

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



PROYECTO DE GRADO

**“GESTIÓN DE ACTIVIDADES DE JÓVENES -
MISIÓN BOLIVIANA OCCIDENTAL”**

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

POSTULANTE: Gladys Fresia Vargas Mamani

TUTOR: Lic. Roberto Vargas Blacutt

REVISOR: Lic. Marcelo Aruquipa Chambi

**LA PAZ – BOLIVIA
2009**

DEDICATORIA



Dedicado con mucho cariño y amor:

A DIOS por amarme y no abandonarme en los momentos que más lo necesito, dándome amor, sabiduría, fortaleza y valentía día a día.

A mis padres, Jorge Vargas Chino y Lorenza Mamani de Vargas por el amor y apoyo incondicional que me brindan y la formación que me dieron para luchar en la vida.

A mis hermanos, Eddy, Richard, Miguel, Silvio, Lucy y Mary por su amor y constante apoyo en las tareas que me propuse realizar

AGRADECIMIENTOS

A Dios, por todo cuanto me a dado y apoyado en todo momento durante toda mi vida.

A mi docente tutor Lic. Roberto Vargas Blacutt, por haberme brindado la colaboración, consejos y tiempo necesarios para la conclusión de mi proyecto de grado.

A mi docente revisor Lic. Marcelo Aruquipa Chambi, por la comprensión, apoyo, tiempo, dedicación, orientación y sugerencias que me condujeron a la finalización del proyecto de grado.

Al personal de la Misión Boliviana Occidental, por su colaboración y confiaron siempre con el desarrollo de este proyecto, a quienes doy mi mas sincera gratitud.

A mis padres, por su amor, apoyo, paciencia y confianza, que siempre me dieron a lo largo de mi vida personal y profesional.

A mis hermanos, por su comprensión y apoyo espiritual que me dieron la fuerza para seguir adelante.

A todos mis amigos y amigas de verdad, por su amor y amistad incondicional, en especial a Hernán Quispe Pañuni, Viviana Mamani Copa, Reyna Marca Poma, Nelly Luna Troche y Benicia Pérez Patzi.

¡¡¡Muchas Gracias!!!

RESUMEN

El proyecto de grado desarrollado es un sistema de Gestión de actividades de Jóvenes, que coadyuvará en la gestión, seguimiento y control de información de iglesias, clubes, eventos, actividades, seguros y miembros de Ministerio joven de la Iglesia Adventista del 7^a día del occidente boliviano que esta a cargo de la Misión Boliviana Occidental (M. B. O.) la misma que atiende a 30 distritos del Occidente Boliviano.

A lo largo del desarrollo del sistema se utilizó la metodología Proceso Unificado Racional (RUP), cuyo marco de trabajo permitió generar un producto software de calidad acorde a los requerimientos de los usuarios, los modelos de esta aplicación han sido elaborados con el lenguaje de Modelado Unificado (UML), que nos permite documentar todo el desarrollo del sistema y el Método de Diseño Hipermedia Orientado a Objetos (OOHDM) para la navegación. La metodología RUP y OOHDM junto a UML constituyen un estándar para el diseño y elaboración de sistemas Orientados a Objetos.

Para la implementación del sistema se utilizó el Modelo Vista Controlador, el lenguaje de programación PHP, la base de datos MySQL, JQuery que es una biblioteca o framework de Javascript., que interactúa con la tecnología AJAX a páginas Web y el servidor Apache, que se adecuan a las necesidades e implementación de la empresa.

El Sistema de Gestión de Actividades de Jóvenes –Misión Boliviana Occidental, beneficia de gran manera en el manejo control y seguimiento y acceso oportuno a toda la información referente a las actividades y eventos del Ministerio Joven, Logrando que este sistema sea un apoyo a la Organización en el trabajo que realiza todos los días, reduciendo el tiempo, en ingreso de datos, elaboración de informes, elaboración de reportes y la verificación de la documentación existente para la toma de decisiones dentro de la organización.

ÍNDICE

CAPITULO I

MARCO INTRODUCTORIO

		Pág.
1.1	Introducción.....	1
1.2	Antecedentes.....	2
1.3	Objeto de estudio.....	5
1.4	Planteamiento del problema.....	6
1.5	Justificación.....	7
1.6	Objetivos.....	7
1.6.1	Objetivo general.....	7
1.6.2	Objetivos específicos.....	7
1.7	Alcance.....	8
1.8	Metodología.....	8
1.9	Importancia del estudio.....	9

CAPITULO II

MARCO TEÓRICO

2.1	Conceptos generales.....	10
2.1.1	Gestión.....	10
2.1.2	Gestión de información.....	10
2.1.3	Indicadores de gestión.....	10
2.1.4	Sistema de información.....	11
2.1.5	Red.....	11
2.1.6	Internet.....	11
2.1.7	Servidor Apache.....	11
2.1.8	Gestores de base de datos.....	12
2.1.9	MySQL.....	12
2.1.10	PHP (Hypertext Pre-processor).....	12
2.2	Modelo Vista Controlador.....	13
2.2.1	Descripción del patrón.....	14
2.2.2	Framework.....	14
2.2.3	Tipos de Frameworks.....	16
2.3	Análisis y diseño orientado a objetos.....	17
2.3.1	Análisis orientado a objetos.....	17

2.3.2	Diseño orientado a objetos.....	18
2.4	Metodología.....	18
2.4.1	Metodología Rational Unified Process RUP.....	18
2.4.2	Características esenciales del RUP.....	19
2.4.3	Estructura Dinámica del proceso. Fases e iteraciones.....	19
2.4.4	Flujos de trabajo	20
2.4.5	Ciclo de vida del RUP.....	22
2.4.6	Fases del RUP.....	23
2.4.6.1	Fase de inicio.....	23
2.4.6.2	Fase de elaboración.....	24
2.4.6.3	Fase de construcción.....	23
2.4.6.4	Fase de transición.....	24
2.5	Metodología OOHDM.....	25
2.6	Lenguaje unificado de modelado-UML.....	26
2.6.1	Tipos de diagramas.....	26
2.6.3	Modelo del negocio.....	27
2.6.3.1	Diagrama de casos de uso.....	27
2.6.3.2	Registro de términos en el glosario.....	30
2.6.3.3	Diagrama de secuencia.....	30
2.6.3.4	Contratos de operaciones.....	31
2.6.3.5	Diagramas de estados.....	32
2.6.3.6	Descripción de Casos de Uso Real.....	32
2.6.3.7	Diagrama de objetos.....	33
2.6.3.8	Diagrama de colaboración.....	33
2.6.3.9	Diagrama de Clases.....	33
2.6.3.10	Diagrama de componentes.....	34
2.6.3.11	Diagrama de despliegue.....	34
2.6.3.12	Diagrama de implementación.....	35
2.7	Modelo de tres capas.....	36
2.8	Arquitectura cliente / servidor.....	37
2.9	Políticas y normas de seguridad de la información.....	37
2.10	Métricas de calidad del software.....	38
2.10.1	Funcionalidad.....	39
2.10.2	Confiabilidad.....	41

2.10.3	Mantenibilidad.....	42
2.10.4	Usabilidad.....	43
2.10.5	Portabilidad.....	43
2.11	Firewall (Cortafuego).....	43
2.12	Estimación del coste del proyecto.....	44
2.12.1	Las métricas orientadas al tamaño.....	45
2.12.2	El método Cocomo II.....	45

CAPITULO III

MARCO APLICATIVO

3.1	Fase de inicio.....	47
3.1.1	Planificación Del proyecto.....	47
3.1.2	Estudio preliminar.....	50
3.1.3	Los requerimientos.....	52
3.1.4	Modelo del negocio.....	53
3.1.4.1	Actores.....	54
3.1.5	Funciones del sistema.....	55
3.2	Fase de elaboración.....	57
3.2.1	Diagrama de actividades.....	57
3.2.2	Modelo de requisitos.....	59
3.2.2.1	Diagramas de casos de uso.....	59
3.2.2.2	Descripción de los diagramas de casos de uso expandidos.....	61
3.2.2.3	Construcción del modelo Entidad relación.....	72
3.2.2.4	Construcción del modelo conceptual.....	72
3.2.2.5	Diagrama de estados.....	75
3.2.3	Modelo de análisis y diseño.....	76
3.2.3.1	Diagrama de secuencias.....	76
3.2.3.2	Diagrama de clases.....	79
3.2.3.3	Diagramas de colaboración	80
3.2.3.4	Modelo de datos.....	82
3.2.3.5	Arquitectura del sistema.....	84
3.2.3.6	Modelo de implementación.....	86
3.3	Fase de construcción.....	88
3.3.1	Modelo de análisis y diseño.....	88

3.3.1.1	Diagrama de componentes.....	89
3.3.1.2	Diagrama de despliegue.....	89
3.3.1.3	Modelo de implementación.....	90
3.3.2	Diseño navegacional.....	91
3.3.3	Diagrama de interfaz abstracta.....	92
3.3.4	Interfaz de usuario.....	95
3.4	Fase de transición.....	98

CAPITULO IV

SEGURIDAD, CALIDAD Y ANÁLISIS DE COSTO / BENEFICIO

4.1	Criterios de seguridad.....	99
4.1.1	Seguridad de la red.....	99
4.1.2	Seguridad de la aplicación.....	100
4.1.3	Seguridad de la Base de Datos.....	102
4.2	Métricas de calidad del software.....	103
4.2.1	Funcionalidad.....	103
4.2.2	Portabilidad.....	106
4.2.3	Mantenibilidad.....	106
4.2.4	Usabilidad.....	107
4.2.5	Confiabilidad.....	107
4.3	Análisis de costos.....	108
4.3.1	Costo del software desarrollado.....	108
4.3.2	Costo de elaboración del proyecto.....	110

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1	Conclusiones.....	111
5.2	Recomendaciones.....	113

BIBLIOGRAFÍA.....	114
--------------------------	------------

ANEXOS

Anexo A	Matriz del marco lógico.....	116
Anexo B	Diagrama de actividades.....	120
Anexo C	Diagramas de casos de uso.....	122

Anexo D	Diagramas de estados.....	128
Anexo E	Diagramas de secuencias.....	130
Anexo F	Diagramas de colaboración.....	132
DOCUMENTACIÓN.....		134



INDICE DE FIGURAS

		Pág.
Figura 1.1	Estructura Orgánica de la Misión Boliviana Occidental.....	2
Figura 2.1	Modelo MVC.....	13
Figura 2.2	Patrón del Modelo vista controlador.....	14
Figura 2.3	Actividades de desarrollo.....	17
Figura 2.4	Ciclos, releases, baseline.....	20
Figura 2.5	Fases e hitos en RUP.....	20
Figura 2.6	Estructura del RUP y esfuerzo en actividades según fase del proyecto	23
Figura 2.7	Casos del negocio	27
Figura 2.8	Representación de caso de uso y actor de UML.....	28
Figura 2.9	Representación de casos de uso de alto nivel.....	28
Figura 2.10	Caso de uso expandido	29
Figura 2.11	Diagrama de Secuencia	31
Figura 2.12	Transición de estados.....	32
Figura 2.13	Diagrama de colaboración	33
Figura 2.14	Representación de un paquete.....	34
Figura 2.15	Diagrama de despliegue	35
Figura 2.16	Modelado del diagrama de implementación.....	35
Figura 2.17	Modelo de tres capas	36
Figura 3.1	Representación en las capas, reservas de eventos.	51
Figura 3.2	Modelo del negocio del Sistema	54
Figura 3.3	Especificación de actores.....	54
Figura 3.4	Diagrama de actividades: Inscripción de nuevos miembros...	57
Figura 3.5	Diagrama de actividades: Inscripción, reserva o confirmación a evento.....	58
Figura 3.6	Diagrama de actividades: Pago de seguro.....	59
Figura 3.7	Muestra un diagrama de casos de uso real	60
Figura 3.8	Muestra un diagrama de casos de uso para el sistema.....	60
Figura 3.9	Diagrama de caso de uso: Inscripción de nuevos miembro....	61
Figura 3.10	Diagrama de caso de uso: Inscripción a evento.....	64
Figura 3.11	Diagrama de caso de uso: Pago de seguro.....	69
Figura 3.12	Diagrama del modelo entidad relación.....	72

Figura 3.13	Diagrama del modelo conceptual del sistema.....	73
Figura 3.14	Diagrama de estados de inscripción de nuevo miembro.....	75
Figura 3.15	Diagrama de estados inscripción de y/o reserva a evento....	75
Figura 3.16	Diagrama de estados de Pago de seguro.....	76
Figura 3.17	Diagrama de secuencia de inscripción de nuevo miembro....	77
Figura 3.18	Diagrama de secuencia inscripción o reserva o confirmación a evento	77
Figura 3.19	Diagrama de secuencia de Pago de Seguro.....	79
Figura 3.20	Diagrama de clases	80
Figura 3.21	Diagrama de colaboración de inscripción de nuevo miembro	81
Figura 3.22	Diagrama de colaboración inscripción o reserva o confirmación a evento	81
Figura 3.23	Diagrama de colaboración inscripción de nueva iglesia.....	82
Figura 3.24	Diagrama Modelo de datos del sistema.....	83
Figura 3.25	Arquitectura lógica del sistema.....	85
Figura 3.26	Diseño físico y arquitectura del Sistema.....	86
Figura 3.27	Diagrama de colaboración de inscripción de nuevo miembro	87
Figura 3.28	Código de inscripción de nuevo miembro.....	88
Figura 3.29	Diseño físico y arquitectura del Sistema.....	89
Figura 3.30	Diseño físico y arquitectura del Sistema.....	89
Figura 3.31	Diseño navegacional del software.....	91
Figura 3.32	AVD Aplicación De la Web Sistema de Gestión de Actividades de Jóvenes.....	92
Figura 3.33	AVD de administración de datos del Sistema, administrador.....	93
Figura 3.34	ADV Inscripción y ADV Usuarios registrados.....	94
Figura 3.35	ADV seguro.....	95
Figura 3.36	Pantalla del Formulario de autenticación.....	96
Figura 3.37	Pantalla de adición de nuevo miembro de Iglesia.....	96
Figura 3.38	Pantalla Modificación o eliminación de datos del miembro de Iglesia.....	97
Figura 3.39	Pantalla Inscripción a evento.....	97
Figura 3.40	Pantalla de creación y emisión de publicaciones.....	8
Figura 4.1	Esquema de Firewall típico entre red local e Internet.....	100

Figura 4.2	Esquema de seguridad del software.....	102
------------	--	-----

INDICE DE TABLAS

		Pág.
Tabla 2.1	Casos de uso de alto nivel	29
Tabla 2.2	Casos de uso expandido.....	30
Tabla 2.3	Elementos del glosario de términos.....	30
Tabla 2.4	Estructura, de un Contrato.....	31
Tabla 2.5	Caso de Uso Real	32
Tabla 2.6	Factores de calidad ISO 9126.....	39
Tabla 2.7	Dominio de información de Puntos Función y su estimación..	39
Tabla 2.8	Dominio de información de Puntos Función.....	40
Tabla 2.9	Categoría de un proyecto en función de sus líneas de código	45
Tabla 2.10	Coeficientes COCOMO.....	46
Tabla 3.1	Tiempos estimados en las fases de RUP.....	47
Tabla 3.2	Plan de actividades de las cuatro fases de RUP.....	48
Tabla 3.3	Descripción de los hitos por fase.....	49
Tabla 3.4	Artefactos de las distintas disciplinas del RUP.....	50
Tabla 3.5	Especificación de actores.....	55
Tabla 3.6	Funciones básica.....	56
Tabla 3.7	Atributos del sistema.....	56
Tabla 3.8	Glosario.....	74
Tabla 3.9	Requerimientos de hardware.....	90
Tabla 4.1	Niveles de acceso a la información de los usuarios.....	101
Tabla 4.2	Cálculo de punto función.....	104
Tabla 4.3	Tabla de factor de ajuste.....	104
Tabla 4.4	Valores Ajuste de Complejidad.....	105
Tabla 4.5	Ponderación de Usabilidad.....	107
Tabla 4.6	Fallas por módulos.....	108
Tabla 4.7	Conversión de Puntos de Función a KLDC.....	109
Tabla 4.8	Coeficientes COCOMO.....	109
Tabla 4.9	Costo de elaboración del proyecto.....	110
Tabla 4.10	Costo Total.....	110



CAPÍTULO I

MARCO INTRODUCTORIO

CAPITULO N° I

MARCO INTRODUCTORIO

1. INTRODUCCIÓN

En la actualidad, vivimos en un mundo donde la tecnología y la información llegan a niveles donde su uso se convierte en una necesidad en todas las áreas, permitiendo reducir precios, costos, aumentar capacidad de procesamiento, almacenamiento y transmisión de información, mejorar las operaciones, tener mayor capacidad de entrega de servicios, optimización de recursos, etc.

Las organizaciones buscan herramientas que les permitan el logro y el cumplimiento de sus objetivos, llegando así a depender de las nuevas tecnologías, haciendo uso de los sistemas de información y el uso del Internet que al ser ejecutados proporcionan apoyo a la toma de decisiones y control de las actividades que desarrollan, rompiendo así fronteras geográficas, acortando el tiempo de comunicación.

El uso de estas nuevas tecnologías proporcionan a las organizaciones mejoras en su imagen y colabora con el desarrollo de una estrategia de promoción y comunicación.

En consecuencia se plantea el Sistema de Gestión de Actividades de Jóvenes - Misión Boliviana Occidental que logra reducir costos y tiempo, dando solución a las necesidades de gestión del Ministerio de joven de la Iglesia Adventista del 7ª día del occidente boliviano que esta a cargo de la Misión Boliviana Occidental (M. B. O.) la misma que atiende a 30 distritos del Occidente Boliviano. Este sistema proporciona mayor exactitud e información actualizada, siendo una herramienta útil para la asistencia a los procesos que se realizan manualmente en la actualidad.

1.2 ANTECEDENTES

La Misión Boliviana Occidental (M.B.O.) de la Iglesia Adventista del Séptimo Día (I.A.S.D.), es una organización privada sin fines de lucro que funciona en la ciudad de La Paz, ubicada en la Calle Rosendo Villalobos Esq. Francisco de Miranda Nro. 1592, Zona: Miraflores, creada en el año 1907. La Misión Boliviana Occidental tiene la siguiente estructura.

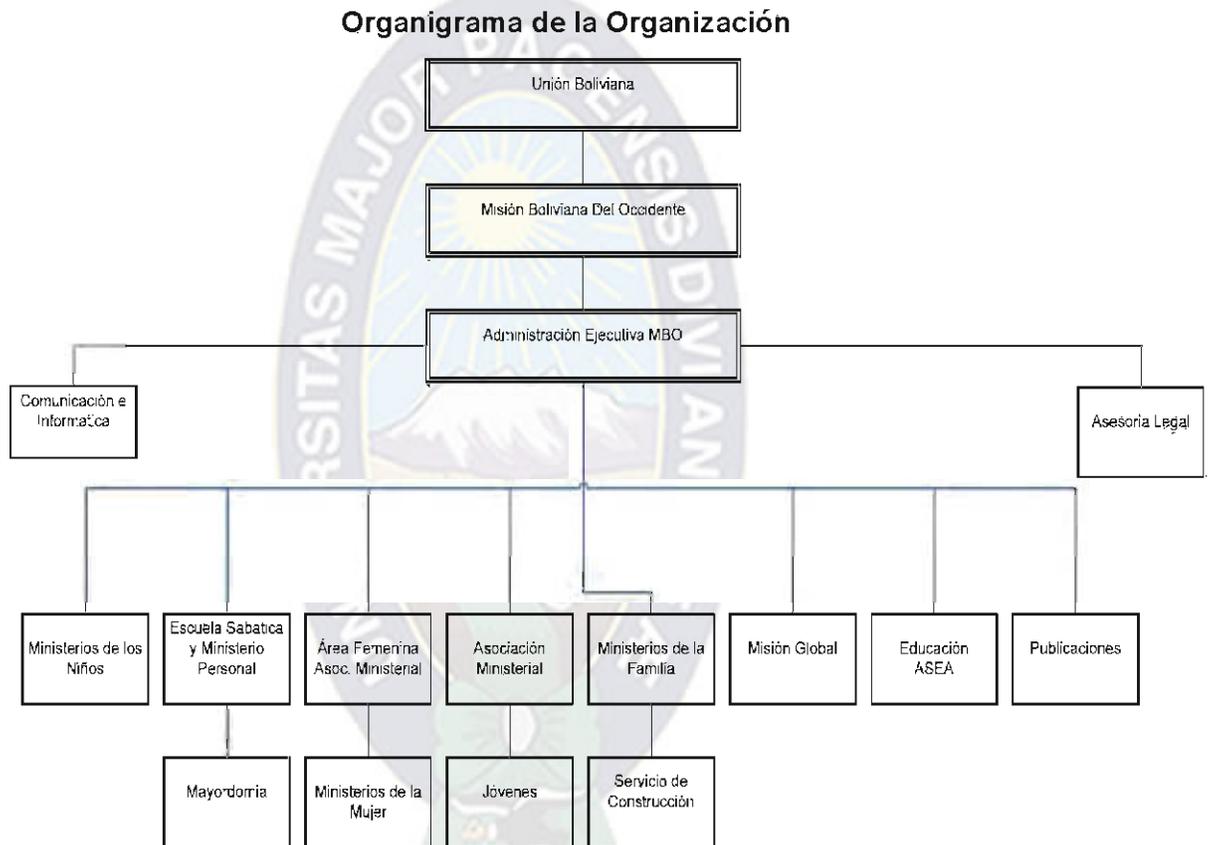


Figura 1.1. Estructura Orgánica de la Misión Boliviana Occidental.

Fuente: [Información proporcionada por el Departamento de Tesorería M.B.O.]

La Misión Boliviana Occidental cuenta con diferentes unidades como ser:

Administración Ejecutiva M.B.O., cuenta con las oficinas administrativas que son los Departamentos de Presidencia, Tesorería y Secretaria (Informe M.B.O., 2009).

- **Departamento de Presidencia,** en este Departamento funciona el máximo nivel jerárquico del orden administrativo de la Misión Boliviana Occidental.

- **Departamento de Tesorería**, en este Departamento se maneja todo lo relacionado a las finanzas y a la administración de la M.B.O.
- **Departamento de Secretaria**, este departamento funciona todo lo referente a las reuniones de la M.B.O., control de los miembros de cada distrito e iglesias con una información pertinente. (Informe M.B.O, 2009).

Asociación Ministerial (Ministerio Joven Adventista), que alberga la Oficina central a cargo de Departamento de Secretaria la cual tiene a su vez que coordinar con los coordinadores a nivel departamental del occidente Boliviano, coordinadores distritales y directores de jóvenes adventistas (J.A.) de iglesia para organizar todas las actividades de jóvenes de la M.B.O. como ser: camporis, encuentros espirituales, encuentros deportivos, reuniones, vigiliyas, ayunos, caminatas, campañas, concilios, conferencias y seminarios,. La función de este departamento es: equipar, fortalecer y asegurar la integración de los jóvenes en todos los aspectos de la vida con la iglesia.

Ministerios de los Niños, alberga la oficina del Departamento de Niños. Es un departamento que tiene la función de coordinar y facilitar actividades que promuevan el fortalecimiento espiritual de los niños.

Área Femenina- Asociación Ministerial (Ministerio de la Mujer), comprende la oficina del Departamento del Ministerio de la Mujer. Es un departamento cuya función es de coordinar y facilitar actividades que promuevan el fortalecimiento de las mujeres adventistas, con seminarios y cursos.

Por las investigaciones que se realizó, se pudo observar que el manejo de la información dentro y fuera de la Misión Boliviana Occidental es de forma manual ya que la información de actividades y proyectos (camporis, retiros espirituales, encuentros, reuniones, vigiliyas, ayunos, caminatas, campañas, concilios, conferencias, capacitaciones, cursos, convenciones y seminarios), el registro de los datos de kardex de los miembros, reserva de inscripciones a las actividades, reservación de material y el acceso a la información es manual y no esta actualizada.

Actualmente la Misión Boliviana Occidental (M.B.O.) no cuenta con un sistema información que le permita hacer un seguimiento de las actividades que se realizan en el Ministerio Joven, así también no cuentan con una base de datos de los miembros.

En cuanto a sistemas similares se han desarrollado e implementado algunos sistemas a nivel internacional como:

- [Sistema para la iglesia Metodista, 2006, Perú, <http://www.iglesiametodista.org.pe>] *Sistema para la iglesia Metodista de Perú*, que es un portal Web que ofrece registro de nuevos interesados, noticias de la iglesia Metodista, noticias metodistas y venta de libros.
- [Sistema Maná, 2009, Bolivia, <http://www.Sistema Mana.org>] *Sistema Maná*, la cual ofrece información de la Iglesia Católica en Bolivia y la pedagogía de la misión permanente. Como también ofrece noticias de actividades a realizarse, convocatorias, mensajes, entrevistas, suscripciones, videos y suscripciones a nivel nacional e internacional.
- [Sitio Web de la Iglesia de los Santos de los últimos días, Argentina, Uruguay; Paraguay, 2009, <http://www.LaIglesia.deJesucristo.de.los.Santos.de.lo.Últimos.Días.org/>,] *Sitio Web de la Iglesia de Jesucristo de los Santos de lo últimos días (Mormones)*, que proporciona información acerca de su organización, actividades, glosario de palabras y preguntas mas frecuentes.
- [Sitio Web de los testigos de Jehová, Bolivia, 2007, <http://www.corazones.org>] *Sitio Web de la Iglesia de los Testigos de Jehová*, está ofrece noticias de la iglesia y actividades, su organización, mensajes, fotos de la iglesia y sus miembros.

En cuanto a sistemas similares al propuesto se han desarrollado algunos proyectos de grado en la carrera de Informática que se relaciona con el presente proyecto, que a continuación se mencionan:

- [Centellas, 2002]: *Sistema de información seguimiento de proyectos Iglesia Evangélica* Gestiona proyectos que se generan en las diferentes comunidades, pertenecientes al distrito de la Iglesia Metodista, los proyectos que contempla es la construcción de postas médicas, de iglesias, casas pastorales, etc.

- [Torrez, 2007]: *Sistema de administración y control de clubes: caso Iglesia Adventista del 7º día* que genera información acerca de la administración de clubes realizando Inscripciones, pagos y control de inscripciones de los grupos denominados A.C.G. (Aventureros, conquistadores, guías mayores).
- [Irahola, 2002]: Sistema de control de Proyectos Inti Watana en este proyecto de grado se desarrollo un sistema de control y seguimiento de los proyectos sociales, culturales y educativos del centro alternativo de desarrollo integral (CADI)

1. 3 OBJETO DE ESTUDIO

La Misión Boliviana Occidental organiza todas las actividades del Ministerio Joven de la Iglesia Adventista del 7º día la cual está dividida en 30 distritos del occidente de Bolivia. En cada distrito existe de 8 a 10 iglesias y cada iglesia tiene un promedio total de 100 a 300 miembros. Este ministerio organiza a los jóvenes que comprenden de los 14 a 30 años. Estos miembros de la Iglesia tienen tres directivos por iglesia que son:

1. Director (a) de jóvenes adventistas
2. Sub-director (a) de jóvenes adventistas
3. Secretario (a) de jóvenes adventistas

Los cuales son notificados para asistir a las reuniones que se llevan acabo en fechas programadas, para planificar e informarles acerca de las actividades y proyectos ya mencionados anteriormente que se realizaran, por lo que la información de estas actividades es por medio del directivo y no así directamente por el miembro del Ministerio Joven. De igual manera la inscripción a nuevas actividades, inscripción de nuevos miembros, entrega de material, realización de consultas e informes de las actividades realizadas son de forma manual.

El presente proyecto de grado coadyuvara en la gestión y seguimiento de actividades del Ministerio Joven de la M.B.O. (Misión Boliviana Occidental), busca aportar un medio de comunicación entre los directivos, miembros de este Ministerio, padres de familia o tutores y sociedad en general.

1. 4 PLANTEAMIENTO DEL PROBLEMA

En la Misión Boliviana Occidental se identificaron los siguientes procesos manuales:

- a) Reservación e inscripciones de los miembros a las actividades planificadas y el registro de seguros anual y por evento se lo realizan en hojas de cálculo y son por medio del administrativo y directivo.
- b) El registro personal de los directivos y miembros del Ministerio Joven de cada distrito e iglesia es voluminosa ya que sobrepasa los 1000 registros personales por distrito.
- c) Dificultad en la centralización de la información de los distritos con su correspondiente al Ministerio Joven, ya que toda esta información es centralizada en formularios impresos en papel.
- d) Demora en la ejecución de proyectos por falta de actualización de registros ya que sus datos no están actualizados permanentemente, la búsqueda y la actualización de los datos tarda al menos una semana por distrito.
- e) Falta de seguimiento y control permanente a las actividades del plan anual del ministerio Joven por lo que no existe un control de las actividades y eventos.
- f) Falta de comunicación de padres de familia o tutores con los directivos existiendo por esto menor participación de los jóvenes en las actividades planificadas.
- g) Dificultad en la generación de informes ya que son inadecuados e inoportunos ya que estos informes están desactualizados.
- h) Dificultad en el acceso a consultas de las actividades, noticias, publicaciones y eventos que se realizan por parte de todos los miembros, ya que no existe un sistema de información al cual puedan acceder todos los miembros del Ministerio Joven.

Por lo tanto el problema ha resolver es:

¿El Sistema de Gestión de Actividades de Jóvenes - Misión Boliviana Occidental vía Web permitirá realizar el manejo y seguimiento de actividades que se realizan en el Ministerio de Jóvenes Adventistas?

1. 5 JUSTIFICACIÓN

La Misión Boliviana Occidental cuenta con 18 computadoras, también cuenta con licencias de Windows Server 2000, con dominio hosting. Además cada distrito tiene 4 computadoras, a si mismo el acceso a Internet lo que permitirá al aprovechamiento optimo de dichos recursos para la implementación del Sistema de Gestión de Actividades de Jóvenes.

Económicamente, el proyecto debe incrementar el rendimiento de la organización, para mejorar el procesamiento y manejo de la información, el cual disminuirá pérdidas económicas al no contar con un sistema automatizado.

Socialmente, la implementación del sistema ayudará a tener mayor participación de personal involucrado además de obtener mejor información para la realización de los proyectos planificados.

1. 6 OBJETIVOS

1. 6.1 Objetivo general

Diseñar e implementar el Sistema de gestión de actividades de Jóvenes vía Web para el Ministerio Joven Adventista de la Misión Bolivia Occidental de la Iglesia Adventista del 7° día que gestione el manejo y seguimiento de las actividades de este Ministerio.

1. 6.2 Objetivos específicos

a) Diseñar e implementar el módulo de inscripción de los miembros, antiguos y nuevos.

- b) Analizar y diseñar una Base de Datos para la centralización de la información de los miembros, iglesias, eventos y actividades.
- c) Diseñar e implementar un portal Web donde se agrupe la información substancial de los proyectos y actividades del Ministerio joven.
- d) Diseñar e implementar el modulo de reportes de miembros registrados.
- e) Diseñar e implementar un módulo de administración de usuarios.

Ver anexo A, matriz del marco lógico

1.7 ALCANCE

El proyecto propuesto contemplara los siguientes alcances y limitaciones:

Se realizará el modulo de inscripciones de los miembros del Ministerio Joven, también se diseñará una base de datos de los miembros, el sistema estará dispuesto para su acceso a través del Internet, el sistema agrupará la información de las actividades, proyectos del Ministerio Joven Adventista y realizará reportes. Todo esto se realizará con el propósito de optimizar el proceso de seguimiento y control de actividades y los roles que se debe cumplir. Además que el proyecto no realizará la parte contable y tampoco la parte de inventarios.

1.8 METODOLOGÍA

Para el desarrollo del sistema se hará uso de la metodología Proceso Unificado Racional RUP, que permitirá generar un producto software de calidad acorde a los requerimientos de los usuarios, los modelos de esta aplicación serán elaborados con el lenguaje de modelado unificado (UML), tomando un enfoque ciclo de vida iterativo incremental, y la metodología del Método de Diseño Hipermedia Orientado a Objetos (OOHDM), que permitirá representar aspectos propios de aplicaciones Web como multimedia y navegación, plasmados en un lenguaje de programación como ser php5 y un gestor de base de datos My SQL, en una plataforma de Windows Server 2000.

1.9 IMPORTANCIA DEL ESTUDIO (APORTES)

El presente proyecto automatizará los procedimientos manuales de inscripción de miembros nuevos y antiguos, reserva de inscripción a las actividades planeadas y realizados por el personal de la institución. Facilitando el acceso a la información necesaria para los administrativos, los directivos, miembros del Ministerio de Jóvenes Adventistas, padres de familia o tutores y sociedad en general, de manera rápida y automatizada por medio de un sistema vía Web.

También se hará uso de FRAMEWORKS que permitirá ayudar en el desarrollo del Sistema de Gestión de actividades de jóvenes - Misión Boliviana Occidental (software) ya que tienen una estructura definida para crear aplicaciones con mayor rapidez. Estos frameworks manejan el patrón de MVC (Modelo- Vista- Controlador) y nos ofrecen la posibilidad de usar formularios, envíos de email, conexión a base de datos, etc.





CAPÍTULO II

MARCO TEÓRICO

CAPITULO N° II

MARCO TEÓRICO

En este capítulo se presenta una teoría básica de los conceptos, herramientas, técnicas y metodologías que se aplicarán en el desarrollo del presente proyecto. Se hará uso de la herramienta UML para la abstracción y especificación de procesos en el desarrollo del software y las metodologías Proceso Unificado Racional "RUP" y el Método de Diseño Hipermedia Orientado a Objetos "OOHDM", ya que estas dos metodologías actúan como modelos que pueden adaptar a cualquier tipo de proyecto y organización.

2.1 CONCEPTOS GENERALES

2.1.1 Gestión

La definición general que se le da a la gestión es la siguiente: Es la acción y al efecto de administrar, planear, organizar, dirigir y controlar el trabajo de los miembros de un negocio, una organización, e incluso un país. A través de una gestión se llevarán a cabo diversas acciones con los recursos disponibles de la organización, las cuales, conducirán al logro de objetivos y metas determinadas [J. Stoner, R Freeman, D Gilbert, 2005].

2.1.2 Gestión de información

Es un proceso que incluye operaciones como: extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma.

2.1.3 Indicadores de Gestión

Una entidad define sus indicadores de gestión para medir, controlar factores de éxito. Tomando las medidas correctivas necesarias en caso de que estos salgan de sus límites, en el menor tiempo posible.

2.1.4 Sistema de información

Un sistema de información es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común [James S, 2008].

Es un conjunto o disposición de elementos que están organizados para lograr un objetivo predefinido, procesando información [Pressman, 2002].

2.1.5 Red

Una red es un numero indeterminado de computadoras que se comunican entre si. Una red de computadoras, es una colección interconectada de computadoras autónomas que se usan para:

- Compartir recursos especialmente la información (los datos).
- Proveer la compatibilidad siendo mas que una fuente para los recursos.

2.1.6 Internet

Internet es una red de redes a escala de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP también se usa este nombre como sustantivo común y por lo tanto en minúsculas para designar a cualquier red de redes las mismas tecnologías que la Internet independientemente de su extensión o de que sea pública o privada, también es un conjunto de ordenadores o servidores conectados en una red de redes mundial que comparten un mismo protocolo de comunicación y que prestan servicios a los ordenadores que se conectan a esa red.

2.1.7 Servidor Apache

Apache es actualmente es el servidor mas utilizado en Internet.

Definiciones de apache en la Web:

- Es el servidor http mas ampliamente disponible en el Internet soporta los lenguajes PERL y PHP.

- Servidor de páginas Web hoy por hoy líder en el mercado de servidores, por delante de soluciones propietarias. Este servidor Web se distribuye libre, fue desarrollado en 1995 y ha llegado a ser el más utilizado en Internet.

2.1.8 Gestores de bases de datos

Los sistemas de gestión de base de datos son programas de software para la administración de las Bases de Datos; y en particular, para almacenar, manipular y recuperar datos en una computadora. El SGBD también se encarga de la comunicación entre el usuario y la base de datos proporcionándole al usuario los medios necesarios para poder obtener información, introducir nuevos datos y actualizar los datos ya existentes.

2.1.9 MySQL

MySQL es uno de los sistemas Gestores de base de datos relacional más populares desarrollado bajo la filosofía de código abierto. La desarrolla y mantiene la empresa MySQL AB, pero puede utilizarse gratuitamente.

