

UNIVERSIDAD AMAZÓNICA DE PANDO
ÁREA DE CIENCIAS Y TECNOLOGÍA
CARRERA DE INGENIERÍA INFORMÁTICA



**“SISTEMA DE ADMINISTRACIÓN DE AFILIADOS Y RECURSOS
ECONÓMICOS DE LA IGLESIA BAUTISTA DE COBIJA”**

**TRABAJO DE TITULACIÓN PRESENTADO EN CONFORMIDAD A LOS REQUISITOS PARA OBTENER EL
TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS**

POSTULANTE: Univ. Reinaldo Rossell Michel
TUTOR: M.Sc. Ing. Freddy Morales Blanco
ASESOR: Lic. Eduardo Zubieta Copeticon

Cobija - Pando – Bolivia
2014

DEDICATORIA

A Dios, por darme la vida, la fortaleza, la salud y el amor para seguir siempre adelante sin decaer.

A mis padres por darme la vida por el apoyo por el ejemplo de perseverancia que me han mostrado en cada momento pero más que nada por su amor.

A mi hija Rosselly Stefany Rossell Cadario quien fue la encargada de darme aliento para seguir adelante luchando por una superación para finalmente obtener el objetivo propuesto, gracias mí niña linda. A Ingrid Cadario Gutiérrez por el apoyo incondicional que me ha dado en momentos difíciles en el trayecto de la vida.

A mis hermanos por el apoyo en todo momento difíciles con ellos aprendí que para todo hay que luchar.

A mis docentes por su gran apoyo y motivación para la culminación de este proyecto de grado.

A mis queridos amigos y compañeros de trabajo que de manera incondicional siempre han estado a mi lado ofreciéndome su apoyo, amor, confianza y aliento para seguir adelante, y quiero que sepan que siempre estarán en mi corazón.

A todas las personas que de alguna u otra forma contribuyeron conmigo, reciban este trabajo como suyo y sépanse acreedores de mi especial agradecimiento: Dios los bendiga.

AGRADECIMIENTOS

En primer lugar a Dios por haberme guiado; en segundo lugar a cada uno de los que son parte de mi familia; a mis hermanos y a todos mis amigos; por haberme dado su fuerza y apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora. A Ingrid Cadario por darme el apoyo incondicional en todo momento.

RESUMEN

Los sistemas de información (SI), han cambiado la forma de operar en las instituciones públicas y privadas, a través de su uso se llegan a obtener primordiales mejoras como la automatización de procesos, donde existe flujo de información y que estas sean confiables para la toma de decisiones.

En el presente proyecto de grado se desarrolló un Sistema de Información que permite administrar los afiliados y recursos económicos de la Iglesia Bautista de Cobija, también se incorporó el módulo de reporte de las asistencias de los afiliados y también de sus donaciones.

La mayor parte de las Iglesias Evangélicas no tienen un sistema que les ayude en la administración de sus afiliados y de sus recursos económicos.

En caso de la Iglesia Bautista de Cobija no contaba con un sistema de información adecuado que le automatice los registros de asistencia y donaciones que cumpla con los requerimientos que le exige la Convención Bautista Boliviana.

Bajo este argumento, se desarrolló un sistema de información que permita administrar los afiliados y recursos económicos de la Iglesia Bautista y al mismo tiempo agilizar y mejorar los procesos de la institución.

Para el cumplimiento del presente proyecto de grado se utilizó la metodología PU (Proceso Unificado), la misma que tiene cuatro etapas para la implementar un sistema.

Índice General

CAPITULO I

1.1.- INTRODUCCION.....	1
1.2 ANTECEDENTES	2
1.3.- DESCRIPCION DEL PROBLEMA:	2
1.4.- SOLUCION PROPUESTA	3
1.5.- OBJETIVOS Y ALCANCE DEL PROYECTO	4
1.5.1.- OBJETIVO GENERAL	4
1.5.2.- OBJETIVOS ESPECÍFICOS:.....	4
1.5.3 ALCANCES.....	4
1.6. METODOLOGIA Y HERRAMIENTAS UTILIZADAS	5
1.7. RESULTADOS OBTENIDOS	8
1.8.- ORGANIZACIÓN DEL DOCUMENTO	9

CAPITULO II

2.1 MARCO INSTITUCIONAL	10
2.2. MARCO LEGAL.....	14
2.3 MARCO TEORICO	14
2.3.1. SISTEMA DE ADMINISTRACIÓN DE IGLESIA	14
2.3.2. ADMINISTRACION DE LA IGLESIA	15
2.3.3. DESCRIPCION DEL SISTEMA ACTUAL	15
2.3.4. SISTEMA DE INFORMACION (SI).....	15
2.3.4.1 SISTEMAS PARA EL PROCESAMIENTO DE TRANSACCIONES	16
2.3.4.2 SISTEMA PARA EL SOPORTE DE DECISIONES	16
2.3.4.3 SISTEMA DE INFORMACIÓN ADMINISTRATIVA	17
2.3.5. BASES DE DATOS	18
2.3.5.1. MODELOS DE BASE DE DATOS.....	19
2.3.5.2. SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD)	19
2.3.5.3. SGBD MYSQL	20
2.3.6. PROCESO UNIFICADO	22
2.3.7. DIRIGIDO POR CASOS DE USO	23

2.3.8. CENTRADO EN LA ARQUITECTURA.....	24
2.3.9.1. LENGUAJE UNIFICADO DE MODELADO (UML).....	25
A). CASOS DE USO	25
B). DIAGRAMA DE CLASES.....	26
C). DIAGRAMA DE ESTADOS.....	26
D). DIAGRAMA DE SECUENCIA.....	27
E).MODELO DE OBJETOS.....	27
F).MODELO DINÁMICO.....	28
H).MODELO FUNCIONAL	29
2.3.10. FASES DE DESARROLLO DE LA METODOLOGÍA PROCESO UNIFICADO...	31
A) INICIO	31
B) ELABORACIÓN.	32
C) CONSTRUCCIÓN.....	32
D) TRANSICIÓN	32
2.3.10.1. HERRAMIENTAS DE DESARROLLO.....	32
2.3.10.2. LENGUAJE DE ETIQUETAS HTML.....	33
2.3.10.3. LENGUAJE DE PROGRAMACIÓN PHP5	33

CAPITULO III

3.1.- FASE DE INICIO.....	35
3.2. FASE DE ELABORACION.....	36
3.2.1. REQUERIMIENTOS.....	36
3.2.2. CASOS DE USO	39
3.3. FASE DE CONSTRUCCION	41
A) CONSTRUCCIÓN.....	41
B) DISEÑO DE LA BASE DE DATOS.....	44
3.4 FASE DE TRANSICIÓN	46
3.4 VALORACIÓN DE CALIDAD DEL SOFTWARE	50
3.4.2 CALIDAD DEL SISTEMA	50
FUNCIONABILIDAD.....	51
FIABILIDAD	52
USABILIDAD	53
EFICIENCIA.....	53
MANTENIMIENTO	54
PORTABILIDAD	55

CAPITULO IV

4.1. CONCLUSIONES.....	57
4.2. RECOMENDACIONES.....	58
BIBLIOGRAFIA	59

INDICES DE FIGURAS

FIGURA 2.1 Iglesia Bautista en General	12
FIGURA 2.2 Proceso de Desarrollo de Software	22
FIGURA 2.3 Historial de Procesos Unificados	23
FIGURA 2.4 Casos de uso que en laza los flujos de trabajo fundamental	24
FIGURA 2.5 Diagrama de Interacción de Objetos para un Modelo Estático.....	28
FIGURA 2.6 Diagrama de Interacción de Objetos con Ventana Gráfica.....	28
FIGURA 2.7. Diagrama de Transición de Estados	29
FIGURA 2.8: Forma básica de DFD.	30
FIGURA 2.9: Fase de la metodología	31
FIGURA 3.1 Función del sistema	35
FIGURA 3.2 Diagrama General	39
FIGURA 3.3 Diagrama casos de uso del Administrador en el Sistema	40
FIGURA 3.4 Diagrama casos de uso Cajero	40
FIGURA 3.5 Diagrama del Proceso	42
FIGURA 3.6 Añadir Afiliado.....	43
FIGURA 3.7 Eliminar Afiliado	44
FIGURA 3.8 Diseño de la base de Datos	45
FIGURA 3.9 Ingresos al Sistema	47
FIGURA 3.10 Menú principal del Sistema	47
FIGURA 3.11 Menú de Agregar Afiliado	48
FIGURA 3.12 Reporte de Asistencia a la Célula.....	48
FIGURA 3.13 Reporte de Ingreso	49
FIGURA 3.14 Afiliados Agregados	49
FIGURA 3.15 Registro de afiliados donantes	50

INDICE DE TABLAS

TABLA 3.1 Requerimientos Generales.....	36
TABLA 3.2 Requerimientos de Autenticación de Usuarios.....	37
TABLA 3.3 Requerimientos de Administrar parámetros del sistema.....	37
TABLA 3.4 Requerimientos de Administrar Personas que asiste a la congregación.....	37
TABLA 3.5 Requerimientos de Administrar Ingresos y Egresos	38
TABLA 3.6 Requerimientos de Reportes.....	38
TABLA 3.7 Descripción del caso de uso Cajero	41
TABLA 3.8 Escala de calificación para el Sistema	51
TABLA 3.9 Resultado de la medida Funcionabilidad	51
TABLA 3.10 Resultado de la medida Fiabilidad.....	52
TABLA 3.11 Resultado de la medida Usabilidad.....	53
TABLA 3.12 Resultado de la medida Eficiente	53
TABLA 3.13 Resultado de la medida Mantenibilidad.....	54
TABLA 3.14 Resultado de la medida Portabilidad	55
TABLA 3.15 Resultados generales de la medida estándar ISO/IEC 9126.....	56

CAPÍTULO I
INTRODUCCIÓN

1.1.- INTRODUCCION

Un sistema de información es el conjunto formal de procesos que, actúan sobre una colección de datos estructurada de acuerdo con las necesidades de una institución y debido al avance de la tecnología que se da en el mundo entero, hoy en día gran parte de las organizaciones e empresas públicas y privadas, han optado por mejorar el control de su organización automatizando sus procesos.

En la actualidad los sistemas de información están tendiendo a automatizarse, para estas permitan tomar decisiones oportunas, confiables y efectivas en cuanto a técnicas de planificación y administración con el fin de garantizar su éxito, limitar el riesgo y reducir costos y aumentar las ganancias. Debido a esta razón, nace la idea de automatizar las actividades cotidianas en las organizaciones; cabe mencionar el vertiginoso avance de las telecomunicaciones y el progreso que han experimentado las ciencias informáticas que obliga a entrar al moderno mundo de la tecnología, al ser competitivos y no quedarse relegados en las tareas que proporcionan beneficios.

En este sentido esta institución que es la IGLESIA BAUTISTA DE COBIJA, no quedando atrás en los pasos que da la tecnología situación que se ha hecho evidente en la forma como se ejecutan los procesos y funciones propias en esta institución. Por lo tanto, estos aspectos son importantes para la elaboración y diseño de sistemas de información, y así satisfacer los requerimientos de los usuarios y mejorar las tareas de registro de los afiliados y de los recursos económicos sabiendo que en la actualidad no cuenta con un sistema de información que le ayude en la administración de afiliados de y sus recursos económicos ya que ha comenzado a crecer en el número de persona como también en los ingresos llegando a sobre pasar a las 500 (quinientas) personas y teniendo un ingreso por semana de 14000 Bs. (Catorce mil Bolivianos). (Ver anexos E)

Considerando lo expuesto en el siguiente proyecto de grado, se propone la elaboración de un “Sistema de administración de afiliados y de los recursos económicos”, que servirá de apoyo a los que están a cargo de la institución la cual carece de información precisa y oportuna.

1.2 ANTECEDENTES

Hace 10 años la Iglesia Bautista de Cobija que comenzó con una forma nueva de evangelismo con el objetivo de crecimiento de la comunidad cristiana, dando como resultado un gran número de afiliados y aumentando de gran manera los recursos económicos llegando a tener 14000 Bs (Catorce mil Bolivianos) por semana.

Considerando la relación de este proyecto con otros similares se describe a continuación proyectos nacionales e internacionales.

- Sistema de registro de asistencia y de ingresos y egresos, se trata de un sistema que registra a los asistentes y su donación, que fue empleado en la Iglesia de Dios Pentecostal del Brasil (Manaos) en el año 1995 cuyos autores fue la misma institución (I.D.P.B).
- Sistema de registro de ingresos y egresos, se trata de un sistema que registra solamente los ingresos y egresos, que fue empleado en la Iglesia Universal de la ciudad de Rio Branco Brasil en el año 2005 cuyo autor fue José Dasilva.
- Sistema de control de asistencia es un sistema que registra a las personas que asiste a las reuniones de los domingos que fue empleado en la Iglesia 2ª Bautista de la Ciudad de Santa Cruz en el año 2008, cuyo autor fue la Empresa Tecnología de Sistemas a cargo del Ing. Raúl Rocha Vargas.

1.3.- DESCRIPCION DEL PROBLEMA:

La Iglesia Bautista de Cobija es una entidad privada controlada por una supervisión en la sede de Santa Cruz y está sujeta a cumplir sus reglamentos que se exige. Esta institución tiene la obligación de cumplir con registros de asistencias y financieros para su presentación en cada mes como indica la normativa de estas instituciones afiliadas.

