegacionales.
3. Se facilito la comunicación entre Los Cónsules y la Cancillería dentro el Sistema via WEB.

5.2 RECOMENDACIONES

- Se recomienda el “versionamiento” es decir en cada gestión implementar nuevos requerimientos debido a que los mismos son volubles.
- En cuanto a la Plataforma de .NET migrar a nuevas versiones, ya que cada versión incluye reducción de código y mejora en la calidad de interfaz para el usuario.
- Usar las nuevas tecnologías y herramientas AJAX's, XML, LINQ y otros que son compatibles con la plataforma .NET

REFERENCIAS BIBLIOGRÁFICAS



En esta parte se hace referencia a los textos, direcciones electrónicas, etc. las cuales tuvieron fuentes de información.

BIBLIOGRAFÍA

LARMAN, 1999 Larman. Craig. *UML y Patrones*, Primera Edición. Prentice Hall. México IW.

[PRESSMAN, 2005] Pressman, R. *Ingeniería de Software*, Quinta edición. Editorial Me Graw . México. 2004,

[JACOBSON, 1999] Jacobson í. *Pi proceso unificado de desarrollo de software*. Me.

Grave Ilii Interamericana de España 2000

[BUNGK, 1979] Bunge, Mario. *La investigación científca*. Editorial Ariel de Barcelona

(KENDALL, 1997} Kendal. Keneth & Kendalh Julie E. *Análisis y Diseño de*

Sistemas". Tercera Edición. Prentice Malí Hispanoamericana S.A.

[FOW99] Martin Fowler, Kendall Scott(1999), *UML Gota a Gota*, Addison Wesley Logman de México, S.A. México 1999. UMSA Biblioteca de Informática.

[PER04] Pérez Vargas José Luis(2005), *Gestión de Traspasos Via Web*. UMSA Biblioteca de Informática.

[PRE 98] Presuman S. Roger, *Ingeniería del Software*, Edición en Español Cuarta Edición 1998.

[RJBOO] Jame Rumbaugh, Ivar Jacobson , Grady Booch(2000), *El lenguaje unificado de Modelo*. Manual de Referencia. Pearson Educación S.A. Madrid.

[BoochO] Grady Booch: *The Architecture of Web Applications*. DeveloperWorksJBM developer solutions. June 2001.

REFERENCIAS WEB

[GOO06] Google (2006). *Definición de Web en la Internet*. Extraído el 20 de abril de 2006. [ww/w.google.com](http://www.google.com).

[PRO03] Proyecto Pixel (2003), Extraído el 29 de abril de 2006. <http://www.proyectopixel.com/qlosario.html>.

[SIS05] Sistemas Estratégicos(2005), extraído el 2 de abril de 2006. <http://www.telesoft-crm.com/>

[MAR 97] Gerardo Moreno Martinez(1997), *Ingeniería del Software UML*, extraído el 2 de junio de 2006.

<http://www.monografias.com/trabajos5/insoft/insoft.html>.

[PHP03] Php, manuales de programación. Extraído el 12 de febrero de 2006. <http://www.php.arch.com>.