Características de MySQL

- Facilidad de exportación e importación de datos, incluso de la base de datos completa.
- Posibilidad de ejecutar conjuntos de instrucciones guardadas en ficheros externos a la base de datos.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.

2.1.10 PHP (Hypertext Pre-processor).

PHP es un lenguaje de programación interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor Web,

tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Características de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Biblioteca nativa de funciones sumamente amplia e incluida, permitiendo las técnicas de Programación Orientada a Objetos.

2.2 MODELO VISTA CONTROLADOR (MVC)

El Modelo-Vista-Controlador es un patrón o una guía para el diseño de arquitecturas software para aplicaciones que ofrecen una fuerte interactividad con usuarios, que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos; el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. El patrón MVC se ve frecuentemente en aplicaciones Web, **Ver la figura 2.1**

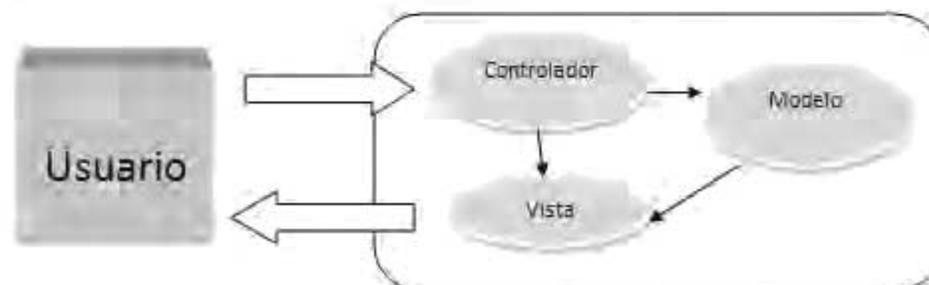


Figura 2.1 Modelo MVC
Fuente: Modelo-Vista-Controlador [Javier J. Gutiérrez, 2008]

2.2.1 Descripción del patrón

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera manejando el Sistema de Gestión de Base de Datos y la Lógica de negocio.
- **Vista:** Es la página HTML y el código que provee de datos dinámicos a la página, este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos de entrada, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista o en los modelos según sea necesario. Siendo capaz de manejar rutas, archivos, clases, métodos y funciones, **Ver la figura 2.2**

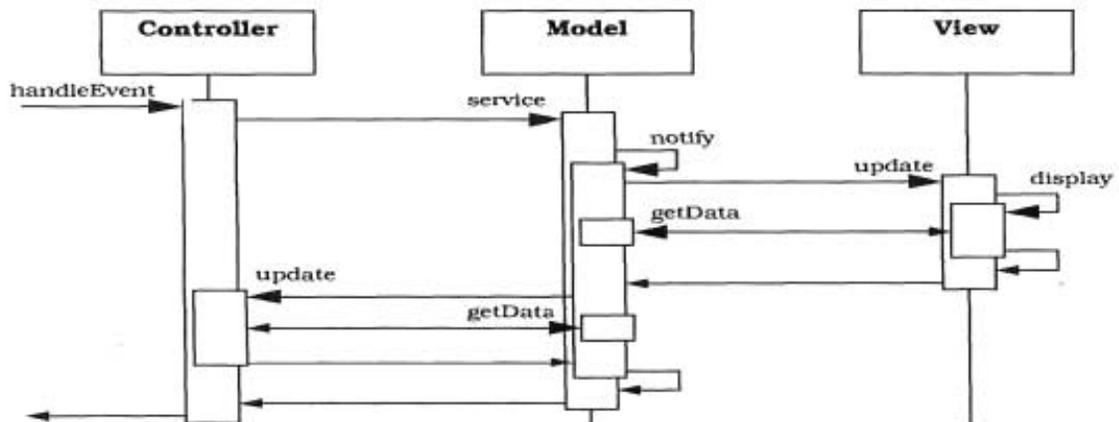


Figura 2.2 Patrón del Modelo vista controlador
Fuente. Modelo-Vista-Controlador [Javier J. Gutiérrez, 2006]

2.2.2 Framework

Un framework, es una estructura software de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, además de que define una filosofía de trabajo, proporciona librerías y funciones que deberían hacer la vida del programador más feliz, ahorra trabajo y tiempo, produce aplicaciones más fáciles de mantener, evita código duplicado y crea aplicaciones Multi-Capa.

Los Frameworks ofrecen una infraestructura que permite a los desarrolladores tener un código más ordenado, limpio, fácil de actualizar, código más seguro, robusto y mucho más eficiente.

Características de los frameworks

- **Rehúso de diseño y de código:** Al adoptar un framework estamos rehusando tanto el artefacto físico (código ejecutable) como el diseño que éste trae incorporado
- **Abstracción de URLs y sesiones:** No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
- **Acceso a datos:** Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos.
- **Separación entre diseño y contenido mediante controladores.**
- **Autenticación y control de acceso:** Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.

Ventajas en el uso de Frameworks

- **Modularidad y reducción de la complejidad.** La aplicación esta formada por subsistemas especializados en distintos aspectos fundamentales de toda aplicación (persistencia, presentación, manejo de log)
- **Fortaleza al cambio.** Los módulos pueden ser evolucionados o cambiados conservando la arquitectura global de la aplicación.
- **Documentación.** La documentación del framework promueve el uso correcto del mismo y disminuye el esfuerzo necesario para el mantenimiento.
- **Estructura.** El desarrollo basado en frameworks establece una estructura sobre la cual las aplicaciones pueden ser construidas, liberando al desarrollador de tomar el 100% de las decisiones de diseño.
- **Distribución de funciones.** Permite paralelizar el trabajo de desarrollo ya que la solución puede desarrollarse como un conjunto de piezas independientes que encajarán en el framework usado.
- **Eficiencia.** El desarrollador puede concentrarse en los requerimientos funcionales de la aplicación.
- **Aplicaciones ricas.** Posibilidad de dar más funcionalidad a los usuarios de la aplicación.

2.2.3 Tipos de frameworks

En la actualidad existen distintos tipos de Frameworks para el desarrollo Web en distintos lenguajes. Pero los Frameworks para PHP basados en el Modelo Vista Controlador (MVC) se encuentran en un gran momento. A continuación listamos algunos de ellos.

- Symfony:
- Seagull:
- Prado:
- CakePHP (framework creado sobre los conceptos de Ruby on Rails)
- PHP on TRAX
- Zoop Framework:
- WACT:
- AKELOS PHP FRAMEWORK:
- Qcodo:
- PHPOpenBiz:
- Codelgniter
- PHPDevShell
- Akelos
- QPHP
- Zend Framework
- Seagull
- PhpMVC
- JQuery

A continuación las características de algunos de los Frameworks para PHP más usados.

- **Zend Frameworks:** El Zend Frameworks es simple, no necesita instalación especial, requiere PHP 5 e incorpora el patrón MVC.
- **Symfony:** Diseñado con el objetivo de optimizar la creación de las aplicaciones Web, con el uso de sus características. Posee una librería de clases que permiten reducir el tiempo de desarrollo.
- **Seagull:** El framework Seagull, nos permite realizar una programación modular, posee un Sistema de Gestión de Contenidos (CMS).
- **JQuery:** Es una biblioteca o framework de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas Web.
- **Prado:** Prado está basado en componentes eventos con el objetivo de acelerar el desarrollo de aplicaciones Web usando PHP 5.

- **CakePHP:** Es un Framework similar a CodeIgniter de desarrollo rápido. Es una estructura de librerías y clases para programar aplicaciones Web. Su base es el Framework de Ruby.
- **PHP on TRAX:** El PHP on TRAX es compatible con PHP5, incluye el patrón MVC, incluye validación y AJAX, múltiples bases de datos. Podemos descargarlo y descomprimirlo hacia un directorio en nuestro servidor web.
- **Zoop Framework:** Es un Framework PHP Orientado a Objeto basado en el modelo MVC, sus desarrolladores lo caracterizan por ser rápido, eficiente y fácil destinado para programadores.
- **Akelos Php framework:** Es una plataforma de desarrollo de aplicaciones web basado en el MVC (Modelo Vista Controlador), patrón de diseño.

2.3 ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS.

La parte fundamental que realiza el análisis y diseño orientado a objetos consiste en situar el dominio de un problema y su solución lógica dentro de la perspectiva de los objetos, (cosas, conceptos o entidades), durante el cual se debería identificar y describir los objetos dentro del dominio del problema, ver en la figura 2.3.

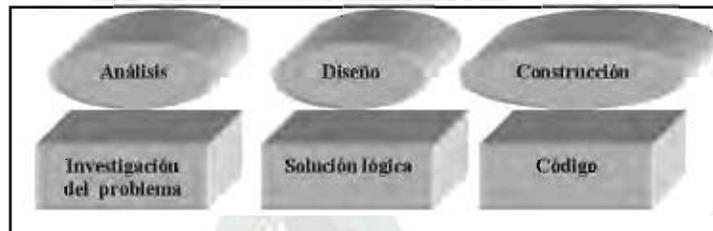


Figura 2.3 Actividades de desarrollo

Fuente: [Larman, 1999]

2.3.1 Análisis orientado a objetos

El análisis se centra en una investigación del problema, no en la manera de definir una solución.

El propósito del análisis orientado a objetos es definir todas las clases que son relevantes al problema que se va a resolver, las operaciones y atributos asociados,

las relaciones y comportamiento asociadas con ellas. Para cumplirlo se deben ejecutar las siguientes tareas:

- ✓ Los requisitos básicos del usuario, el cual debe comunicarse entre el cliente y el usuario.
- ✓ Identificar clases, donde se deben definir atributos y métodos
- ✓ Representar las relaciones objeto a objeto.
- ✓ Modelar el comportamiento del objeto
- ✓ Repetir iterativamente estas tareas hasta completar el modelo.

2.3.2 Diseño orientado a objetos

Durante el diseño orientado a objetos, se define los objetos lógicos del software que finalmente será implementado. El diseño es el primer paso en la fase de diseño orientado a objetos, transforma el modelo de análisis creado usando el análisis orientado a objetos en un modelo de diseño que sirve como un anteproyecto para la construcción del software.

2.4 METODOLOGÍA

2.4.1 Rational Unified Process (Proceso unificado de Rational – RUP)

El Rational Unified Process (Proceso unificado de Rational – RUP) está basado en una integración del trabajo de tres metodologistas, Ivar Jacobson, Grady Booch and James Rumbaugh. Estos metodologistas, fueron reunidos por Rational para formar un marco de metodologías unificadas, cohesivas y comprensivas de desarrollo de sistemas de software.

Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

2.4.2 Características esenciales del RUP

El Proceso Unificado tiene las siguientes características distintivas:

- **Centrado en los modelos:** Los diagramas son el vehículo de comunicación más expresivo que las descripciones en el lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.
- **Dirigido por Casos de Uso.** Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba
- **Centrado en Arquitectura:** Los modelos son proyecciones del análisis y diseño constituye la arquitectura del producto a desarrollar.
- **Iterativo e Incremental.** El proceso reconoce que es práctico dividir grandes proyectos en proyectos más pequeños o mini-proyectos. Cada mini-proyecto comprende una iteración que resulta en un incremento. Una iteración puede abarcar la totalidad de los flujos del proceso. Las iteraciones son planificadas en base a los Casos de Uso.

2.4.3 Estructura Dinámica del proceso. Fases e iteraciones

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable. Ver figura 2.4

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los hitos para cada una de las fases son:

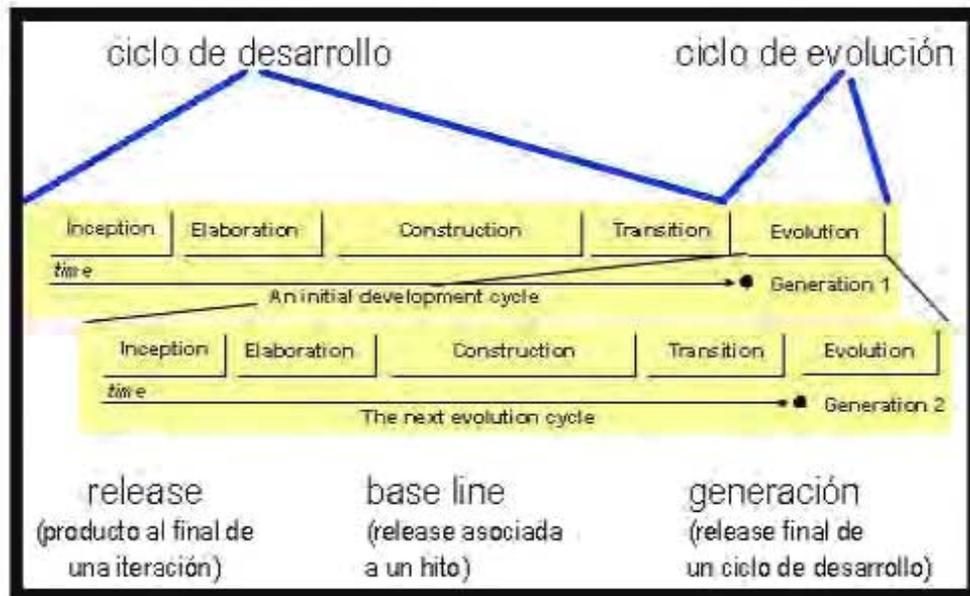


Figura 2.4: Ciclos, releases, baseline
Fuente: El lenguaje unificado de Modelado [Booch,1999]

Los hitos para cada una de las fases son: Inicio - Lifecycle Objectives, Elaboración - Lifecycle Architecture, Construcción - Initial Operational Capability, Transición - Product Release. Las fases y sus respectivos hitos se ilustran en la Figura 2.5

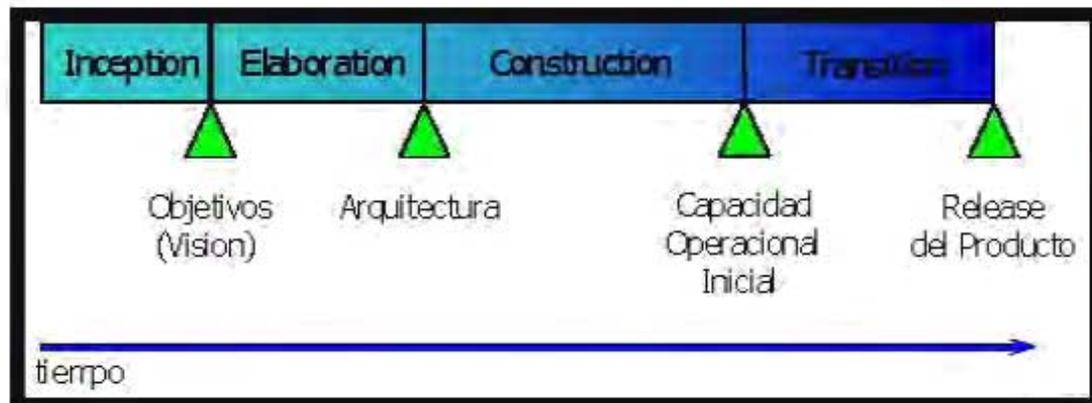


Figura 2.5: Fases e hitos en RUP
Fuente: [Web RUP introducción ,2007]

2.4.4 Flujos de trabajo

Una disciplina o flujo de trabajo es una colección de actividades relacionadas con un área de atención dentro de todo el proyecto con una perspectiva clásica de cascada. El RUP consta de nueve flujos de trabajo que se organizan como: flujo de

trabajo de procesos (modelo de negocio, requisitos, análisis y diseño, implementación, prueba, instalación o despliegue) los cuales detallaremos mas adelante y flujo de trabajo de gestión o soporte (administración de configuración y cambios, administración de proyecto, ambiente o entorno).

Modelado del negocio: Describe la estructura dinámica de la organización, los propósitos son:

- Entender los problemas de la organización desea solucionar e identificar mejoras potenciales.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento compartido del problema.
- Derivar los requisitos del sistema necesarios para apoyar a la organización.

Requisitos: Describe el método basado en casos de uso para extraer los requisitos, tiene el propósito de:

- Establecer y mantener un acuerdo entre clientes y otros interesados sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir los límites del sistema
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.

Análisis y Diseño: Describe las diferentes vistas arquitectónicas, el propósito del análisis y diseño es:

- Transformar los requisitos al diseño del futuro sistema.
- Desarrollar una arquitectura robusta del el sistema.
- Adaptar el diseño para hacerlo corresponder con el ambiente de implementación, y ajustarla para un desempeño esperado.

En esta etapa nos apoyaremos del modelo **OOHDM** con sus actividades de:

- Diseño conceptual donde elaboraremos el modelo de clases.
- Diseño navegacional, donde propone dos modelos diseño de las clases navegacionales y los contextos navegacionales.

- Diseño de interfaz abstracta, donde usamos un lenguaje gráfico del modelado abstracto de interfaces.

Implementación: Tiene en cuenta el desarrollo del software, la prueba de unidades y la integración, el propósito es:

- Implementar el diseño de elementos en términos de los elementos (archivos fuente, binarios, ejecutables y otros)
- Probar los componentes desarrollados en las unidades

Pruebas: Esta disciplina actúa como proveedor de servicios a las otras disciplinas en muchos aspectos. Las pruebas se enfocan principalmente en la evaluación y aseguramiento de la calidad del producto, desarrollado a través de las siguientes prácticas:

- Encontrar y documentar defectos en la calidad del software.
- Recomendar sobre la calidad percibida en el software
- Validar y probar las suposiciones hechas durante el diseño y la especificación de requerimientos de forma concreta.
- Validar si los requerimientos son implementados apropiadamente.

Despliegue: Cubre las configuraciones del sistema entregable.

- Describe las actividades asociadas con el aseguramiento de la entrega disponibilidad del producto de software hacia el usuario final.
- Existe un énfasis en probar el software en el sitio de desarrollo, realización de pruebas beta del sistema antes de su entrega final al cliente.

2.4.5 Ciclo de vida del RUP

El ciclo de vida del RUP organiza el desarrollo de las tareas en fases e iteraciones. El resultado de cada fase e iteración son productos tangibles del proceso.

Ver figura 2.6

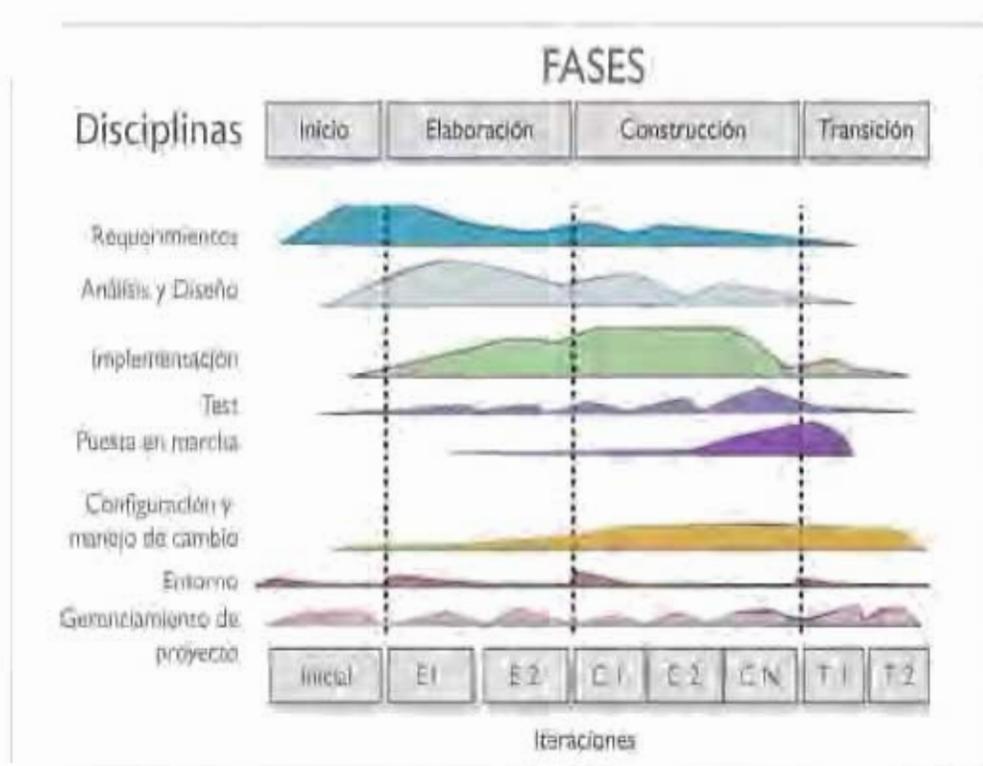


Figura 2.6: Estructura del RUP y esfuerzo en actividades según fase del proyecto
Fuente: El lenguaje unificado de Modelado [Booch, 1999]

2.4.6 Fases del RUP

El RUP incluye las cuatro etapas importantes que son: **Iniciación, elaboración, construcción y transición**, las cuales muestran que para producir una versión del producto en desarrollo se aplican todas las actividades de ingeniería pero con diferente énfasis; en las versiones preliminares, como además indica la intuición, hay más énfasis actividades de modelado del negocio, requisitos, análisis y diseño; conforme se producen versiones el énfasis pasa a las actividades de implementación, pruebas y despliegue.

2.4.6.1 FASE DE INICIO

En esta fase vamos a realizar un plan de desarrollo del proyecto; también se ve cual va a ser su alcance, antes de que el proyecto comience.

Una vez completada la recaudación de información se presenta un plan general del desarrollo del sistema, definiendo el modelo del negocio y el alcance del proyecto. Se

identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales. Se desarrolla, un plan de negocio para determinar que recursos deben ser asignados al proyecto.

2.4.6.2 FASE DE ELABORACIÓN

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

Como producto de esta fase tenemos un modelo de casos de uso con los principales actores.

2.4.6.3 FASE DE CONSTRUCCIÓN

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Por lo que el producto requiere varias iteraciones para su construcción, para cada iteración se selecciona algunos casos de usos, se proponen nuevos o se desechan, luego se procede a su implementación y pruebas. Se comienza la elaboración de material de apoyo del usuario.

2.4.6.4 FASE DE TRANSICIÓN

Esta fase toma los productos de las fases anteriores y los traduce en el sistema final, los factores de riesgo y los errores, son pequeños, pero eso no significa que ya no se realicen mas pruebas.

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

2.5 METODOLOGÍA OOHDM

OOHDM, Método de Diseño Hipermedia Orientado a Objetos (Object Oriented Hypermedia Design Methodology) es una metodología propuesta por Rossi y Schwabe para la elaboración de aplicaciones multimedia.

OOHDM define las pautas de trabajo, centrado principalmente en el diseño, para desarrollar ampliaciones multimedia de forma metodológica. OOHDM ha evolucionado bastante desde su nacimiento. Actualmente esta siendo utilizado por sus autores para el desarrollo de aplicaciones en la Web. OOHDM comprende cuatro actividades de diferentes denominadas:

- Diseño conceptual "Conceptual Design"
- Diseño de navegación "Navigational Design"
- Diseño de interfaz abstracta "Abstract Interface Design"
- Implementación "Implementation"

Durante cada actividad un conjunto de modelos orientados a objetos que describen aspectos de diseño en particular es construido o enriquecido por la iteración previa. Considerar los diseños conceptuales, de navegación y de interfaz como actividades separadas no solamente nos permite concentrarnos en diferentes aspectos, de uno en uno, sino principalmente obtener un "Framework" para razonar sobre el proceso de diseño, encapsulamiento experiencia de diseño especificada en cada actividad.

2.6 UML (LENGUAJE UNIFICADO DE MODELADO)

UML es un lenguaje gráfico que sirve para modelar, diseñar, estructurar, visualizar, especificar, construir y documentar el software. UML proporciona un vocabulario común para toda la cadena de producción, desde quien recaba los requisitos de los usuarios, hasta el último programador responsable del mantenimiento. Es un lenguaje estándar para crear los planos de un sistema de forma completa y no ambigua. Fue creado por el Object Management Group, OMG, un consorcio internacional sin ánimo de lucro, que asienta estándares en el área de computación distribuida orientada a objetos. [Schneider G., Winters J.P., 2001]

2.6.1 Tipos de diagramas

UML trabaja con de nueve diagramas para representar las distintas vistas de un sistema que son los siguientes:

- **Diagrama de Casos de Uso:** Modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- **Diagrama de Clases:** Muestra las clases (descripciones de objetos que comparten características comunes) que componen el sistema y como se relacionan entre sí.
- **Diagrama de Objetos:** Muestra una serie de objetos (instancias de las clases) y sus relaciones.
- **Diagrama de Secuencia:** Enfatiza la iteración entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.
- **Diagrama de Colaboración:** Igualmente, muestra la iteración entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- **Diagrama de estados:** Modela el comportamiento de acuerdo con los eventos.
- **Diagrama de Actividades:** Simplifica el diagrama de Estados modelando el comportamiento mediante flujos de actividades.
- **Diagrama de Componentes:** Muestra la organización y las dependencias entre un conjunto de componentes.
- **Diagrama de despliegue:** Muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo.

2.6.3 MODELO DEL NEGOCIO

Este modelo describe la funcionalidad de la institución o empresa para la cual se desarrolla el sistema, en términos de casos de uso y actores del negocio.

En este modelado se hace una descripción de los subsistemas en los que se ha dividido el sistema central. De acuerdo al análisis y diseño se debe tomar en cuenta tres diagramas principales: el modelo de casos de uso de negocio, el modelo de dominio y los modelos de objetos del negocio.

- **El modelo de casos de uso del negocio**, describe los procesos de negocio y los clientes. **Ver figura 2.7**

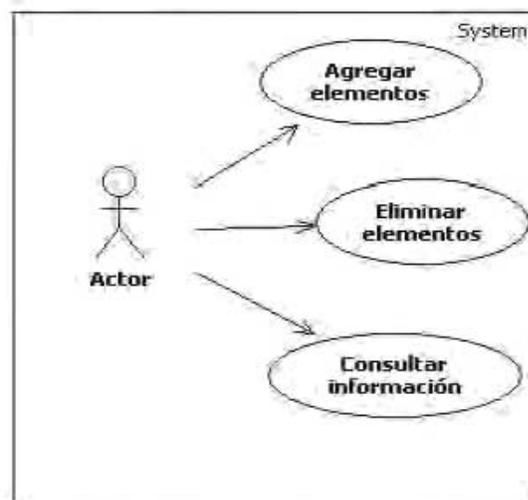


Figura 2.7: Casos del negocio

Fuente: Metodología Orientada a Objetos y dirigida por eventos [FA&LMN,2005]

2.6.3.1 Diagrama de casos de uso

Casos de Uso es una técnica para capturar información acerca de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje (ver figura 2.1). No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

Los casos de uso, según Ivar Jacobson, describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario. Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.

Este diagrama contiene descripciones de la funcionalidad del sistema independiente de la implementación, cubren la carencia existente en métodos previos (OMT, Booch) en cuanto a la determinación de requisitos.

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. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.). Los casos de uso pueden ser expresados en dos formatos: **Ver figura 2.8**

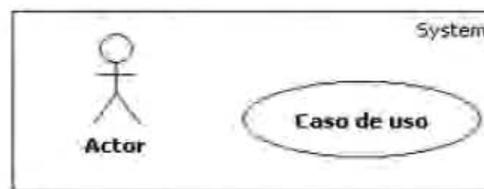


Figura 2.8 Representación de caso de uso y actor de UML
Fuente: Modelisation Object avec UML [Muller Pierre Alain,2000]

Casos de uso de alto nivel: Generalmente estos casos de uso son muy breves describiendo procesos en dos o tres acciones. Este caso de usos se puede representar de dos maneras gráfica (**Ver figura 2.9**) y de acuerdo a su estructura. **Ver tabla2.1**

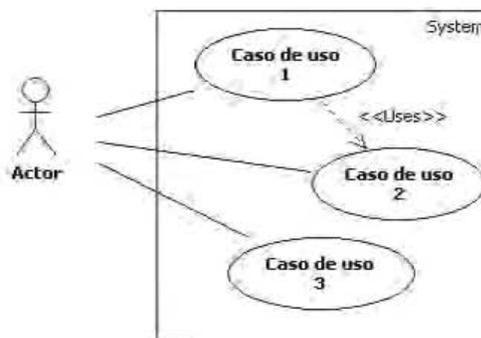


Figura 2.9 Representación de casos de uso de alto nivel
Fuente: Modelisation Object avec UML [Muller Pierre Alain,2000]

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
Tipos:	1.Primario, secundario u opcional (a explicar) 2.Esencial o real (a explicar)
Descripción:	Explicar brevemente el proceso que realizan los actores en el Caso de Uso.

Tabla2.1: Casos de uso de alto nivel
Fuente: Modelisation Object avec UML [Muller Pierre Alain, 2000]

Casos de uso expandidos

Casos de uso con formato expandido: Son descripciones más extensas que pueden contener cientos de oraciones con las cuales se realiza la descripción de un proceso completo. Este caso de uso se representa gráficamente como se muestra en la **figura 2.10**, y su estructura de descripción se describe en la **tabla 2.2**.

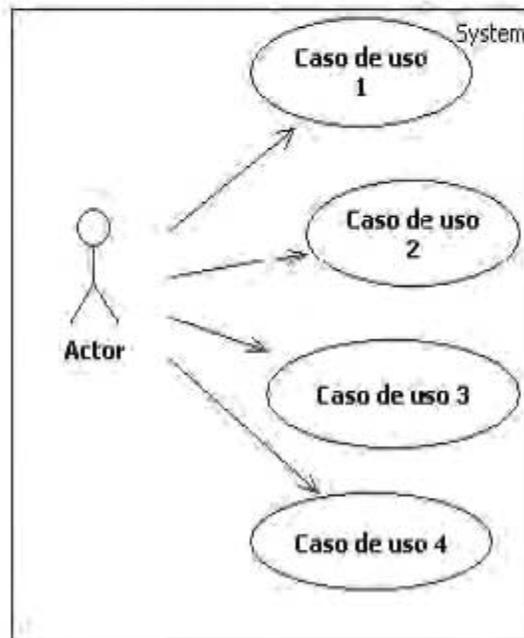


Figura 2.10 Caso de uso expandido
Fuente: Modelisation Object avec UML [Muller Pierre Alain, 2000]

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.
Tipo:	1 .Primario, secundario un opcional. 2. Esencial o real.
Referencia Cruzada	Casos relacionados de uso y funciones también relacionadas del sistema.
Curso normal de acciones:	
Acción de actor:	Respuesta del sistema:
CURSO NORMAL DE ACCIONES:	
ACCIÓN DEL ACTOR	ACCIÓN DEL SISTEMA
Acciones numeradas de los actores:	Descripciones numeradas de las respuestas del sistema.

Tabla2.2: Casos de uso expandido
Fuente: Modelisation Object avec UML [Muller Pierre Alain,2000]

2.6.3.2 Registro de términos en el glosario

El glosario es también un diccionario de datos que incluye todos los términos que requieren explicación para mejorar la comunicación y aminorar el riesgo de interpretación de términos. Estos términos dan como resultado una buena comprensión del problema, que ocurrirá en el desarrollo de las aplicaciones, siendo utilizado como documento referencia, ver tabla 2.3

TERMINO	CATEGORÍA	COMENTARIO
Nombre del término	Caso de uso/atributo/Tipo	Breve explicación del término

Tabla2.3: Elementos del glosario de términos
Fuente: Modelisation Object avec UML [Muller Pierre Alain,2000]

2.6.3.3 Diagrama de secuencia

Es el diagrama que muestra iteraciones ordenadas según la secuencia temporal de eventos. Ver Figura 2.11

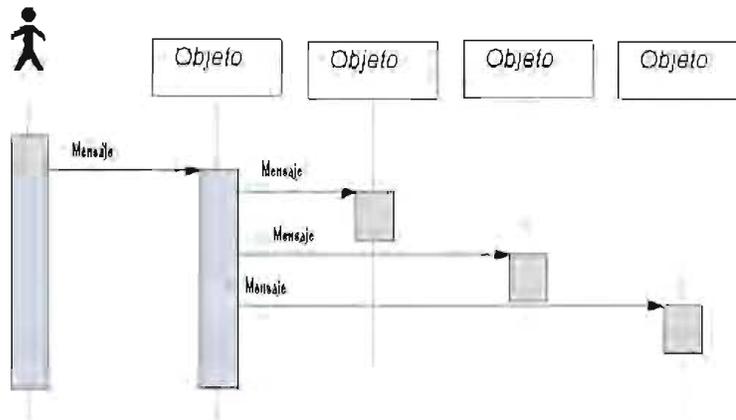


Figura 2.11 Diagrama de Secuencia.
Fuente: [LARMAN, 1999]

2.6.3.4 Contratos de operaciones

Los contratos contribuyen a definir el comportamiento de un sistema a partir de como cambia el estado de un sistema cuando se llama una operación suya, es decir que describen el efecto que sobre él tiene las operaciones. La preparación de los contratos de operación depende del desarrollo previo del modelo conceptual, de los diagramas de la secuencia del sistema.

A continuación presentamos en la tabla 2.5 la estructura de un contrato de operaciones:

Contrato	
Nombre	Nombre de la operación y parámetros.
Responsable	Descripción formal de la responsabilidad que debe cumplir la operación.
Tipo	Nombre del tipo: concepto, clase software, interfaz)
Referencias cruzadas	Número de referencia de las funciones del sistema.
Notas	Notas de diseño, información a fin.
Excepciones	Casos excepcionales
Salida	Salidas de mensajes, registros de salida.
Precondición	Suposición acerca del estado del sistema antes de ejecutarse
Poscondición	Estado del sistema después de ejecutarse la operación.

Tabla 2.4 Estructura, de un Contrato

Fuente: [Larman, 1999]

2.6.3.5 Diagramas de estados

Los diagramas de estados muestran los posibles estados en que puede encontrarse un objeto y las transiciones que pueden causar un cambio de estado. El estado de un objeto depende de la actividad que esté llevando a cabo o de alguna condición, ver figura 2.12 [Grady Booch, James Rumbaugh, Ivar Jacobson, 1996]

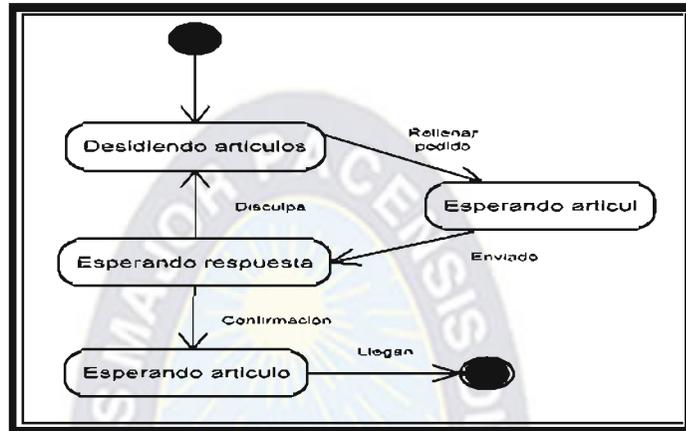


Figura 2.12 Transición de estados
Fuente: [Jacobson I., Booch G., 3ª edición]

Un diagrama de estado presenta el ciclo de vida de un objeto, los eventos que el ocurren, sus transiciones y los estados que median entre esos eventos. Estos diagramas pueden aplicarse a varios elementos.

2.6.3.6 Descripción de Casos de Uso Real

Un caso de uso real describe el diseño concreto del mismo a partir de una tecnología particular de entrada y salida, así como de su implementación global. Ver tabla 2.5

Caso de Uso:	Nombre del caso de uso
Actores:	Lista de Actores
Propósito:	Intención del caso de uso
Resumen:	Repetición del caso de uso de alto nivel a alguna síntesis similar
Tipo:	1.- Primario, secundario u opcional. 2.- Esencial o real.

Tabla 2.5 Caso de Uso Real
Fuente: [LARMAN, 1999]

2.6.3.7 Diagrama de objetos

Pertenece a la clasificación de los diagramas que dan una vista estática del sistema. Contiene un conjunto de instancias de elementos encontrados en un diagrama de clases.

2.6.3.8 Diagrama de colaboración

Son artefactos que explican gráficamente como los objetos interactúan entre sí a través de mensajes para realizar las tareas del sistema desarrollado **ver figura 2.13**

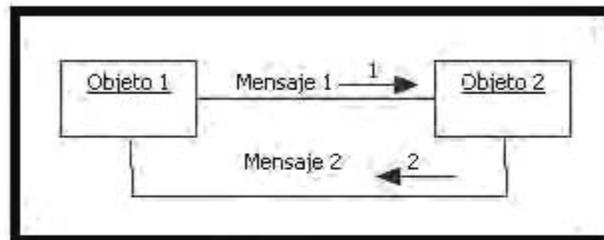


Figura 2.13 Diagrama de colaboración
Fuente: Modelización Object avec UML [Muller Pierre Alain, 2000]

2.6.3.9 Diagrama de Clases

Es la representación de los aspectos estáticos del sistema, utilizando diversos mecanismos de abstracción (clasificación, generalización y agregación). El diagrama de clases recoge las clases de objetos y sus asociaciones. En este Diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestran información temporal.