En base a la recopilación de la información obtenida a través de entrevista con los involucrados, se pudo comprobar un conjunto de problemas. De acuerdo a esto se detectó como causa los

Siguientes problemas: Deficiente uso de medios informáticos en la administración de la información en apoyos a los procesos, inexistencia de una base de datos organizada de ingresos y egresos, elaboración manual de informes, inexistencia de una base de datos de control de asistencia de los afiliados.

Estos problemas considerados como causas ocasionan el siguiente problema principal:

“Deficiente administración de Afiliados y Recursos Económicos de la Iglesia Bautista de Cobija”.

Ocasionando como efecto los siguientes problemas: Demora y errores en la elaboración de reportes, deficiencia en la generación de la información oportuna y confiable, deficiente notificación de los ingresos y egresos, desconocimiento de los ingresos que se recibe a diario, deficiencia en la administración de los recursos, deficiente control de asistencia de los afiliados a las reuniones.

1.4.- SOLUCION PROPUESTA

Como existe una deficiente administración de afiliados y recursos económicos dentro de la Iglesia Bautista de Cobija y estas deficiencias existen porque no hay una lista actualizada de los afiliados, tampoco existe un control de los recursos económicos de cada uno de los afiliados. Por esta razón la solución propuesta es que se desarrolle e implemente un sistema de información que permita centralizar toda la información en un solo formato y que todos tengan acceso a la información necesaria. Lo que se pretende llegar utilizando el sistema son:

- Ayudar a mejorar el control de los afiliados que llegan y de los que se van.
- Que exista un registro detallado de todos los afiliados.
- Tener los recursos económicos actualizados.
- Realizar de forma automática la lista de los afiliados y cierre de gestión de los recursos económicos de manera automática.

1.5.- OBJETIVOS Y ALCANCE DEL PROYECTO

1.5.1.- Objetivo General

“Implementar un sistema de información utilizando la metodología Unified Process (Proceso Unificado) mejorando la administración de los afiliados y los recursos económicos para la toma de decisiones de la Iglesia Bautista de Cobija”.

1.5.2.- Objetivos Específicos:

- Realizar las especificaciones de análisis y diseño para el sistema de control de las personas que asisten a la celebración y de los ingresos y egresos en base a la metodología Unified Process (Proceso Unificado).
- Desarrollar el diseño del sistema para establecer el tamaño y los módulos necesarios.
- Desarrollar los módulos de diseño, para lograr la funcionalidad del sistema en una arquitectura física y un entorno de implementación.
- Aplicar pruebas de funcionamiento del sistema y evaluar la calidad del mismo aplicando las métricas de calidad de software determinadas en la norma ISO 9126.

1.5.3 Alcances

El sistema de administración de afiliados y de los recursos económicos (SAMMIB), alcanzará un manejo eficiente de la información en los siguientes procesos:

- Módulo de Administrar Parámetros del Sistema se van a generar el tipo de moneda, Tipos de Entrada, Tipos de Salida, Tipos de Documentos de comprobantes.

- Módulo de Administrar Personas que Asiste la Congregación. Usuarios del Sistema, Personas Habilitadas para tomar la asistencia, Listado de todos los asistentes.
- Módulo de Administración de Ingresos y Egresos. Tipo de ingreso, Tipo de Egresos, Cantidad de lo ingresado, Cantidad de lo egresado.
- Módulo Reportes de ingresos, egresos, asistencia. Busca a las personas, mostrar asistencia, Salidas Trimestrales o mensuales, Entradas Trimestrales o mensuales.

1.6. METODOLOGIA Y HERRAMIENTAS UTILIZADAS

La metodología para la implementación del sistema, está basada en el proceso unificado (P.U) y el lenguaje unificado de modelado (UML 2.0).

Esta metodología es un proceso de desarrollo de software, es un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema de software. Este marco de trabajo de proceso genérico que por especialista separa cada clase de sistema de software, para diferentes áreas de aplicación, tipo de organización, niveles de competencia, tamaño de proyectos. La cual se divide en cuatro etapas.

1.6.1. Fase de Inicio

Durante la fase de inicio se desarrolla una descripción del producto final, y se presenta el análisis del negocio. Esta fase responde las siguientes preguntas:

¿Cuáles son las principales funciones del sistema para los usuarios más importantes?

¿Cómo podría ser la mejor arquitectura del sistema?

¿Cuál es el plan del proyecto y cuánto costará desarrollar el producto?

En esta fase se identifican y priorizan los riesgos más importantes.

El objetivo de esta fase es ayudar al equipo de proyecto a decidir cuáles son los verdaderos objetivos del proyecto. Las iteraciones exploran diferentes soluciones posibles, y diferentes arquitecturas posibles.

1.6.2. Fase de Elaboración

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura.

Las iteraciones en la fase de elaboración:

- Establecen una firme comprensión del problema a solucionar.
- Establece la fundación arquitectural para el software.
- Establece un plan detallado para las siguientes iteraciones.
- Elimina los mayores riesgos.

El resultado de esta fase es la línea base de la arquitectura.

En esta fase se construyen típicamente los siguientes artefactos:

- El cuerpo básico del software en la forma de un prototipo arquitectural.
- Casos de prueba
- La mayoría de los casos de uso (80%) que describen la funcionalidad del sistema.

1.6.3. Fase de Construcción

Durante la fase de construcción se crea el producto. La línea base de la arquitectura crece hasta convertirse en el sistema completo.

Al final de esta fase, el producto contiene todos los casos de uso implementados, sin embargo puede que no esté libre de defectos.

Los artefactos producidos durante esta fase son:

- El sistema software

- Los casos de prueba
- Los manuales de usuario

La fase de construcción finaliza con el software acabado.

Esta fase termina cuando el equipo programador llega a un acuerdo sobre:

- El producto es estable para ser usado
- El producto provee alguna funcionalidad de valor
- Todas las partes están listas para comenzar la transición.

1.6.4. Fase de Transición

La fase de transición cubre el período durante el cual el producto se convierte en la versión de prueba.

Las iteraciones en esta fase continúan agregando características al software. Sin embargo las características se agregan a un sistema que el usuario se encuentra utilizando activamente.

Los artefactos construidos en esta fase son los mismos que en la fase de construcción. El equipo se encuentra ocupado fundamentalmente en corregir y extender la funcionalidad del sistema desarrollado en la fase anterior.

La fase de transición finaliza con el lanzamiento del Producto.

Este hito se alcanza cuando el equipo de desarrollo llegan a un acuerdo sobre:

- Se han alcanzado los objetivos fijados en la fase de Inicio.
- El usuario está satisfecho.

Para el desarrollo del sistema se utilizó las herramientas o tecnologías de desarrollo las cuales son: HTML, CSS, PHP5, MYSQL 5.0, APACHE 2.0

Estas herramientas permitieron estructurar la arquitectura del software como cliente/servidor, donde HTML, CSS, son parte del cliente, mientras que PHP5, MYSQL 5 y APACHE 2, forman parte del servidor.

1.7. RESULTADOS OBTENIDOS

Lo que se llegó a obtener con el cumplimiento del proyecto fueron:

- Se llegó a identificar los procesos que lleva la institución para administrar las entradas y salidas de los afiliados.
- Se diseñó la estructura estática y dinámica del sistema en base al análisis realizado.
- Se llegó a construir los módulos correspondientes al Sistema que responde a las funcionalidades requeridas por los involucrados.
- Se realizó la etapa de prueba durante ocho meses del Sistema en la Iglesia Bautista de Cobija antes del producto final.

Después de la implementación del sistema en la Iglesia Bautista de Cobija, todos los afiliados y recursos económicos en proceso de regularización en el Sistema (aspecto que no sucedía anteriormente), lo que se logró alcanzar con Sistema fueron los siguientes.

- Los afiliados ya se encuentran registrados en el sistema con sus respectivos códigos.
- Ya se realizó el primer cierre de gestión en base a los recursos económicos que habían en la Iglesia.
- El registro de nuevos afiliados y recursos económicos ya se realizan por medio del Sistema
- Cada movimiento de los afiliados o de los recursos económicos se realizan por medio del sistema
- La existencia de los afiliados y recursos económicos se va actualizando automáticamente conforme el movimiento que realiza la Iglesia.

El grado de satisfacción por parte de los destinatarios es de un 90 %, porque se resolvió la mayor parte del problema, la parte cómo manejar sus informes es la que aún tiene dificultades dentro de la Iglesia Bautista de Cobija.

1.8.- ORGANIZACIÓN DEL DOCUMENTO

El documento del presente proyecto se organiza en cuatro capítulos los cuales son.

- En el **capítulo I** se describe la parte introductoria al proyecto de grado.
- En el **capítulo II**, se describe toda la teoría que fundamenta el desarrollo y las herramientas aplicadas.
- En el **capítulo III**, se muestra el desarrollo del proyecto aplicando la metodología adoptada.
- En el **capítulo IV**, se encuentran las conclusiones y recomendaciones sobre el uso del Sistema.

CAPÍTULO II
MARCO INSTITUCIONAL
MARCO LEGAL
MARCO TEÓRICO

2.1 MARCO INSTITUCIONAL

La Iglesia Bautista fundada el 5 de mayo de 1979 en la ciudad de Cobija, teniendo como objetivo el crecimiento de la comunidad cristiana, ha optado nuevas formas de evangelismo en donde ha crecido de gran manera, resultando complicado la administración de los afiliados y de los recursos económicos, la Iglesia lleva sus registros de forma manual en fichas de asistencia que luego son guardados en un archivador de palanca y los registro económicos en hoja de Excel que son impresas mensualmente y son guardadas en un archivador de palanca.

Internamente, la Iglesia tiene una estructura jerárquica, subdividida en ministerios, y células. Podría decirse que hay diferentes tipos de personas en la iglesia: los asistentes, quienes regularmente van a la iglesia, los miembros Oficiales, quienes ya han sido bautizados en la iglesia y asisten constantemente, los Líderes, que son bautizados, han tomado el curso de discipulado practico y Liderazgo y además forman parte de un ministerio (de servicio), los Líderes cabeza de Ministerio y los Líderes de célula que son los directamente encargados de los demás Líderes de servicio en cada ministerio y cumplen con los requisitos de todo líder.

El Pastor, que es la cabeza principal tiene un sueldo, al igual que sus Asistentes Pastorales y Secretaria.

Los Ministerios están subdivididos en departamentos. Cada líder pertenecer a una célula y/o a un ministerio.

Las células tienen a cargo Líderes. Cada uno de ellos está a cargo de una célula semanales, a las que asisten personas nuevas (de las cuales hay que tomar el registro) o antiguas.

Los afiliados de la iglesia pueden acudir a dos cultos en la semana. El domingo en dos horarios distintos.

Cada semana, luego de terminarse una célula, un líder de célula, se encarga de recoger los datos de las personas nuevas en tarjetas impresas, que luego son entregadas a la secretaria para ser

Guardadas. El líder de célula, es el encargado de llamar a estas personas o de asignarlas a su propio auxiliar Líder y entregar informe de esas llamadas, puesto que es necesario mantener en contacto con los nuevos asistentes.

Cualquier persona, sea asistente, puede pedir una célula. Las células son realizadas en los hogares de quien los solicita (anfitrión). Para cada célula es asignado un líder. De esta persona necesita saberse, el nombre, la dirección y el teléfono.

Si es una persona recién convertida, se desea saber qué líder estaba encargado en su fecha de conversión de tomar sus datos. De este modo a ese líder se le asignara la responsabilidad del grupo familiar.

En cualquier otro caso, ya sea que la persona sea antigua y no tenga asignado un líder, o que el líder responsable no pueda tomar la célula, se le asignara a otro líder, que pueda tomarla. De ese momento en adelante, el líder asignado es responsable de la célula.

Una persona es miembro oficial de la iglesia el día que se bautiza. De estos bautismos y de las fechas no se lleva un registro. Para cada ocasión se entregan los certificados a cada persona.

Es indispensable para el sistema poder manejar todas las jerarquías establecidas dentro de la iglesia, a fin de permitir un pleno control de la misma, en cuanto registro de personas, su estado dentro de la iglesia, los cargos que tiene en los diferentes ministerios, o en la célula, si pertenece a alguna.

El Sistema, conforme a lo anteriormente descrito, debe mejorar la calidad de los registros de los datos de las personas, como su nombre, dirección, teléfono, email, y los ministerios células a las que pertenece. De acuerdo a esto debe permitir búsquedas de personas, por su estado o por su nombre, a la vez, debe proporcionar la generación de certificados correspondientes a los cambios de los estados de las personas.