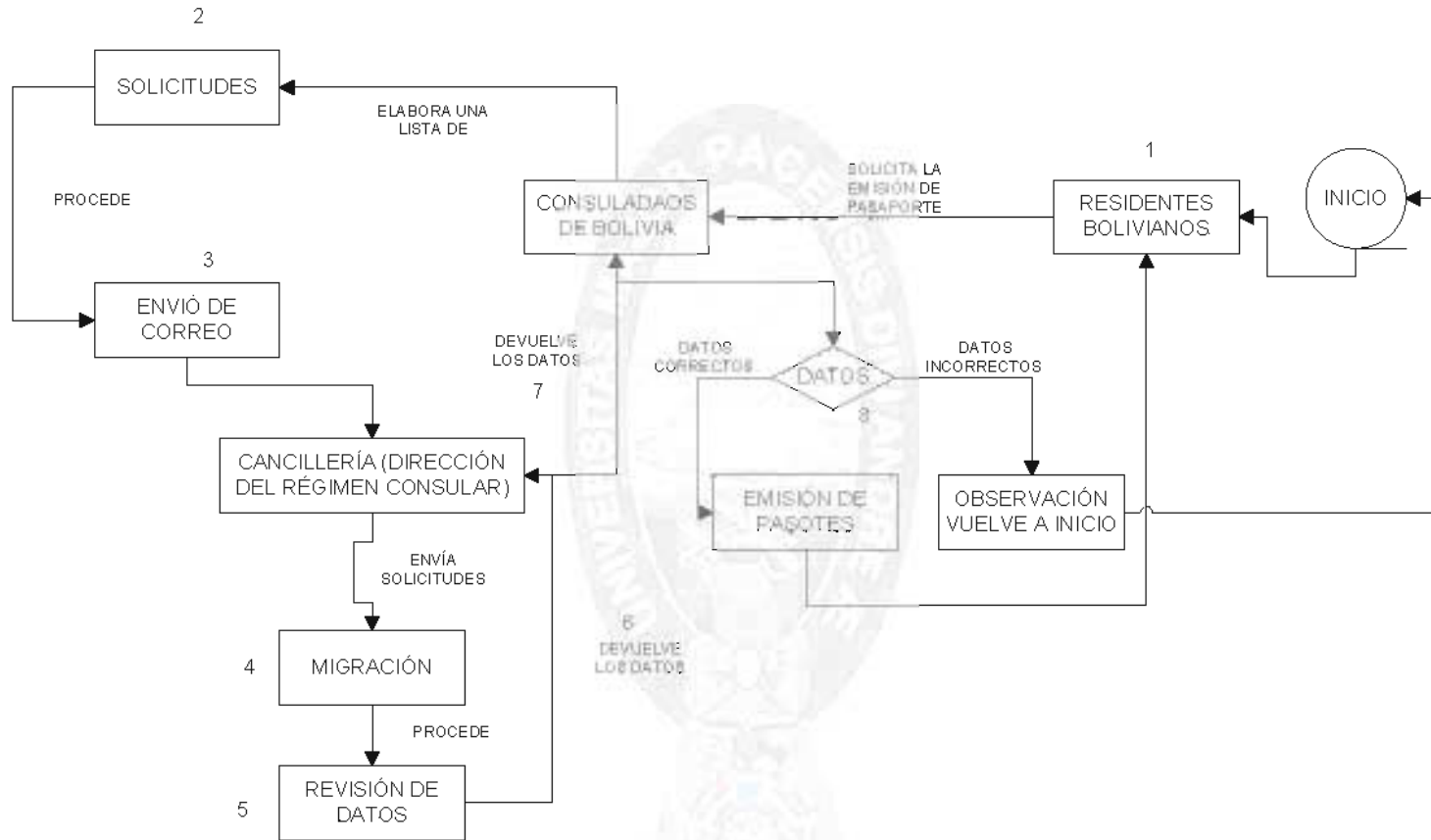
[ABR05] Silva Abrahao(2005), *Calidad de sistemas de información web*, extraído el 1 de junio de 2006. <http://www.dsic.upv.es>.

[GUE06] Luis A. Guerrero(2006), *Modelando Aplicaciones Web con UML* extraído el 20 de abril de 2006. <http://www.luis.com>.