Se advierten los siguientes tipos de relación:

1. **Asociación.** Estas representan un conjunto de enlaces o instancias de clases.
2. **Herencia.** Indica que una subclase hereda los métodos y atributos especificados por una superclase hereda los métodos por ende la subclase, además de poseer sus propios métodos y atributos poseerá las características y atributos visibles de la superclase.
3. **Agregación.** Es un puntito de relación jerárquica entre un objeto que representa la totalidad de ese objeto y las partes que la componen. Permite el agrupamiento físico de estructuras relacionadas lógicamente.

2.6.3.10 Diagrama de componentes

Muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Los elementos de modelado dentro serán componentes y paquetes, ya sean las instancias especificadas de cada tipo se encuentran en el diagrama de despliegue [Muller, 2000].

Cada componente debe ser documentado con el diagrama de componentes más detallado; un diagrama de clases, o un diagrama de casos de uso. **Ver figura 2.14**

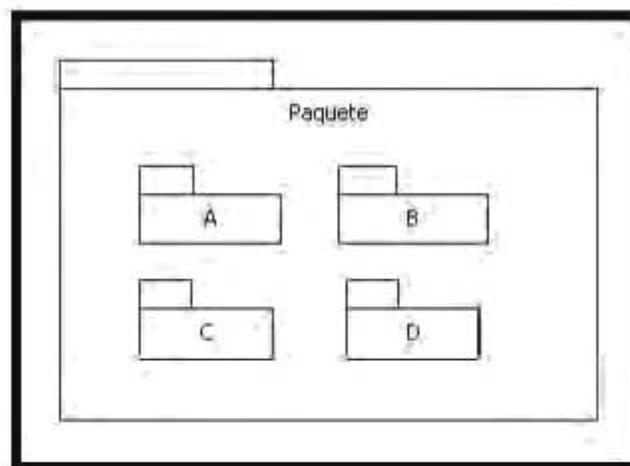


Figura 2.14 Representación de un paquete
Fuente: Modelización Object avec UML [Muller Pierre Alain, 2000]

2.6.3.11 Diagrama de despliegue

En el diagrama de despliegue indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa en software y en el hardware que lo contiene. Cada hardware representa un nodo.

Un nodo representa como un cubo, un nodo es un elemento donde se ejecutan los componentes, representan el despliegue físico de estos componentes.

En el ejemplo se muestra el nodo de la Terminal de un usuario. **Ver figura 2.15**



Figura 2.15 Diagrama de despliegue
Fuente: Elaboración propia

2.6.3.12 Diagrama de implementación

Son aquellos que se usan para modelar sólo componentes que existen como entidades en tiempos de ejecución, no se usan para modelar solo de tiempo de compilación o de un tiempo de enlazado. Puedes también modelar componentes que migran de nodo a nodo u objetos que migran de componente a componente usando una relación de dependencia estereotipo. **Ver figura 2. 16** [Muller Pierre Alain, 2000]

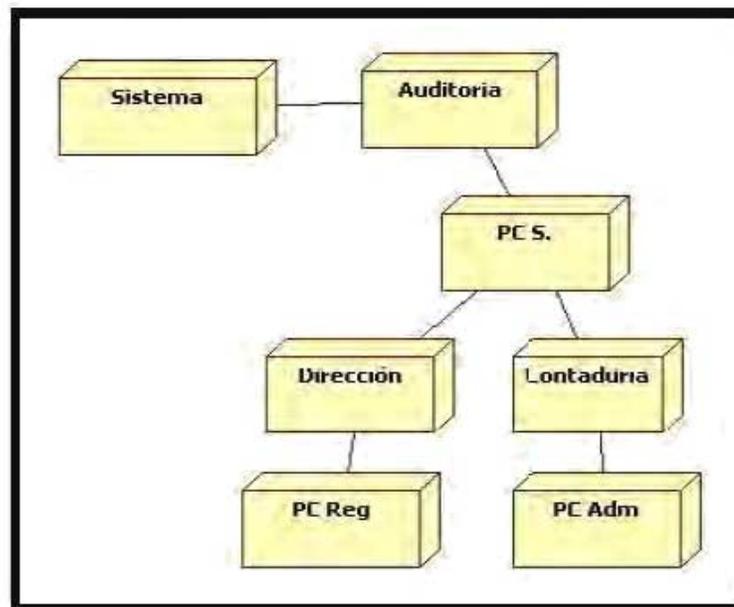


Figura 2.16 Modelado del diagrama de implementación
Fuente: Modelisation Object avec UML [Muller Pierre Alain, 2000]

2.7 MODELO DE TRES CAPAS

Como es la propia empresa la encargada de obligar a que se cumplan algunas reglas de negocio, es conveniente encontrar la manera de centralizar la gestión de estas reglas en un único lugar, de modo que todo el código necesario no se haya de duplicar en cada una de las aplicaciones. [Agulló, 1997]

El modelo de tres capas consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

- Cuando se requiera mucho procesamiento de datos en la aplicación.
- En aplicaciones donde la funcionalidad este en constante cambio.
- Cuando los procesos no están relativamente muy relacionados con los datos.
- Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.
- Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.
- Esta muy adecuada para utilizarla con la tecnología orientada a objetos. Ver figura 2.17

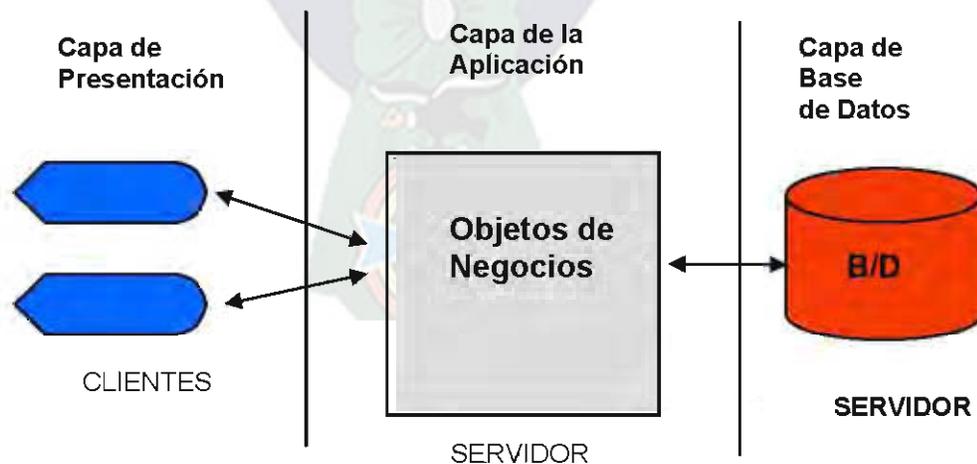


Figura 2.17 Modelo de tres capas
Fuente: Sistemas Distribuidos. [Coulouris, 2001]

2.8 ARQUITECTURA CLIENTE / SERVIDOR

La arquitectura cliente / servidor que describe como la forma en que se estructura la aplicación, en este sentido permite una división del trabajo en tres áreas especializadas de aplicación.

En el modelo cliente / servidor, se denomina cliente al proceso que inicia el diálogo o solicita los recursos de un determinado servidor, y servidor al proceso que responde a las solicitudes enviadas por el cliente. Para adoptar una arquitectura Cliente / servidor se poseen las características siguientes: La funcionalidad de la aplicación migra hacia todas las computadoras cliente. Y las funciones de Bases de Datos se le asignan al servidor.

2.9 POLÍTICAS Y NORMAS DE SEGURIDAD DE LA INFORMACIÓN

Uno de los aspectos más importantes es la seguridad de la información, en pocos años han cobrado un gran auge y mas aun ya el mundo globalizado con el Internet, en particular relación con la Web Las políticas de seguridad en informática emergen como el instrumento para concienciar a sus miembros acerca de la importancia y sensibilidad de la información y servicios críticos, de la superación de fallas y de las debilidades de tal forma que permitan a la institución mantener la integridad y confiabilidad de la información involucrada a sus procesos, en este sentido surgen normas estándares como la norma UNE-ISO/IEC 17799 que define diez dominios de control que cubren la gestión de seguridad de la información:

1. Política de seguridad.
2. Aspectos organizativos para la seguridad.
3. Control de acceso.
4. Seguridad física del entorno.
5. Seguridad relacionada con los usuarios.
6. Control de reservas de eventos.
7. Gestión de comunicación y operaciones.
8. Desarrollo y mantenimiento del sistema.
9. Gestión de continuidad del negocio.

10. conformidad.

2.10 METRICAS DE CALIDAD DEL SOFTWARE

Los desarrolladores de software estarán de acuerdo que el software de alta calidad es una de las metas más importantes. La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según el cliente que las pidan.

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

El estándar ISO 9126 (ver Anexo D), desarrollado en un intento de identificar los atributos clave de calidad: Ver tabla 2.6

Factor de Calidad	Subatributos del factor
Funcionabilidad	Idoneidad Corrección Interoperatividad Conformidad Seguridad
Confiabilidad	Madurez Tolerancia de fallos Facilidad de recuperación
Usabilidad	Facilidad de comprensión Facilidad de aprendizaje Operatividad
Eficiencia	Tiempo de uso Recursos utilizados
Facilidad de mantenimiento	Fácil de análisis Fácil de cambio

	Estabilidad Fácil de prueba
Portabilidad	Facilidad de instalación Fácil de ajuste Facilidad de adaptación al cambio

Tabla 2.6 Factores de calidad ISO 9126
Fuente: [PRESS - 03]

2.10.1.1 FUNCIONABILIDAD

El grado en que el software satisface las necesidades. Los puntos de función (PF) se describen como medidas básicas desde donde se calculan métricas de productividad. Los datos de PF se utilizan de dos formas durante la estimación del proyecto software:

- Como una variable de estimación que se utiliza para dimensionar cada elemento del software.
- Como métricas de línea base recopilada de proyectos anteriores y utilizados junto con variables de estimación para desarrollar proyecciones de costo y esfuerzo.

Para estimaciones de PF, la descomposición funciona de la siguiente manera: Ver tabla 2.7

Dominio de información	Descripción
Numero de entradas de usuario	Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
Número de salidas de usuario	Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a reportes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un reporte no se cuenta de forma separada
Número de archivos	Se cuenta cada archivo maestro lógico () esto es un grupo lógico de datos que pueden ser una parte de una gran base de datos o un archivo independiente
Número de interfaces externas	Se cuenta todas las interfaces legibles por la maquina

Tabla 2.7 Dominio de información de Puntos Función y su estimación
Fuente: [PRESS - 03]

Los puntos función se calculan complementando a la tabla 2.8

Parámetro de medición	Cuenta	Factor de ponderación			resultado
		Simple	Medio	Complejo	
Número de entradas de usuario	N1	3	4	6	N1* factor
Número de salidas de usuario	N2	4	5	7	N2* factor
Número de peticiones de usuario	N3	3	4	6	N3* factor
Número de archivos	N4	7	10	15	N4* factor
Número de interfaces externas	N5	5	7	10	N5* factor
CUENTA TOTAL					$\Sigma(Ni*factor)$

Tabla 2.8 Dominio de información de Puntos Función

Fuente: [PRESS - 03]

Para calcular los PF se utiliza la relación siguiente:

$$PF = CUENTA TOTAL * (0.65 + 0.01 * \Sigma(Fi))$$

CUENTA TOTAL = Sumatoria de todas las entradas de la tabla 2.3

Fi = son valores de ajuste a la complejidad según las respuestas a las preguntas.

1. ¿Requiere el sistema copias de seguridad?
- 2.- ¿Se requieren comunicaciones de datos?
- 3.- ¿Existen funciones procesamiento distribuido?
- 4.- ¿Es crítico el rendimiento?
- 5.- ¿Será ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?
- 6.- ¿Requiere el Sistema entrada interactiva?
- 7.- ¿requiere la entrada de datos interactivo que las transacciones de entrada se lleven a cabo sobre múltiples pantallas y variadas operaciones?

- 8.- ¿Se actualizan los archivos maestros de forma interactiva?
- 9.- ¿Son complejas las entradas, las salidas, los archivos y peticiones?
- 10.- ¿Es complejo el procesamiento interno?
- 11.- ¿Se ha diseñado el código para ser reutilizable?
- 12.- ¿Están incluidas en el diseño la conversión y la instalación?
- 13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones?
- 14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Cada una de las preguntas es respondidas usando una escala con rangos desde cero (no importante o aplicable) hasta 5 (absolutamente esencial)

2.10.1.2 CONFIABILIDAD

Cantidad de tiempo que el software esta disponible para su uso. La confiabilidad está definida como la probabilidad del sistema libre de fallos en un contexto determinado y durante un periodo de tiempo.

Este es el grado de confianza esperado por parte del usuario en la operación adecuada del sistema.

Fiabilidad: El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.

Seguridad: Representa la capacidad de que el sistema no afecte ni sea afectado por su entorno y de quien los utiliza.

Disponibilidad: Define la probabilidad de que el sistema este funcionando en un tiempo determinado.

Protección: Representa la capacidad del sistema para protegerse a si mismo de instrucciones accidentales o programadas.

La confiabilidad del sistema es directamente proporcional a la calidad de sus componentes y se calcula de la siguiente manera:

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

Donde:

$R(t)$: la función de confiabilidad en un tiempo t

λ : Error de tasa constante de fallo

t : tiempo de operación del sistema

2.10.1.3 MANTENIBILIDAD

Este aspecto de calidad involucra los elementos que simplifica la labor de prevención, corrección o ampliación del código del programa.

La mantenibilidad es la detección de corrección de fallas, también se basan en los cambios requeridos por el usuario, el tiempo del sistema es indefinido, pero está sujeta a las nuevas necesidades del usuario.

El IEEE Std.982 1-1998 sugiere un índice de madurez de software (IMS) que proporcione una indicación de la estabilidad de un producto de software (basados en los cambios que ocurren en cada versión del producto) Se determina la siguiente información:[Presuman, Roger S.;2006].

M_T = el número de módulos en la versión actual

F_c = el número de módulos cambiados en la versión actual

F_a = el número de módulos añadidos a la versión actual

F_d = El número de módulos en la versión anterior que se eliminaron en la actual

El índice de la madurez del software se calcula de la siguiente manera:

$$IMS = [M_T - F_c + F_a + F_d] / M_T$$

A medida que el IMS se acerca a 1.0 el producto empieza a estabilizarse.

2.10.1.4 USABILIDAD

La usabilidad esta definido como el esfuerzo necesario que el usuario debe realizar para aprender a utilizar el sistema, esto incluye preparación de datos de entrada y la interpretación de las salidas de un sistema. La evaluación de la usabilidad suele ser subjetiva ya que se le mide indirectamente, lo que hace que su valoración sea subjetiva.

Para evaluar la usabilidad del sistema, se puede tomar en cuenta el manual del usuario y las medidas cuantificables.

Las medidas típicas cuantificables de Niedsen sientan una buena base para la evaluación de la usabilidad.

- Tiempo en que los usuarios se tomar para completar tareas específicas.
- El número de tareas de diversos tipos que puedes ser completadas dentro de un tiempo limite dado.
- La relación entre las interacciones exitosas y los errores.
- El número de errores del usuario.
- El número de acciones erróneas inmediatamente posteriores.
- La frecuencia de uso de manuales del sistema resolviendo el problema del usuario.
- El número de ocasiones que el usuario se desvía de la tarea rea.

I

2.10.5 PORTABILIDAD

La portabilidad de un sistema de información se define como la facilidad de transferir un producto a diferentes entornos de hardware/software, sin necesidad de aplicar acciones o mecanismos distintos. También es considerado como la capacidad del producto software para ser usado en lugar de otro producto software para el mismo propósito, dentro del mismo entorno. Las características más importantes que se consideran para este factor son: la facilidad de instalación, facilidad de ajuste y facilidad de adaptación al cambio.

2.11 FIREWALL (Cortafuego)

Un firewall es un sistema que protege a un ordenador o a una red de ordenadores contra intrusiones provenientes de redes de terceros (generalmente desde internet). Un sistema de firewall filtra paquetes de datos que se intercambian a través de internet. Por lo tanto, se trata de una pasarela de filtrado que comprende al menos las siguientes interfaces de red:

- una interfaz para la red protegida (red interna)
- una interfaz para la red externa.

El sistema firewall es un sistema de software, a menudo sustentado por un hardware de red dedicada, que actúa como intermediario entre la red local (u ordenador local) y una o más redes externas. Un sistema de firewall puede instalarse en ordenadores que utilicen cualquier sistema siempre y cuando:

- La máquina tenga capacidad suficiente como para procesar el tráfico
- El sistema sea seguro
- No se ejecute ningún otro servicio más que el servicio de filtrado de paquetes en el servidor

En caso de que el sistema de firewall venga en una caja negra (llave en mano), se aplica el término "aparato".

Cómo funciona un sistema Firewall

Un sistema firewall contiene un conjunto de reglas predeterminadas que le permiten al sistema:

- Autorizar la conexión (*permitir*)
- Bloquear la conexión (*denegar*)
- Rechazar el pedido de conexión sin informar al que lo envió (*negar*)

Todas estas reglas implementan un método de filtrado que depende de la política de seguridad adoptada por la organización. Las políticas de seguridad se dividen generalmente en dos tipos que permiten:

- la autorización de sólo aquellas comunicaciones que se autorizaron explícitamente: "Todo lo que no se ha autorizado explícitamente está prohibido"
- el rechazo de intercambios que fueron prohibidos explícitamente: El primer método es sin duda el más seguro. Sin embargo, impone una definición precisa y restrictiva de las necesidades de comunicación.

2.12 ESTIMACIÓN DEL COSTE DEL PROYECTO

Existen una serie de métricas propuestas por la Ingeniería del Software para determinar el esfuerzo de un proyecto, el alcance del mismo y la productividad de sus programadores. Vamos a aplicar algunas de las mismas a este desarrollo.

2.12.1 Las métricas orientadas al tamaño

Las métricas orientadas al tamaño proporcionan medidas directas del software y del proceso por el cual se desarrolla. Se basan en la medición del número de Líneas De Código - LDC - que contiene el desarrollo, entendiendo por línea de código una sentencia del lenguaje de programación (se excluyen comentarios y líneas en blanco de los fuentes). Una forma de clasificarlos es atendiendo al número de líneas de código, como se muestra en la **tabla 2.9**

Categoría	Programadores	Duración	Líneas de código	Ejemplo
Trivial	1	0 - 4 semanas	<1k	Utilidad de ordenación
Pequeño	1	1 - 6 meses	1k-3k	Biblioteca de funciones
Media	2 - 5	0,5 - 2 años	3 k-50 k	Compilador de C
Grande	5 - 20	2 - 3 años	50k - 100k	SO pequeño
Muy grande	100 - 1000	4 - 5 años	100 k - 1 M	Grandes SO
Gigantes	1000 - 5000	5 - 10 años	>1M	Sistemas de distribución

Tabla 2.9 Categoría de un proyecto en función de sus líneas de código

2.12.2 El método Cocomo II

Una metodología que se encarga de medir proyectos software es COCOMO. La metodología COCOMO (COConstructive COSt MOdel) se debe a Barry Boehm, y está orientada a líneas de código.

Hay una jerarquía de modelos COCOMO: básico, intermedio y avanzado, la cual se aplica a tres tipos diferentes de software:

Orgánico: Proyectos relativamente sencillos, menores de 50.000 líneas de código. Se tiene experiencia en proyectos similares y se encuentra en un entorno estable.

Semiacoplado: Proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable, y las restricciones intermedias.

Empotrado: proyectos bastante complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica. Se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Dado que sólo se va a emplear una variable para la estimación (la línea de código), se empleará COCOMO básico, ya que es un modelo uní variable estático, con lo que se obtiene una valoración objetiva del esfuerzo realizado. Este proyecto será considerado como software orgánico, ya que posee menos de 50.000 líneas de código. La ecuación del esfuerzo de COCOMO básico tiene la siguiente forma:

$$E = \text{Esfuerzo} = (a) * (\text{KLDC})^b * (\text{persona x mes})$$

Donde KLDC es el número de líneas de código, distribuidas en millares, para el proyecto. La ecuación del tiempo de desarrollo es:

$$T = \text{Tiempo de duración del desarrollo} = (c) * (\text{Esfuerzo})^d * (\text{meses})$$

Por su parte los coeficientes a, b, c y d se obtienen empíricamente del estudio de una serie de proyectos, y sus valores son: Ver **tabla 2.10**

Proyecto software	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Tabla 2.10 Coeficientes COCOMO



CAPÍTULO III

MARCO APLICATIVO

CAPITULO N° III

MARCO APLICATIVO

RESUMEN

Este capítulo comprende poner en práctica lo mencionado en los anteriores capítulos, desarrollar el Sistema de Gestión de Actividades de Jóvenes - Misión Boliviana Occidental y para esto se hará uso de dos metodologías: Proceso Unificado Racional "RUP" y el Método de Diseño Hipermedia Orientado a Objetos "OOHDM" las cuales nos ayudaran a construir modelos para comprender mejor el sistema y los fundamentos teóricos para el desarrollo del proyecto.

3.1 FASE DE INICIO

3.1.1 Planificación Del proyecto

Para cumplir con los objetivos establecidos en el proyecto y desarrollar todas las fases que contemplan la metodología RUP y OOHDM, es necesario realizar una planificación temporal respecto al tiempo de trabajo que tomará la conclusión del Sistema de Gestión de Actividades de Jóvenes. La planificación estará en base al tiempo estimado por la Misión Boliviana Occidental.

La organización del desarrollo del sistema se base en fases e iteraciones. La siguiente tabla muestra la distribución de tiempos y el número de iteraciones de cada fase de la metodología RUP. Ver tabla 3.1

FASES	NÚMERO DE ITERACIONES	DURACIÓN	FECHAS
Fase de inicio	2	4 semanas	Del 10/08/09 al 4/09/09
Fase de elaboración	4	8 semanas	Del 24/08/09 al 16/10/09
Fase de construcción	5	13 semanas	Del 31/08/09 al 20/11/09
Fase de transición	3	6 semanas	Del 19/10/09 al 27/11/09

Tabla 3.1 Tiempos estimados en las fases de RUP
Fuente: Elaboración propia

Las actividades a realizar encada fase se detallan en la tabla 3.2.

FASE	ACTIVIDADES
FASE DE INICIO	<ul style="list-style-type: none"> • Elaborar la planificación del proyecto • Elaborar la visión del proyecto <ul style="list-style-type: none"> ✓ Estudio preliminar ✓ Requerimientos • Desarrollar el modelo del negocio <ul style="list-style-type: none"> ✓ Identificar actores ✓ Funciones del sistema
FASE DE ELABORACIÓN	<ul style="list-style-type: none"> • Realizar el modelo del negocio <ul style="list-style-type: none"> ✓ Diagrama de actividades • Realizar el Modelo de requisitos que incluye: <ul style="list-style-type: none"> ✓ Identificación de los casos de usos ✓ Descripción de los casos de usos ✓ Construcción del modelo Entidad relación ✓ Construcción del modelo conceptual ✓ Diagramas de estados • Realizar el modelo de Análisis y Diseño que incluye: <ul style="list-style-type: none"> ✓ Diagrama de secuencia ✓ Diagrama de clases ✓ Diagrama de colaboración ✓ Modelo de datos ✓ Arquitectura del sistema • Realizar el modelo de Implementación
FASE DE CONSTRUCCIÓN	<ul style="list-style-type: none"> • Diagrama de componentes • Diagrama de despliegue • Modelo de implementación del esquema de base de datos • Codificación del sistema • Interfaz de usuario
FASE DE TRANSICIÓN	<ul style="list-style-type: none"> • Desarrollar manuales de usuario • Instalación del sistema • Capacitación a los usuarios • Corregir fallar de implementación

Tabla 3.2 Plan de actividades de las cuatro fases de RUP
Fuente: Elaboración propia

En el desarrollo de la etapa de Análisis y Diseño nos apoyaremos de la metodología OOHDM con sus actividades. Ver tabla 3.3 de:

METODOLOGÍA	ACTIVIDADES
OOHDM	<ul style="list-style-type: none"> • Diseño conceptual <ul style="list-style-type: none"> ✓ Modelo de clases • Diseño de navegación • Diseño de interfaz abstracta • Implementación

Tabla 3.3 Plan de actividades de OOHM del proyecto
Fuente: Elaboración propia

Los hitos que marcan el final de cada fase se describen en la tabla 3.4.

FASE	DESCRIPCIÓN	HITO
FASE DE INICIO	En esta fase se hace un plan de desarrollo del proyecto que incluye: una planificación inicial respecto al tiempo, los requerimientos, se trabaja la primera versión del caso inicial de negocio.	La aceptación de todo este plan de desarrollo del proyecto por parte de los supervisores.
FASE DE ELABORACIÓN	Se desarrolla esencialmente el prototipo de la arquitectura, casos de usos completo al menos el 80%, un modelo de análisis y diseño al menos de un 10%.	La revisión y aceptación del prototipo de la arquitectura marcan el final de esta fase.
FASE DE CONSTRUCCIÓN	Se terminada de analizar y diseñar todos los casos de uso refinando el modelo de Análisis y Diseño. El producto se realiza en base a iteraciones, cada una da una versión actualizada del sistema.	Obtener la versión final del sistema con una capacidad de operación inicial.
FASE DE TRANSICIÓN	Preparación de la versión final del sistema asegurando su implementación de manera adecuada incluyendo la capacitación de los usuarios.	Entrega de toda la documentación del proyecto, manuales de usuario y la finalización de la capacitación a los usuarios y la instalación del producto final.

Tabla 3.4 Descripción de los hitos por fase
Fuente: Elaboración propia

La tabla 3.5 se muestra un resumen de las principales tareas del proyecto en sus distintas disciplinas variando en cada fase el trabajo a realizar.

DISCIPLINAS	ARTEFACTOS GENERADOS O MODIFICADOS
MODELADO DEL NEGOCIO	Modelo del negocio ✓ Diagrama de casos de uso del negocio ✓ Diagrama de actividades.
REQUISITOS	Requisitos funcionales Requisitos no funcionales Glosario Modelo conceptual
ANÁLISIS / DISEÑO	Modelo de análisis / diseño ✓ Diagrama de CU del sistema ✓ Descripción de los casos de usos ✓ Diagramas de secuencias ✓ Diagramas de estado ✓ Diagramas de clases Modelo de datos
IMPLEMENTACIÓN	Prototipos de interfaces
PRUEBAS	Casos de prueba funcionales
DESPLIEGUE	Modelo de despliegue ✓ Diagrama de componentes ✓ Diagrama de despliegue
GESTIÓN Y CONFIGURACIÓN DE CAMBIOS	Manual de usuario

Tabla 3.4 Artefactos de las distintas disciplinas del RUP

Fuente: Elaboración propia

3.1.2 Estudio preliminar:

Análisis de la institución: La Misión Boliviana Occidental, institución dependiente la Iglesia Adventista de la Unión Boliviana es la organización encargada de administrar toda la información de los ministerios de la Iglesia Adventista del Séptimo Día que comprenden de 30 distritos del Occidente Bolivia.

El propósito del presente proyecto es diseñar, desarrollar e implementar un sistema de información para el Ministerio Joven de la Iglesia Adventista organizada

por la Misión Boliviana Occidental, para la gestión, control y seguimiento de actividades y proyectos o eventos (camporis, retiros espirituales, encuentros, reuniones, vigias, ayunos, caminatas, campañas, concilios, conferencias, capacitaciones, cursos, convenciones y seminarios), el registro de los datos de kardex de los miembros, reserva de inscripciones a las actividades, reservación de material a través de un sitio Web, empleando información confiable y segura.

- **Capas de arquitectura y énfasis en el caso de estudio.** El Presente sistema vía Web con su entorno gráfico y su acceso a la base de datos se representa con un diseño de arquitectura de varias capas, como la siguiente representación, figura 3.1.

Presentación: Interfaz gráfica, ventanas.

- **Lógica de aplicación - Objetos denominados del problema:** Conceptos del problema. Ejemplo, objeto reservas de eventos, debe cumplir con los requisitos de aplicación.
- **Lógica de aplicación - Objetos de servicio:** Objetos denominados no relacionados con el problema que presentan servicios de reportes.
- **Almacenamiento:** Un mecanismo persistente de almacenamiento. *Ejemplo,* representación de una base de datos relacional.

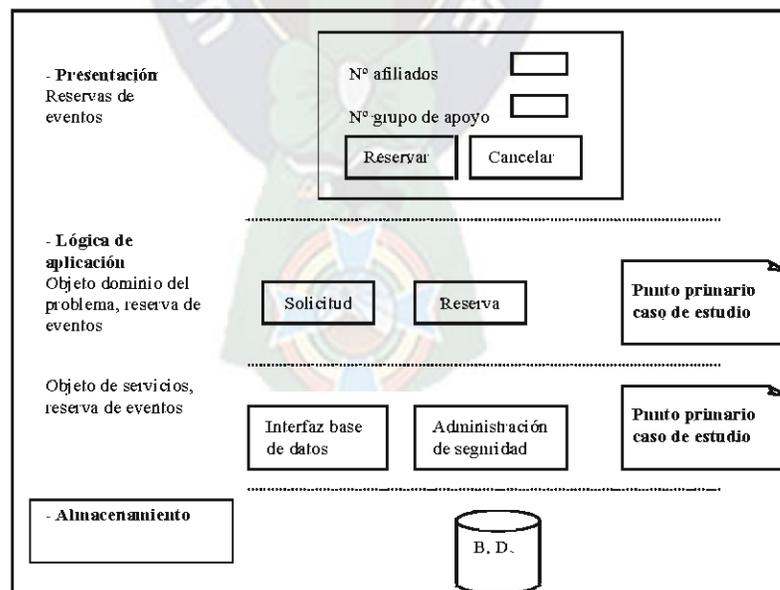


Figura 3.1 Representación en las capas, reservas de eventos.

Fuente: Elaboración propia

- **La estrategia de desarrollo iterativo.** El primer ciclo se destina a una simple aplicación de las funciones básicas, el segundo amplía las funciones del sistema.
- **Conociendo los requerimientos:** Objetivo crear artefactos de la fase de requerimientos como ser especificación de funciones, identificar y clasificar las funciones del sistema, identificar los atributos del sistema y relacionarlos. Un proyecto no puede ser exitoso sin una especificación correcta y exhaustiva de los de los requerimientos [Craig Larman, 1999].

3.1.3 Los requerimientos

Se recomienda los siguientes artefactos: *Panorama general, clientes, metas, funciones del sistema, atributos del sistema.* Lo cual se detalla a continuación.

➤ **Panorama general**

El objetivo de este proyecto es implementar el Sistema de Gestión de Actividades de Jóvenes - Misión Boliviana Occidental que logrará reducir costos y tiempo, utilizando conceptos y principios de ingeniería Web cuya información sea gestionada y administrada dinámicamente por una base de datos.

➤ **Clientes**

Los denominados clientes, en este caso la organización Misión Boliviana Occidental (M. B. O.) que atiende a gestión al Ministerio Joven de la Iglesia Adventista del 7^a día del occidente boliviano, institución sin fines de lucro, que ayuda a la sociedad, con actividades socioculturales.

Los actores involucrados en el sistema son:

Administrativos

Esta área consta del coordinador general, coordinadores distritales, secretaria requiere la siguiente información:

- ✓ Reportes del ministerio joven de las iglesias.
- ✓ Inscripción de eventos y actividades.
- ✓ Control de reportes del ministerio joven.
- ✓ Realizar actualizaciones de tarjetas de los miembros del ministerio joven.
- ✓ Reporte de seguros.

Director de Jóvenes Adventistas

Los directores de Jóvenes Adventistas necesitan actualizar los registros de sus afiliados al ministerio joven.

- ✓ Actualización de datos de los miembros del ministerio joven.
- ✓ Listado de miembros de Jóvenes adventistas de cada iglesia.
- ✓ Inscripción de nuevos miembros del ministerio joven.
- ✓ Datos personales de afiliados.
- ✓ Reserva de eventos y actividades como por ejemplo a campamentos.
- ✓ Listado de miembros que cuentan con el seguro respectiva a cada evento

Padre de familia o apoderado

Necesitan saber información inmediata de sus hijos como ser

- ✓ Información sus eventos a realizarse, en que fechas, tiempo de estancia fechas de retorno de encuentros y/o eventos.

Miembros del ministerio joven

Los miembros que necesitan saber:

- ✓ Saber las actividades del plan anual de la institución.
- ✓ Compartir experiencias de eventos.

➤ Metas

La automatización del sistema con los siguientes atributos.

- ✓ Reportes de ministerio Joven por iglesia.
- ✓ Reportes de inscritos a eventos.
- ✓ Inscripciones de miembros de los ministerios jóvenes de cada iglesia.
- ✓ Inscripciones reservas a eventos y actividades.
- ✓ Llenado de formularios de forma automatizada y no así manualmente.
- ✓ Emisión de comunicados vía Web.
- ✓ Pago de seguros.

3.1.4 MODELO DEL NEGOCIO

El modelo del negocio describe los procesos en términos de casos de uso y actores del contexto que se corresponden con los procesos del sistema y los usuarios. El modelado del negocio del proyecto esta enmarcado dentro de las delimitaciones del contexto del sistema y describe los procesos exactos relacionados con los actores y casos de uso encontrados. Ver figura 3.2.

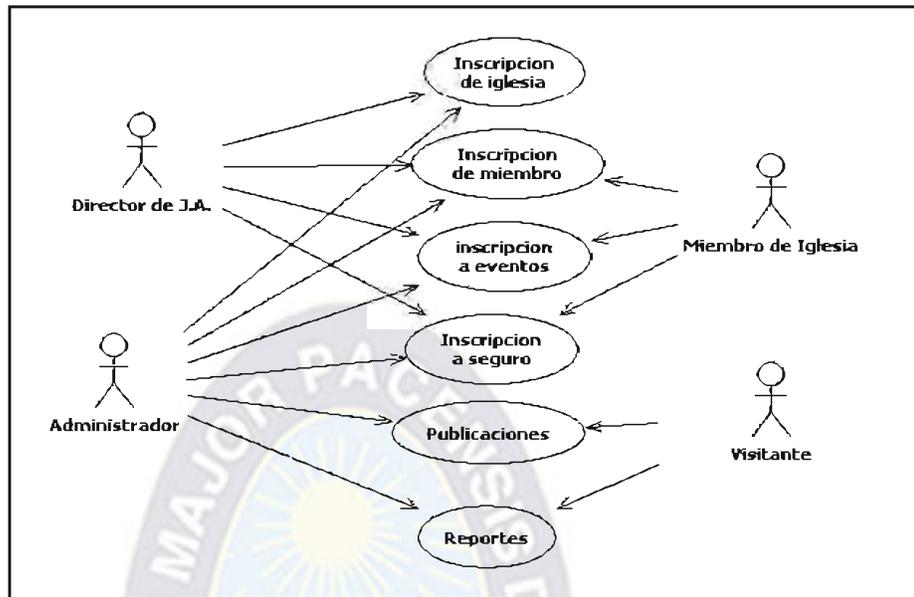


Figura 3.2 Modelo del negocio del Sistema
Fuente: Elaboración propia

3.1.4.1 Actores

A continuación detallaremos a los diferentes actores que interactuarán con el sistema en los diferentes procesos, especificando también las acciones realizadas por éstos.