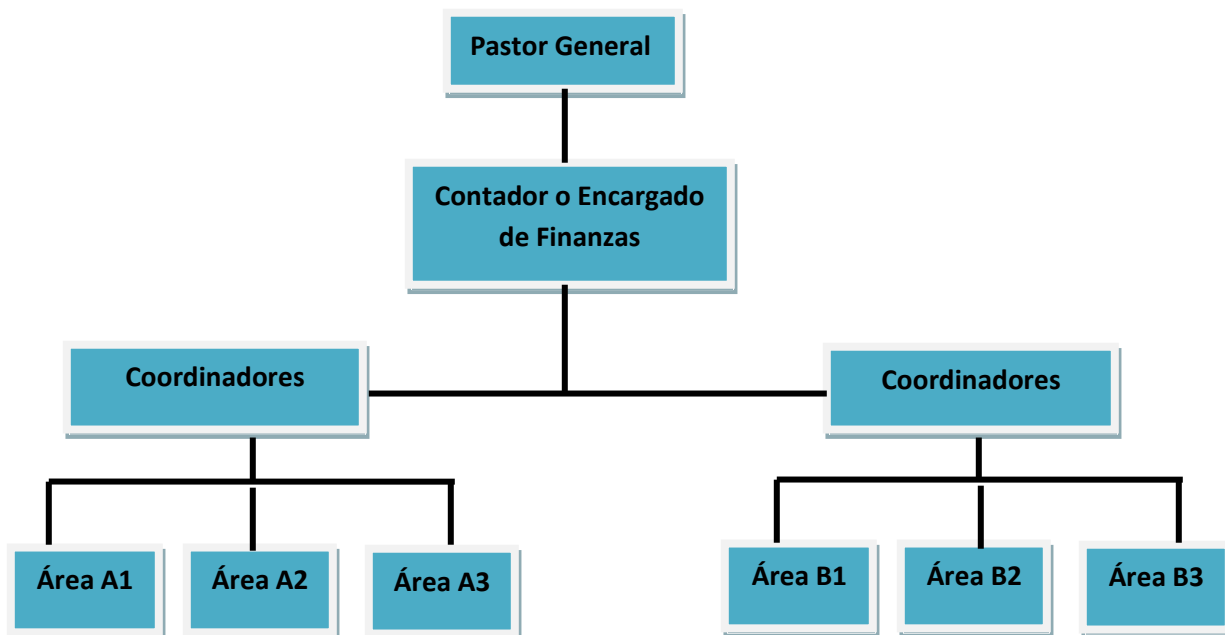
Además debe posibilitar el control de las células, sus tipos, y las personas que están registradas en esta, su función dentro de la misma, la cantidad de líderes que existen, el número de células que trabaja cada líder, y la información general de cada una de ellas, como nombre del anfitrión que la realiza, nombre del líder que la dirige, y el día de su realización (VER ANEXOS E).

De acuerdo al rol del usuario en el sistema, se debe autorizar el acceso a las células, y a las diferentes opciones para la célula como: agregar líderes, eliminar líderes, y acceder a búsquedas sobre esas personas.

Las entradas y salidas de dinero deben ser registradas por fecha, y debe llevarse cuenta de la última persona que hizo la modificación a fin de que sea más confiable el acceso. El Sistema debe permitir, la generación de un informe de tesorería por dos fechas.

De acuerdo a lo descrito organizacionalmente esta institución tiene una estructura la cual se describe a continuación.

FIGURA 2.1 Iglesia Bautista en General



Fuente: [Alejandro Rivero, Estatutos internos 2004]

Donde las funciones de cada uno de ellos explicaremos a continuación:

El pastor General es que toma todas las decisiones de la iglesia.

El encargado de finanzas o contador es el que recibe todos ingresos que se dan en las diferentes áreas y los procesas para dar los distintos egresos.

La función de los coordinadores es recibir las asistencias y los ingresos que se suscitan en las diferentes áreas y posteriormente pasar al contador los ingresos.

Las áreas constan de una secretaria donde se realizan las asistencias de cada afiliado y registró de los ingresos para posteriormente pasar al coordinador.

Los procesos más importantes que se realiza en esta institución son:

Líder

El líder es que recolecta la información de los nuevos afiliados y también de sus donaciones y los lleva secretaria para ser registrado.

Registros de nuevos afiliados

La secretaria es la encargada recibir los registros de los nuevos afiliados como también los registros financieros.

Registro económico

El contador es el que recibe los registros financieros luego de registrarlos y pasar al pastor.

Autorización económica

El pastor es el encargado de verificar los registros financieros y dar autorización para su uso.

Reporte

La secretaria es que lleva el control de nuevos afiliados y registra utilizando el Excel para después sacar los cuadros estadístico mensuales.

El contador es el encargado de sacar los registros de ingresos en Excel para después llevarlo al pastor general de la Iglesia.

2.2. MARCO LEGAL

Las instituciones públicas de Bolivia se rigen a la ley financiera de autonomía N. 455 que estipula que es lo que se debe o no debe gastar del presupuesto.

Los ministerios evangélicos son organizaciones privadas que no se rigen a esta ley de autonomía en Bolivia no existe aún una norma, ley o decreto que rijan a las iglesias como institución en la pública, pero como cada iglesia tiene su propia asociación nacional entonces ellas se rigen a los estatutos internos de la Convención Bautista Boliviana en la cual son afiliados, señala en el artículo IV párrafo quinto y sexto de forma clara “Que cada Iglesia Asociada a la convención debe presentar un informe mensual de los avances que se está haciendo en la congregación en cuanto a los afiliados y de los recursos económicos de la congregación”.

2.3 MARCO TEORICO

2.3.1. SISTEMA DE ADMINISTRACIÓN DE IGLESIA

El sistema de administración de la iglesia comprende en administrar los afiliados y los recursos económicos que tienen las iglesias, esto se comprende mejor con la definición de administración que tiene la iglesia que se detalla a continuación.

2.3.2. ADMINISTRACION DE LA IGLESIA

La Iglesia Bautista de Cobija hasta el día de hoy no ha llevado un registro actualizado de las personas, nuevas o antiguas, de los afiliados.

Tan solo se lleva cuenta de los líderes de los diferentes ministerios y células y solo si estos llevan bastante tiempo asistiendo a la Iglesia.

Al principio el registro de las personas se realizaba con un lapso de dos años entre actualizaciones, donde la secretaria, repartía entre el Liderazgo tarjetas de membrecía que eran llenadas, la mayor parte de las veces no en su totalidad, por las personas a las que les eran entregadas.

Cuando las tarjetas eran devueltas, los registros eran guardados en un archivo Excel.

No se han vuelto a hacer registros desde el 2007 en Enero hasta el día de hoy. La iglesia sigue en constante crecimiento.

2.3.3. DESCRIPCION DEL SISTEMA ACTUAL

Toda la información es solicitada a través de tarjetas de afiliados cada una de las cuales contiene la información correspondiente a los datos principales de las personas como nombre, dirección de residencia, teléfono y datos concernientes a la Iglesia como si se encuentra bautizada o no, sin embargo, carece de información en cuanto a si trabaja en algún ministerio o las diferentes áreas en las que se desenvuelve dentro de la Iglesia.

2.3.4. SISTEMA DE INFORMACION (SI)

Un Sistema de Información es el conjunto formal de procesos que, actúan sobre una colección de datos estructurada de acuerdo con las necesidades de una institución. Recopila, elabora y distribuye parte de la misma en las operaciones de ella y en las actividades de dirección y control correspondientes, apoyando al menos en parte en la toma de decisiones necesaria para

Desempeñar las funciones y procesos de negocio de la institución de acuerdo con su estrategia”. Los sistemas de información se dividen en las siguientes categorías.

2.3.4.1 Sistemas para el procesamiento de transacciones

Sustituye los procedimientos manuales por otros basados en computadoras, trata con procesos de rutina bien estructurados.

2.3.4.2 Sistema para el soporte de Decisiones

Proporciona información a las autoridades que deben tomar decisiones sobre situaciones particulares, apoyan la toma de decisiones en circunstancias que no están bien estructuradas, una decisión se considera no estructurada si no existen procedimientos claros para tomarla.

El sistema de información que se desarrolló en el presente trabajo de grado comparte características tanto de tipo administrativo como de apoyo a la toma de decisiones, por que apoya las autoridades a tomar decisiones y resolver problemas en base a la información que está registrada. Al mismo tiempo proporciona información en un tiempo pertinente para el apoyo en la toma de decisiones.

Las solicitudes de sistemas de información están motivadas por uno de los siguientes tres objetivos generales:

- a) **Resolver un problema.** Actividades, procesos o funciones que en la actualidad, o quizá en el futuro, no satisfacen los estándares de desempeño o las expectativas.

- b) **Aprovechar una oportunidad.** Un cambio para ampliar o mejorar el rendimiento de la institución y su competitividad.
- c) **Dar respuesta a directivos.** Proporcionar información en respuesta a órdenes, solicitudes o mandatos originados por una autoridad legislativa o administrativa, llevar a cabo tareas de cierta manera, o también cambiar la información o tal vez el desempeño.

2.3.4.3 Sistema de Información Administrativa

Ayudan a las autoridades a tomar decisiones y resolver problemas. Se recurren a los datos almacenados como consecuencia del procesamiento de las transacciones, pero también de información, sobre tendencias económicas, demandas, costos de préstamos y otros.

Los sistemas de administración ayudan a los directivos a tomar decisiones y resolver problemas. Los directivos recurren a los datos almacenados como consecuencia del procesamiento de las transacciones, pero también emplean otra información.

Dado que los procesos de decisión están claramente definidos, entonces se puede identificar la información necesaria para formular las decisiones. Cada vez que se necesite la información, esta se prepara y presenta en una forma y formato diseñados con anterioridad.

Los especialistas en sistemas de información describen las decisiones apoyadas por estos sistemas como decisiones estructuradas. El aspecto estructurado se refiere al hecho de que los administradores conozcan de antemano los factores que deben tenerse en cuenta para la toma de decisiones así como las variables con influencia más significativa sobre el resultado de una decisión (buena o mala) A su vez, los analistas de sistemas desarrollan reportes bien estructurados que contienen la información necesaria para las decisiones o que indican el estado de las variables importantes.

El sistema de información administrativa, o sistema de informes para la administración, presentara reportes basados en las actividades de nivel de transacción. Con frecuencia la información proporcionada se combina con otra de naturaleza externa. La necesidad de tomar

Cada una de estas decisiones se presenta con frecuencia y, por tanto, la información necesaria para ello debe prepararse con regularidad.

La información que fluye dentro del SI se almacena en una base de datos para su posterior uso.

2.3.5. Bases de Datos

La Base de Datos (BD) es justamente la colección de datos recopilados y estructurados que existe durante un periodo de tiempo. Por ejemplo, un libro contable, debido a su estructura, se puede considerar como tal. Una novela, por el contrario, no tiene estructura, y no se suele considerar. Generalmente, un sistema de información consta de una o más BD, junto con los medios para almacenarlas y gestionarlas, sus usuarios y sus administradores.

Una base de datos es un conjunto de datos almacenados de manera que se pueda acceder fácilmente a ellos, con la posibilidad de ordenarlos y relacionarlos. Es un conjunto de información estructurada en registros y almacenada en un soporte electrónico legible desde un ordenador. Cada registro constituye una unidad autónoma de información que puede estar a su vez estructurada en diferentes campos o tipos de datos que se recogen en dicha BD.

La información contenida en una base de datos se distribuye normalmente en registros, la cual puede dividirse en unidades de datos de una misma categoría llamadas campos. Por ejemplo, en un directorio de miembros de una asociación, un registro será la ficha completa de cada uno de los socios. En cada registro se recogerán determinados datos, como el nombre, la profesión, la dirección o el teléfono, cada uno de los cuáles constituye un campo.

Las bases de datos se representan en un modelo de base de datos que se explica en el siguiente apartado.

2.3.5.1. Modelos de Base de Datos

Existen diferentes modelos de datos dentro de una base de datos como ser.

- Modelo Distribuidas
- Modelo Relacional
- Modelo Orientado a Objetos

El modelo que se utilizó en el proyecto de grado es el Modelo Relacional, es la que permite representar los datos y la relación de manera simple, para representar y manipular los datos.

La estructura fundamental del modelo relacional es la "relación", es decir una tabla bidimensional constituida por filas (tuplas) y columnas (atributos), las relaciones representan las entidades que se consideran interesantes en la base de datos, cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representan las propiedades de la entidad.

En la actualidad las bases de datos se asocian con las computadoras, y su gestión no suele ser manual, sino altamente automatizada. La tecnología actual exige la delegación de la gestión de una base de datos a unos tipos de aplicaciones software específicas denominadas Sistemas de Gestión de Bases de Datos (SGBD).

2.3.5.2. Sistema de Gestión de Base de Datos (SGBD)

Los sistemas de Gestión de Bases de Datos, son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporciona un acceso controlado a la misma.

Los SGBD es una aplicación que interactúa con los usuarios de los programas de aplicación y la base de datos.

Algunos de los SGBD más conocidos son: SQL, ACCESS, POSTGRESS, MYSQL y otros, para el presente proyecto de grado se utilizara el manejador de base de datos MYSQL.

2.3.5.3. SGBD MySql

MySql Data base Server es un administrador de Base de Datos (DB) de código fuente. Para agregar, acceder y procesar datos guardados en un computador, se necesita un administrador de DB como el MySql. Dado que las computadoras son muy buenas manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de los sistemas de información.

A si como todo SGBD, MySql utiliza el Lenguaje de Definición de Datos y el Lenguaje de Manejo de Datos que se definen a continuación.

MySql utiliza el lenguaje de consultas para manipular los datos que están dentro de la BD.