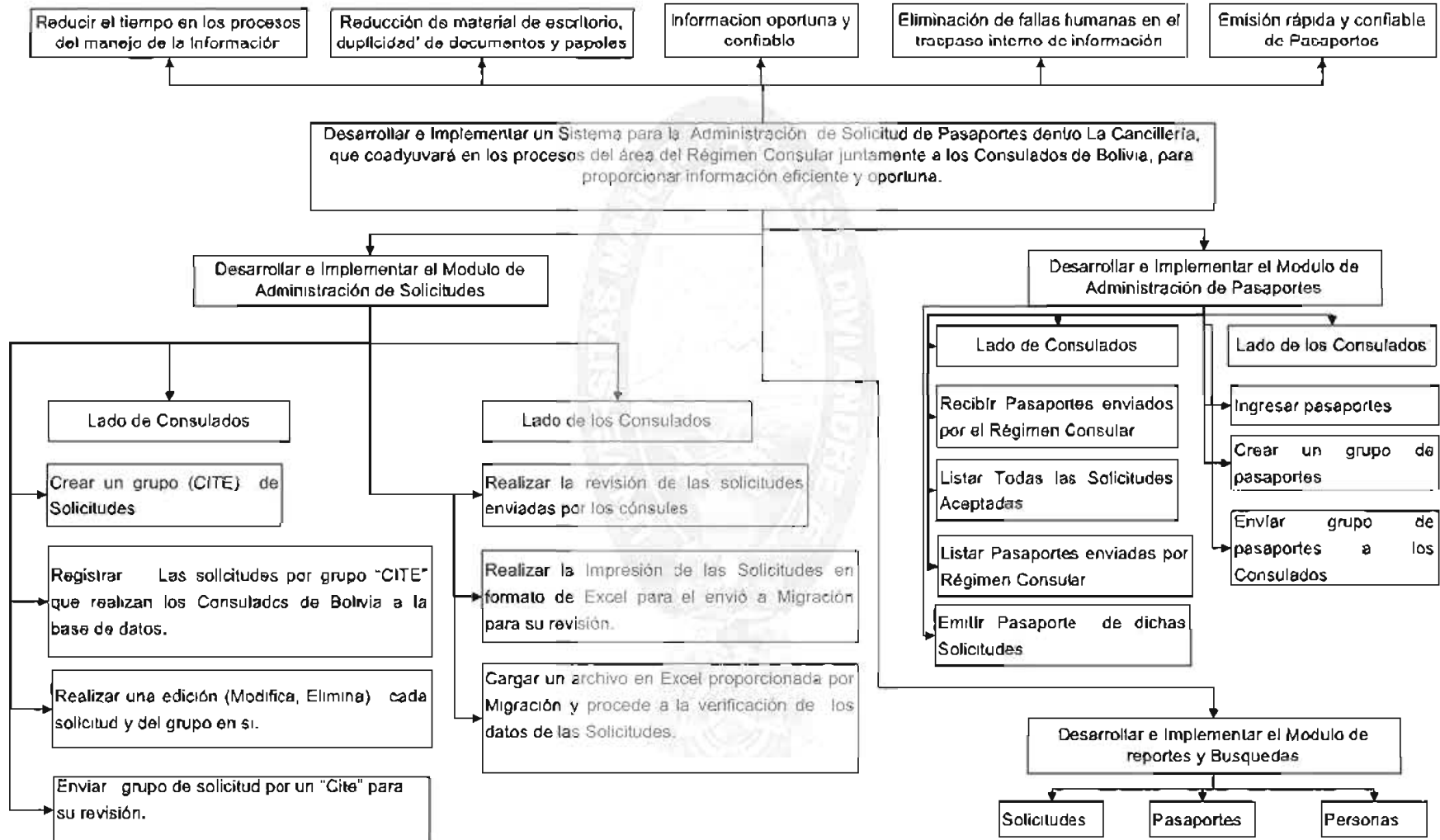
[ACE06] Cesar Fernández Acebal. *Usabilidad*. Extraído el 12 de abril de 2006 www.cesaracebal.com.

En esta parte condene el árbol de problemas árbol de objetivos y el marco teórico, glosario de términos

SITUACIÓN DEL SISTEMA ACTUAL



Árbol de Objetivos



Glosario de Términos

Sistema:

Es la combinación de software y hardware que da una solución a un problema de negocios.

El desarrollo de sistemas:

Es la creación de un programa para un cliente o *ente*, este último es quien tiene el problema que debe ser resuelto.

Analista:

Es el que documenta el problema del cliente y lo comunica a los *desabolladores*, que son los programadores que generarán el programa que resolverá el problema y lo distribuirán en equipos de computación.

Actor:

Es una persona, un componente de hardware, un lapso u otro sistema