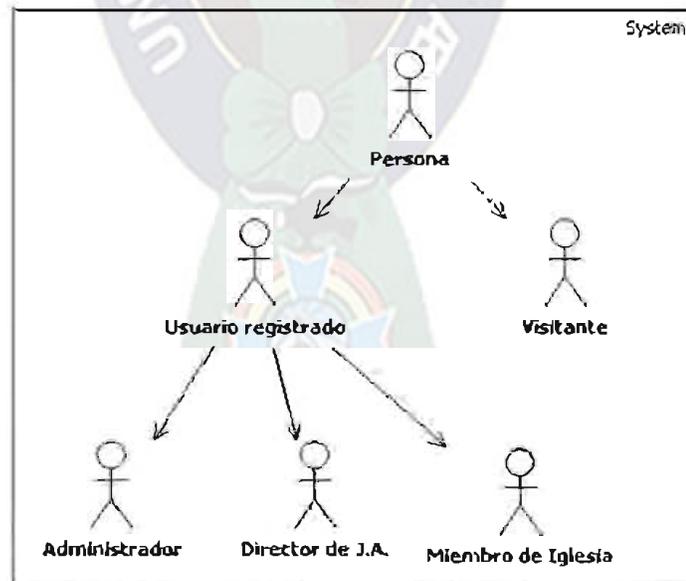


Figura 3.3 Especificación de actores
Fuente: Elaboración propia

Actor	Descripción
 Administrador	Es el encargado en administrar y gestionar el sistema con actualizaciones de los datos existentes dentro del Ministerio Joven de la iglesia Adventista del séptimo día.
 Director de J.A.	Es aquel actor que realiza reservas, inscripciones a eventos y actividades de sus miembros del club de jóvenes adventistas de cada iglesia al que pertenece, como también algunas actualizaciones de su club de J.A.
 Miembro de iglesia	Es el usuario que accede a las consultas, inscripciones a eventos y actividades a realizarse por el Ministerio Joven.

Tabla 3.5 Especificación de actores
Fuente: Elaboración Propia

3.1.5 Funciones del sistema

Las funciones o requerimientos del sistema son lo que este deberá realizar, dicho sistema se detallará a continuación. Para un mejor estudio de nuestro sistema lo dividiremos en distintas categorías.

Funciones básicas: En la tabla 3.6 observamos las funciones mínimas que debe realizar el sistema, para esto se facilitará con vistas de pantallas sencillas y de fácil manejo.

Ref. N°	Función	Categoría
R.1	Inscripción de miembros de iglesia al ministerio joven.	Oculto
R.2	Inscripción de iglesias al ministerio joven.	Oculto
R.3	Inscripción y/o reserva de eventos y/o actividades vía Web.	Oculto
R.4	Control de pago de seguros	Oculto
R.5	Seguimiento y reporte de seguros de los miembros del ministerio joven.	Evidente
R.6	Emisión de reportes para el seguimiento y control del ministerio joven	Evidente
R.7	Actualización de datos de los involucrados	Oculto
R.8	Mecanismos de seguimiento y control de almacenamiento persistente.	Evidente
R.9	Ofrece mecanismos de comunicación entre los procesos.	Evidente
R.10	Actualización de datos de Las iglesias	evidente
R.11	Permitir altas, bajas y cambios de toda la información registrada en el sistema.	Oculto
R.12	Ofrece registros de todos los involucrados al sistema.	evidente
R.13	Sistema almacena datos	oculto

Tabla 3.6 Funciones básicas

➤ **Atributos del sistema.**

Atributos del sistema que son requisitos que por defecto debe de tener ver tabla 3.7

ATRIBUTOS	DETALLES Y RESTRICCIONES DE FRONTERA
Facilidad de uso	Cuando ingresa al sistema cada usuario tendrá un menú distinto. Diferentes roles para la facilidad de su uso.
Tiempo de respuesta	La actualización de los datos requeridos: inscripción al Ministerio joven, mensajes, actividades, datos personales afiliados, apoderados y instructores. Las opciones aparecerán solo segundos.
Metáfora de interfaz	La interfaz será orientada a la metáfora de una forma accesible y comprensible al usuario con mensajes de acciones a cada evento del sistema según el modelo la navegación. Maximiza una navegación fácil mediante páginas dinámicas Php.
Tolerancia a fallas	Borrar en la duplicación de datos.
Plataforma	Microsoft Windows con navegador Internet Explorer o mozilla

Tabla 3.7 Atributos del sistema

3.2 FASE DE ELABORACIÓN

Le fase de elaboración establece la arquitectura base del Sistema de Gestión de Actividades de Jóvenes - Misión Boliviana Occidental y define las líneas que se seguirán para el desarrollo del proyecto.

3.2.1 Diagramas de actividades

Hacemos uso de los diagramas de actividades para describir las actividades de los procesos de negocio que se van a asistir o automatizar con el sistema.

- Diagrama de actividades "INSCRIPCIÓN DE NUEVO MIEMBRO"

En la figura 3.4 se muestra el diagrama de actividades para la inscripción de nuevo miembro.

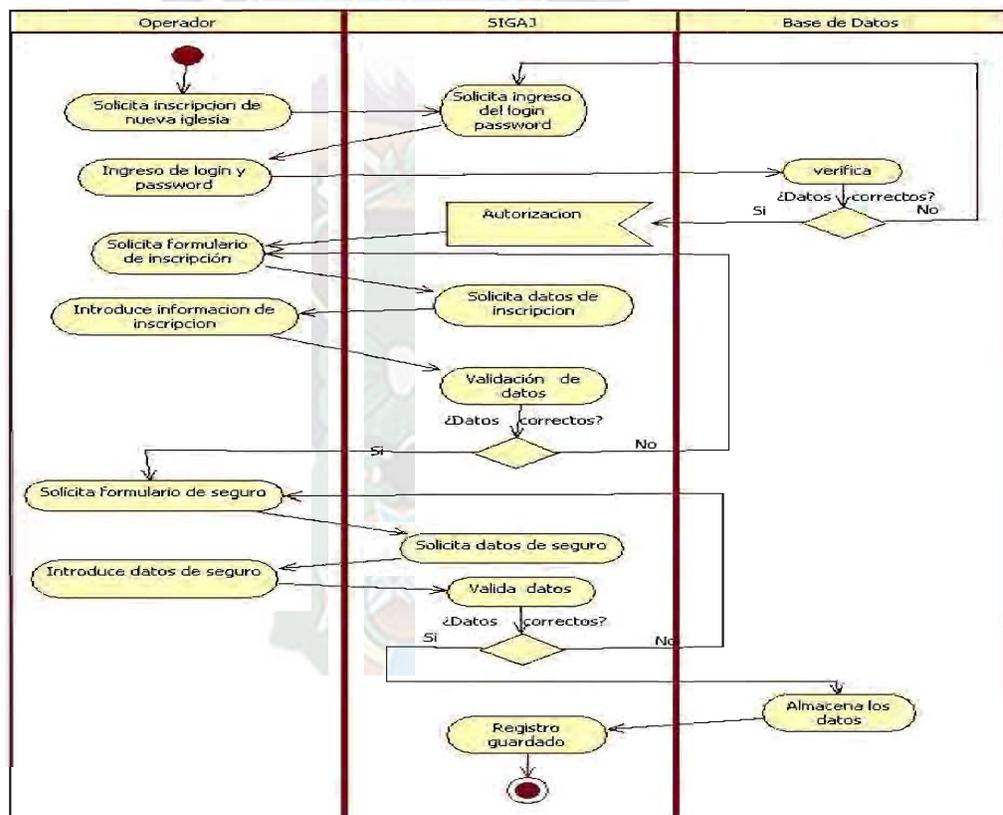


Figura 3.4 Diagrama de actividades: Inscripción de nuevos miembros
Fuente: Elaboración Propia

- Diagrama de actividades "INSCRIPCIÓN , RESEVA O CONFIRMACIÓN A EVENTO"

En la figura 3.5 se muestra el diagrama de actividades para inscripción, reserva confirmación a evento.

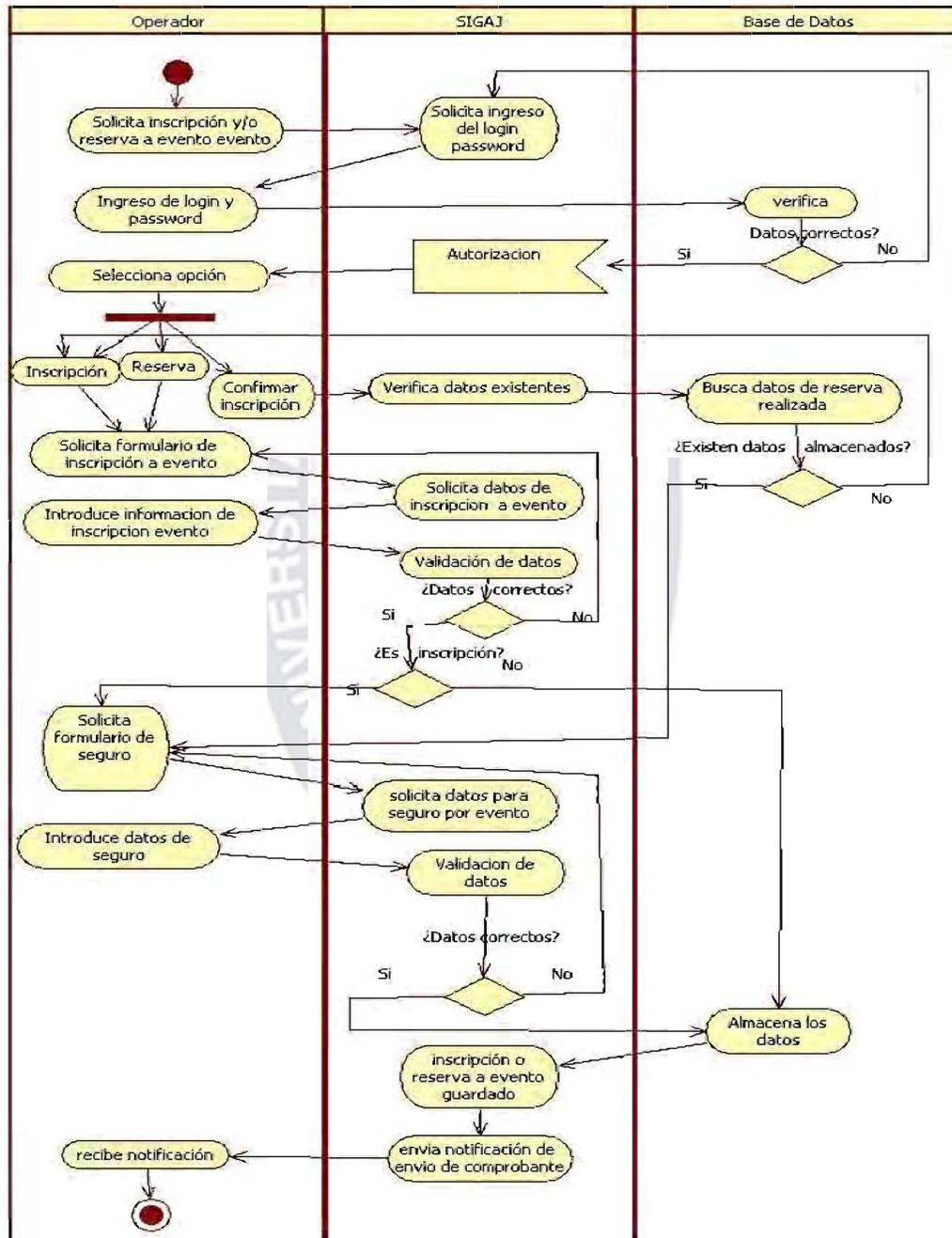
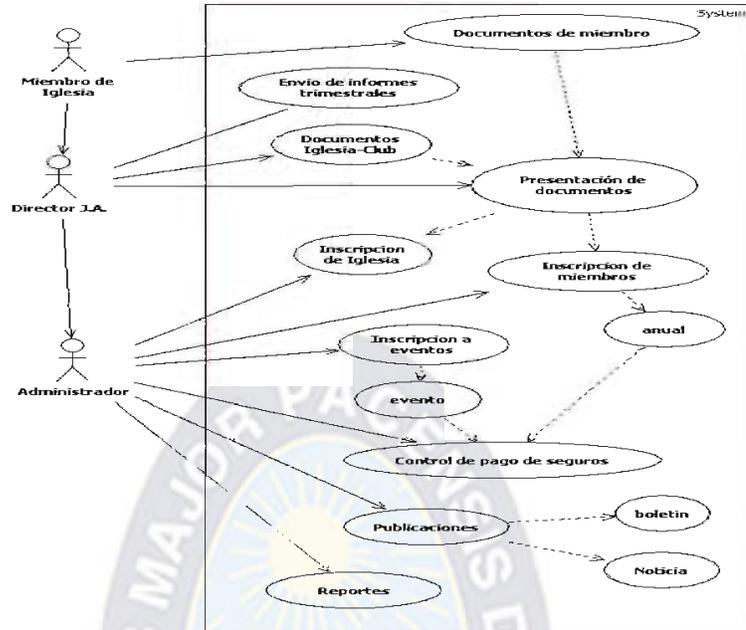
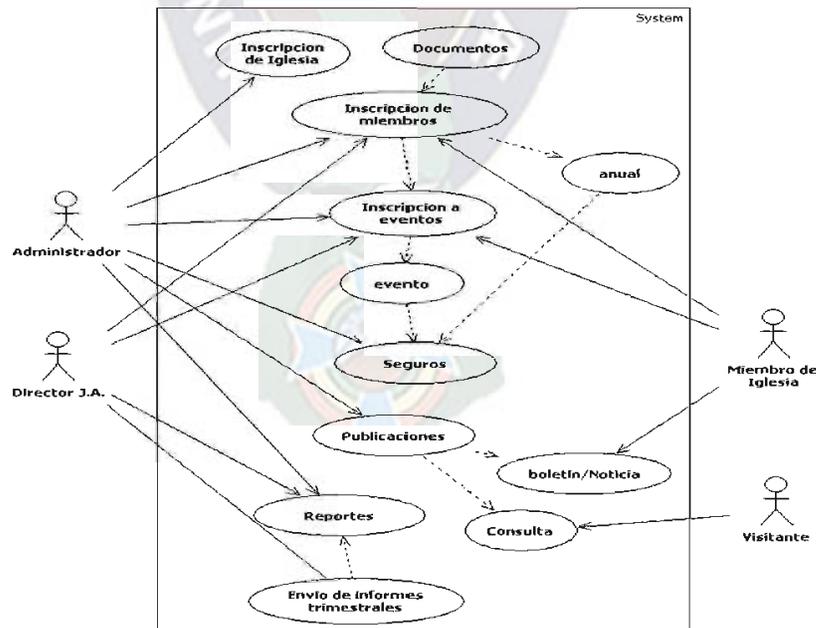


Figura 3.5 Diagrama de actividades: Inscripción, reserva o confirmación a evento.
Fuente: Elaboración Propia



En la figura 3.7 Muestra un diagrama de casos de uso real.
Fuente: Elaboración Propia

- **Diagramas de casos de uso general del sistema:** Los casos de uso, permiten identificar las clase de análisis cuyo objetos son necesarios para llevar a cabo el flujo de sucesos explican a detalle como comienzan como interactúan con sus actores involucrados. Ver figura 3.8



En la figura 3.8 Muestra un diagrama de casos de uso para el sistema.
Fuente: Elaboración Propia

3.2.2.2 Descripción de los diagramas de casos de uso expandidos

A continuación veremos los casos de uso más importantes. Los demás procesos y diagrama de casos de uso se detallan en el anexo A.

a) Caso de uso “INSCRIPCIÓN DE NUEVO MIEMBRO”

En la figura 3.9 se muestra el diagrama de caso de uso para la inscripción de nuevo miembro.

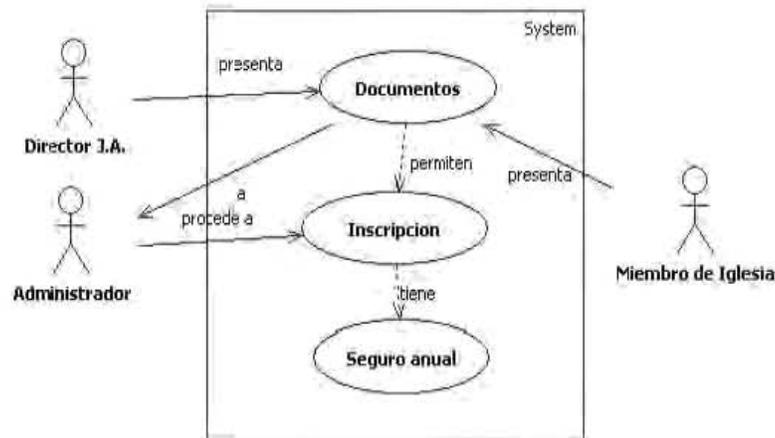


Figura 3.9 Diagrama de caso de uso: Inscripción de nuevos miembros
Fuente: Elaboración Propia

• Caso de uso de alto nivel: “INSCRIPCIÓN DE NUEVO MIEMBRO”

Caso de uso:	Inscripción de nuevo miembro
Actores:	Administrador, Director de J.A., Miembro de iglesia
Tipo:	Primario
Descripción:	El administrador o director de J.A. o miembro de iglesia inscriben a un nuevo miembro al ministerio joven, llenando el formulario necesario y presentando la documentación correspondiente.

- Diagrama de caso de uso expandido “INSCRIPCIÓN DE NUEVO MIEMBRO”

Caso de uso	Inscripción de nuevo miembro
Actores	Administrador, Director de J.A., Miembro de iglesia
Propósito	Registro e inscripción de nuevo miembro al ministerio joven.
Resumen	El usuario autorizado (administrador, Director de J.A. Y Miembro de iglesia) registra a un nuevo miembro al Ministerio Joven, llenando primeramente el formulario de inscripción de nuevo miembro, después de llenar el formulario se procede a la validación de los datos. Luego se llena el formulario de pago de seguro anual validándolo de igual manera, después se el usuario recibe la notificación de envió o presentación de deposito bancario hasta fecha una determinada. Con esto se podrá asignar funciones como: director, administrador, usuario miembro de iglesia. Se debe presentar documentación requerida y enviar el depósito bancario, caso contrario el sistema eliminara el registro dentro de un periodo de tiempo.
Tipo:	Primario
Referencias Cruzadas	R1, R4, R7, R8, R9, R11,R13

- Curso normal de INSCRIPCIÓN DE NUEVO MIEMBRO

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresa al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- Ingresa los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de registro de nuevo usuario..	6.- Despliega el formulario de registro.
	7.- Introduce los datos de: nombre completo, ci, edad, e-mail, fecha de nacimiento, nombre de iglesia.	8.- Valida los campos introducidos, para almacenar los datos de nuevo miembro del Ministerio Joven..
	9.- Selecciona la opción de pago de seguro anual.	10.- Despliega formulario de seguro anual.

	11.- Llena formulario de seguro anual.	12.- Valida campos introducidos, para luego almacenar los datos correspondientes a inscripción de nuevo miembro.
Curso alternativo de eventos	13.- El usuario introduce datos erróneos en el momento de ingreso al sistema.	14.- El sistema despliega nuevamente el formulario de login y password.
	15.- El usuario introduce datos erróneos del miembro.	16.- El sistema despliega un mensaje de error y manda al paso 6.
	17.- El usuario introduce datos erróneos de seguro anual.	18.-El sistema despliega un mensaje de error y envía al paso 10.

- **Contratos de operaciones: “INSCRIPCIÓN DE NUEVO MIEMBRO”**

Contrato: Inscripción Nuevos miembros	
Nombre:	Inscripción de Nuevos miembros
Responsabilidades	Administrador, Director de J.A., Miembro de iglesia Ingreso con contraseña tipo de usuario, el sistema valida tipo código y tipo de usuario, si son válidos el sistema permite realizar listado la inscripción según tipo de usuario.
Tipo:	Sistema
Referencias Cruzadas	R1, R4, R7, R8, R9, R11,R13
Notas	Realiza la búsqueda de registros en la Base de Datos según tipo de usuario y tipo de seguro.
Excepciones	Si la introducción de datos del nuevo usuario y seguro anual no es correcta el sistema devuelve un mensaje de datos erróneos.
Salida	Registro de nuevo miembro y seguro anual aceptado y almacenado.
Precondición	Debe presentar la documentación y comprobante de depósito bancario correspondiente para su registro hasta una fecha especificada.

Post-condición	Caso de no presentar la documentación requerida el sistema eliminara los registros.
-----------------------	---

b) Caso de uso “INSCRIPCIÓN O RESERVA O CONFIRMACIÓN DE INSCRIPCIÓN A EVENTO”

En la figura 3.10 se muestra el diagrama de casos de uso para el Incripción o reserva o confirmación de inscripción a evento.

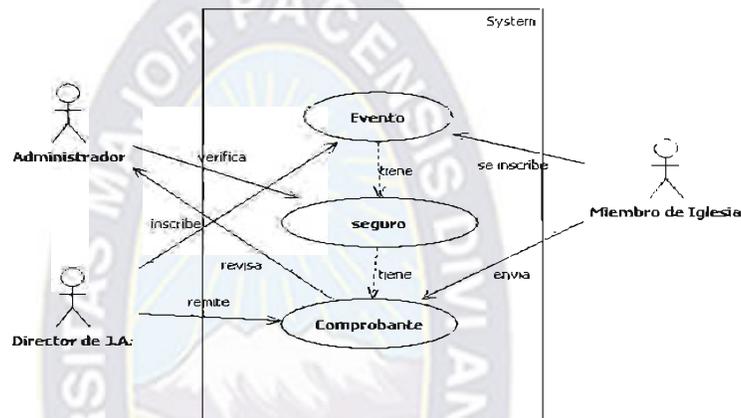


Figura 3.10 Diagrama de caso de uso: Inscripción a evento

Fuente: Elaboración Propia

- **Caso de uso de alto nivel: “INSCRIPCIÓN O RESERVA O CONFIRMACIÓN DE INSCRIPCIÓN A EVENTO”**

Caso de uso:	Inscripción o reserva o confirmación de inscripción a evento
Actores:	Administrador, Director de J.A., Miembro de iglesia
Tipo:	Primario
Descripción:	El usuario selecciona la opción que quiere realizar, esta opción puede ser el de inscribirse a un evento u actividad, reservar cupos al evento y confirmar la inscripción de reserva a evento realizada anteriormente, que pertenece al Ministerio Joven.

- **Diagrama de caso de uso expandido “INSCRIPCIÓN O RESERVA O CONFIRMACIÓN DE INSCRIPCIÓN A EVENTO”**

Nombre	<i>Inscripción a evento</i>
Actores	Administrador, Director de J.A., Miembro de iglesia
Propósito	Registro e inscripción del miembro usuario de cada una de las distintas

	iglesias pertenecientes al ministerio joven a uno de los diferentes eventos, el cual es organizado por la M.B.O. pagando debidamente su seguro por evento.
Resumen	El sistema verifica los datos del miembro, para dar permiso a la inscripción o reserva o confirmación de inscripción a un evento y/o actividad. El usuario miembro al realizar su inscripción debe asegurarse de haber realizado su depósito bancario ya que se debe presentar o enviar el comprobante de depósito bancario. Si en el caso de elegir la opción de reserva el sistema verificara los datos introducidos menos el de seguro por lo que envía notificación de confirmación de inscripción hasta una fecha determinada. Y por ultimo si se elige la opción de confirmación de inscripción el sistema verifica si existe historial de datos y si confirma pasa a la opción de pago de seguro y se llenan los datos de este formulario y se almacenen los datos. Por ultimo se envía otra notificación de envío de comprobante de deposito bancario hasta fecha determinada
Tipo:	Primario
Referencias Cruzadas	R3,R4,R5,R7, R8, R11, R12, R13

- **Curso normal de “INSCRIPCIÓN O RESERVA O CONFIRMACIÓN DE INSCRIPCIÓN A EVENTO”**

- 1) **Caso de inscripción**

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresa al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- Ingresa los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de evento y selecciona la opción inscripción a evento.	6.- Despliega el tipo de formulario de inscripción a evento.
	7.- Introduce los datos necesarios para el caso de inscripción a evento.	8.- Valida los campos introducidos, para almacenar los datos, que posteriormente son almacenados, luego va a la opción de seguro.
	9.- Ingresa la opción de pago de seguro por evento.	10.- Despliega formulario de seguro por evento.

	11.- Llena formulario de seguro por evento con los datos necesarios para el caso.	12.- Valida campos introducidos, para luego almacenar los datos correspondientes a pago de seguro por evento.
	14.- Recibe notificación de envío de comprobante bancario hasta fecha determinada, si el usuario quiere puede enviar su comprobante vía fax en ese instante.	13.- Envía notificación de presentación o envío de comprobante hasta fecha determinada.
Curso alternativo de eventos	14.- El usuario introduce datos erróneos en el momento de ingreso al sistema.	15.- El sistema despliega nuevamente el formulario de login y password.
	16.- El administrador introduce datos erróneos en el formulario de inscripción a evento.	17.- El sistema retorna al paso 6.
	18.- El administrador introduce datos erróneos en el formulario de seguro por evento.	19.- El sistema retorna al paso 10.

Caso de reserva

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresar al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- Ingresar los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de evento y selecciona la opción reserva a	6.- Despliega el formulario necesario de inscripción a evento.
	7.- Introduce los datos necesarios para el caso de reserva a evento.	8.- Valida los campos introducidos, para almacenar los datos, que posteriormente son almacenados.
	9.- Recibe notificación de confirmación de inscripción.	10.- Envía notificación de confirmación de inscripción hasta fecha determinada.
Curso alternativo de eventos	11.- El usuario introduce datos erróneos en el momento de ingreso al sistema.	12.- El sistema despliega nuevamente el formulario de login y password que esta en el paso 2.
	13.- El administrador introduce datos erróneos.	14.- El sistema retorna al paso 6.

2) Caso de confirmación de inscripción

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresa al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- Ingresa los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de evento confirmación de inscripción.	6.- Despliega el formulario de confirmación de inscripción.
	7.- Introduce los datos necesarios para el caso de confirmación de inscripción.	8.- Valida los campos introducidos, y busca historial de datos de reserva realizada y va a la opción de seguro.
	9.- Ingresa la opción de pago de seguro por evento.	10.- Despliega formulario de pago de seguro por evento.
	11.- Llena formulario de seguro por evento con los datos necesarios para el caso.	12.- Valida campos introducidos, para luego almacenar los datos correspondientes a pago de seguro por evento.
	14.- Recibe notificación de envío de comprobante bancario hasta fecha determinada, si el usuario quiere puede enviar su comprobante vía fax en ese instante.	13.- Envía notificación de presentación o envío de comprobante hasta fecha determinada.
Curso alternativo de eventos	15.- El usuario introduce datos erróneos en el momento de ingreso al sistema.	16.- El sistema despliega nuevamente el formulario de login y password.
	17.- El administrador introduce datos erróneos en el formulario de confirmación de inscripción.	18.- El sistema retorna al paso 6.
	19.- El administrador introduce datos erróneos en el formulario de seguro por evento.	20.- El sistema retorna al paso 10.

- **Contratos de operaciones: “INSCRIPCIÓN O RESERVA O CONFIRMACIÓN DE INSCRIPCIÓN A EVENTO”**

• Contrato: “Inscripción o reserva o confirmación de inscripción a evento”	
Nombre:	Inscripción o reserva o confirmación de inscripción a
Responsabilidades	Administrador, Director de J.A., Miembro de iglesia Ingreso con contraseña tipo de usuario, el sistema valida tipo código y tipo de usuario, si son válidos el sistema permite realizar la inscripción, reserva o confirmación de inscripción de acuerdo a la opción que eligió el usuario.
Tipo:	Sistema
Referencias Cruzadas	R3,R4,R5,R7, R8, R11, R12, R13
Notas	Realiza la búsqueda de registros en la Base de Datos según tipo de usuario.
Excepciones	Si la introducción de datos de usuario no es correcta el sistema retorna al formulario de partida. Introducción de datos erróneos en los formularios de inscripción, reserva y confirmación de inscripción, ya que estos son validados en el momento del llenado del formulario o registro. Por lo que el sistema retorna a los formularios ya mencionados.
Salida	Inscripción o reserva o confirmación de inscripción aceptada así como el envío de notificación de presentación de depósito bancario.
Precondición	El usuario miembro debe ser miembro del Ministerio de Jóvenes Adventistas, para poder acceder a la opción de <u>inscripción, reserva o confirmación de inscripción a evento.</u>
Post-condición	Miembro de iglesia deberá presentar comprobante de depósito bancario, caso de no presentar el comprobante de depósito bancario hasta fecha especificada el sistema eliminará la inscripción al evento.

c) Caso de uso “PAGO DE SEGURO”

En la figura 3.11 se muestra el diagrama de caso de uso para EL PAGO DE SEGURO.

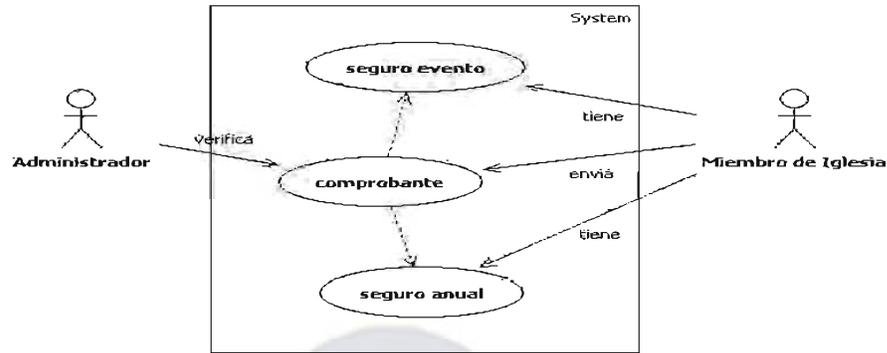


Figura 3.11 Diagrama de caso de uso: "Pago de seguro"
Fuente: Elaboración Propia

- **Caso de uso de alto nivel: " PAGO DE SEGURO"**

Caso de uso:	Pago de seguro
Actores:	Administrador, Director de J.A
Tipo:	Primario
Descripción:	El administrador realiza el pago del seguro ya sea este anual o por evento teniendo el depósito bancario de los inscritos como miembro al Ministerio de Jóvenes Adventistas o el depósito bancario por el costo de evento al cual esta inscrito el miembro del Ministerio.

- **Diagrama de caso de uso expandido " PAGO DE SEGURO"**

Nombre	<i>Seguro anual y por evento</i>
Actores	Administrador, Director de J.A.
Propósito	El sistema verifica y valida numero de comprobante de depósito bancario por cuestión de seguro.
Resumen	El sistema verifica los datos del deposito bancario con al boleta de deposito original del miembro enviado vía fax, detallando si es para el seguro anual o caso contrario para seguro por evento y/ actividad. Esta verificación de número de depósito bancario se la puede realizar hasta una fecha especificada.
Tipo	Primario
Referencias Cruzadas	R4, R5, R6, R7, R8, R10, R11, R12, R13

- **Curso normal de “PAGO DE SEGURO”**

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresa al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- ingresa los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de pago de seguro.	6.- Despliega el formulario de pago de seguro. Especificando si es para seguro por evento o seguro anual.
	7.- Introduce el número de depósito bancario, además de C.I. y elige el tipo de seguro.	8.- Valida los campos introducidos, y va y busca historial de miembro inscrito al Ministerio de Jóvenes Adventista o inscrito a evento para almacenar los datos del número de depósito bancario, que están siendo comprobados con la boleta de depósito bancario original. Y retorna a la pantalla principal.
Curso alternativo de eventos	9.- El usuario introduce datos erróneos del depósito bancario.	10.- El sistema despliega un mensaje de error y manda al paso 1.
	11.- El administrador introduce datos erróneos en el formulario de pago de seguro.	12.- El sistema retorna al paso 6.
	14.- El administrador recibe aviso para realizar nuevamente inscripción de miembro o a evento	13.- El sistema no encuentra historial por lo que envía aviso para realizar nuevamente inscripción de miembro y luego a la opción evento.

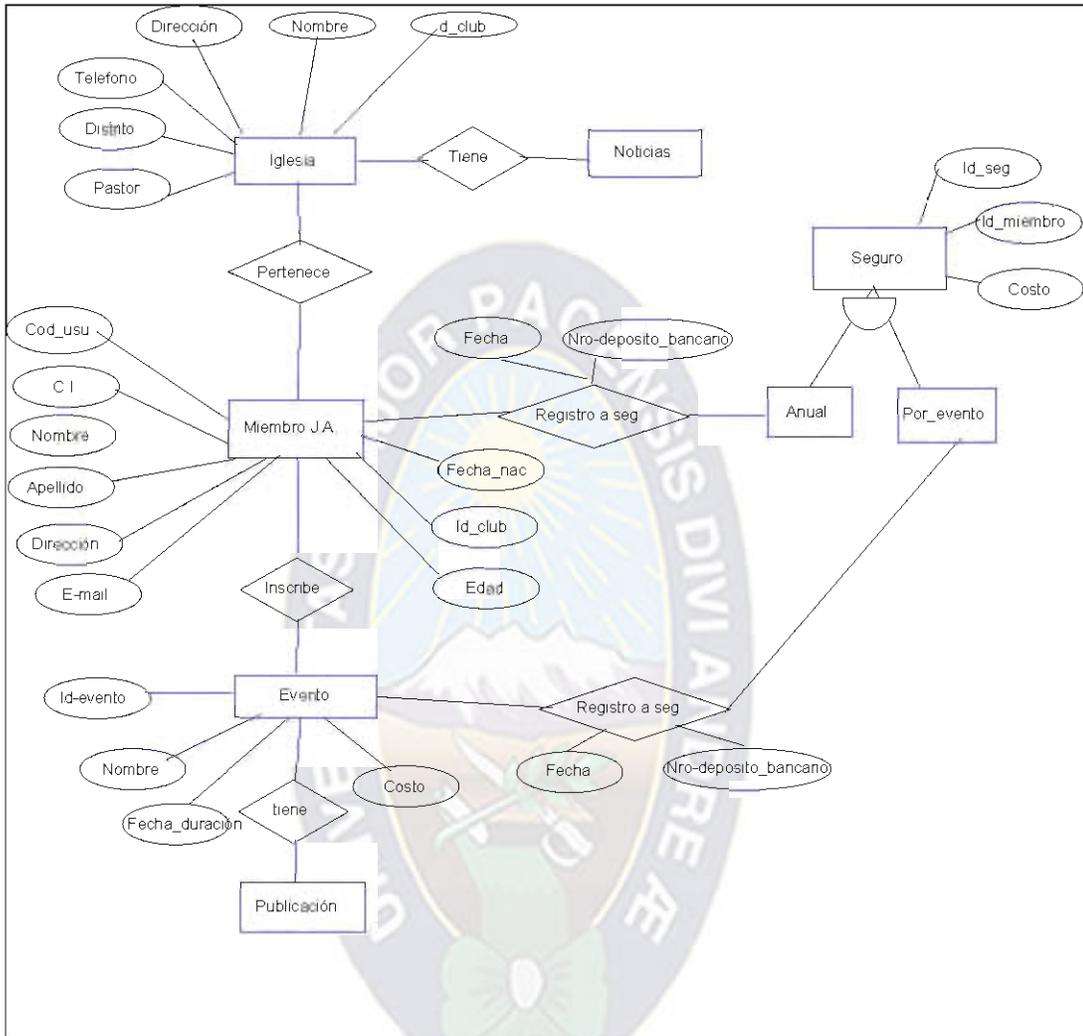
- **Contratos de operaciones: “PAGO DE SEGURO”**

Contrato: Pago de seguro	
Nombre:	Pago de seguro
Responsabilidades	El Administrador ingresa con contraseña tipo de usuario, el sistema valida tipo código y tipo de usuario, si son válidos el sistema permite realizar el pago de segura ya sea este anual o por

Tipo:	Sistema
Referencias	R4, R5, R6, R7, R8, R10, R11, R12, R13
Cruzadas	
Notas	Realiza la búsqueda de formularios de pago de seguro en la Base de Datos.
Excepciones	Si la introducción de datos del depósito bancario no es correcta, el sistema regresa al formulario inicial. Y si no existe historial de miembro inscrito al Ministerio de Jóvenes Adventistas o miembro inscrito a evento, el sistema muestra mensaje de realizar
Salida	Registro de pago de seguro es guardado en la base de datos.
Precondición	El usuario miembro debe enviar el depósito bancario vía fax o caso contrario ser presentado hasta una fecha especificada para verificar el número de depósito bancario enviado al momento de la
Post-condición	Numero de depósito bancario verificado. Caso de no presentar el comprobante de depósito bancario original, el sistema eliminará el registro de inscripción.

Los demás casos de uso se encuentran en el anexo C.