A) Lenguaje de Definición de Datos (LDD)

Un lenguaje de definición de datos es un lenguaje de programación proporcionado por el sistema de gestión de base de datos que permite definir estructuras de datos utilizando el lenguaje de consultas SQL, las instrucciones que se utilizan en esta categoría son.

- CREATE DATABASE
- CREATE TABLE
- DROP TABLE
- ALTER TABLE

B) Lenguaje de Manejo de Datos (LMD)

Un lenguaje de manejo de datos así como el LDD es un lenguaje de programación para manipular los datos que están almacenados dentro de la DB, utilizando el lenguaje de consultas SQL, las instrucciones que se utilizan en esta categoría son.

- INSERT
- SELECT
- UPDATE
- DELETE

El software que administra la BD, tiene los siguientes objetivos que se muestran a continuación.

- Definir la Base de Datos mediante el Lenguaje de Definición de Datos, el cual permite especificar la estructura, tipo de datos y las restricciones sobre los datos, almacenándolo todo en la base de datos.
- Separar la descripción y manipulación de datos, permitiendo un mayor entendimiento de los objetos, además de flexibilidad de consulta y actualización de los datos.
- Permitir la inserción, eliminación, actualización, consulta de los datos mediante el Lenguaje de Manejo de Datos, lo que permite resolver el problema que presentan los sistemas de archivos, donde hay que trabajar con un conjunto fijo de consultas o la necesidad de tener muchos programas de aplicaciones.
- Proporcionar acceso controlado a la base de datos.
 - a) Seguridad: los usuarios no autorizados no pueden acceder a la base de datos.
 - b) Integridad: mantiene la integridad y consistencia de la base de datos.
 - c) Control de Recurrencia: permite el acceso compartido a la base de datos.
 - d) Control de Recuperación: restablece la base de datos después de producirse un fallo de software o hardware.
- Gestionar la estructura física de los datos y su almacenamiento, proporcionando eficiencia en las operaciones de la base de datos y el acceso al medio de almacenamiento.
- Proporcionar un mecanismo de vistas, que permita a cada usuario tener su propia vista o visión de la base de datos.
- Permitir una fácil administración de los datos.

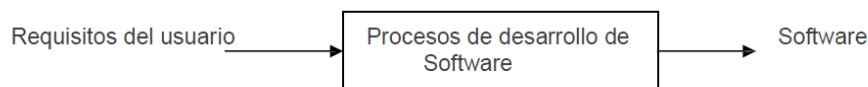
Un Sistema de Información además de orientarse hacia una categoría de información y administrar sus datos con un SGBD, funciona en una intranet dentro de la Institución.

2.3.6. PROCESO UNIFICADO

Un proceso define quien está haciendo que, y cuando, además dice como alcanzar un determinado objetivo. En la ingeniería de software el objetivo es construir un producto de software [Jacob, 2000], vale decir, que todos los proyectos necesitan de un proceso que guie sus actividades.

Según Jacobson “El procesos Unificado de desarrollo de Software”, unos procesos efectivos proporcionan normas para el desarrollo eficiente de Software de calidad, captura y presenta las mejores prácticas que la tecnología permite. Por tanto reduce el riesgo y hace el proyecto más predecible (ver Figura. 2.1).

FIGURA 2.2 Proceso de Desarrollo de Software

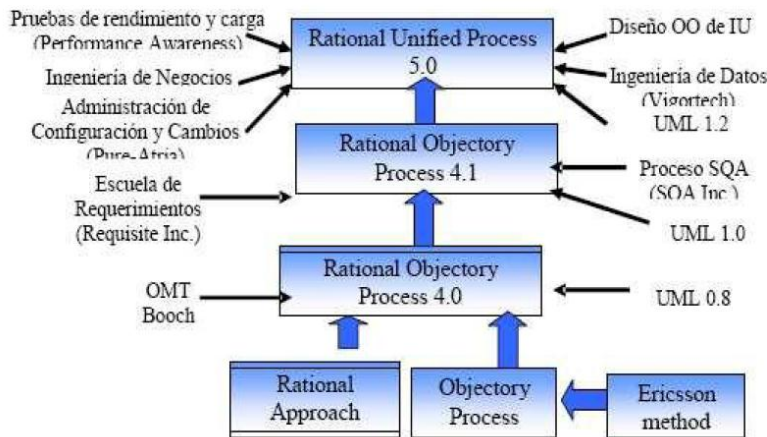


Fuente: [Jacob, 2000]

Entre muchos investigadores de la orientación a objetos hay tres autores que se han destacado por sus contribuciones al uso del paradigma en todo el proceso de desarrollo: Ivar Jacobson, Grady Booch y James Rumbaugh. Luego de muchos años de trabajo individual desarrollado y difundido sus propios métodos, han unido sus teorías y su experiencia, y se han puesto a la cabeza de un formidable grupo de investigadores para contribuir dos herramientas con las cuales buscan estandarizar y por ende facilitar el uso de los objetos en la programación: El lenguaje Unificado de Modelo (UML Unified Modeling Language) y el proceso unificado rotacional para el desarrollo de programas (RUP, Rational Unified Process) mientras que UML, es ya un lenguaje maduro que ha logrado la aceptación de amplios sectores de la industria y la academia, RUP sigue siendo una propuesta que deberá depurarse y templarse al calor de la

experiencia de su aplicación en el campo y los portes de los casos de estudio [Jacob,2000] (ver Figura. 2.2).

FIGURA 2.3 Historia de Procesos Unificados



Fuente: [Jacob, 2000]

RUP y UML están estrechamente relacionados entre sí, pues mientras el primero establece las actividades y los criterios para conducir un sistema desde su máximo nivel de abstracción, el segundo ofrece la notación grafica necesaria para representar los sucesos, modelos, que se obtienen de procesos de refinamiento. RUP se define como unos procesos dirigidos por:

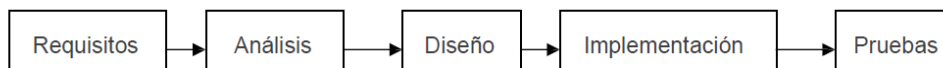
- Casos de Uso
- Centrado en la Arquitectura
- Iterativo e Incremental

2.3.7. DIRIGIDO POR CASOS DE USO

Procesos de desarrollo de software utiliza los casos de uso como una herramienta para la obtención de requisitos de usuario. Donde los casos de uso son para definir la funcionalidad del sistema, y guían al desarrollador en la construcción de la arquitectura del sistema.

La descripción obtenida de los requerimientos debe ser comprendida por casos de uso que nos ayudan a recopilar la información acerca de la interacción que tiene los usuarios en este caso actores con el sistema. Un caso de uso es una secuencia, reacciones que el sistema lleva a cabo para ofrecer un resultado de valor a algún actor, que sirven para realizar pruebas sobre los componentes desarrollados (ver Figura. 2.3). Los casos de uso enlazan los flujos de trabajo fundamentales. El proyecto progresa a través de estos flujos de trabajo, que inician en los casos de uso [Jacob 2000].

FIGURA 2.4 Casos de uso que enlaza los flujos de trabajo fundamental



Fuente: [Larma, 1999]

2.3.8. CENTRADO EN LA ARQUITECTURA

En el caso de software la arquitectura se refiere a un conjunto de decisiones significativas acerca de la organización de un sistema, la elección de los elementos acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de las cuales se componen el sistema con su respectivo comportamiento y las interacciones entre esos elementos y la composición de esos elementos estructurales.

La necesidad de una arquitectura radica en poder comprender el sistema, es decir que todos los que están involucrados con su desarrollo deben entender el problema al cual va enfocado el sistema de software para satisfacer las demandas individuales y de la organización mediante la utilización de los diagramas definidos por UML.

La organización es un punto muy importante ya que cuanto mayor sea la organización del proyecto software mayor será la comunicación entre los desarrolladores para coordinar sus esfuerzos dividiendo el sistema en subsistemas definiendo las interfaces correctas.

Al conocer el dominio de problema y con qué componentes se piensa en cómo conectar esos componentes para cumplir con los requisitos del sistema y realizar los modelos de casos de uso reutilizando dichos componentes.

2.3.9.1. Lenguaje Unificado de Modelado (UML)

UML tiene una notación gráfica que permite representar en mayor o menor medida todas las fases de un proyecto de software basado en computadoras: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos y otros, hasta la implementación y configuración con los diagramas de despliegue.

Existen diversos métodos y técnicas orientadas a objetos, con muchos aspectos en común pero utilizando distintas notaciones, se Presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques y la falta de estandarización en la manera de representar en forma gráfica un modelo, impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre estos distintos diseñadores, lo que genero la creación del UML como estándar para el modelado de sistemas de software, pero con posibilidades de ser aplicado a todo tipo de proyectos.

En el proyecto de grado se utilizó los siguientes diagramas especificados en la metodología.

- Diagrama de Casos de Uso
- Diagrama de Clases
- Diagrama de Estados
- Diagrama de Secuencia

a). Casos de Uso

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo. En este tipo de diagramas intervienen algunos conceptos nuevos como.

Actor.- Es una entidad externa al sistema que se modela y que puede interactuar con él; un ejemplo de actor podría ser un usuario o cualquier otro sistema.

b). Diagrama de Clases

Los diagramas de clases representan un conjunto de elementos del modelo que son estáticos, como las clases y los tipos, sus contenidos y las relaciones que se establecen entre ellos.

Una clase representa un conjunto de objetos que tienen una estructura, un comportamiento y unas relaciones con propiedades parecidas. Describe un conjunto de objetos que comparte los mismos atributos, operaciones, métodos, relaciones y significativo. Los componentes de una clase son:

Atributo. Corresponde con las propiedades de una clase o un tipo.

Operación. También conocido como método, es un servicio proporcionado por la clase que puede ser solicitado por otras clases y que produce un comportamiento en ellas cuando se realiza.

c). Diagrama de Estados

Representan la secuencia de estados por los que un objeto o una interacción entre objetos pasa durante su tiempo de vida en respuesta a estímulos (eventos) recibidos. Un estado en UML es cuando un objeto o una interacción satisfacen una condición, desarrolla alguna acción o se encuentra esperando un evento.

Cuando un objeto o una interacción pasa de un estado a otro por la ocurrencia de un evento se dice que ha sufrido una transición, existen varios tipos de transiciones entre objetos: simples y

Complejos. Como todas las metodologías Orientados a Objetos se envían mensajes, en este caso es la acción de la pueden enviar mensajes a uno o varios objetos destino.

d). Diagrama de Secuencia

Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tiene lugar. En los diagramas de este tipo intervienen objetos que son instancias concretas de una clase que participa en la interacción. Un diagrama de secuencia representa una forma de indicar el periodo durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento.

Los diagramas de secuencia permiten indicar cuál es el momento en el que se envía o se completa un mensaje mediante el tiempo de transición, que se especifica en el diagrama.

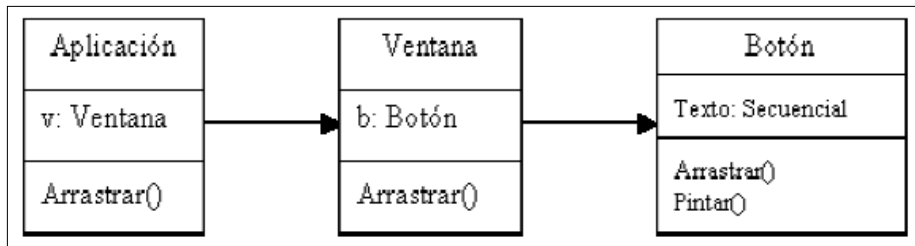
e). Modelo de Objetos

El modelo de objetos describe la estructura estática (de datos), de los objetos del sistema (identidad, atributos y operaciones) y también sus relaciones.

Este modelo de objetos contiene **diagramas de objetos** (Ver Figura 2.4), que son unos grafos, cuyos nodos son clases de objetos y cuyos arcos son relaciones entre las clases. El diagrama contiene clases de objetos organizados en jerarquías, que comparten una estructura y comportamiento comunes y que están asociadas a otras clases, las cuales definen los atributos que lleva cada instancia de objeto y las operaciones que efectúa o sufre cada uno. En cada instancia de la clase se guardan los valores de esos atributos.

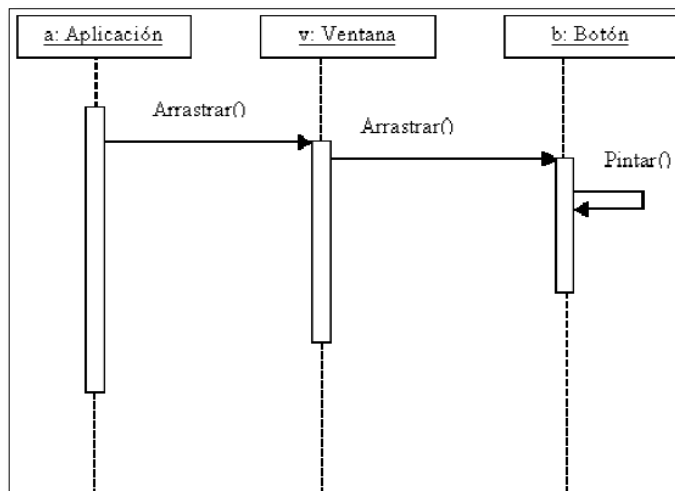
El diagrama de comunicación representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen claramente las responsabilidades. La interacción entre objetos se modela gráficamente mediante un diagrama de interacción entre Objetos (Ver Figura 2.2).