3.2.2.3 Construcción del modelo Entidad relación

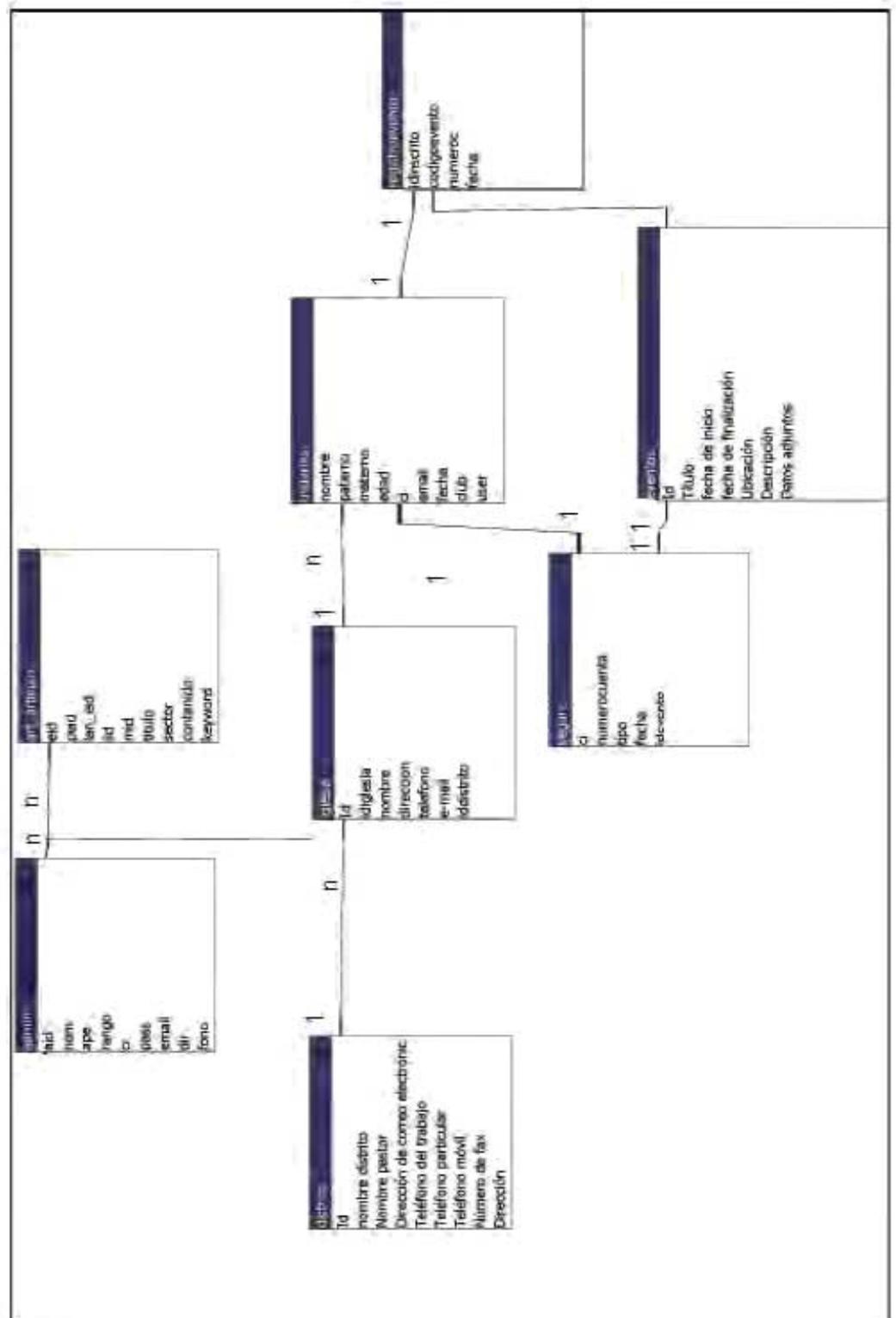


La figura 3.12. Diagrama del modelo entidad relación.

Fuente: Elaboración propia

3.2.2.4 Construcción del modelo conceptual

Para construir este modelo se hace uso de una lista de informaciones obtenida a partir de los diagramas de actividades. Además es una cualidad esencial que debe ofrecer un modelo conceptual es que representa cosas del mundo real.



La figura 3.13 Diagrama del modelo conceptual del sistema.
Fuente: Elaboración propia

✓ **Registro de los términos del glosario**

Es un documento simple en el cual se definen términos, la tabla 3.4 se muestra para el sistema planteado.

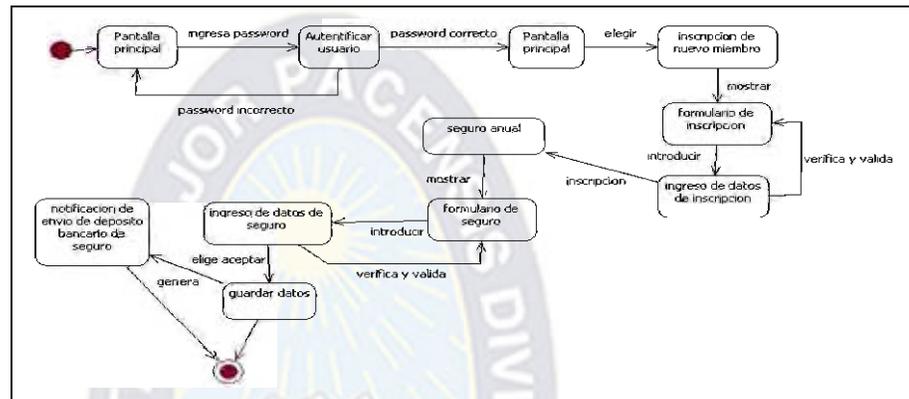
Termino	Categoría	Comentarios
Conexión al sistema	Caso de uso	El usuario ya sea el administrativo, director de J.A O Miembro de Iglesia, hace una conexión al sistema, introduciendo código y contraseña, el sistema el sistema devuelve la conexión admitiéndole el ingreso al mismo siempre y cuando estos datos sean correctos.
Especificación de datos	atributo	Descripción breve de un dato específico ya sea de Miembro de iglesia, Director de J.A., Administrador.
Nombre	Tipo	Un nombre que tiene cada dato ya sea de Miembro de iglesia, Director de J.A., Administrador
Inscripción Ministerio joven	Caso de uso	Esta función es la mas importante, esta deben estar bien registrados los datos, para que el transcriptor pueda realizar distintas operaciones
Inscripción a eventos y/o actividades	Caso de uso	Esta función cumple cada director de J.A., reservando cupo para dicho evento, introduciendo datos personales directiva, grupo de apoyo, Miembros de iglesia.
Verificación de seguros anuales y/o por eventos	Caso de uso	Esta función la realiza el administrador, revisando y controlando que miembros del ministerio joven cuentan con el seguro.
Visualización de reportes e informes anuales y trimestrales.	Caso de uso	Los usuarios administrativos, coordinador general, coordinadores distritales ingresan código y contraseña y el sistema valida los datos, si están correctos les permite el ingreso como tipo autoridad y así pueda acceder a la información de control de iglesias miembros del Ministerio aoven que desee acerca de rendimiento, permanencia en el ministerio joven.
Ver datos personales	Caso de uso	Los administrativos, directores de J.A., afiliados, introducen su clave, el sistema valida su clave y luego despliega la información datos personales, según el rol correspondiente al usuario, el los usuarios verifican los datos si sus datos son correctos el sistema confirma datos del usuario.
Actualización de tarjeta de especialidad	Atributo	Es un proceso que realiza después de la inserción de notas mediante una carta al instructor de club que le solicitan, pudiendo error del director de J.A.
Cambio contraseña	Caso de uso	Los administrativos, directores de J.A., Miembros de iglesia introduce su clave actual, el sistema valida la clave, el usuario lee las recomendaciones para introducir una nueva clave y luego el usuario introduce su nueva clave para que luego el sistema actualiza la base de datos con la nueva clave introducida.

Tabla 3.8 Glosario

3.2.2.6 Diagrama de estados

Un diagrama de estados describe todas las secuencias de estados y acciones por las que puede pasar cada uno de los subsistemas durante su ciclo de vida como reacción a distintos eventos.

- a) Diagrama de estados, para el caso de uso “INSCRIPCIÓN DE NUEVO MIEMBRO”, ver figura 3.14.



La figura 3.14 Diagrama de estados de inscripción de nuevo miembro.
Fuente: Elaboración propia

- b) Diagrama de estados, para el caso de uso de “INSCRIPCIÓN O RESERVA O CONFIRMACIÓN DE INSCRIPCIÓN A EVENTO” ver figura 3.15.

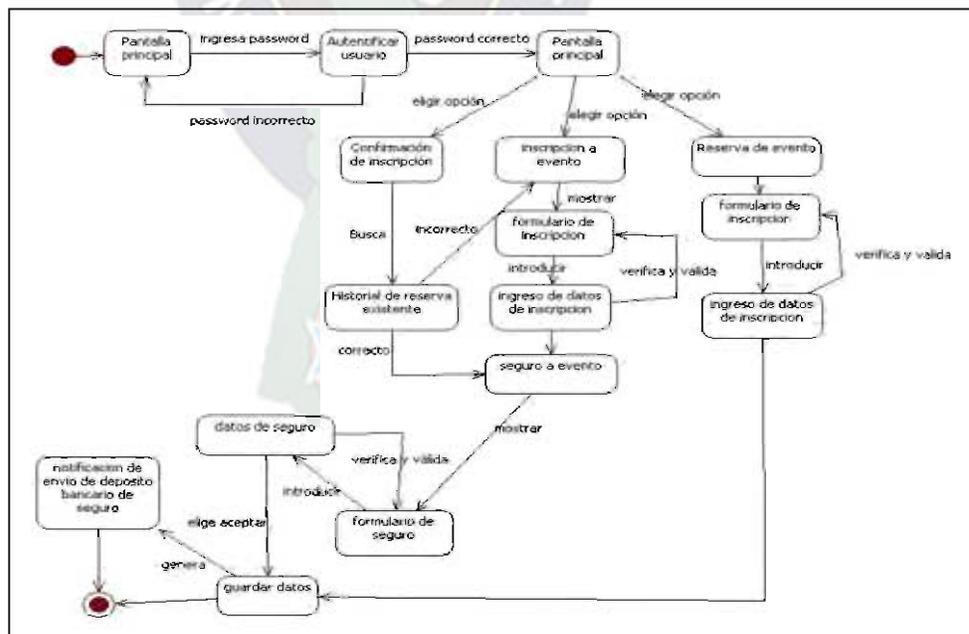


Figura 3.15 Diagrama de estados inscripción de y/o reserva a evento.
Fuente: Elaboración propia

c) Diagrama de estados, para el caso de uso de “ PAGO DE SEGURO” ver figura 3.16

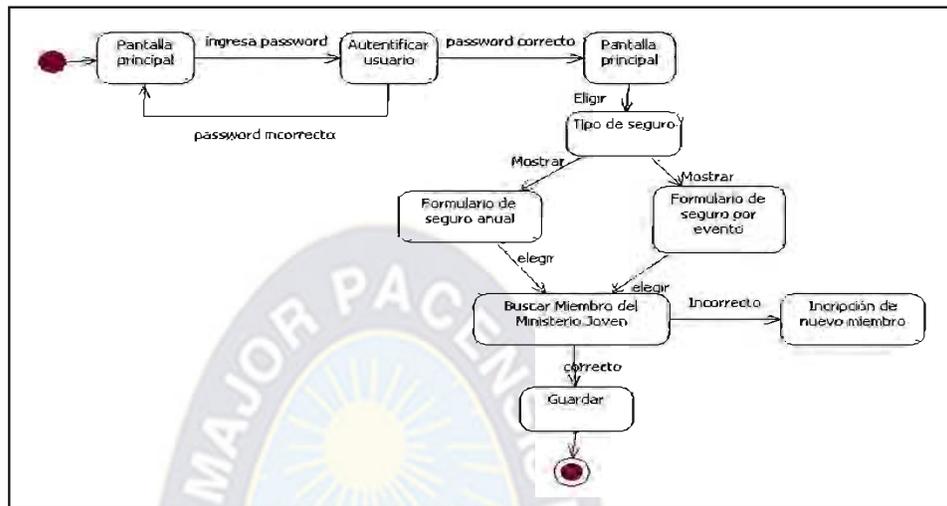


Figura 3.16 Diagrama de estados de Pago de seguro.

Fuente: Elaboración propia

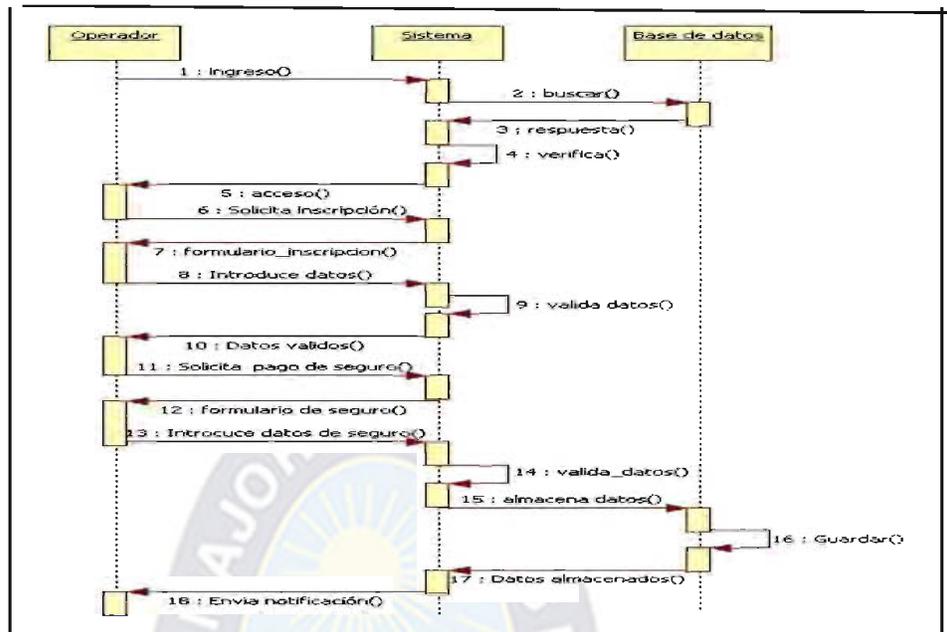
Los demás diagramas de actividades se encuentran en el anexo D.

3.2.3 MODELO DE ANALISIS Y DISEÑO

3.2.3.1 Diagrama de secuencias

En este punto tratamos de determinar las operaciones que demandan y reclama cada actor del sistema. El diagrama de secuencia es un sistema que debería prepararse para el curso normal de los eventos de un caso de uso posiblemente también de otros, teniendo en cuenta los cursos opcionales más interesantes [Craig Larman, 1999].

a) Diagrama de secuencia, para el caso de uso “INSCRIPCIÓN DE NUEVO MIEMBRO”, ver figura 3.17



La figura 3.17 Diagrama de secuencia de inscripción de nuevo miembro.
Fuente: Elaboración propia

b) Diagrama de secuencia, para el caso de uso de **“INSCRIPCIÓN O RESERVA O CONFIRMACIÓN DE INSCRIPCIÓN A EVENTO”** ver figura 3.18.1, figura 3.18.2 y figura 3.18.3

1. Caso de inscripción a evento

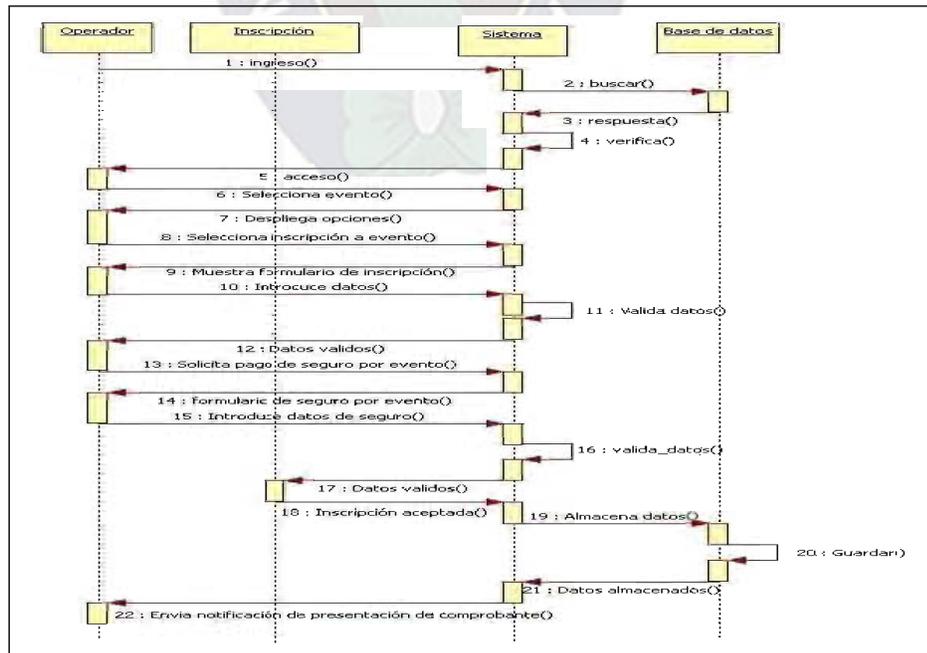


Figura 3.18.1 Diagrama de secuencia inscripción a evento.
Fuente: Elaboración propia

2. Caso de reserva a evento

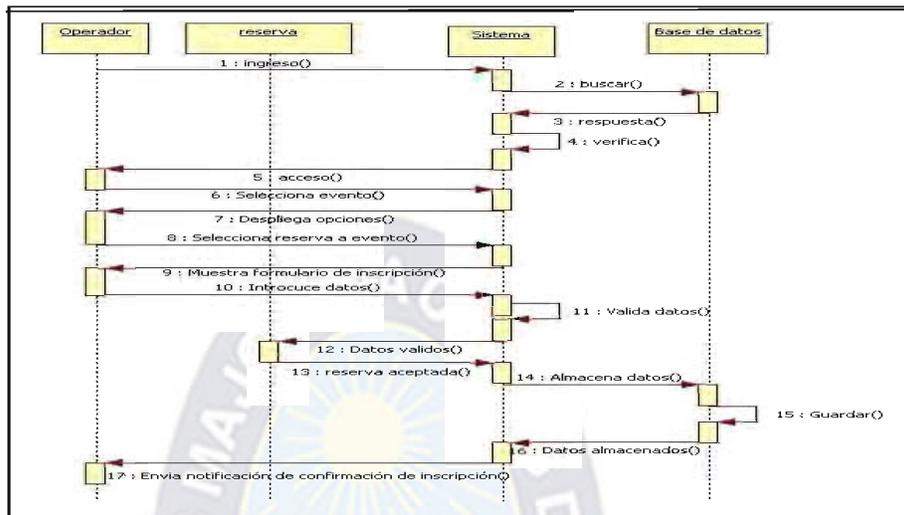


Figura 3.18.2 Diagrama de secuencia de reserva a evento.
Fuente: Elaboración propia

3. Caso de Confirmación de inscripción

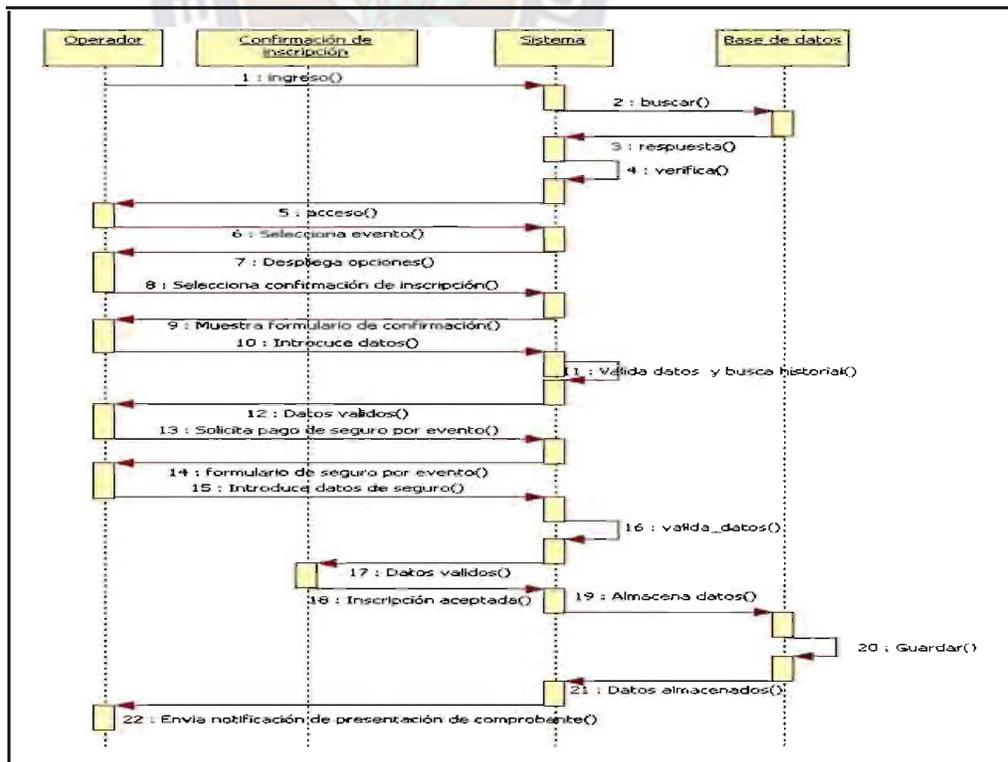


Figura 3.18.3 Diagrama de secuencia confirmación de inscripción a evento.
Fuente: Elaboración propia

c) Diagrama de secuencia, para el caso de uso de “PAGO SEGURO” ver figura 3.19

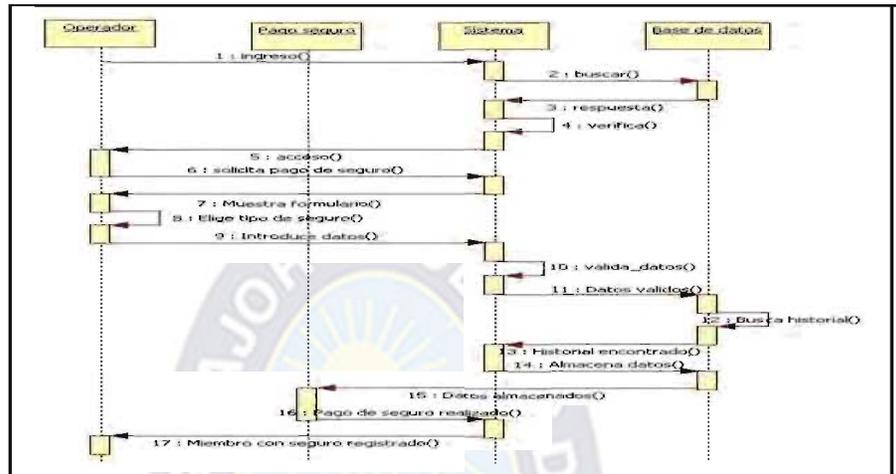


Figura 3.19 Diagrama de secuencia de Pago de Seguro.
Fuente: Elaboración propia

Los demás diagramas de actividades se encuentran en el anexo E.

3.2.3.2 Diagrama de clases

El diagrama de clases muestra la estructura estática del sistema modelado. Construimos el diagrama de las definiciones de las clases del modelo conceptual y diagramas de secuencia.

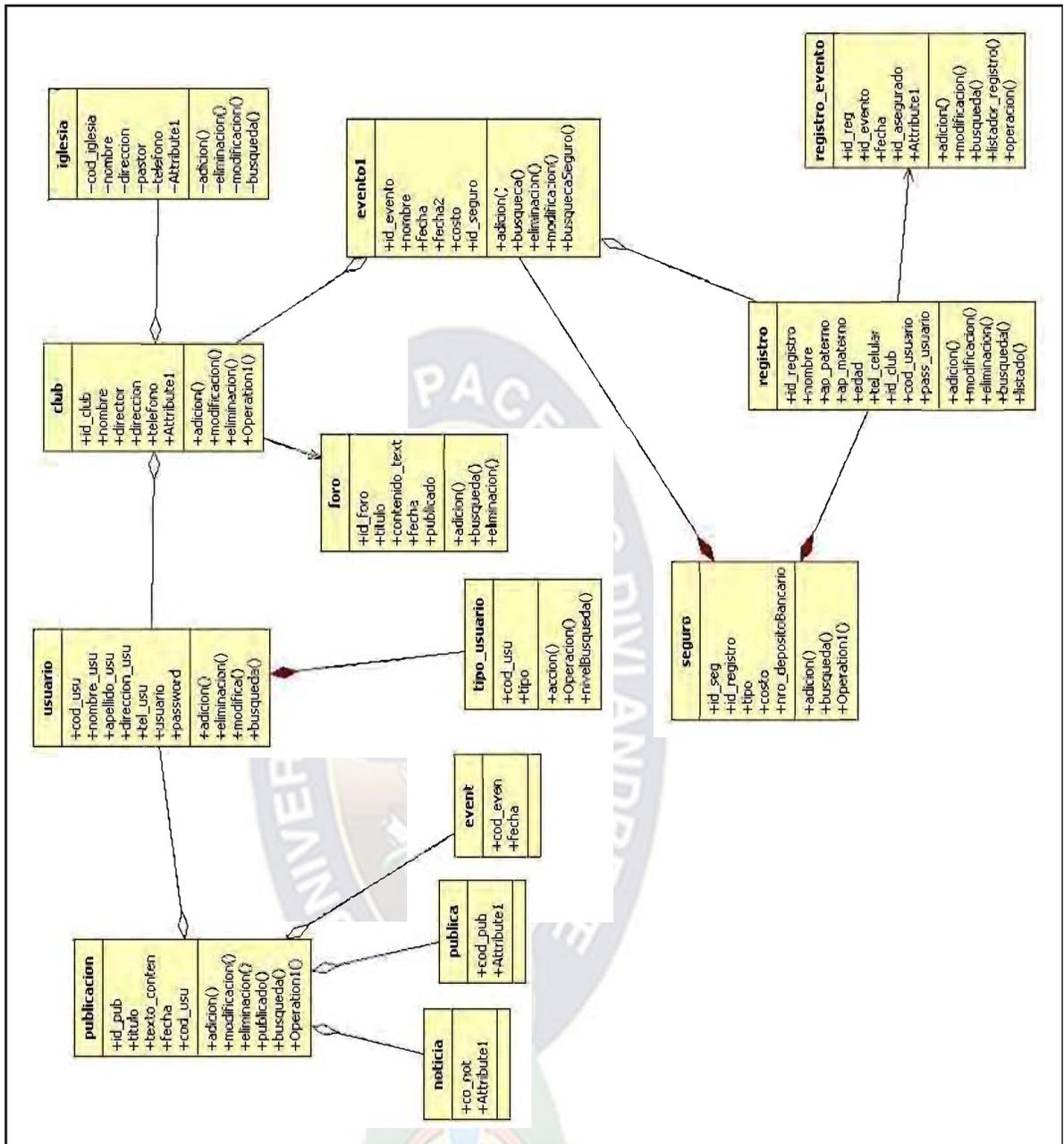
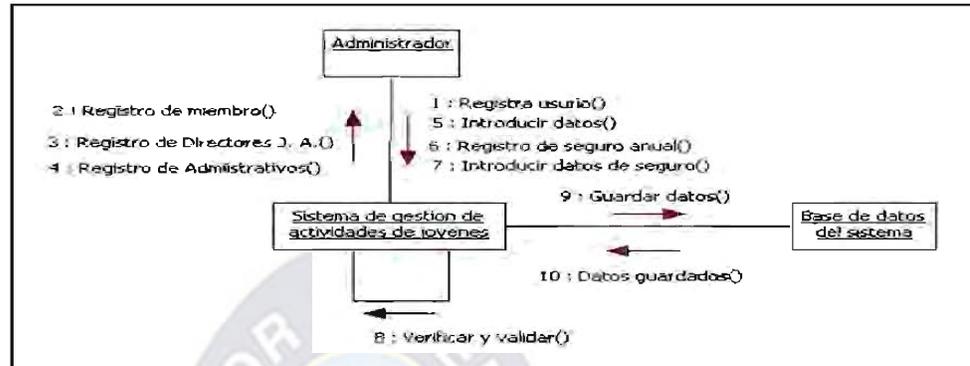


Figura 3 20 Diagrama de clases
Fuente: Elaboración propia

3.2.3.3 Diagramas de colaboración

Los diagramas de colaboración que destacan la organización de los objetos que participan en una interacción además se destaca el nuercero que se le asigna a cada mensaje para ordenar a si la actividad que se realiza.

a) Diagrama de colaboración, para el caso de uso “**INSCRIPCIÓN DE NUEVO MIEMBRO**”, ver figura 3.21



La figura 3.21 Diagrama de colaboración de inscripción de nuevo miembro.
Fuente: Elaboración propia

b) Diagrama de colaboración, para el caso de uso de “**INSCRIPCIÓN O RESERVA O CONFIRMACIÓN A EVENTO**” ver figura 3.22.1, figura 3.22.2 y figura 3.22.3

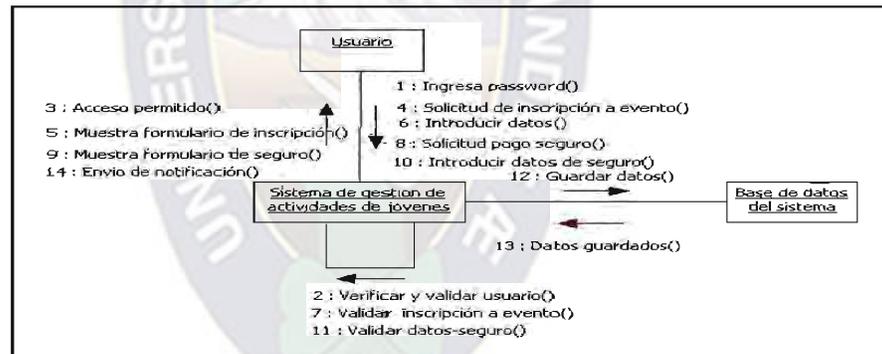


Figura 3.22.1 Diagrama de colaboración inscripción a evento.
Fuente: Elaboración propia

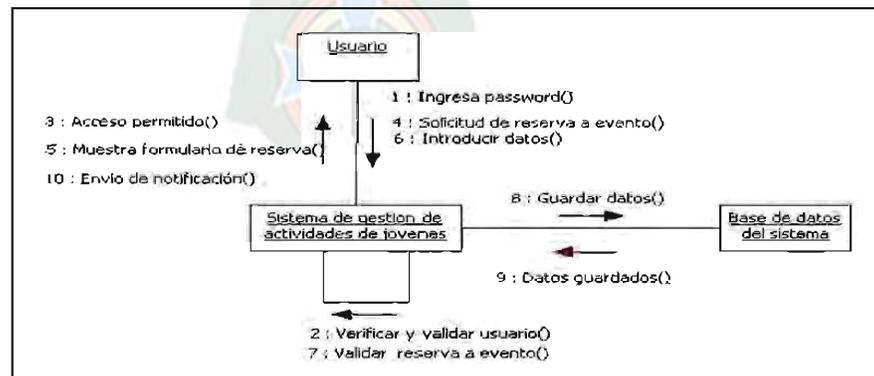


Figura 3.22.2 Diagrama de colaboración reserva a evento.
Fuente: Elaboración propia

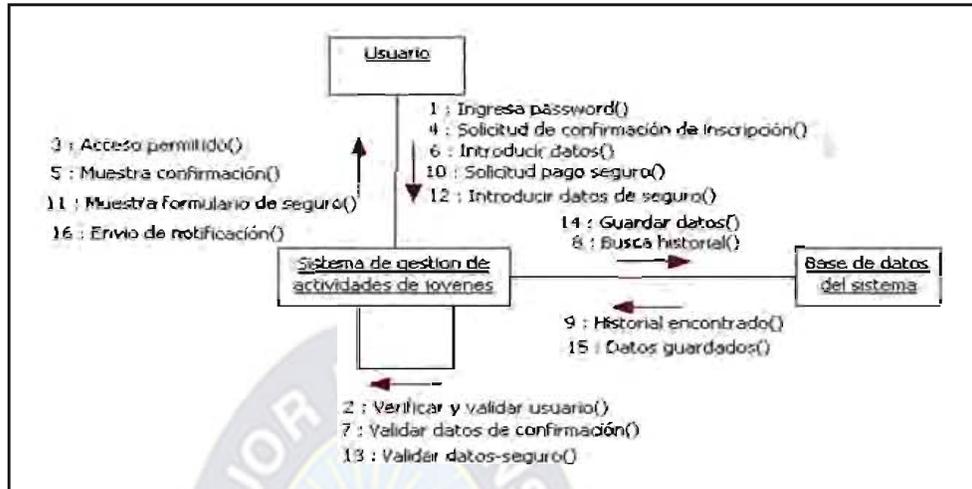


Figura 3.22.3 Diagrama de colaboración Confirmación de inscripción a evento.
Fuente: Elaboración propia

c) Diagrama de colaboración, para el caso de uso de “PAGO DE SEGURO” ver figura 3.23

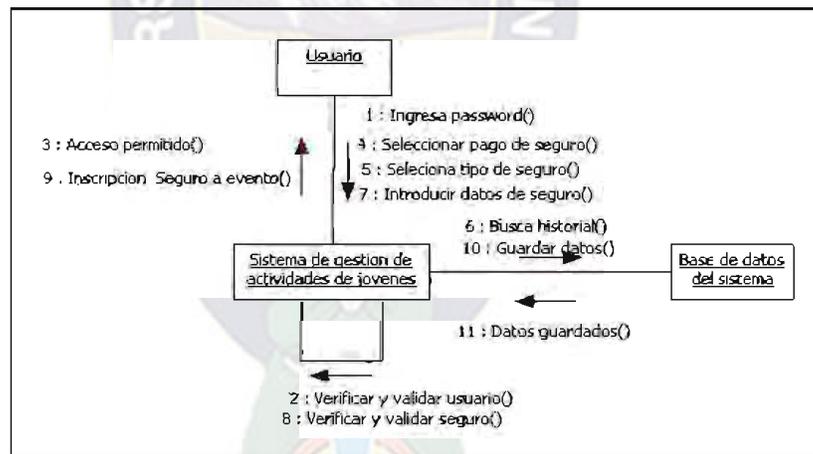


Figura 3.23 Diagrama de colaboración inscripción de nueva Iglesia.
Fuente: Elaboración propia

Los demás diagramas de actividades se encuentran en el anexo F.

3.2.3.4 Modelo de datos

En esta actividad se define la estructura física de datos que utilizará el sistema a partir del modelo de diseño de clases, obteniendo así un modelo de base de datos relacional.

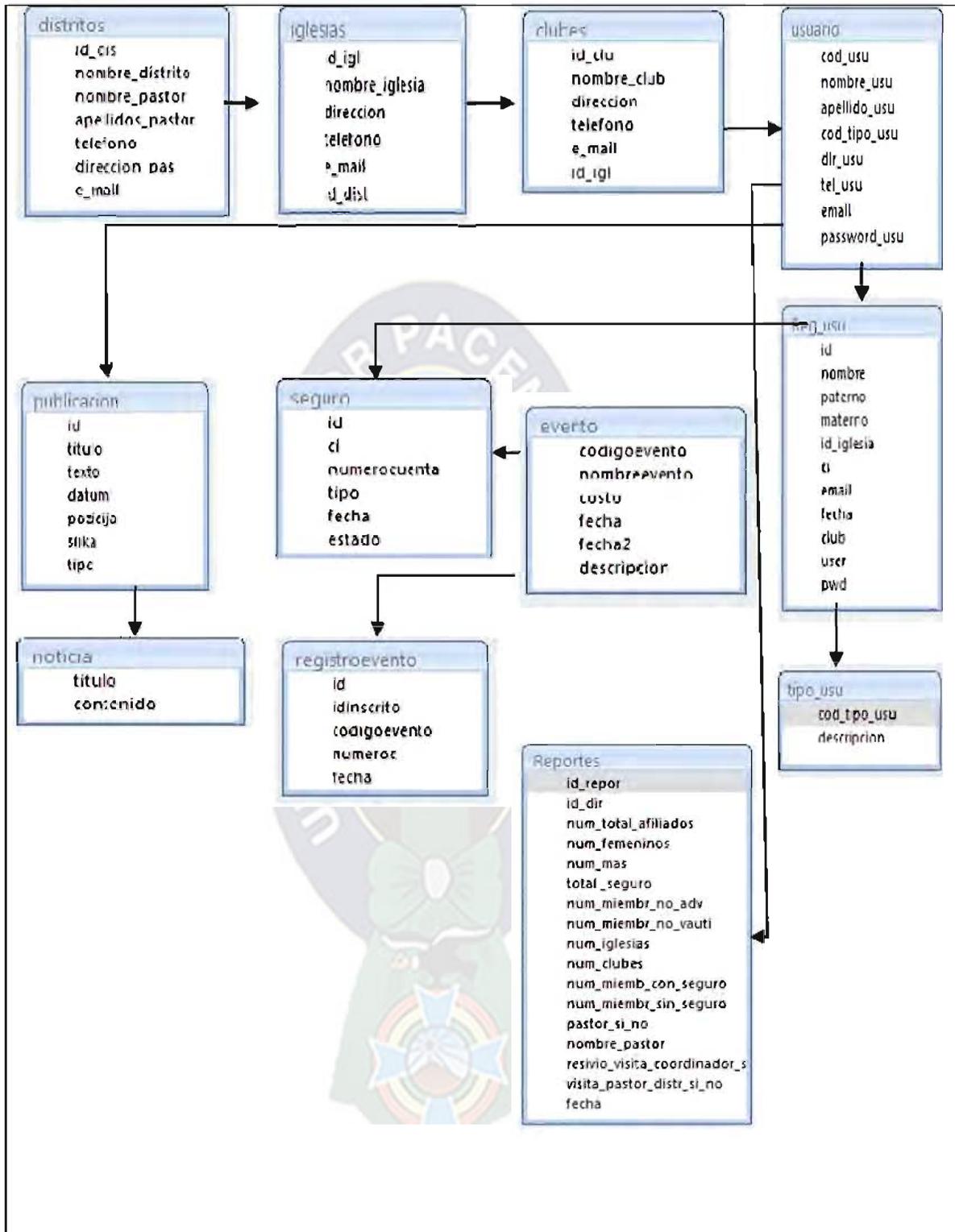


Figura 3 24 Diagrama Modelo de datos del sistema
Fuente: Elaboración propia

3.2.3.5 Arquitectura del sistema

Se presenta en un entorno multiusuario. Ya que al diseñar el Sistema de Gestión de actividades de jóvenes – Misión Boliviana Occidental, se tomó en cuenta los dos tipos de arquitectura que sugiere la metodología RUP: La arquitectura física y lógica, así como la tecnología e infraestructura que se empleará para implementar sus funcionalidades.

✓ Arquitectura lógica

Para la arquitectura lógica el sistema esta basado en **Model, View, Controller**, o bien: **Modelo, Vista y Controlador (MVC)**, que es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, separando el sistema en “tres capas”; el **Modelo, la Vista y el controlador**, como se muestra a continuación en el diagrama con la vista lógica de la arquitectura del Sistema de Gestión de Actividades de Jóvenes.

- **Capa de vista**, es la capa donde se desarrolla la interfaz para el usuario final en la que se presenta la información obtenida con el modelo en un formato adecuado para interactuar.
- **Capa del controlador**, es la capa donde ocurren la mayoría de los procesos, esta capa responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista, dependiendo de la acción solicitada por el usuario, siendo el que pide al modelo la información necesaria e invoca a la plantilla (de la vista) que corresponda para que la información sea presentada.
- **Capa del modelo**, el Modelo se encarga de todo lo que tiene que ver con la persistencia de datos. Guarda y recupera la información del medio persistente que utilizemos, ya sea una base de datos, ficheros de texto, XML, etc. Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Entonces el producto software esta desarrollado para que funcione en entornos Web, esto implica que su mantenimiento debería ser fácil, para este efecto se usa el modelo vista controlador, donde los procesos, las interfaces (formularios y reportes) y

el mismo modelo que es la abstracción de las clases, y estructuras estáticas se puedan mantener independientemente de las demás. Ver figura 3.25.

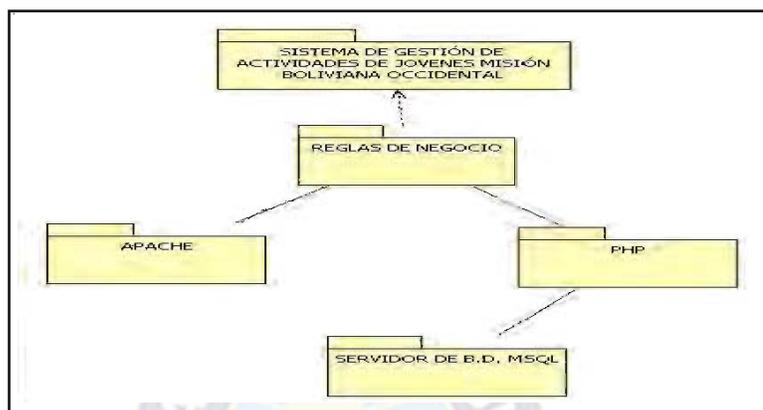


Figura 3.25 Arquitectura lógica del sistema
Fuente: Elaboración propia

✓ **Arquitectura física**

Trata de la utilización de recursos en términos de hardware y software necesarios para el funcionamiento del sistema. La implementación es la puesta en marcha del sistema de Gestión de actividades de jóvenes –Misión Boliviana Occidental, realizando la integración total de todos los subsistemas que se plantearon e identificaron. Para esto es necesario cumplir con los requerimientos descritos a continuación. Ver figura 3.25

- **Browser cliente:** En la aplicación de los clientes necesita el browser para solicitar al servidor Web páginas HTML y PHP. Estos son Internet Explorer, firefox y otros.
- **Servidor Web:** Es el lugar donde los browsers de los clientes acceden al sistema y esto solo es a través del servidor Web, en este caso utilizaremos el servidor Apache 2.2.
- **Conexión HTTP:** Es el protocolo utilizado para la conexión entre los browsers y el servidor Web.
- **Páginas HTML:** Son aquellas páginas Web que tienen una interfaz para el cliente que no tienen procesamiento en el lado del servidor anteriormente mencionado.

- **Página servidor:** Son todas las páginas Web que realizan procesamiento en el servidor, para este proyecto se ha utilizado páginas desarrolladas en PHP.
- **Servidor de aplicación:** Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor (servidor de aplicación Apache), siendo el encargado de ejecutar el código de las páginas servidor.
- **Base de datos:** Es el punto donde se almacenada todos los datos de los usuarios del sistema, se ha utilizado la base de datos MySQL.
- **Script cliente:** Usamos Java Script y el framework JQuery embebido dentro de algunas páginas HTML y Hojas de Estilo CSS para conseguir una interfaz mejorada.

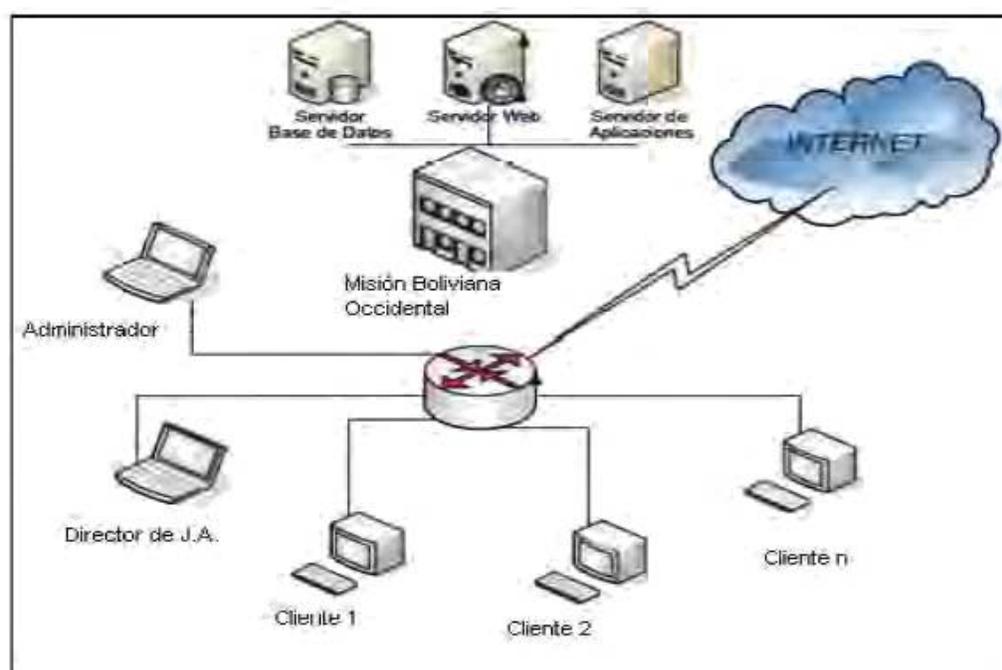


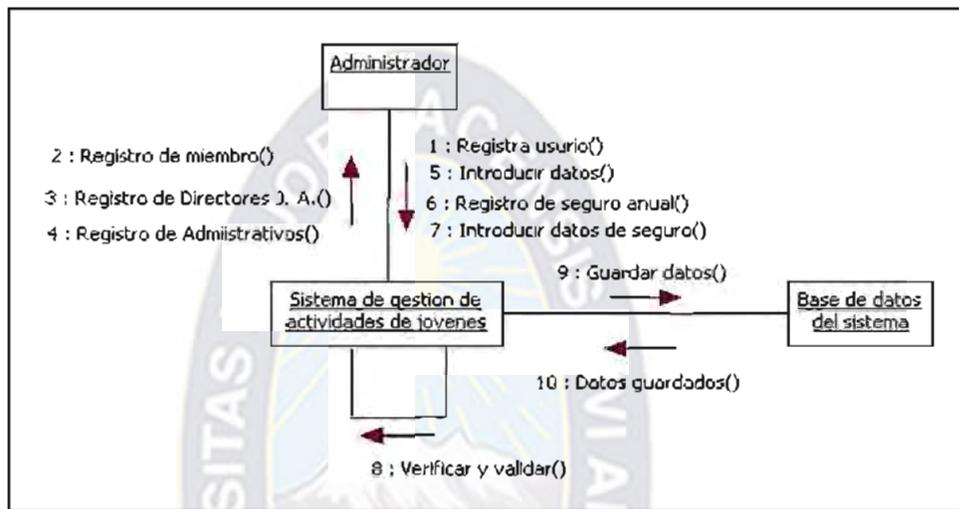
Figura 3.26 Diseño físico y arquitectura del Sistema
Fuente: Elaboración Propia

3.3.3.6 Modelo de implementación

En la implementación las clases de diseño son convertidas en ficheros fuentes, y ejecutables código utilizando un lenguaje de programación. La programación es una fase separada durante la cual los modelos son convertidos a código y el sistema realizando a través de la implementación (archivos de código fuente, archivos de encabezados, etc.)

Para eso se tendrá el siguiente ejemplo de Implementación.

En el siguiente gráfico se observa el diagrama de colaboración, para el caso de uso “INSCRIPCIÓN DE NUEVO MIEMBRO”, ver figura 3.27



La figura 3.27 Diagrama de colaboración de inscripción de nuevo miembro.
Fuente: Elaboración propia

La figura 3.28 muestra el código de este diagrama de colaboración.

```

<fieldset id="form">
  <legend>Nuevo Usuario</legend>
  <form name="comentariosFrm" id="comentariosFrm" method="post" action="ameusuario.php">
  <ol>
    <li>
      <label>Nombre: </label><input type="text" name="nombre_usu" value="{? echo $nombre; ?}"
      class="required" />
    </li>
    <li>
      <label>Apellido(s): </label><input type="text" name="apellido_usu" value="{? echo $apellido; ?}"
      class="required" />
    </li>
    <li>
      <label>Telefono: </label><input type="text" name="tel_usu" value="{? echo $telefono; ?}"
      class="required number" />
    </li>
    <li>
      <label>Direccion: </label><input type="text" name="dir_usu" value="{? echo $telefono; ?}"
      class="required" />
    </li>
  </ol>
  </form>
  </fieldset>
  
```

Formulario de registro

3.3.1.1 Diagrama de componentes

Este diagrama muestra la organización y las dependencias de los componentes lógicos del sistema de Gestión de actividades de jóvenes – Misión Boliviana Occidental. Ver figura 3.29

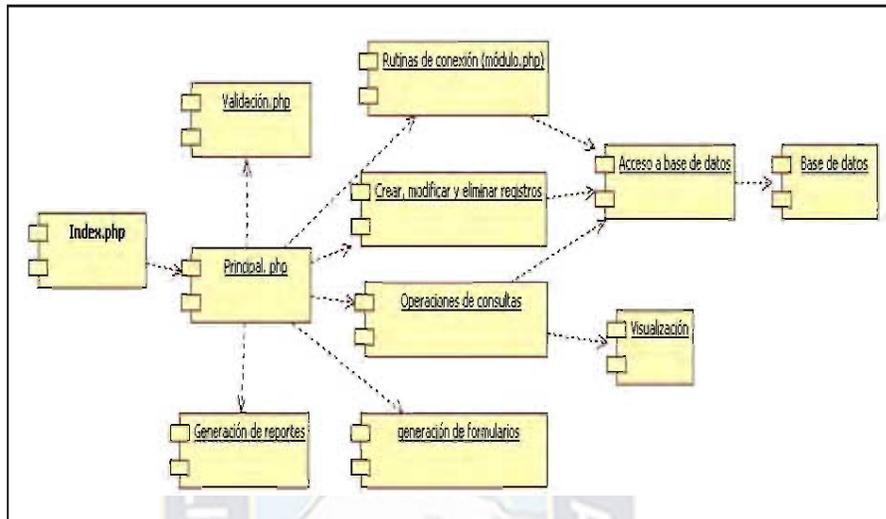


Figura 3.29 Diseño físico y arquitectura del Sistema
Fuente: Elaboración Propia

3.3.1.2 Diagrama de despliegue

La vista del diagrama de despliegue representa la disposición de componentes de ejecución en instancias de nodos. Un nodo es un recurso de ejecución en el sistema, tal como una computadora, un dispositivo, o memoria. Esta vista permite determinar las consecuencias de la distribución y asignación de recursos. Ver figura 3.30

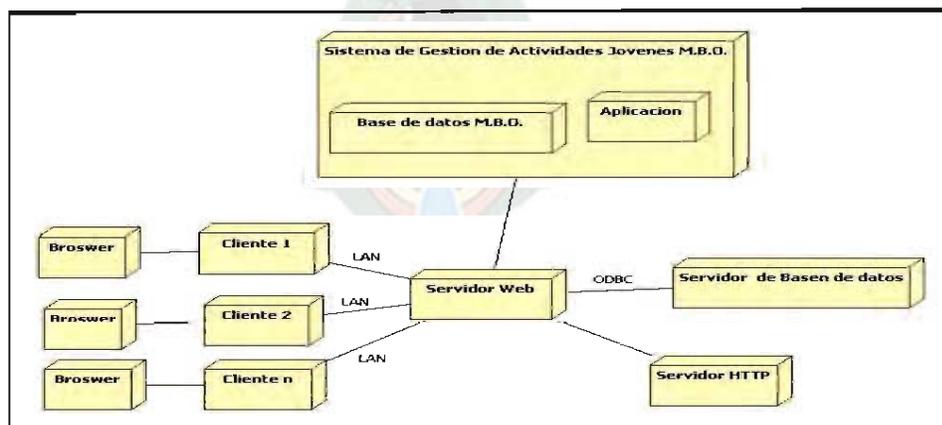


Figura 3.30 Diseño físico y arquitectura del Sistema
Fuente: Elaboración Propia

3.3.1.3 Modelo de implementación

La implementación es la puesta en marcha del Sistema de Gestión de Actividades de Jóvenes de la Misión Boliviana Occidental, realizando una integración total de los subsistemas que se identificaron. Esta implementación dependerá del cumplimiento de los siguientes requerimientos:

➤ **Requerimientos de Software**

Para el buen funcionamiento del sistema se requieren las siguientes características:

- ✓ **Herramientas para el desarrollo.**
- ✓ **Plataforma de trabajo**

Instalación del sistema, primeramente tiene que tener un servidor Web instalado sistema operativo Windows Server 2000, servidor Apache y páginas dinámicas Php.

- ✓ **Gestor de Base de Datos**

El Gestor de Base de Datos, proyecto fue elaborado entorno a Mysql versión 5.

- ✓ **Herramienta de programación**

➤ La herramienta de programación que utilizamos para este proyecto es basado en el paginas dinámicas Php y JQuery que es una biblioteca o framework de Javascript., que interactúa con la tecnología AJAX a páginas Web.

Instalación. Tipo de hardware como de software en la cual pueda radicar y su rendimiento sea deseable.

✓ **Software**

- Sistema Operativo Windows Server 2000/XP/2003, O Linux Fedora Core 5
- Servidor de página Web Apache 2.2
- Servidor de base da datos MySQL

➤ **Requerimiento de hardware**

dispositivo	características
procesador	Adiante de Pentium IV 1333, 1600, 2800 (superior) mhz (velocidad)
Disco duro	80. Giga bites o superior
memoria ram	1 GB o superior
Monitor	Svga
tarjeta de red	Pci para rj 45 o cable coaxial
Cable de red	Utp

Tabla 3.9 Requerimientos de hardware

- **Comunicaciones**

Protocolo de comunicaciones de Internet TCP/IP

- ✓ **Requerimiento de Internet**

- Conexión a Internet < línea dedicada
- Ancho de banda 512 KB o superior

3.3.2 Diseño navegacional

En este diseño se define el contexto en el cual se desenvolverá el usuario, es decir la información a la que tendrá acceso y las posibilidades permitidas para su manejo. Ver figura 3.31

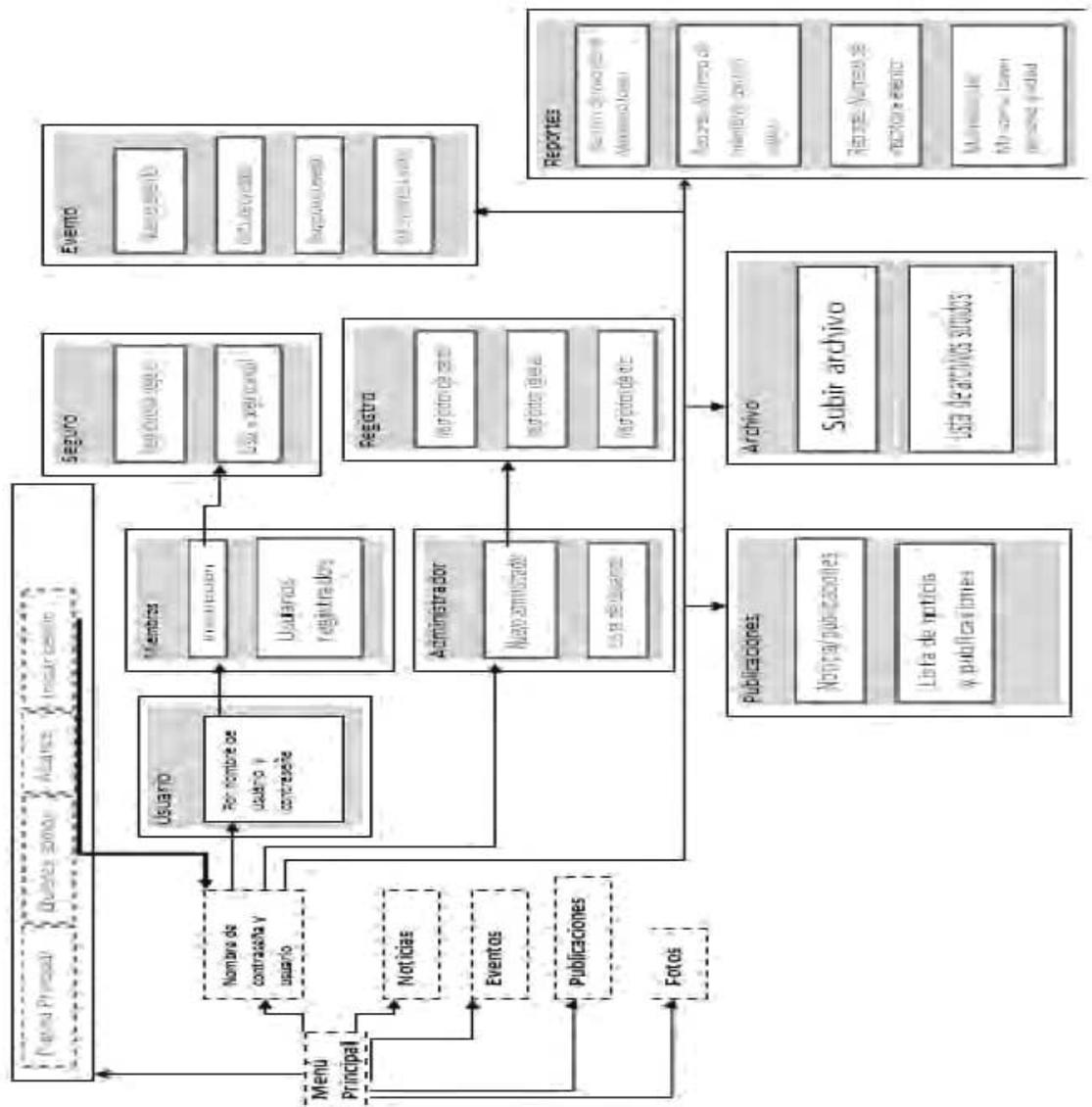


Figura 3.31 Diseño navegacional del software
Fuente: Elaboración Propia

3.3.3 Diagrama de interfaz abstracta

Para describir la interfaz del usuario de la aplicación en OOHDM se utiliza el modelo Vista de Datos Abstractos (ADV) que especifica la organización y el comportamiento de la interfaz. Esto significa la manera en que diferentes objetos aparecerán y activarán vistas de datos u otros objetos de la interfaz.

- AVD Aplicación De la Web Sistema de Gestión de Actividades de Jóvenes. Ver figura 3.32

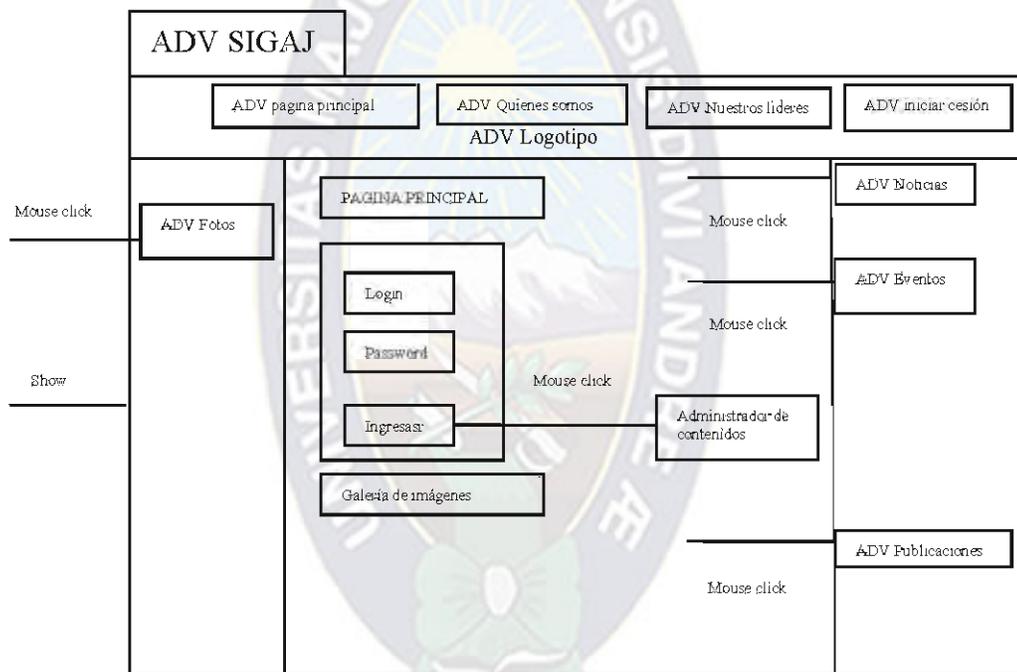


Figura 3.32 AVD Aplicación De la Web Sistema de Gestión de Actividades de Jóvenes.
Fuente: Elaboración Propia

- AVD Administración de datos del Sistema de Gestión de Actividades de Jóvenes al cual accede el administrador. Ver figura 3.33

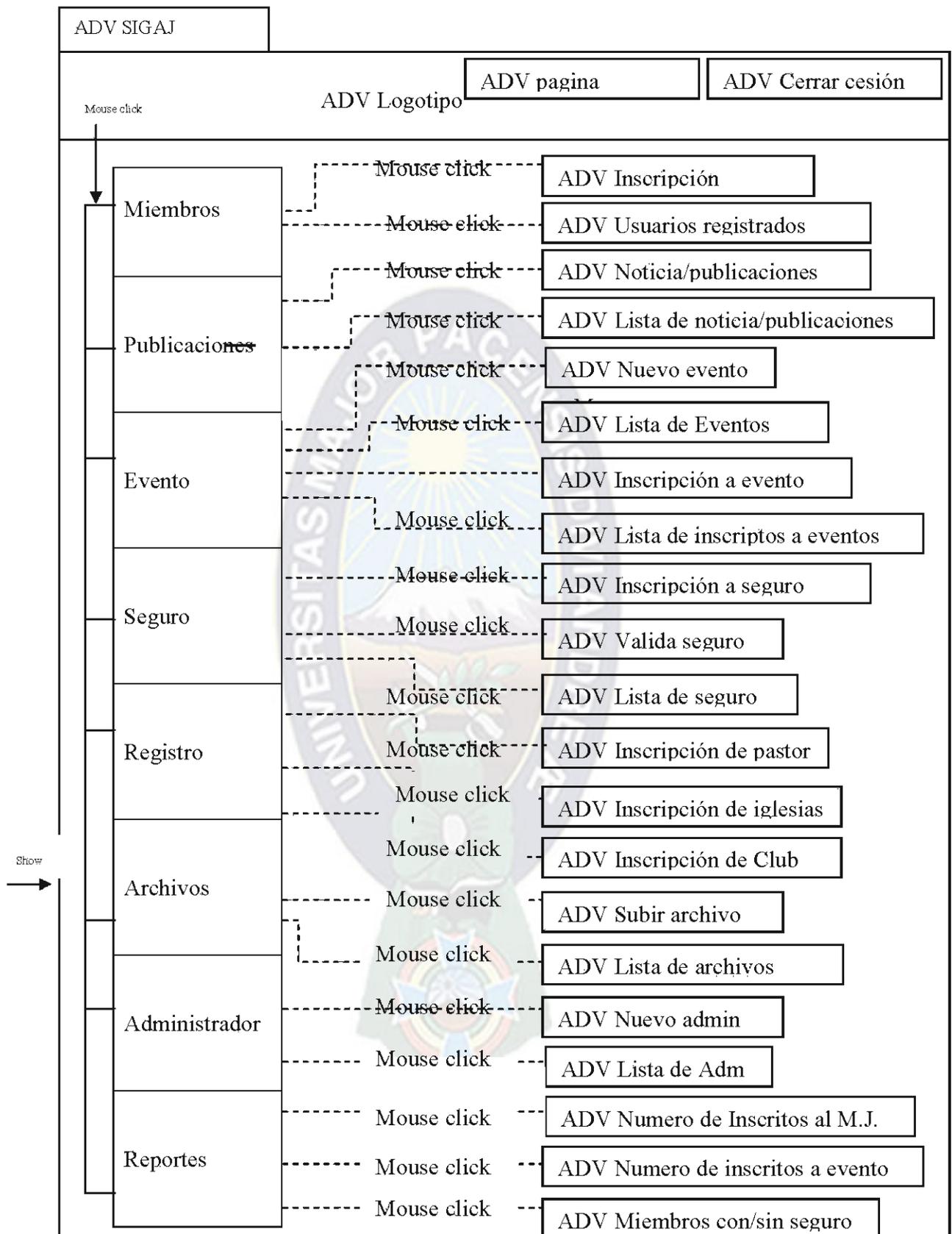
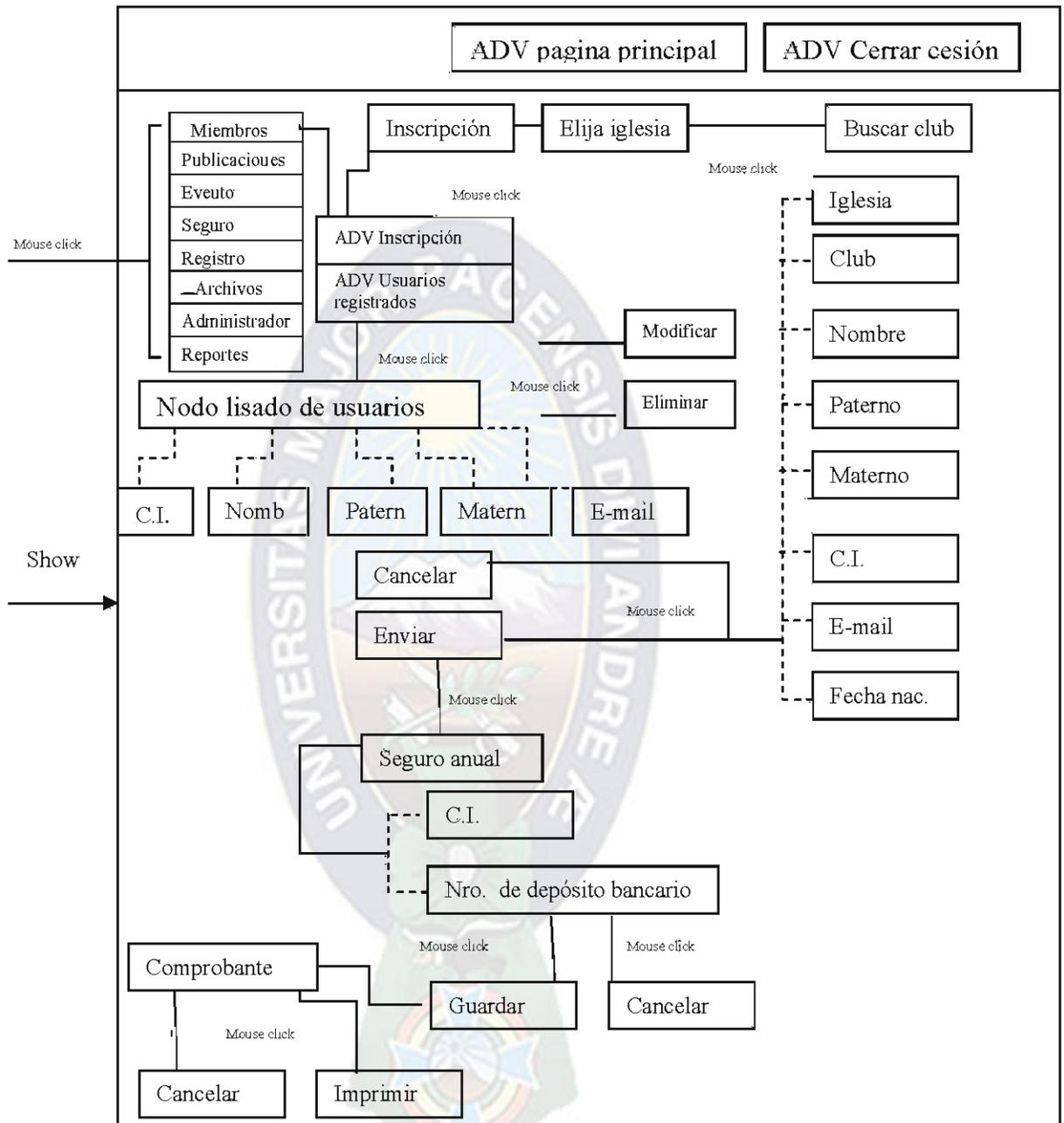


Figura 3.33 AVD de administración de datos del Sistema, administrador.
Fuente: Elaboración Propia

- “ADV Inscripción y ADV Usuarios registrados”, ver figura 3.34.



La figura 3.34 ADV Inscripción y ADV Usuarios registrados.

Fuente: Elaboración propia

- "ADV seguro", ver figura 3.35

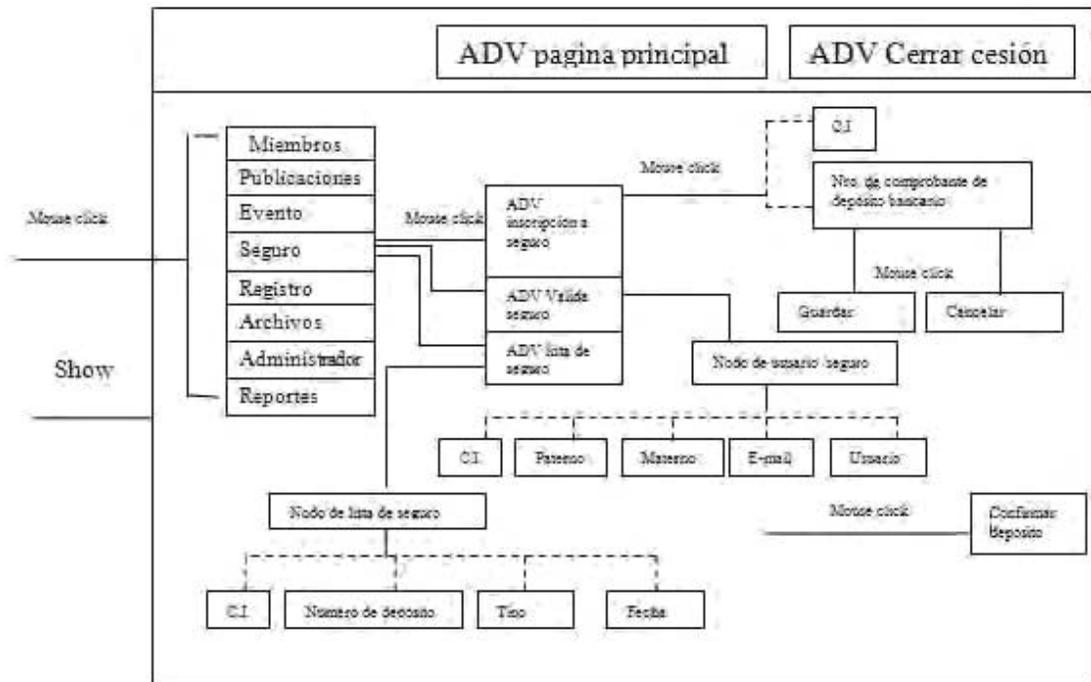


Figura 3.35 ADV seguro.
Fuente: Elaboración propia

3.3.4 Interfaz de usuario

A continuación se muestra la figura 3.35 la pantalla de autenticación de usuarios, también vemos la interfaz de la pagina Web de la institución, cuenta con links como ser inicio, quienes somos, publicaciones, noticias, foro, quienes son los miembros de las diferentes Iglesias que forman parte del Ministerio Joven del Occidente Boliviano. Luego de autenticarse podemos observar otro menú pero por tipo de usuario, inscripción a evento, pago de seguro, crear publicación, crear noticia, además que se tiene el menú de los directores, que pueden inscribir a sus afiliados, pueden realizar reservas de eventos como ser campamentos. Ver figura 3.38

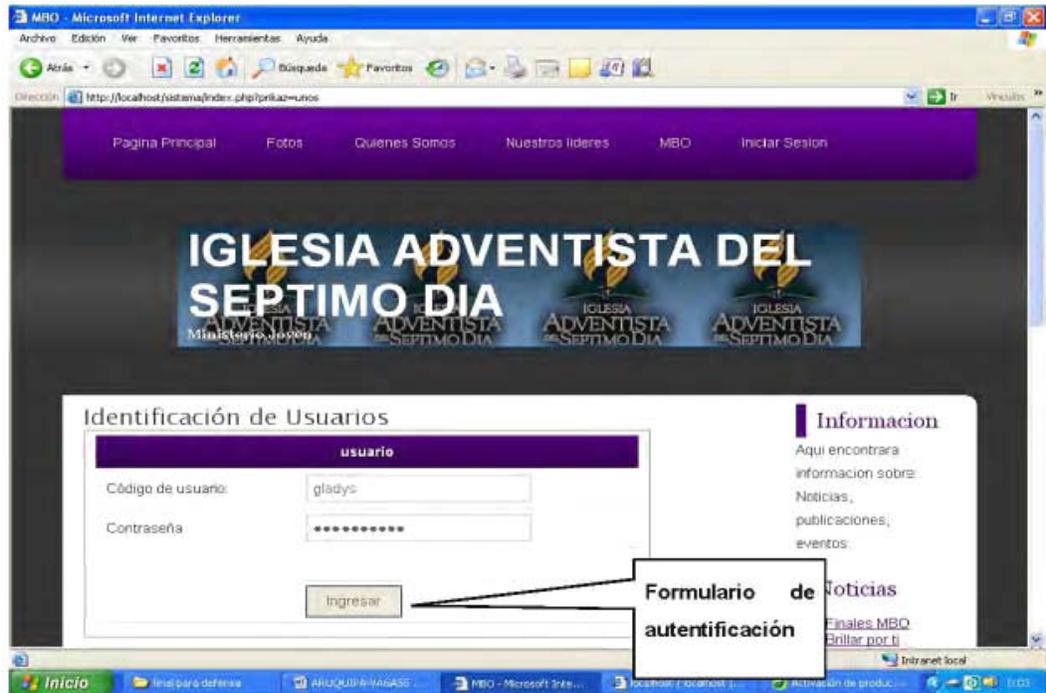


Figura 3.36 Formulario de autenticación.
Fuente: Elaboración propia

A continuación se muestra la figura 3.37, la pantalla de inscripción de nuevos miembros, adicionar, editar datos del nuevo miembro de Iglesia. Además que se observa la validación de los datos que son ingresados.