FIGURA 2.5 Diagrama de Interacción de Objetos para un Modelo Estático.



Fuente: (James Rumbaugh, 1996)

FIGURA 2.6 Diagrama de Interacción de Objetos con Ventana Gráfica.



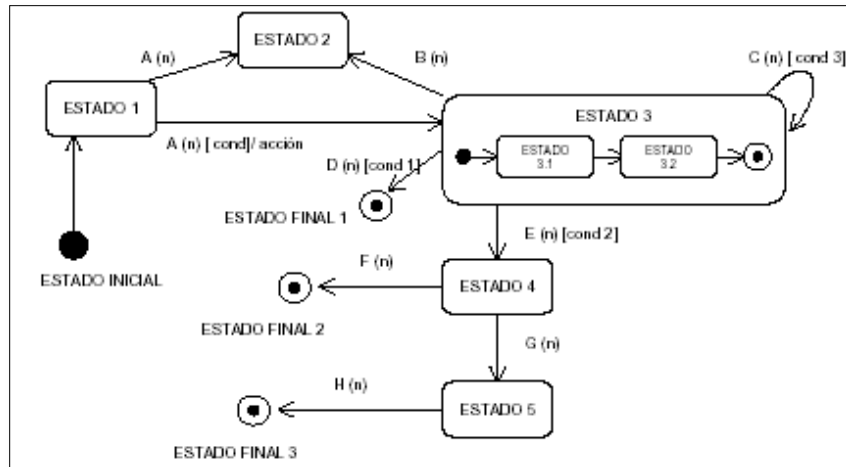
Fuente: (James Rumbaugh, 1996)

f).Modelo Dinámico

El modelo dinámico describe los aspectos de comportamiento (de control) de un sistema que cambian con el tiempo. El modelo dinámico se utiliza para especificar e implementar los aspectos del control del sistema. Los modelos dinámicos contienen **diagramas de estados** (Ver Figura 2.3). Un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos son transiciones entre estados causadas por sucesos o eventos.

Los Diagramas de Transición de Estados (DTE) se utilizan para describir el comportamiento de los objetos estableciendo sus vidas posibles. Una vida válida es una secuencia de eventos que caracteriza un comportamiento correcto para todos los objetos de la clase. Los principales componentes del diagrama son los estados y las flechas, que representan los cambios de estado.

FIGURA 2.7. Diagrama de Transición de Estados



Fuente: (James Rumbaugh, 1996)

Se especifican en este modelo la temporización y secuencia de operaciones (sucesos que marcan los cambios, secuencias de los mismos, estados que definen el contexto para los sucesos), y la organización de sucesos y de estados. El modelo dinámico captura el control, aquel aspecto de un sistema que describe las secuencias de operaciones que se producen sin tener en cuenta lo que hagan las operaciones, aquello a lo que afecten o la forma en la que estén implementadas.

h).Modelo Funcional

El modelo funcional describe las transformaciones (de función), de valores de datos que ocurren dentro del sistema, captura lo que hace el sistema, independientemente de cuándo se haga o de la forma en que se haga. Las funciones se invocan como acciones en el modelo dinámico y se

muestran como operaciones que afectan a objetos en el modelo de objetos. El modelo funcional contiene **diagramas de flujo de datos** (Ver Figura 2.6).

El Diagrama de Flujo de Datos (DFD) es una técnica gráfica que describe el flujo de información y las transformaciones que se aplican a los datos, conforme se mueven de la entrada a la salida. Es un grafo cuyos nodos son procesos y cuyos arcos son flujos de datos, se muestra las dependencias entre los valores y el cálculo de valores de salida a partir de los de entrada y de funciones, sin considerar cuando se ejecutan las funciones, ni siquiera si llegan a ejecutarse. También se le conoce como un grafo de flujo de datos o un diagrama de burbujas.

FIGURA 2.8: Forma básica de Diagrama de Flujo de Datos.



Fuente: (James Rumbaugh, 1996)

i).Diagrama de Flujo de Datos

Los diagramas de flujo de datos son un tipo de herramienta de modelado, permiten modelar todo tipo de sistemas, concentrándose en las funciones que realiza, y los datos de entrada y salida de esas funciones.

Un sistema puede representarse empleando varios diagramas de flujos de datos, cada flujo de datos puede representar una parte "más pequeña" del sistema.

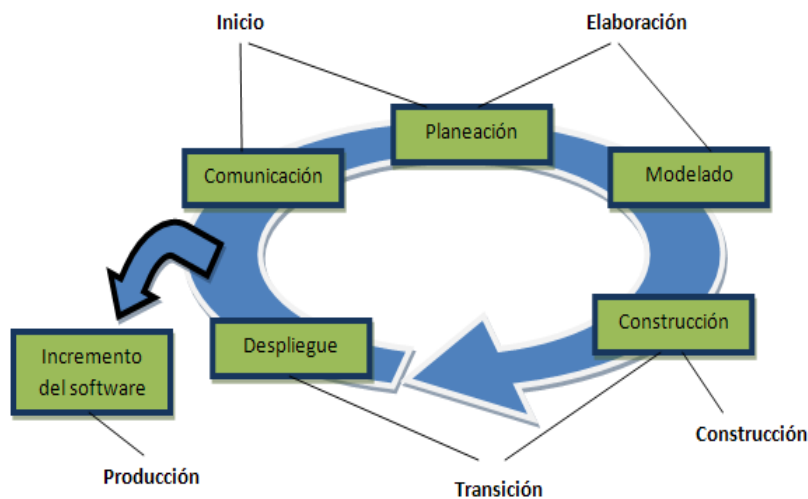
Los DFD permiten una partición por niveles del sistema. El nivel más general se representa con un DFD global llamado diagrama de contexto.

El diagrama de contexto DFD representa a todo el sistema con una simple burbuja o proceso, las entradas y salidas de todo el sistema, y las interacciones con los usuarios.

2.3.10. FASES DE DESARROLLO DE LA METODOLOGÍA PROCESO UNIFICADO

Esta metodología se divide en cuatro etapas las cual se describe a continuación.

FIGURA 2.9: Fase de la metodología



Fuente: (James Rumbaugh, 1996)

a) **Inicio**

En esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

b) Elaboración.

En esta fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

c) Construcción

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

d) Transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

Esta metodología no ha sido aplicada probablemente por su complejidad de administración o desconocimiento de la misma, desaprovechando sus considerables ventajas respecto a los métodos tradicionales. Por esto, es necesario entonces desarrollar mecanismos de apropiación tecnológica más eficaces, que permitan mantener actualizadas las prácticas organizacionales y los marcos de referencia aquí mencionados.

2.3.10.1. Herramientas de Desarrollo

De acuerdo a la tecnología adoptada para el funcionamiento del Sistema, las herramientas de desarrollo que se utilizaron para el presente proyecto fueron los que se describen a continuación.

2.3.10.2. Lenguaje de etiquetas HTML

Se optó por utilizar HTML para desarrollar las vistas del sistema con el que usuario interactuará constantemente por ejemplo con los formularios y las listas y reportes.

HTML, siglas de Hyper Text Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Según (Miguel Ángel Alvares, 2009) HTML es el lenguaje con el que se definen las páginas web, básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

HTML se creó en un principio con el objetivo de divulgar información con texto y algunas imágenes. No se pensó que llegara a ser utilizado para crear área de ocio y consulta con carácter multimedia (lo que es actualmente la web), de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, se ha incorporado modificaciones con el tiempo, esto es lo que se conoce como los estándares del HTML.

2.3.10.3. Lenguaje de programación PHP5

El lenguaje de programación PHP (Hypertext Preprocessor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web.

PHP fue desarrollado originalmente por (Rasmus Ledford, 1994) como un CGI escrito en Perl que permitía la interpretación de un número limitado de comandos.

El sistema fue denominado Personal Home Page Tools y consiguió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas.

La última versión es PHP5 fue lanzado el 13 de julio del 2004, que utiliza el motor Zend-2 y presenta mejoras significativas y un entorno de programación orientado a objetos mucho más completo, que permite que el PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas J2EE y .NET.

El lenguaje de Programación a partir de su versión 5 adopta la mayor parte de las características de la Programación Orientada a Objetos.

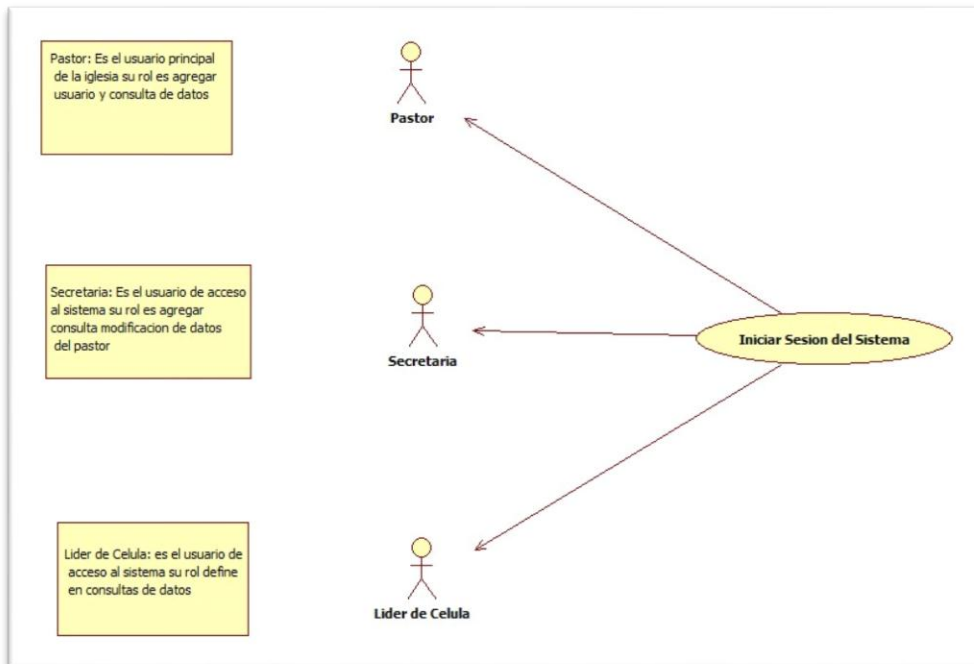
CAPÍTULO III
DESARROLLO DEL PROYECTO

3.1.- FASE DE INICIO

En esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

Usuarios que intervienen en el Sistema

FIGURA 3.1 Función del sistema



Fuente: Elaboración Propia

3.2. FASE DE ELABORACION

En esta fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

3.2.1. Requerimientos

En esta etapa se identificó los requerimientos en la Iglesia Bautista de Cobija.

Los requerimientos generales pueden verse en la siguiente Tabla.

TABLA 3.1 Requerimientos Generales

Ref.	Requerimientos	Tipo
1	Reporte de las personas de la Iglesia	Evidente
2	Realice los de ingreso de la Iglesia	Evidente
3	Realice los Egresos de la iglesia	Evidente
4	Reportar los saldos en caja en moneda nacional	Evidente
5	Reporte de los donadores de la iglesia	Evidente
6	Reportar del seguimiento de las personas	Evidente
7	Incorporar la técnica de evaluación de cada personas haciendo cuadros comparativos	Evidente
8	Que el reporte finales se puedan tener en cuadros comparativos del mes anterior con el mes actual	Evidente
9	Los reportes de los donadores deben ser por categoría (Diezmo, Ofrenda)	Evidente

Fuente: Elaboración Propia

A partir de los requerimientos generales se realizó un análisis para poder obtener los requerimientos específicos para identificar las funcionalidades del sistema.

TABLA 3.2 Requerimientos de Autenticación de Usuarios

Ref.	Requerimientos	Tipo
1	Cada usuario que interactúe con el sistema de tener una identificación de usuario y su contraseña	Evidente
2	Una vez ingresado al sistema el usuario tiene acceso al menú.	Evidente
3	Cada usuario debe tener diferentes opciones de acuerdo al rol del usuario	Evidente

Fuente: Elaboración Propia

TABLA 3.3 Requerimientos de Administrar parámetros del sistema.

Ref.	Requerimientos	Tipo
1	En el menú del sistema se debe mostrar las opciones para ingresar datos de registro de las personas.	Evidente
2	El sistema debe permitir registrar los datos como: Asistencia, Donaciones por personas, Seguimiento de las personas y otros	Evidente
3	El sistema debe permitir ver los datos ingresados.	Evidente

Fuente: Elaboración Propia

TABLA 3.4 Requerimientos de Administrar Personas que asiste a la congregación

Ref.	Requerimientos	Tipo
1	El sistema debe permitir registrar a las personas que asiste a la congregación.	Evidente
2	El sistema también debe permitir registrar funcionarios de la iglesia, esto para realizar los registros de los datos.	Evidente
3	El sistema debe permitir ver los datos ingresados.	Evidente

TABLA 3.5 Requerimientos de Administrar Ingresos y Egresos

Ref.	Requerimientos	Tipo
1	El sistema permitirá registrar los ingresos de los donadores por grupo familiar, elaborando una nota de entrada donde se registra los siguientes datos; monto expresado en las tres monedas, fecha de entrega, nombre del que entrego, nombre del que recibió.	Evidente
2	También permitirá registrar el monto por persona.	Evidente
3	Después de ingresar los datos, el sistema debe permitir calcular el sub total y el total de manera automática.	Evidente
4	El sistema debe permitir hacer el tipo de cambio a la cotización actual.	Evidente
5	El sistema debe permitir imprimir la nota de ingreso.	Evidente
6	El sistema debe permitir registrar los egresos por categoría.	Evidente
7	El sistema debe permitir realizar cierre de los ingresos y de los ingresos mensualmente.	Evidente

Fuente: Elaboración Propia

TABLA 3.6 Requerimientos de Reportes

Ref.	Requerimientos	Tipo
1	Debe permitir buscar el nombre de las personas y mostrar la cantidad que ha donado y sus asistencias.	Evidente
2	Debe reportar todos los donadores en grupo familiar y en individual.	Evidente
3	Debe permitir reportar los Egresos mensualmente.	Evidente
4	Debe permitir reportar los ingresos mensualmente.	Evidente
5	El sistema debe permitir reportar egresos por rubros.	Evidente
6	Debe permitir el cierre mensualmente de los ingreso y de los egresos y de la asistencia.	Evidente
7	El sistema debe permitir imprimir todos los reportes de ingresos y egresos y asistencia en un informe mensual	Evidente

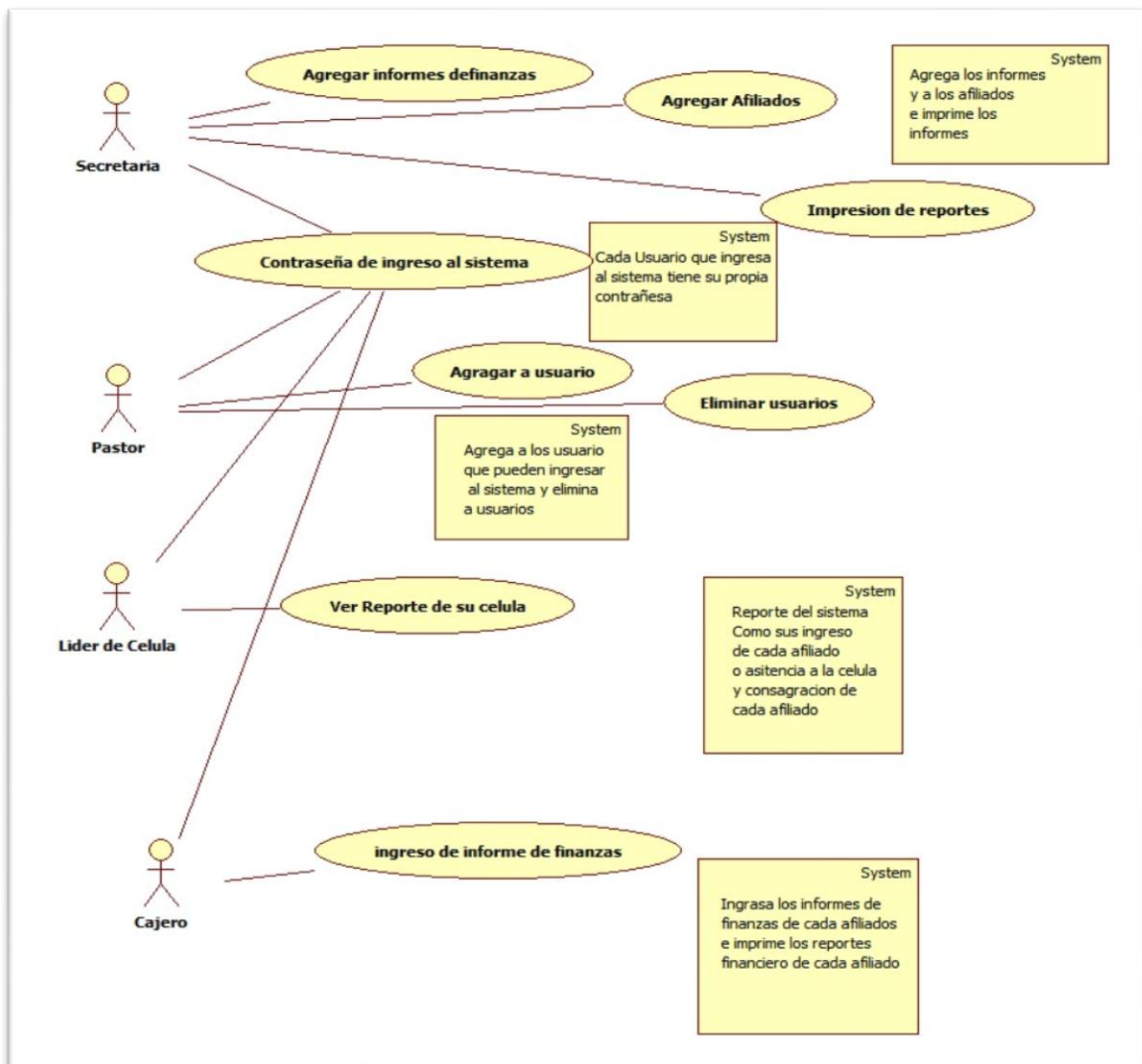
Fuente: Elaboración Propia

3.2.2. Casos de uso

Para poder ver el panorama general del sistema se elaboró un Caso de Uso General donde muestran los actores que intervienen en el sistema y que opciones tienen cada uno con sus respectivos roles.

A continuación se muestra el caso de uso general.

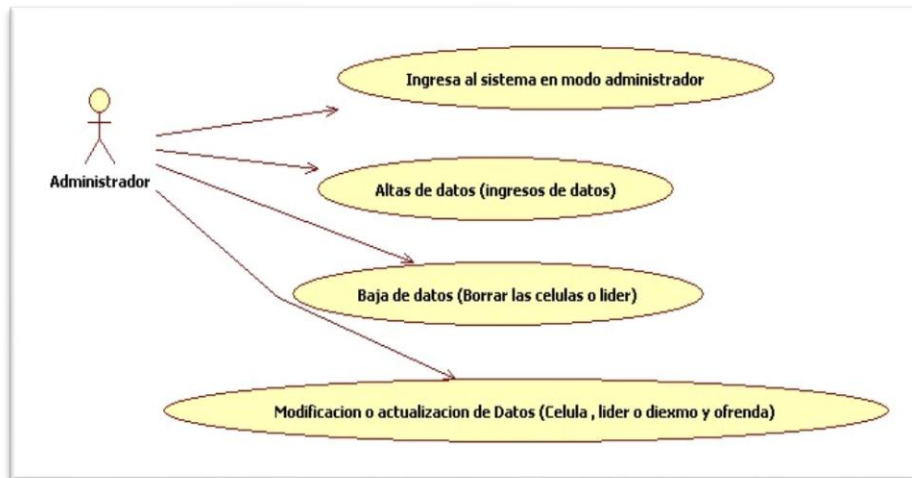
FIGURA 3.2 Diagrama General



Fuente: Elaboración Propia

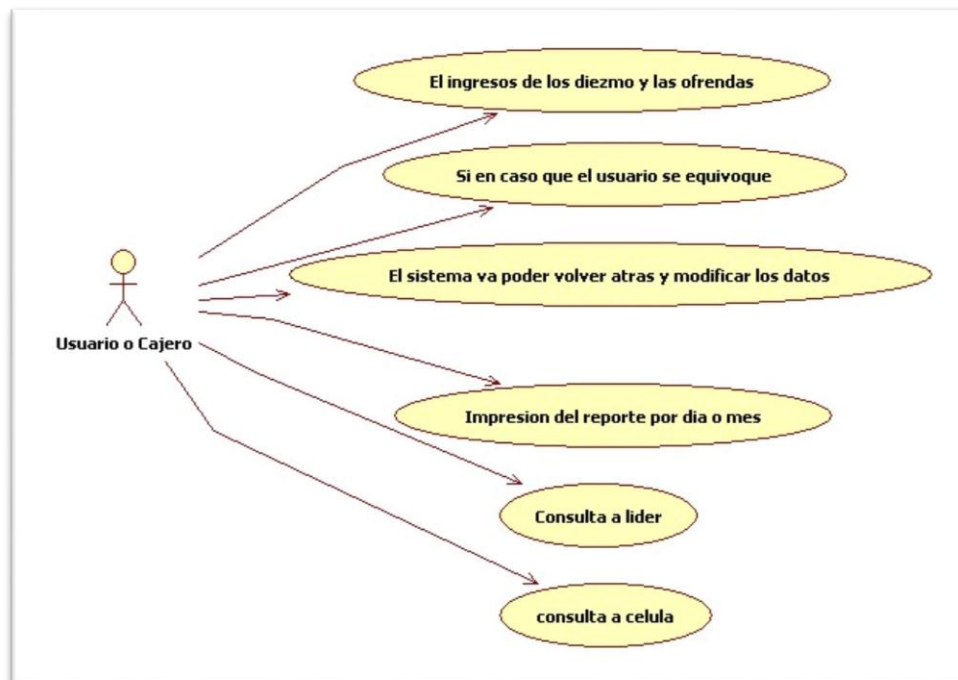
Después de elaborar el caso de uso general se elaboran los casos de uso más detallados que se muestran a continuación.

FIGURA 3.3 Diagrama casos de uso del Administrador en el Sistema



Fuente: Elaboración Propia

FIGURA 3.4 Diagrama casos de uso Cajero



Fuente: Elaboración Propia

TABLA 3.7 Descripción del caso de uso Cajero

Caso de uso	Descripción
Registra los datos ingresados de Diezmo y Ofrendas	Cuando se trae un nuevo informe el usuario cajero va ingresar los datos que trae el líder en las fichas registradas con los diferentes montos y monedas.
Listar los montos que han sido agregados	En la lista de montos registrado se listara por nombre de la persona por fecha y por mes.
Editar datos que se han ingresado	Permite editar los montos de la lo que se ingresó en caso de monedas o de personas.
Eliminar el registro	Elimina la persona o el ingreso de la base de datos.