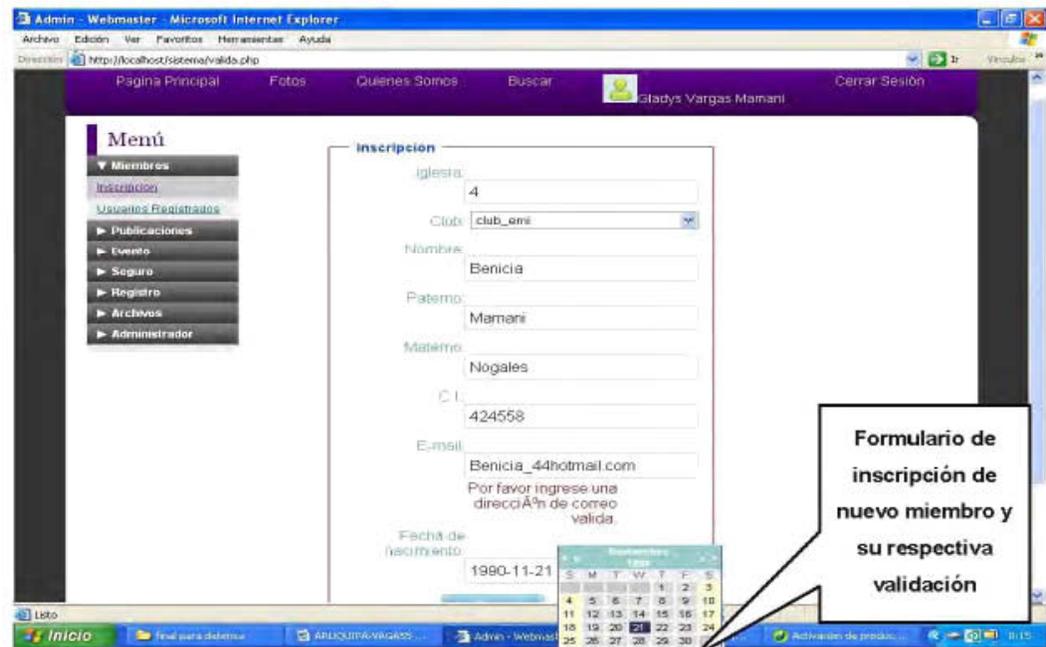


Figura 3.37 adición de nuevo miembro de Iglesia
Fuente: Elaboración propia

La figura 3.38 muestra la administración de afiliado, como ser editar o eliminar:

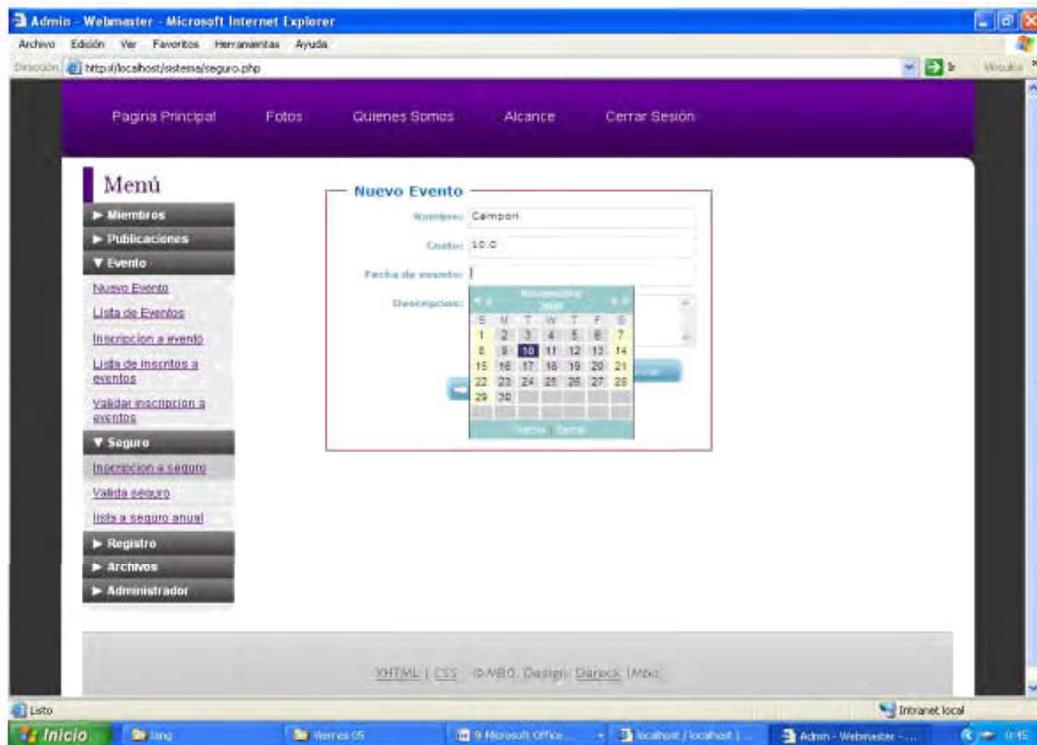


CÓDIGO	NOMBRE (S)	APELLIDOS	TIPO	DIRECCIÓN	TELÉFONO	
3456789	gladys	vargas	mamani	gladys@mail.com	34567890	Modificar Eliminar
567890	marco	vegas	pabone	h@k.com	567890	Modificar Eliminar

mostrar 1 - 5 de 2 Vease por pagina: 12 15 50 ALL
[Insertar Nuevos Miembro](#)

Figura 3.38 Modificación o eliminación de datos del miembro de Iglesia.

En la siguiente figura 3.39, se observa la inscripción a eventos.



Admín - Webmaster - Microsoft Internet Explorer

Arquivo Edición Ver Favoritos Herramientas Ayuda

Dirección: http://localhost/sistema/seguro.php

Página Principal Fotos Quiénes Somos Alcance Cerrar Sesión

Menú

- Miembros
- Publicaciones
- Evento
 - [Nuevo Evento](#)
 - [Lista de Eventos](#)
 - [Inscripción a evento](#)
 - [Lista de inscritos a eventos](#)
 - [Validar inscripción a eventos](#)
- Seguro
 - [Inscripción a seguro](#)
 - [Validar seguro](#)
 - [Lista a seguro anual](#)
- Registre
- Archivos
- Administrador

Nuevo Evento

Nombre:

Categoría:

Fecha de evento:

Descripción:

HTML | CSS | DHTML | Design | Doreck | IMB

Figura 3.39 Inscripción a evento.
Fuente: Elaboración propia

La figura 3.40 siguiente muestra el listado de avisos, la publicación de noticias que informara como boletín de información a los involucrados.

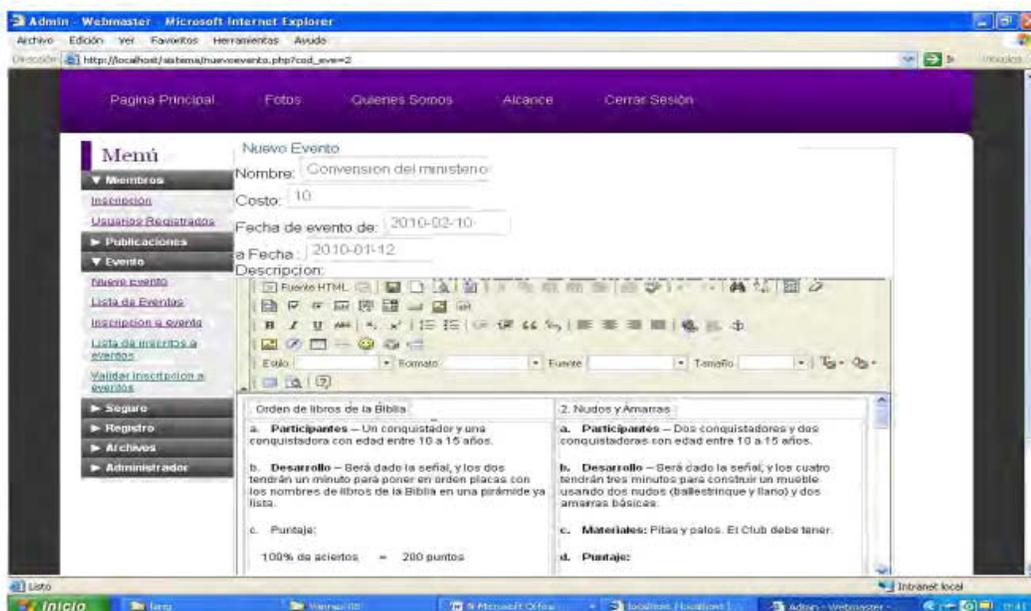


Figura 3.40 Creación y emisión de publicaciones.
Fuente: Elaboración propia

3.4 FASE DE TRANSICIÓN

En esta fase se muestra que el trabajo ha sido desarrollado y completado en las fases anteriores ya que se tiene un sistema estable, integrado y comprobado, por lo que se encuentra listo para entregar e implementar a los usuarios.

En esta fase el sistema puede ser instalado y utilizado, ya que además cuenta con la documentación completa. Además en un periodo de 20 días se han realizado las pruebas y correcciones necesarias para el correcto funcionamiento de este, con lo que se cuenta con un sistema que cubre todas las expectativas y requisitos establecidas por la Misión Boliviana Occidental.

Las capacitaciones a los usuarios se encuentran en proceso de desarrollo, ya que se los esta capacitando en el manejo del Sistema de Gestión de Actividades de Jóvenes – Misión Boliviana Occidental, además se contará con la guía de instalación y manual de usuario.



CAPÍTULO IV

SEGURIDAD, CALIDAD Y ANÁLISIS DE COSTO / BENEFICIO

CAPITULO IV

SEGURIDAD, CALIDAD Y ANALISIS COSTO/BENEFICIO

4.1 CRITERIOS DE SEGURIDAD

El control de acceso a los sistemas de informáticos constituye una herramienta para proteger la información y la entrada a un sitio Web por completo. De acuerdo con la norma UNE-ISO/IEC 177799: existen dos aspectos que son de vital importancia para la protección de información de software estos puntos son:

- Los aspectos organizativos para el resguardo de la información
- Los controles de acceso

Existen otros ocho aspectos que el estándar toma en cuenta, pero a estos se les interpreta como complementos que se utilizarán de acuerdo con las necesidades de resguardo. Es decir que la Seguridad consiste en la “aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo.”

4.1.1 Seguridad en la red

La seguridad ha sido el principal concerniente a tratar cuando una organización desea conectar su red privada al Internet.

Firewall (Cortafuego): Dentro la red, la gestión de seguridad es importante al momento de acceder a Internet y proveer conexión a los servidores de la MBO, para realizar esa tarea existen herramientas que ayudan a configurar firewall lógicos entre ellos se puede mencionar dos importantes como ser Iptables y Ipchains encargados de crear reglas de accesos tanto para la entrada y salida de paquetes dentro la institución.

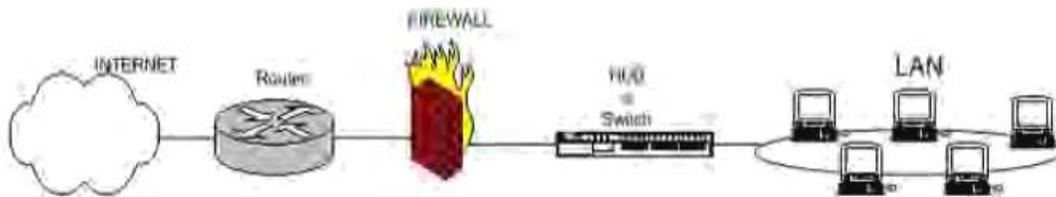


Figura 4.1 Esquema de Firewall típico entre red local e Internet
Fuente: IPTables manual práctico por, Ing. Pello Xavier Altadill Izura

En la misión se eligió IPTables el cual permite autorizar, bloquear y denegar la conexión con permisos ya que IPTables es un sistema de firewall bajo Linux Fedora y FreeBSD. Dentro de la Misión Boliviana Occidental están establecidas Listas de Control de Acceso que definen la red y las máquinas en particular. A cada lista se le asigno Reglas de Control de Acceso de acuerdo a las políticas de la organización, que permiten o deniegan el acceso a la red y al presente sistema.

4.1.2 Seguridad de la aplicación

a) Autenticación

- **Autenticación de usuarios:** Esta propiedad permite asegurar el origen de la información, es la autenticación que identifica al usuario o la máquina que trata de acceder a los recursos protegidos o no. La identidad del usuario es verificada mediante la validación del usuario y su contraseña, determinando que operaciones puede realizar el usuario.
- **Autenticación de datos:** En los sistemas de procesamiento autorizado de datos, muchas veces es imposible que alguien determine si los datos han sido modificados.

b) Confidencialidad

- **Seguridad de acceso:** Para esto se definen los niveles de usuarios, que son los únicos autorizados para el acceso a la información. Cada nivel de usuario esta asociado a un rol otorgado por el administrador del sistema.

USUARIO	CARACTERISTICAS Y PERMISOS
Administrador	<ul style="list-style-type: none"> ➤ Accede a la base de datos ➤ Crea, elimina, modifica y actualiza cuentas de usuario ➤ Crea, actualiza y elimina miembros del Ministerio Joven ➤ Publica, modifica y elimina documentos, con el fin de descargarlos por la web ➤ Publica, modifica y elimina noticias, eventos, imágenes que se crean necesarios ➤ Crea, modifica, actualiza y elimina iglesias, clubs, pastores ➤ Crea, modifica, actualiza y elimina inscripción a eventos y eventos ➤ Actualiza miembros con su respectivos seguros
Directores	<ul style="list-style-type: none"> ➤ Crea, actualiza y elimina miembros del Ministerio Joven ➤ Publica, modifica y elimina documentos, con el fin de descargarlos por la web ➤ Publica, modifica y elimina noticias, eventos ➤ Crea, modifica, actualiza y elimina iglesias, clubs, pastores ➤ Crea, modifica, actualiza y elimina inscripción a eventos y eventos ➤ Actualiza miembros con su respectivos seguros
Asistente	<ul style="list-style-type: none"> ➤ Crea, actualiza y elimina miembros del Ministerio Joven ➤ Crea, modifica, actualiza y elimina inscripción a eventos y eventos ➤ Inscribe a seguros. ➤ Publica, modifica y elimina documentos, con el fin de descargarlos por la web ➤ Accede a la información publicada en el portal
Miembro de J.A.	<ul style="list-style-type: none"> ➤ Se inscribe como nuevo miembro ➤ Se inscribe a evento ➤ Accede a la información publicada en el portal ➤ Accede a imágenes publicadas en el portal
Navegante del sitio Web	<ul style="list-style-type: none"> ➤ Accede a la información publicada en el portal ➤ Accede a imágenes publicadas en el portal

Tabla 4.1 Niveles de acceso a la información de los usuarios

El sistema, identifica al usuario antes de acceder a la base de datos solicitando un identificador de usuario y contraseña para poder iniciarse. La contraseña esta protegida mediante un código de 32 caracteres, generados por un algoritmo de encriptación (MD5). Dentro de la base de datos, todos los datos utilizados para el acceso de usuario se encuentran encriptados en cadenas únicas de 32 bits mediante el algoritmo de encriptación hash, MD5.

La figura 4.2 siguiente muestra un panorama de acceso al sistema.



Figura 4.2 Esquema de seguridad del software
Fuente. Elaboración propia

c) Disponibilidad:

Para esto se hace uso de variables de sesión, las cuales permiten que después de la autenticación del usuario y una vez autorizado el ingreso al sistema, este tiene acceso a la información sin restricciones, sin embargo una vez que no se realiza ninguna transacción durante un periodo de tiempo determinado, automáticamente se niega la estadía en el sistema.

De igual manera se toma en cuenta lo siguiente:

- Para el resguardo de la información se establece un monitoreo de ingreso a las funciones del software de tal manera se tenga un registro de usuarios que entran y realizan actividades en el sistema formalmente descrito el procedimiento descrito, es parte de las actividades planteadas por ISO 15408.
- El ISO 15408 sugiere generar procesos que habiliten algunas funciones del sistema de acuerdo con el usuario que entra el software.

4.1.3 Seguridad de la base de datos

- **Seguridad Lógica:** Se establecieron controles de accesos y sesiones de los usuarios a las plataformas de procesamiento informático y a los datos que estas gestionan con el fin de evitar las irregularidades que obstaculicen la

confidencialidad, exactitud y disponibilidad de la información y las mejoras Web fueran factibles de ejecutarse.

- **Seguridad Física:** Se realizan Backus de la base de datos de forma periódica (una copia por semana)

4.2 MÉTRICAS DE CALIDAD DEL SOFTWARE

La norma ISO-9126 define un modelo general para determinar la calidad del software a desarrollar que son: funcionalidad, portabilidad, mantenibilidad, usabilidad y confiabilidad.

4.2.1 Funcionalidad

Para hallar la funcionalidad del sistema utilizaremos las métricas orientadas a la función. Y para calcular los puntos de función (PF) se utiliza la siguiente relación:

$$PF = Cuenta\ Total * [X + Min(Y) * xFi]$$

Donde:

PF: Medida de Funcionalidad entregada a la aplicación como valor de normalización

CT: La suma de los siguientes datos.

X: Nivel de confiabilidad del sistema.

Y: Nivel de significación de error

Fi: Valores de complejidad.

Para hallar la funcionalidad del sistema se consideran los siguientes datos:

Entrada de Usuario:

- Panel de identificación de usuarios
- Panel de Registro de miembros del Ministerio de Jóvenes Adventistas
- Panel de Registro de inscripción de nuevos miembros
- Panel de Registro de iglesias
- Panel de Registro de inscripción de nueva iglesia
- Panel de Registro de eventos
- Registro de inscripción de nuevo club

- Panel de registro de Seguro
- Panel de solicitud de pago de seguro
- Registro publicación
- Formulario de solicitud de nueva publicación

Salidas de Usuario.

- Listado de usuarios
- Listado de iglesias
- Listado de eventos
- Reporte de miembros inscritos al Ministerio de jóvenes adventistas
- Listado de miembros de iglesia inscritos con seguros
- Listado de miembros de iglesia inscritos sin seguro
- Reporte miembros por sexo y edad
- Reporte de informes trimestrales
- Reporte de informes anuales
- Listado de eventos
- Listado de publicaciones
- Listado de noticias

Parámetro de Medición	Numero	Factor de ponderación			
		Simple	Medio	Complejo	
Nro de Entradas de Usuario	11 *	3	4	6	= 44
Nro de Salidas de Usuario	12 *	4	5	7	= 60
Nro de Peticiones de Usuario	6 *	3	4	6	= 18
Nro de archivos	100 *	3	10	15	= 1000
Nro de interfaces externos	1 *	1	1	1	= 1
Total					= 1023

Tabla 4.2 Cálculo de punto función

La determinación del valor de complejidad es subjetiva. A continuación detallamos los datos necesarios para ubicar los parámetros anteriormente mencionados.

Los factores de ajuste responde en el rango de (0 - 5) de acuerdo a la siguiente tabla:

Escala	Sin importancia	Incidental	Moderado	Medio	Significativo	Esencial
Factor (Fi)	0	1	2	3	4	5

Tabla 4.3 tabla de factor de ajuste

Valores de ajuste de Complejidad	Ponderación
1.- ¿Requiere el sistema copias de seguridad?	4
2.- ¿Se requieren comunicaciones de datos?	5
3.- ¿Existen funciones procesamiento distribuido?	2
4.- ¿Es crítico el rendimiento?	2
5.- ¿Será ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?	5
6.- ¿Requiere el Sistema entrada interactiva?	4
7.- ¿requiere la entrada de datos interactivo que las transacciones de entrada se lleven a cabo sobre múltiples pantallas y variadas operaciones?	3
8.- ¿Se actualizan los archivos maestros de forma interactiva?	5
9.- ¿Son complejas las entradas, las salidas, los archivos y peticiones?	4
10.- ¿Es complejo el procesamiento interno?	5
11.- ¿Se ha diseñado el código para ser reutilizable?	5
12.- ¿Están incluidas en el diseño la conversión y la instalación?	3
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones?	5
14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5

Tabla 4.4 Valores Ajuste de Complejidad

Por lo tanto:

$$F_i = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7 + F_8 + F_9 + F_{10} + F_{11} + F_{12} + F_{13} + F_{14}$$

$$F_i = 4 + 5 + 2 + 2 + 5 + 4 + 3 + 5 + 4 + 5 + 5 + 3 + 5 + 5 = 57$$

Posteriormente reemplazamos en la ecuación de PF y se tiene lo siguiente:

$$PF_{\text{real}} = \text{Cuenta Total} * [X + \text{Min}(Y) * F_i]$$

$$PF_{\text{real}} = 1023 * [0.65 + 0.01 * 57]$$

El punto de función real obtenido es = **1248.06**

La funcionalidad con un nivel de confianza de 100% será:

$$PF_{\text{esperado}} = 1023 * [0.65 + 0.01 * 70]$$

$$PF_{\text{esperado}} = 1381.05$$

Calculando el porcentaje de funcionalidad, se tiene:

$$\%PF = PF_{\text{real}} / PF_{\text{esperado}} = 1248.06 / 1381.05 = 0.90$$

Por lo tanto concluimos que el Sistema de Información tiene una funcionalidad del **90%**

4.2.2 Portabilidad

La portabilidad es el esfuerzo necesario para transferir el programa d un entorno de sistema hardware o software a otro El sistema tiene los atributos relacionados con la habilidad del software para ser transferido desde un entorno a otro.

A nivel de Software el “Sistema de Gestión de Actividades de Jóvenes – Misión Boliviana Occidental” es portable bajo los siguientes sistemas operativos: Windows 98, Windows 200, Windows XP. Además como el programa esta desarrollado con software libre, es compatible llevar de un entorno a otro, si quisiéramos llevar a un entorno Linux, debemos levantar los servicios de Linux, para trabajar con Apache, Mysql. Se debe copiar los archivos Php y HTML a la carpeta del servidor de páginas Web Apache.

A nivel Hardware el Sistema es portable bajo las siguientes características: Pentium IV, Memoria RAM de 256 MB.

4.2.3 Mantenibilidad

Existen métricas diseñadas para actividades de mantenimiento de software. El estándar IEEE 982 sugiere un índice de madurez del software que proporciona una estabilidad del producto. Se determina la siguiente información:

MT: Número de módulos de la versión

FC: Número de módulos de la versión actual que se han cambiado.

Fa: Número de módulos de la versión actual que se han dañado.

Fd: Número de módulos de la versión anterior que se han borrado con la versión actual

Una vez que se cuenta con la información mencionada se calcula el índice de madurez del software de la siguiente manera:

$$IMS = [MT - (Fa + FC + Fd)] / MT$$

Entonces tendremos:

$$IMS = [12 - (0 + 2 + 0)] / 12$$

$$IMS = 0.83\%$$

4.2.4 Usabilidad

Existen diferentes métodos para medir la usabilidad, haciendo estudios con usuarios finales, de manera que para evaluar el se utilizo un cuestionario, donde los mismos respondieron asignando una ponderación, como se observa en la siguiente tabla.

La Función de usabilidad se calcula de la siguiente manera:

$$FU = [(SUM x_i / n)*100]/n$$

Preguntas	Pésimo (1)
1.- ¿El sistema satisface los requerimientos de manejo de información?	4
2.- ¿Las salidas del sistema están de acuerdo a sus requerimientos?	4
3.- ¿Es fácil de aprender para el usuario el sistema?	3
4.- ¿El interfaz del sistema es agradable?	3
5.- ¿El sistema facilita el trabajo que realiza?	5
6.- ¿el sistema muestra formas de mensajes que ayuden al usuario?	3
Total	22

Tabla 4.5 Ponderación de Usabilidad

Por lo tanto tenemos:

$$FU = [(22/ 5)*100]/5$$

$$FU = 88\%$$

2.2.5 Confiabilidad

La confiabilidad del sistema se define en términos estadísticos, es la probabilidad de operación de fallos del sistema en un entorno determinado en un lapso de tiempo. Siendo la confiabilidad del sistema es directamente proporcional a la confiabilidad de sus componentes.

De un total de 20 pruebas realizadas a los diferentes módulos del software se obtuvo el siguiente cuadro de fallas, como se muestra tabla 4.5

Modulo	Numero de fallas
Administración de usuarios	2
Modulo de inscripción de miembros	1
Modulo de inscripción,	2

reservas o confirmación de eventos	
Pago de seguros	0
Publicaciones	0
Reportes y sitio Web	0
Total	5

Tabla 4.6 Fallas por módulos

Como se puede observar en el gráfico de 20 pruebas realizadas a los distintos módulos, se produjo 5 fallas, lo que significa esto se puede apreciar en la siguiente relación de confiabilidad:

Confiabilidad = $(5 / 20) * 100\% = 25\%$ de fallos, gracias a este resultado se puede interpretar que existe un 75% de confiabilidad o fiabilidad.

4.3 ANÁLISIS DE COSTOS

Para determinar el costo total del proyecto se tomará en cuenta los siguientes costos:

- Costo del Software desarrollado.
- Costo de implementación del Sistema. Conformado por: costo de software propietario y costo de hardware.
- Costo de elaboración del proyecto.

4.3.1 Costo del Software desarrollado

Para la determinación del costo del Software desarrollado, se hará uso del Modelo Constructivo de Coste CÓCONO II, orientado a los Puntos de Función. La metodología de este modelo se encuentra explicada en el capítulo 2.

- **Conversión de los Puntos de Función a KLDC**

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

Las formulas serán las siguientes:

$$E = \text{Esfuerzo} = a \text{ KLDC } e * \text{FAE (persona x mes)}$$

$$T = \text{Tiempo de duración del desarrollo} = c \text{ Esfuerzo } d \text{ (meses)}$$

$$P = \text{Personal} = E/T \text{ (personas)}$$

Para calcular el Esfuerzo necesitamos hallar la variable KDLC (Kilo líneas de código), donde los PF son **1248.06** (dato obtenido de nuestra base de datos) y las líneas por cada PF equivalen a 29 según vemos en la tabla que se ilustra a continuación:

LENGUAJE	LDC/PF
Ensamblador	320
C	150
COBOL	105
Pascal	91
Prolog/LISP	64
C++	64
Visual Basic	32
SQL	12

Tabla 4.7 Conversión de Puntos de Función a KLDC [JONES96]

Por lo tanto tendremos:

$$KLDC = (PF * \text{Líneas de código por cada PF})/1000 = (1248.06 * 12)/1000 = 14.09 \text{ KLDC}$$

En nuestro caso el tipo orgánico será el más apropiado ya que el número de líneas de código no supera los 50 KLDC, por consiguiente, los coeficientes que usaremos serán las siguientes:

Tipo de proyecto	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Tabla 4.8 Coeficientes COCOMO

- **Cálculo del esfuerzo del desarrollo:**

Los coeficientes se obtienen de la tabla 4.7, sobre proyectos de los cuales no se tiene experiencia, por lo tanto la ecuación queda de la siguiente manera:

$$E = aKLDC^b$$

$$E = 2.4 * 14.09^{1.05}$$

$$E = 38 \text{ personas/mes}$$

- **Cálculo tiempo de desarrollo:**

$$T = c \text{ Esfuerzo } d$$

$$T = 2,5 * (38)^{0,38}$$

$$T = 9.96 \text{ meses}$$

Personal promedio:

$$P = E/T$$

$$P = 38/9.96 = 3.81 \text{ personas}$$

Según estas cifras serán necesarias cuatro personas trabajando alrededor de 10 meses y medio, para concluir el trabajo

Costo: Suponiendo que al personal se les pague un promedio de 200 \$us / mes y se trabaje los 10 meses y medio entonces el costo del proyecto será:

$$\text{COSTO} = 100 * 9.96 * 4 \text{ (salario * meses de trabajo * numero de personas)}$$

$$\text{COSTO} = 3984 \text{ $us.}$$

4.3.2 Costo de elaboración del proyecto

Los costos de elaboración del proyecto se refieren a los costos del estudio del sistema, en la etapa de recopilación y análisis principalmente, estos costos se presentan en la siguiente tabla 4.8.

Detalle	Importe (\$us)
Análisis y diseño del proyecto	50
Bibliografía	20
Material de Escritorio	10
Internet	10
Otros	10
TOTAL 100	

Tabla 4.9 Costo de elaboración del proyecto

Costo Total

El costo total es la sumatoria del costo del Software desarrollado, costo de implementación del Sistema y costo de elaboración del proyecto, detallados en la siguiente tabla 4.9.

Detalle	Importe (\$us)
Costo del Software desarrollado	3984
Costo de elaboración del proyecto	100
Costo hardware y software	900
Total	4984

Tabla 4.10 Costo Total



CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CAPITULO N° V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Al haber concluido el presente proyecto de grado “Sistema de Gestión de actividades de jóvenes – Misión Boliviana Occidental” llegamos a las siguientes conclusiones:

- a) El “Sistema de Gestión de actividades de jóvenes – Misión Boliviana Occidental” ha sido concluida satisfactoriamente, con lo que se logró automatizar el manejo y la gestión de la información de una manera rápida y eficiente de atención a los distintos usuarios. realizando por ejemplo el registro de nuevos miembros del Ministerio Joven de la M.B.O., como también el seguimiento de los eventos y las actividades que realizan en el Ministerio Joven de la M.B.O.
- b) Este sistema desarrollado ayuda a realizar el registro de nuevos miembros del Ministerio Joven de la M.B.O., registro de nuevas iglesias, así como el seguimiento de los eventos y las actividades que realizan en el Ministerio Joven de la M.B.O. solucionando los problemas existentes haciendo de esta una herramienta que permite objetividad y transparencia en la toma de decisiones.
- c) Se implemento un sistema de información vía Web, para el control y administración del Ministerio Joven de la Misión Boliviana Occidental que administra a la Iglesia Adventista del Séptimo Día del occidente Boliviano.
- d) La aplicación de TIC' S, Programación Orientada a Objetos, arquitectura tres capas Modelo Vista Controlador (MVC), se logro optimizar el presente trabajo, como ser la gestión de archivos, hace que cumpla con las métricas de calidad, haciendo el sistema cumpla con la mantenibilidad y flexibilidad.

- e) Se logró el cumplimiento tanto del objetivo principal como también los objetivos específicos planteados inicialmente.
- f) Proporcionar información oportuna, es lo que se logro con la implementación de este sistema, el problema de falta o perdida de información será casi nulo, permitiendo tener toda la información en el instante preciso ya que se establecieron mecanismos de seguridad para precautelar el acceso a la información que brinda el sistema, estos mecanismos son:
- La autenticación utilizando nombres de usuario y contraseñas.
 - Disponibilidad utilizando sesiones
 - La confidencialidad utilizando roles para los usuarios.
- g) El sistema se encarga de generar reportes de forma automática permitiendo optimizar el tiempo de respuesta de la solicitud de los usuarios, incluso disminuyendo las horas de trabajo.
- h) Se puede observar que este sistema es de reutilización, ya que si se deseara aumentar en un futuro algunas mejoras o reacondicionar a sistema de gestión de información tranquilamente se podrá adaptar.
- i) El sistema tiene una interfaz amigable reduciendo el tiempo de respuesta para las consultas, además de ser un sistema multiplataforma.

5.2 RECOMENDACIONES

Las recomendaciones que se pueden dar partir de la utilización del sistema son las siguientes:

- a) Se recomienda establecer mecanismos y controles de autoprotección al servidor contra algún posible virus, debido a que el sistema envía información al disco duro del servidor.
- b) Se recomienda realizar copias de seguridad de la base de datos periódicamente, para evitar perdida de la información.

- c) Para las futuras actualizaciones del software se recomienda se siga usando programación orientada a objetos, ya que el uso de esta metodología optimiza el desarrollo del software, ayudado del framework JQuery y el Modelo Vista Controlador ya que ayuda a la reutilización de componentes.
- d) Se recomienda a los futuros proyectos el uso de estos métodos, pues los beneficios son enormes así podemos nombrarlos: Optimización del tiempo, flexibilidad de cambio, reutilización de componentes, que conducen a la construcción de software sea de calidad.
- e) También se recomienda que se pueda adicionar un modulo de control financiero, que es muy importante, con el fin de centralizar toda la información existente en la organización.





BIBLIOGRAFÍA

BIBLIOGRAFÍA

[Larman, 1999] Larman G. 1999, Uml Y Patrones, primera edición, editorial Prentice-Hall, México

[Pressman, 2003] Ingeniería del software un enfoque practico, editorial McGraw-Hill/Interamericana de España, S.A.U. Madrid.

[Jacobson I, Rumbaugh J., Booh G., 2001] Unified Modelling Language, 3ª edición Adison-Wesley

[Senn, 1992] Análisis y diseño de sistemas de información. Segunda edición McGraw-Hill, México

[Garcia, 2005] Rational Unified Process (RUP). E.E.U.U: Quinta Edición.

[Ibarra A., 2004] Rational Unified Process. Chile: Cuarta Edición.

[Grady Booch I. ,2000] El Proceso Unificado de Desarrollo de Software, primera edición Pearson Educación, España.

[Vernazza L., 1996] Estado del Arte del desarrollo sistemático de Interfaces Gráficas de usuario, México.

[Taha Hamdy A., 1997] Investigación de operaciones, 6ta edición, Prentice-Hall, México.

[Roberto Hernández Sampieri, Dr. Carlos Fernández Callado, Dra. Pilar Baptista Lucio, 1991] Metodología de la investigación, Editorial McGraw-Hill.

[Somerville, I, 2002] Ingeniería de software: Un enfoque práctico, Quinta Edición. Editorial McGraw – Hill.

WEB

Sánchez, A. V., Saorín, P.T. 2002: Comunidades virtuales y portales de ocio y entretenimiento:

<http://qt1.edu.um.es:8080/PORTALES-cvocio.pdf>

Portillo, L., Millán c., 2002 Portales académicos en Internet:

<http://www.udo.edu.ve/botonesexternos/Eventos/vicoloquio/>

Gestión de la tecnología. 2006: ciclo de vida. Madrid: Universidad Politécnica de Madrid:

<http://www.getec.etsit.upm.es/docencia/gtecnologia/transferencia/inmaduras.htm>

OOHDM (Object Oriented Hypermedia Design Method): Chile:

<http://www.geocities.com/cocheli99/hipermedi/http>.

Conceptos de diagramas de estados, secuencia, colaboración, etc. Bolivia:

http://www.cs.ualberta.ca/~pfiguero/soo/uml/casos_uso01.html

Modelo Vista Controlador, Wikipedia:

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Using the MVC design pattern, Adobe Systems 2007, Live Docs:

<http://www.adobe.com/support/documentation/en/flash/>



ANEXOS

ANEXO A

MATRIZ DEL MARCO LÓGICO

TEMA: GESTIÓN DE ACTIVIDADES DE JÓVENES - MISIÓN BOLIVIANA OCCIDENTAL

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p>FIN:</p> <p>Mejorar la gestión de actividades del Ministerio de Jóvenes Adventistas., reduciendo el trabajo manual que se realiza actualmente. Optimizando los procesos administrativos y de comunicación de información.</p>	<ul style="list-style-type: none"> ➤ El sistema automatizara el trabajo manual, mejorando los procesos de gestión, comunicación de información acortando distancias geográficas. ➤ El uso del sistema cumplirá con las metas trazadas, pues el acceso al sistema sobre pasa en un 80% del total de los interesados. ➤ Se disminuirá en un 50% el trabajo del personal administrativo en tiempo y esfuerzo. 	<ul style="list-style-type: none"> ➤ Entrevistas y encuestas a los miembros del Ministerio de Jóvenes Adventistas., administrativos de la M.B.O., padres de familia sobre el impacto del sistema implantado. 	<ul style="list-style-type: none"> ➤ No exista resistencia al cambio y al uso de las nuevas tecnologías ➤ Existirá un crecimiento exponencial de los miembros del Ministerio de Jóvenes Adventistas..