Fuente: Elaboración Propia

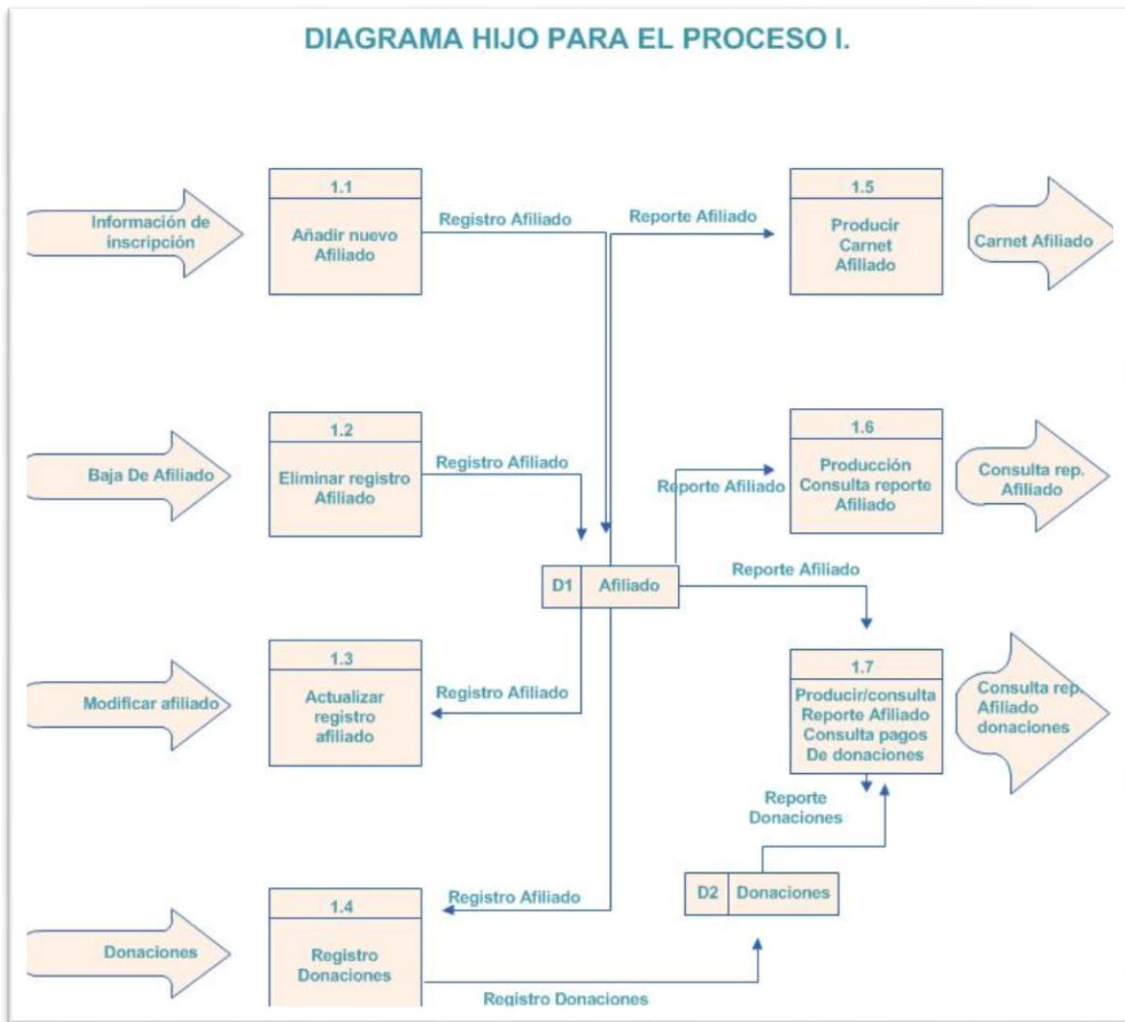
3.3. FASE DE CONSTRUCCION

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

a) Construcción

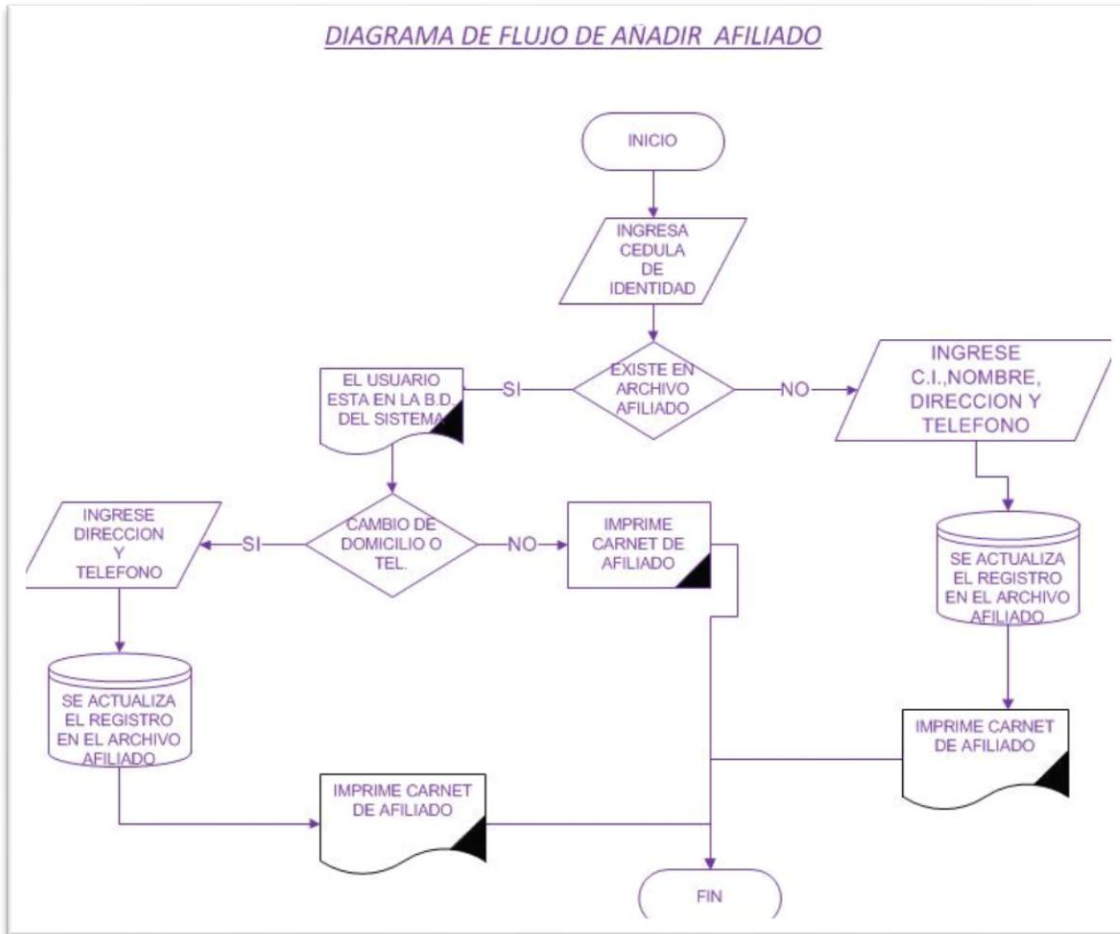
El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto. Se realizas los siguientes diagramas para la construcción del Sistema.

FIGURA 3.5 Diagrama del Proceso



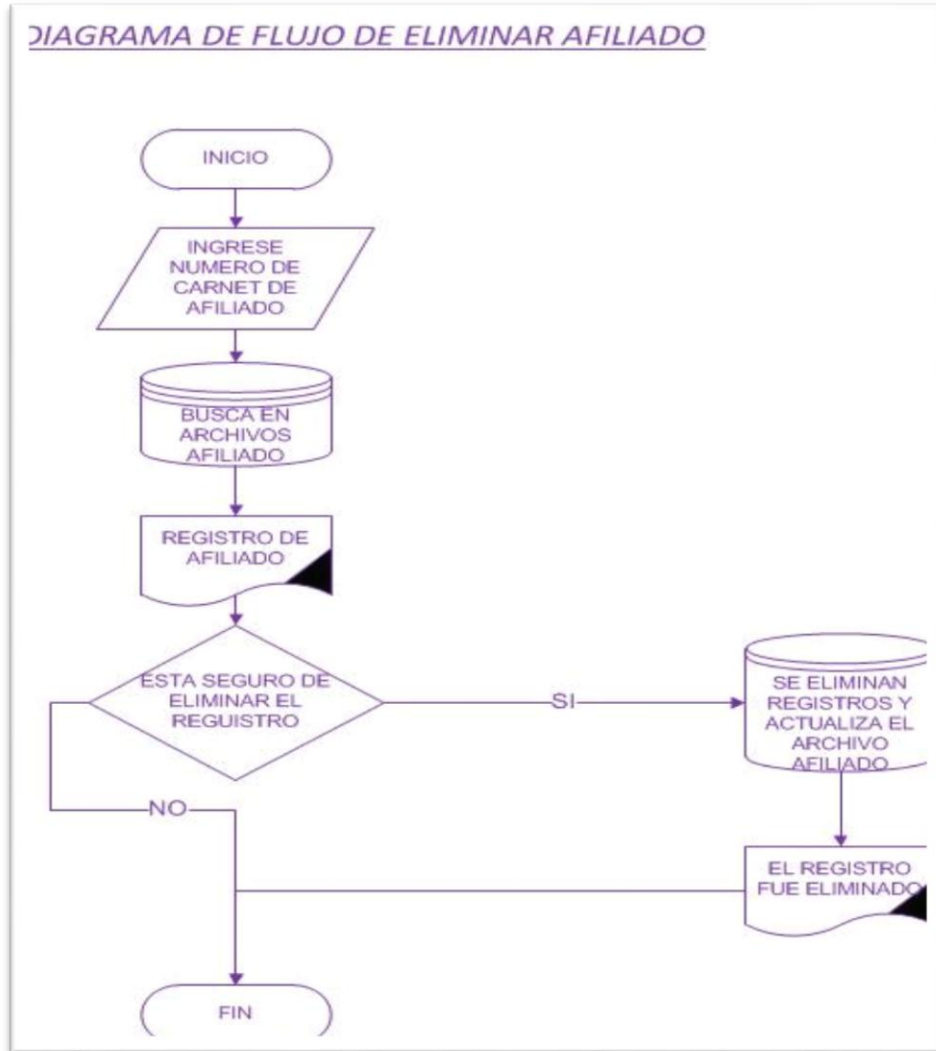
Fuente: Elaboración Propia

FIGURA 3.6 Añadir Afiliado



Fuente: Elaboración Propia

FIGURA 3.7 Eliminar Afiliado

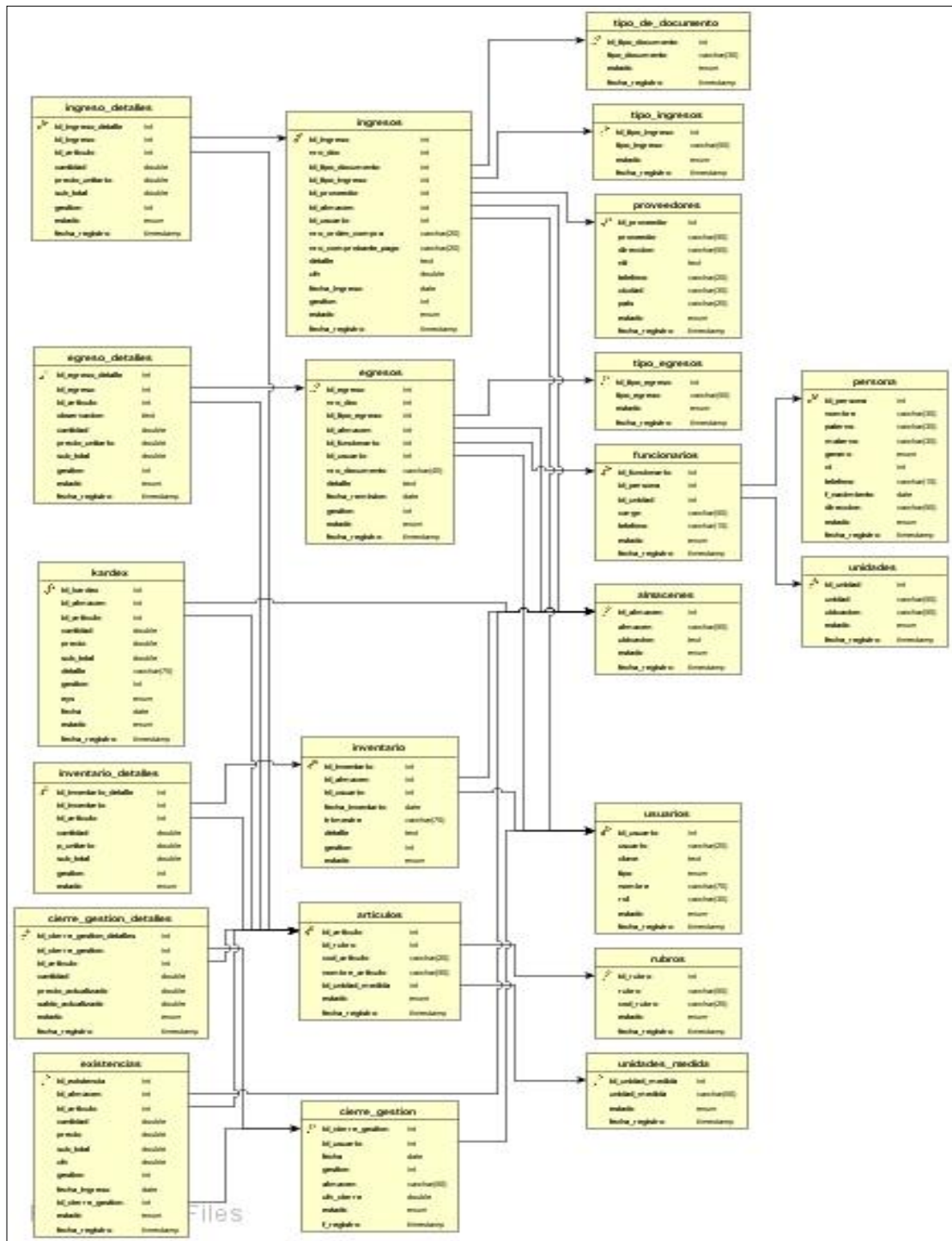


Fuente: Elaboración Propia

b) Diseño de la base de datos

Para que el sistema guarde toda la información, se diseñó la base de datos con el nombre de *ib_sammib*, la misma que cuenta con 38 tablas y está diseñada en el manejador de base de datos *MYSQL*, la base de datos se encuentra en la tercera forma normal así como se ve en la siguiente Figura.

FIGURA 3.8 Diseño de la base de Datos



Fuente: Elaboración Propia

3.4 FASE DE TRANSICIÓN

La fase de **transición** cubre el período durante el cual el producto se convierte en la versión de prueba para la utilización.

Las iteraciones en esta fase continúan agregando características al sistema. Sin embargo las características se agregan a un sistema que el usuario se encuentra utilizando activamente.

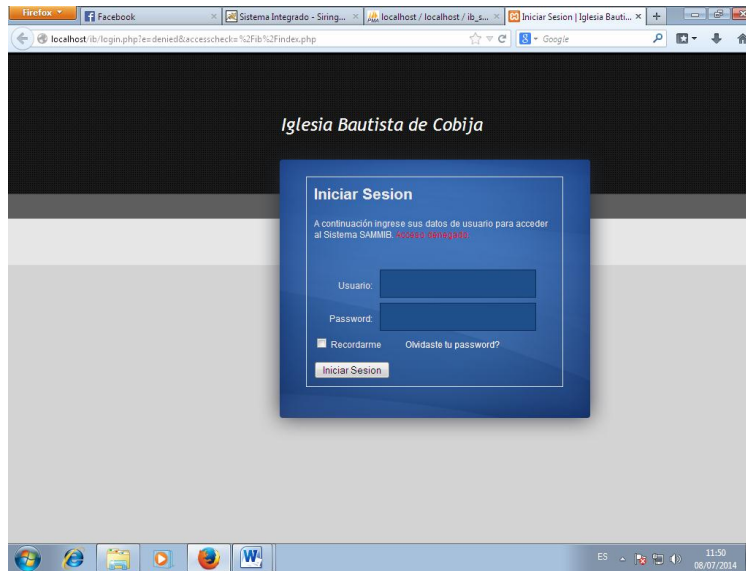
Los artefactos construidos en esta fase son los mismos que en la fase de construcción. El equipo se encuentra ocupado fundamentalmente en corregir y extender la funcionalidad del sistema desarrollado en la fase anterior.