<p>PROPÓSITO:</p> <p>Desarrollar e implementar el Sistema de Gestión de Actividades de Jóvenes para la Misión Boliviana Occidental.</p>	<ul style="list-style-type: none"> ➤ Los miembros del Ministerio de Jóvenes Adventistas estarán inscritos en un 80 % del total de miembros. ➤ Un 85% de los miembros del Ministerio de Jóvenes Adventistas. estarán correctamente informados de las actividades, reuniones, noticias y proyectos programados. ➤ La cantidad de reclamos será menor al 30%. ➤ Los miembros del Ministerio de Jóvenes Adventistas. será de un 80% después de la implementación del sistema. 	<ul style="list-style-type: none"> ➤ Inspección de listas ➤ Mayor participación de los Miembros del Ministerio de Jóvenes Adventistas. a las actividades programadas ➤ Encuestas a los administradores. ➤ Inspección de la página Web en línea. 	<ul style="list-style-type: none"> ➤ La M.B.O. continué con sus actividades. ➤ Que el sistema haya sido aceptado por todos los usuarios
---	---	---	---

<p>COMPONENTE:</p> <ul style="list-style-type: none"> ➤ Subsistema de inscripciones ➤ Subsistema de pre-inscripciones a las actividades programadas ➤ Diseño del portal Web para el Ministerio de Jóvenes Adventistas. M.B.O. ➤ Subsistema de PRE-inscripciones a las actividades planificadas. 	<ul style="list-style-type: none"> ➤ Sistema de Gestión de actividades de jóvenes Misión Boliviana Occidental funcionará en óptimas condiciones después de un mes de su implantación. ➤ Cumplimiento de cronograma en un 90% ➤ Documentación u manuales del sistema vía Web en dos semanas después de terminar el proyecto. 	<ul style="list-style-type: none"> ➤ Acta de entrega ➤ Manuales ➤ Listado de resultados de prueba de los subsistemas ➤ Portal Web en línea 	<ul style="list-style-type: none"> ➤ Asignación del presupuesto de forma oportuna ➤ Observaciones del sistema o prototipos con una evaluación de requerimientos oportuno ➤ Navegar en el portal Web del Ministerio de Jóvenes Adventistas. M.B.O.
<p>ACTIVIDADES:</p> <ul style="list-style-type: none"> ➤ Ingeniería de requerimientos ➤ Análisis del sistema ➤ Diseño ➤ Implementación ➤ Pruebas 	<ul style="list-style-type: none"> ➤ Costos ➤ 1 analista de sistemas sueldo por un mes 2000 Bs. ➤ 2 desarrolladores por cuatro meses sueldo total de 19200 Bs., sueldo por 	<ul style="list-style-type: none"> ➤ Seguimiento del cronograma. ➤ Seguimiento de actividades por parte de: <ul style="list-style-type: none"> ○ Desarrollador del proyecto ○ Tutor del proyecto 	<ul style="list-style-type: none"> ➤ Se cuenta con los medios de investigación para la elaboración del proyecto ➤ Se cuenta con el suficiente material de trabajo para el diseño e implementación de los procesos básicos del

<p>➤ Implantación</p>	<p>desarrolladores 2400 Bs./mes.</p> <p>➤ Una computadora central que funciones como Servidor de Páginas Web. Pentium 4 de 3Ghz, 1 Gbts de memoria 4800 Bs.</p> <p>➤ Material de escritorio 200 Bs.</p> <p>TOTAL 26200 Bs.</p>	<ul style="list-style-type: none"> ○ Revisor del proyecto ○ Tesorero de la M.B.O. <p>➤ Documentación y diseño del sistema redactados en proyecto para la carrera de informática entregados antes de la defensa.</p> <p>➤ Recibos de pago de sueldos</p> <p>➤ Recibo de compra de material de escritorio</p> <p>➤ Recibo de compra de equipo de computación.</p>	<p>sistema</p> <p>➤ Se cuenta con el dominio propio para la publicación de la página en Internet.</p>
-----------------------	--	---	---

ANEXO B

DIAGRAMA DE ACTIVIDADES

- Diagrama de actividades “INSCRIPCIÓN DE NUEVA IGLESIA”

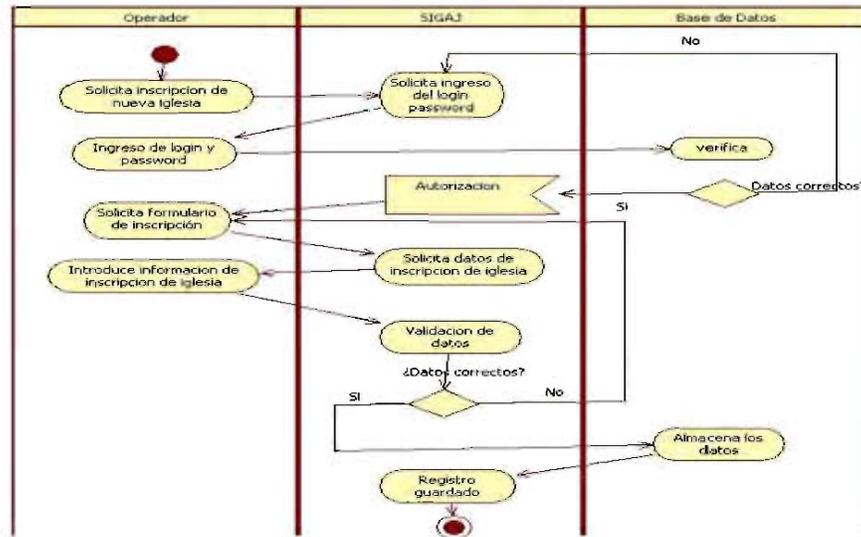


Diagrama de actividades: Inscripción de nueva Iglesia
Fuente: Elaboración Propia

- Diagrama de casos de uso expandidos “PUBLICACIÓN”

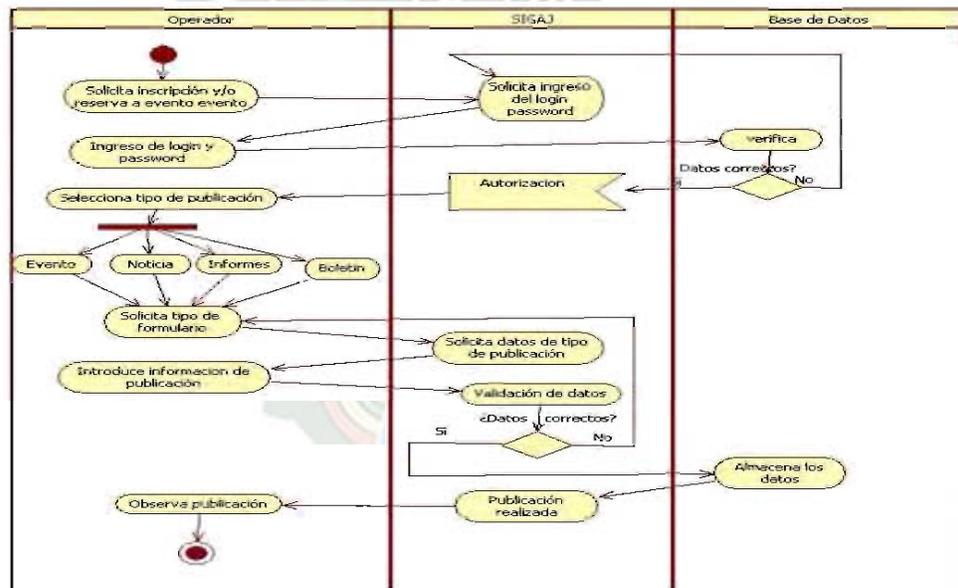


Diagrama de actividades: Publicaciones
Fuente: Elaboración Propia

- Diagrama de caso de actividades “REPORTES”

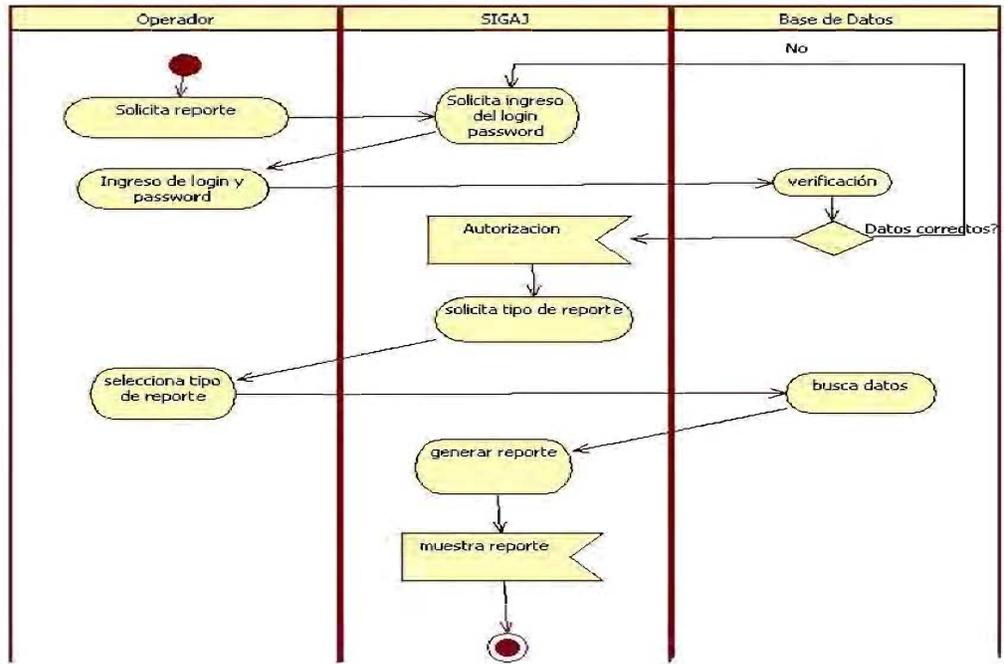


Diagrama de actividades: Reportes.

Fuente: Elaboración Propia

ANEXO C

DIAGRAMAS DE CASOS DE USO

➤ Caso de uso “INSCRIPCIÓN DE NUEVO DE NUEVA IGLESIA”

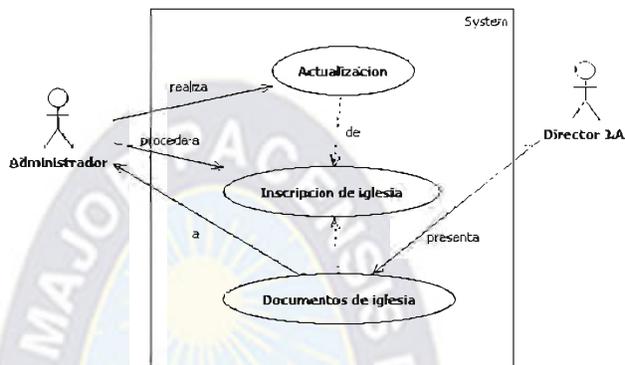


Diagrama de caso de uso: Inscripción de nueva iglesia

Fuente: Elaboración Propia

• Caso de uso de alto nivel: “INSCRIPCIÓN DE NUEVO MIEMBRO”

Caso de uso:	Inscripción de nuevo Iglesia
Actores:	Administrador, Director de J.A.
Tipo:	Primario
Descripción:	El administrador inscribe a una nueva iglesia al ministerio joven con la documentación presentada por e Director de J.A. de la iglesia que desea ser inscrita.

• Diagrama de caso de uso expandido “INSCRIPCIÓN DE NUEVA IGLESIA”

Caso de uso	<i>Inscripción de nuevo miembro</i>
Actores	Administrador, Director de J.A.
Propósito	Registro e inscripción de nueva iglesia al Ministerio Joven.
Resumen	El administrador o usuario ingresa al sistema y selecciona la opción de inscripción de nueva iglesia, el sistema muestra formulario de inscripción de nueva iglesia y luego el administrador procede al llenado de datos para luego validarlos Una vez validado se podrá registrar e inscribir a un nueva iglesia guardando los datos de iglesia introducidos y validados.
Tipo	Primario
Referencias Cruzadas	R2, R8, R9, R10, R11, R12, R13

- **Curso normal de INSCRIPCIÓN DE NUEVA IGLESIA**

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresa al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- ingresa los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de registro de nueva iglesia	6.- Despliega la opción de inscripción de nueva iglesia mostrando el formulario de nuevo registro.
	7.- Introduce los datos de: nombre completo de iglesia, dirección, edad, e-mail, fecha de nacimiento, nombre de pastor y distrito al cual pertenece.	8.- Valida los campos introducidos, para almacenar los datos, y retorna al formulario de registro de iglesia.
Curso alternativo de eventos	9.- El usuario introduce datos erróneos en el momento de ingreso al sistema.	10.- El sistema despliega nuevamente el formulario de login y password.
	11.- El administrador introduce datos erróneos.	12.- El sistema despliega un mensaje de error y manda al paso 6.

- **Contratos de operaciones: “INSCRIPCIÓN DE NUEVA IGLESIA”**

Contrato:	Inscripción Nuevas Iglesias
Nombre:	Inscripción de Nuevas Iglesias
Responsabilidades	Administrador, Director de J.A., Ingresa con contraseña tipo de usuario, el sistema valida tipo código y tipo de usuario, si son válidos el sistema despliega el menú de opciones.
Tipo:	Sistema
Referencias	R2, R8, R9, R10, R11, R12, R13
Notas	Realiza la búsqueda de registros en la Base de Datos según tipo de usuario.
Excepciones	Si la introducción de datos de la nueva iglesia no es correcta el sistema devuelve un mensaje de datos erróneos.
Salida	Registro de nueva iglesia aceptado y almacenado.
Precondición	Debe presentar la documentación correspondiente para su registro hasta una fecha especificada.
Post-condición	Caso de no presentar la documentación requerida el sistema eliminara los registros.

➤ **Diagrama de casos de uso: “EMISIÓN DE PUBLICACIÓN”**

En la figura 3.15 se muestra el diagrama de caso de uso para emisión de publicación.

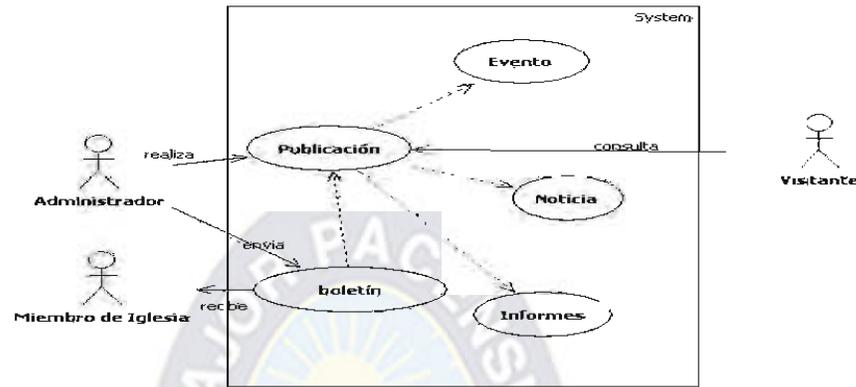


Figura 3.15 Figura Diagrama de caso de uso: Emisión de publicación

Fuente: Elaboración Propia

• **Caso de uso de alto nivel: “EMISIÓN DE PUBLICACIÓN”**

Caso de uso:	Emisión de publicación
Actores:	Administrador
Tipo:	Primario
Descripción:	El administrador del sistema actualiza y publica boletines, Informes, noticias, eventos y/o actividades realizados y organizados por la M.B.O.

• **Diagrama de caso de uso expandido “EMISIÓN DE PUBLICACIÓN”**

Nombre	Emisión de publicación
Actores	Administrador, Miembro de iglesia, visitante.
Propósito	El administrador del sistema actualiza y publica boletines, Informes, noticias, eventos y/o actividades realizados y organizados por la M.B.O.
Resumen	El administrador, a través del sistema introduce nuevos datos actualizados, generando nuevas publicaciones como ser: nuevos eventos y/o actividades, noticias respecto a las distintas iglesias, reuniones, nuevos miembros, boletines y informes sobre las actividades y eventos realizados.
Tipo	Primario
Referencias Cruzadas	R5, R6, R8, R9, R12

- **Curso normal de eventos “EMISIÓN DE PUBLICACIÓN”**

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresa al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- ingresa los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de publicaciones y tipo de publicación y elige tipo de formulario para llenarlo.	6.- Despliega el tipo de formulario de la opción elegida.
	7.- Llena el formulario y lo envía al sistema	8.-El sistema recibe los datos y los publica.
	10.- Observa la publicación desplegada.	9.- el sistema despliega las publicaciones actualizadas.
Curso alternativo de eventos	11.- El usuario introduce datos erróneos del al ingresar al sistema.	12.- El sistema despliega retorna al paso 1.
	13.- El usuario introduce datos erróneos en el formulario de publicación.	12.- El sistema despliega retorna al paso 6.

- **Contratos de operaciones: “EMISIÓN DE PUBLICACIÓN”**

Contrato: Emisión de publicación	
Nombre:	Emisión de publicación
Responsabilidades	El usuario Ingresa con contraseña tipo de usuario, el sistema valida tipo código y tipo de usuario, si son válidos el sistema permite realizar listado la inscripción según tipo de usuario.
Tipo:	Sistema
Referencias Cruzadas	R5, R6, R8, R9, R12
Notas	Realiza la búsqueda de tipo de publicación.
Excepciones	Si los datos son introducidos son erróneos no se podrá realizar la publicación.
Salida	Vista de las publicaciones deseadas por el usuario

➤ **Caso de uso de alto nivel: “GENERA REPORTES”**

En la figura 3.16 se muestra el diagrama de caso de uso para los reportes.

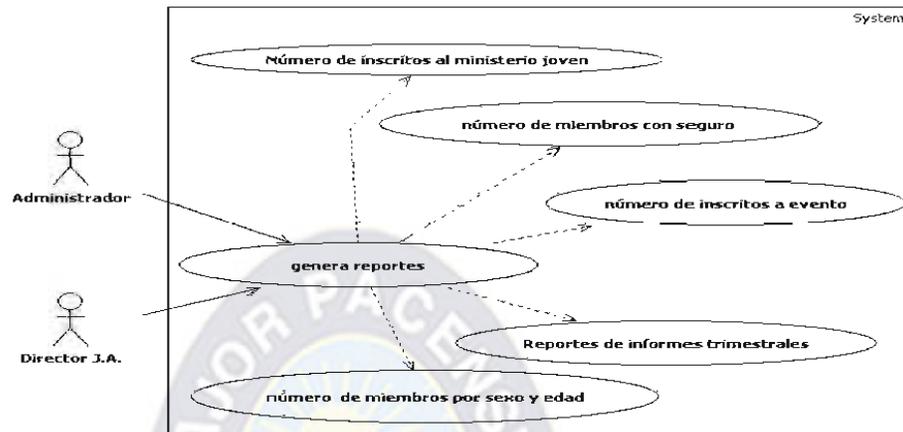


Figura 3.16 Figura Diagrama de caso de uso: Genera reportes

Fuente: Elaboración Propia

• **Caso de uso de alto nivel: “GENERA REPORTES”**

Caso de uso:	Pago de seguro
Actores:	Administrador, Director de J.A.
Tipo:	Primario
Descripción:	El administrador o director de J.A. elige el la opción reporte del menú y luego tipo de reporte, a lo que el sistema genera y lo muestra.

• **Diagrama de caso de uso expandido “GENERA REPORTES”**

Nombre	<i>Genera Reportes</i>
Actores	Administrador, Director de J.A.
Propósito	Presentar reportes actualizados gracias a los informes enviados por las distintas iglesias pertenecientes al Ministerio de Jóvenes Adventistas.
Resumen	El sistema recibe informes al igual que usa la base da datos para generar reportes como número de miembros inscritos al Ministerio joven, numero de de miembros por iglesias, por edades, por sexo, además de reportar número de inscritos a los eventos y/o actividades, numero de miembros que cuentan con seguro anual y/o por evento.
Tipo	Primario
Referencias Cruzadas	R6, R7, R8, R9, R12

- **Curso normal de ““GENERA REPORTES”**

	Acción de actores	Respuesta del sistema
Curso Normal de eventos	1.- Ingresar al sistema	2.- Solicita que ingrese el nombre de usuario y contraseña
	3.- Ingresar los datos de usuario y contraseña.	4.- Valida e inicia sesión de usuario, despliega el menú correspondiente al nivel de acceso que tiene el usuario.
	5.- Escoge el menú de reportes por trimestre o anuales y se elige el tipo de reporte.	6.-Despliega de los reportes e informes a través de listas y gráficos.
	7.- Elige la opción imprimir o guardar.	8.- Imprime
	9.- Elige la opción cerrar sesión.	10.- El sistema cierra sesión.
Curso alternativo de eventos	11.- El usuario busca incorrectamente.	12.- El sistema despliega un mensaje de error y manda al paso 1.

- **Contratos de operaciones: “GENERA REPORTES”**

Contrato: Genera reportes	
Nombre:	Genera reportes
Responsabilidades	Administrador ingresa con contraseña tipo de usuario, el sistema valida tipo código y tipo de usuario, si son válidos el sistema permite realizar listado la inscripción según tipo de usuario.
Tipo:	Sistema
Referencias Cruzadas	R6, R7, R8, R9, R12
Notas	Se selecciona el reporte que se quiere observar.
Excepciones	Si no existen datos actualizados, entonces se mostraran los reportes más recientes.
Salida	Vista del reporte seleccionado por el usuario.
Precondición	El usuario debe ser un administrador para poder ingresar a los reportes.
Post-condición	Despliegue de los reportes e informes.

ANEXO D

DIAGRAMAS DE ESTADOS

- Diagrama de estados, para el caso de uso inscripción de “NUEVA IGLESIA”.

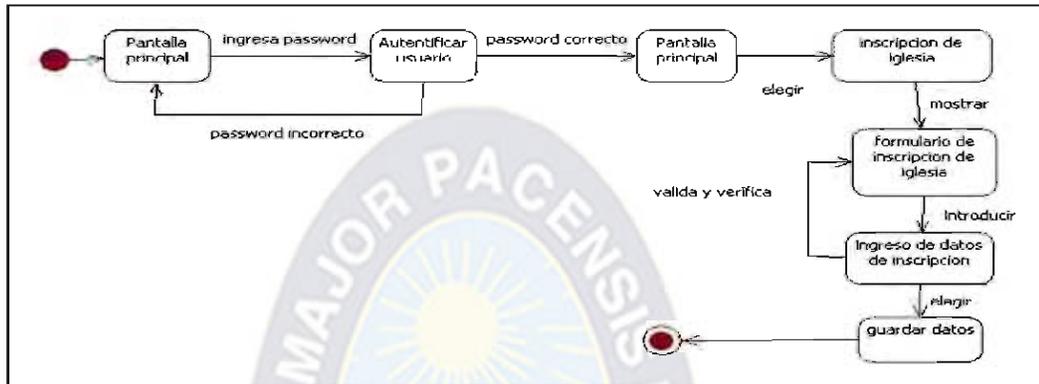


Diagrama de estados inscripción de nueva iglesia.

Fuente: Elaboración propia

- Diagrama de estados, para el caso de uso inscripción de “PUBLICACIÓN”.

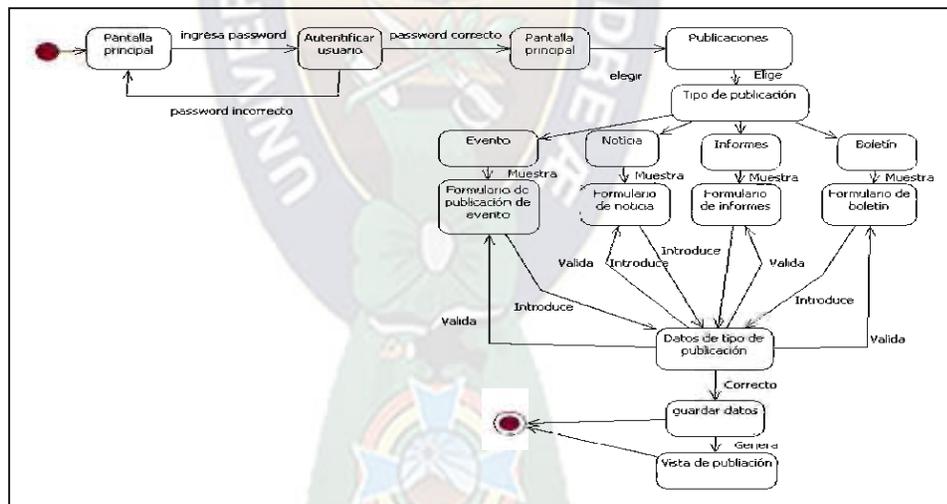


Diagrama de estados de Publicación.

Fuente: Elaboración propia

- Diagrama de estados, para el caso de uso de **“GENERA REPORTES”**.

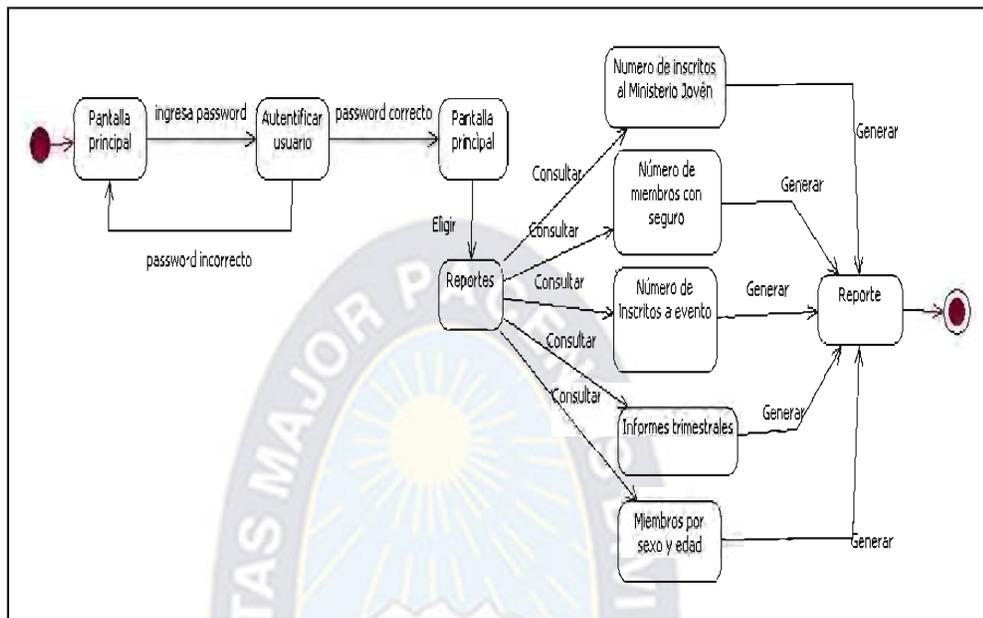


Diagrama de estados inscripción de Genera reportes.
Fuente: Elaboración propia

ANEXO E

DIAGRAMAS DE SECUENCIAS

- Diagrama de secuencia, para el caso de uso inscripción de “NUEVA IGLESIA”.

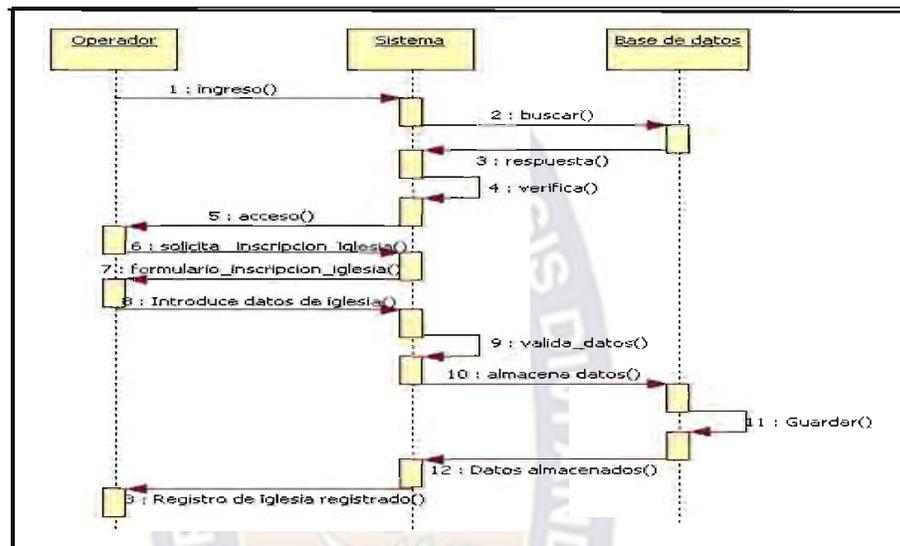


Diagrama de secuencia inscripción de nueva iglesia.

Fuente: Elaboración propia

- Diagrama de secuencia, para el caso de uso inscripción de “PUBLICACIÓN”.

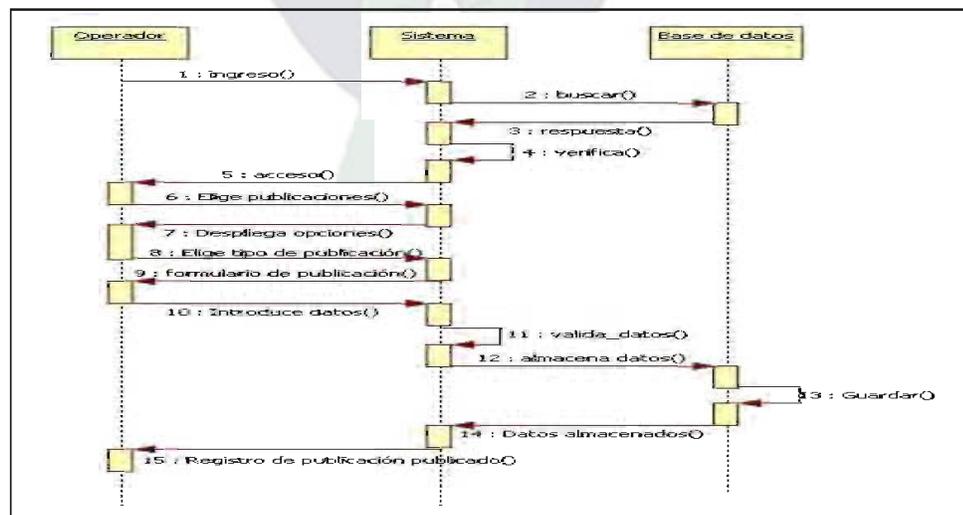


Diagrama de secuencia de Publicación.

Fuente: Elaboración propia

- Diagrama de secuencia, para el caso de uso de “**GENERA REPORTES**”

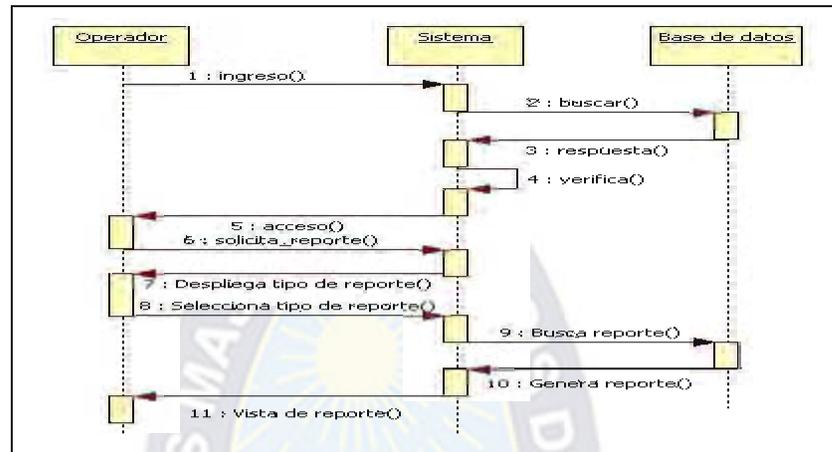


Figura 3.23 Diagrama de secuencia inscripción de Genera reportes.
Fuente: Elaboración propia

ANEXO F

DIAGRAMAS DE COLABORACIÓN

- Diagrama de colaboración, para el caso de uso inscripción de “NUEVA IGLESIA”.

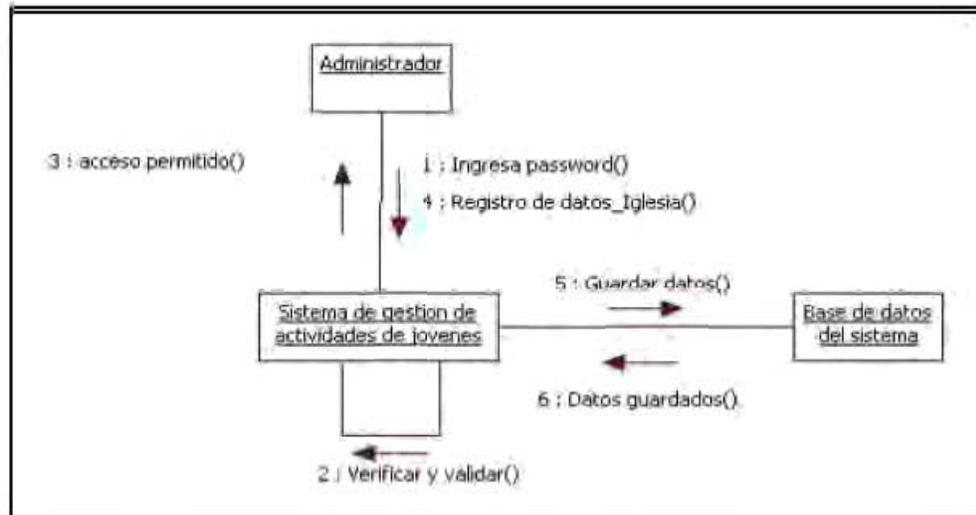


Diagrama de colaboración inscripción de nueva iglesia.
Fuente: Elaboración propia

- Diagrama de colaboración, para el caso de uso de “PUBLICACIONES”.

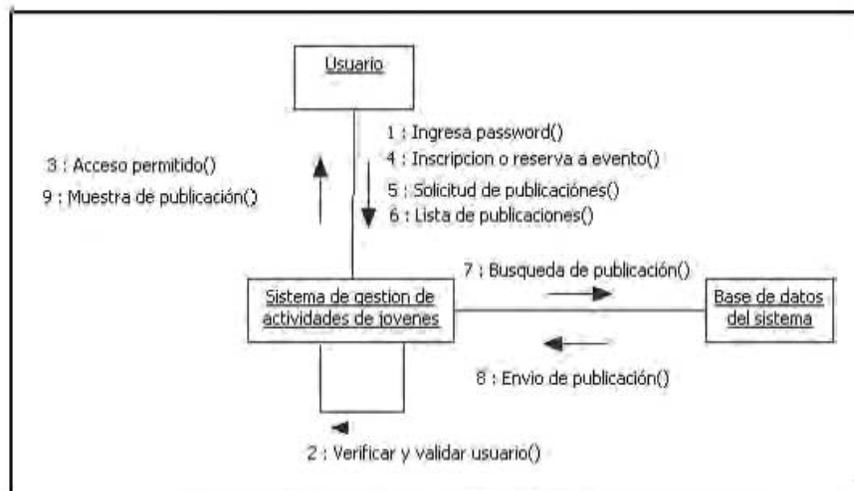


Diagrama de colaboración inscripción de nueva iglesia.
Fuente: Elaboración propia

➤ Diagrama de colaboración, para el caso de uso de "REPORTES".

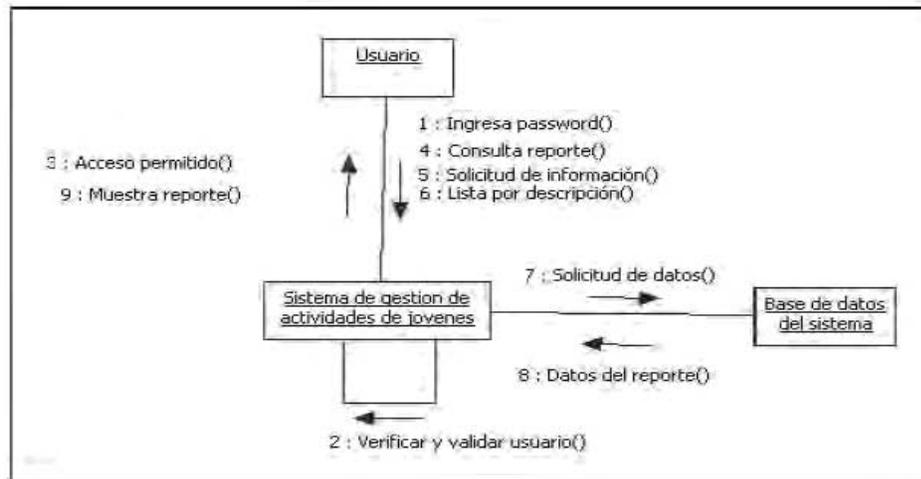


Diagrama de colaboración inscripción de nueva iglesia.
Fuente: Elaboración propia