Esta fase se alcanza cuando el equipo de desarrollo llegan a un acuerdo sobre:

- Se han alcanzado los objetivos fijados en la fase de Inicio.
- El usuario está satisfecho.

En los siguientes gráficos se muestra muchas interfaces con las que cuenta el Sistema de Administración de Afiliado y Recursos Económicos que interactúan los usuarios.

FIGURA 3.9 Ingresos al Sistema



Fuente: Elaboración Propia

FIGURA 3.10 Menú principal del Sistema



Fuente: Elaboración Propia

FIGURA 3.11 Menú de Agregar Afiliado

Agregar miembro

A continuación Ingrese los datos del miembro.

Nombres: * Colocar ambos nombres.

Apellido Paterno: * Campo obligatorio

Apellido Materno:

Genero: * Masculino

Numero de Carnet: Incluir procedencia

Telefono: Fijo o Celular

Domicilio:

Tipo de sangre: * No especificado

Es donante?: * No

Grado de instruccion: * Bachiller

Estado civil: * Soltero(a)

Lugar de trabajo:

Dependencia: * Iglesia Bautista de Cobija


Fuente: Elaboración Propia

FIGURA 3.12 Reporte de Asistencia a la Célula

Llenar nuevo informe

Informe No: 104 fecha de llenado: 2013-06-27 16:01:51

EL PUEBLO QUE SE INTERESA



CADA CASA UNA IGLESIA
CADA MIEMBRO UN MINISTRO

IGLESIA BAUTISTA DE COBIJA

AV. 09 DE FEBRERO 222 TEL: 8422483
e-mail: Iglesiasbautistadecobija@hotmail.com

IIIFORME SEMANAL DE LA Celula 05

LIDER: Lucelia Escalera Merida
SUBAREA: Sub - Area A1
AREA: Area A
DIA Y FECHA DE ENCUENTRO: 2013-04-06

AUXILIAR(ES): Lucelia Escalera Merida
ASESOR/A: Piedades Negrete de Merida
COORD. DE AREA: Dora Sillerico de Vargas
INICIO: 20:10 **CIERRE:** 22:00

DETALLE	RESPONSABLE	REALIZO
Juan 3:16	Erlan Cruz Ramirez	NO
Resp. de NIÑ@os	Erlan Cruz Ramirez	NO
Evangelismo	Lucelia Escalera Merida	SI
Edificacion	Lucelia Escalera Merida	SI
Exaltacion	Rolando Arraya Merida	SI

Fuente: Elaboración Propia

FIGURA 3.13 Reporte de Ingreso

COMPROBANTE DE INGRESOS de:
Celula 01

INFORME	PERSONA	DIEZMOS			OFRENDAS			Fecha
		Bs.	Sus.	RS.	Bs.	Sus.	RS.	
31	Maria Quiroga	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	2013-06-07
31	Simona Montes	0.00.-	0.00.-	0.00.-	10.00.-	0.00.-	0.00.-	2013-06-07
31	Maria Esther Cruz	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	2013-06-07
31	Elda Ramirez	0.00.-	0.00.-	0.00.-	10.00.-	0.00.-	0.00.-	2013-06-07
31	Olivia Ramirez	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	2013-06-07
31	Victoria Vega	180.00.-	0.00.-	0.00.-	20.00.-	0.00.-	0.00.-	2013-06-07
31	Doreide Correta	90.00.-	0.00.-	0.00.-	90.00.-	0.00.-	0.00.-	2013-06-07
32	Maria Quiroga	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	2013-06-07
32	Simona Montes	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	2013-06-07
32	Maria Esther Cruz	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	2013-06-07
32	Elda Ramirez	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	0.00.-	2013-06-07

Fuente: Elaboración Propia

FIGURA 3.14 Afiliados Agregados

A continuacion se listan los miembros activos de la Iglesia Bautista

Hay 371 miembro(s) activo(s).

10 resultados por pagina

Nombres	Apellidos	Telefono	Domicilio	Dependencia	Lugar de Trabajo	Acciones
Abigail	Peredo Crespo	---	Barrio: Pantanal	Celula 17	---	
Abigail	Gonzales	---	---	Celula 22	---	
Adaildo	Nacimiento Andres	---	Km. 28	Celula 21	---	
Adela	Gomez Rufino	---	---	Celula 02	---	
Adelaida	Humaza	---	Barrio: Pantanal	Celula 36	Labores de Casa	
Alejandro	Rivero Nunez Vela	77109716	---	Celula 06	Iglesia Bautista de Cobija	
Alejandro	Abrego Arias	79199699	Barrio: Frontera	Celula 06	Pear Bit'S	

Fuente: Elaboración Propia

FIGURA 3.15 Registro de afiliados donantes

A continuation specify the income of each member

Celula 01	Miembros	COMPROBANTE DE INGRESO			
		Monto	Moneda	Monto	Moneda
	Doreide Correia Vidal	<input type="text"/>	Bs Sus R\$	<input type="text"/>	Bs Sus R\$
	Victoria Vega	<input type="text"/>	Bs Sus R\$	<input type="text"/>	Bs Sus R\$
	Olivia Ramirez	<input type="text"/>	Bs Sus R\$	<input type="text"/>	Bs Sus R\$
	Elda Ramirez	<input type="text"/>	Bs Sus R\$	<input type="text"/>	Bs Sus R\$

Fuente: Elaboración Propia

Las figuras mostradas anteriormente certifican la prueba del funcionamiento del sistema, de esta manera se llega a concluir el desarrollo del sistema de administración de afiliados y recursos económicos.

3.4.1 Valoración de Calidad del Software

La evaluación se refiere a la comprobación de la calidad del sistema elaborado en base a la Metodología PU, para lo cual se aplica las métricas de calidad de la Norma ISO/IEC 9126.

3.4.2 Calidad del Sistema

Basados en la norma de calidad ISO/IEC 9126 (**Ver Anexo B**) se ha elaborado un cuestionario de evaluación (**Ver Anexo E**), establece la forma de medir la calidad del Sistema, del cual se

toma estos resultados el Software fue valorado con índices elevados en los criterios de: Utilización del usuario con acceso al Sistema.

Se ha asignado un peso (valor numérico) a cada atributo del Sistema de acuerdo al grado de importancia o nivel de necesidad exigida por la Iglesia Bautista de Cobija, va desde 1 que significa muy poco importante, hasta el 10 que significa extremadamente importante, se ajustó cada pregunta de la evaluación a un atributo exigido, con estos resultados se aplicó una fórmula para obtener las métricas a cada atributo. El usuario califica un puntaje desde 1 hasta 5, de acuerdo a la siguiente escala:

TABLA 3.8 Escala de calificación del Sistema

Calificación	Puntaje
Muy mal	1
Pésimo	2
Regular	3
Bueno	4
Muy Bueno	5

Fuente: Elaboración propia en base a (Hernaz & Minguet, 2006)

De acuerdo a la norma ISO/IEC 9126, esta se categoriza en seis elementos para realizar la evaluación del sistema. A continuación se detalla estos elementos:

Funcionabilidad

TABLA 3.9 Resultado de la medida Funcionabilidad

Atributo	Peso	Resultado
Adecuación	10	4
Exactitud	10	5
Seguridad	10	5

Fuente: Elaboración propia en base a (Hernaz&Minguet, 2006)

$$\text{Medida de Funcionabilidad} = \frac{10*4 + 10*5 + 10*5}{10+10+10}$$

$$\text{Medida de Funcionabilidad} = 4,6$$

De acuerdo al resultado obtenido, se califica como valor aceptable, por lo que el Sistema si cumple con la métrica de funcionalidad de acuerdo a los requerimientos exigidos por el responsable de la Iglesia Bautista de Cobija.

Fiabilidad

TABLA 3.10 Resultado de la medida Fiabilidad

Atributo	Peso	Resultado
Madurez	10	5
Tolerancia a Fallas	10	5
Recuperabilidad	10	5

Fuente: Elaboración propia en base a (Hernaz&Minguet, 2006)

$$\text{Medida de Fiabilidad} = \frac{10*5 + 10*5 + 10*5}{10+10+10}$$

$$\text{Medida de Fiabilidad} = 5$$

De acuerdo al resultado obtenido, se califica como valor aceptable de la evaluación del Sistema en su métrica de fiabilidad (tiene validaciones sobre los campos, se recupera la información almacenada, etc.) lo cual significa que el sistema es seguro y recupera los datos almacenados fácilmente.

Usabilidad

TABLA 3.11 Resultado de la medida Usabilidad

Atributo	Peso	Resultado
Entendimiento	10	5
Aprendizaje	10	5
Operabilidad	7	5
Atracción	5	5

Fuente: Elaboración propia en base a (Hernaz&Minguet, 2006)

$$\text{Medida de Usabilidad} = \frac{10*5 + 10*5 + 7*5 + 5*5}{10 + 10 + 7 + 5}$$

$$\text{Medida de Usabilidad} = 5$$

De acuerdo al resultado obtenido en su métrica de Usabilidad, se establece que el Sistema de Administración de Afiliados y Recursos Económicos tiene una gran aceptación por parte del usuario, en el manejo de las interfaces y la facilidad de aprendizaje. Por el cual se otorga una buena puntuación, en esta fase es importante estar entre la puntuación de cuatro a cinco.

Eficiencia

TABLA 3.12 Resultado de la medida Eficiente

Atributo	Peso	Resultado
Comportamiento de Tiempos	10	5
Utilización de los recursos	10	5

Fuente: Elaboración propia en base a (Hernaz&Minguet, 2006)

$$\text{Medida de Eficiencia} = \frac{10*5 + 10*5}{10 + 10}$$

$$\text{Medida de Eficiencia} = 5$$

De acuerdo al resultado obtenido, se establece que el Sistema, cumple con los requerimientos exigidos en un determinado tiempo (hace lo que tiene que hacer, no demora mucho tiempo su ejecución), por lo que el software obtiene la puntuación aceptable de acuerdo a la métrica de eficiencia.

Mantenimiento

TABLA 3 13 Resultado de la medida Mantenimiento

Atributo	Peso	Resultado
Capacidad de ser analizado	10	4
Cambiabilidad	7	4
Estabilidad	5	3
Facilidad de Prueba	5	4

Fuente: Elaboración propia en base a (Hernaz&Minguet, 2006)

$$\text{Medida de Mantenimiento} = \frac{10*4 + 7*4 + 5*3 + 5*4}{10 + 7 + 5 + 5}$$

$$\text{Medida de Mantenimiento} = \mathbf{3,8}$$

De acuerdo al resultado obtenido, se establece que el sistema cumple con los requerimientos mínimos exigidos para su mantenimiento y aceptabilidad. Por lo que se obtuvo una puntuación de **3.8**, lo cual significa que es aceptable.

Portabilidad

TABLA 3.14 Resultado de la medida Portabilidad

Atributo	Peso	Resultado
Adaptabilidad	10	5
Coexistencia	5	5
Facilidad de Instalación	10	5

Fuente: Elaboración propia en base a (Hernaz&Minguet, 2006)

$$\text{Medida de Portabilidad} = \frac{10*5 + 5*5 + 10*5}{10 + 5 + 10}$$

$$\text{Medida de Portabilidad} = 5$$

De acuerdo al resultado, se establece el sistema es multiplataforma (Windows, Linux) y puede coexistir con otros sistemas, también es muy fácil su instalación por lo cual obtiene la puntuación más alta.

En conclusión se tiene el siguiente resultado que se visualiza en el Tabla:

TABLA 3.15 Resultados generales de la medida estándar ISO/IEC 9126

Atributos	Puntaje
Funcionalidad	4,6
Fiabilidad	5
Usabilidad	5
Eficiencia	5
Mantenimiento	3,8
Portabilidad	5

Fuente: Elaboración propia en base a (Hernaz&Minguet, 2006)

La Tabla 3.14, indica que el Sistema de Administración de Afiliados y Recursos Económicos cumple con las Métricas de Calidad de la Norma **ISO/IEC 9126**, por lo que **SÍ** es pertinente utilizarlo en la Iglesia Bautista de Cobija.

CAPÍTULO IV
CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

El desarrollo tecnológico actual facilita en gran medida la administración de información en las instituciones, facilitando el almacenamiento de la información en grandes cantidades superando las limitaciones físicas que se imponen. Con la realización de este software que se elaboró en base a los *Requerimiento de la iglesia Bautista de Cobija*, se estableció que es una labor muy importante que debe cumplir la Iglesia Bautista.

De acuerdo a la culminación del proceso del proyecto de grado se elaboran las siguientes conclusiones:

1. Se diagnosticó los procesos identificando los requerimientos de la Iglesia Bautista de Cobija.
2. Se realizó el respectivo análisis del sistema construyendo el modelo de objetos, modelo dinámico y el modelo funcional.
3. Se realizó el diseño del sistema en base a la metodología PU (Proceso Unificado).
4. Se desarrolló los módulos del sistema con sus respectivas funcionalidades o procesos.
5. Se realizaron las pruebas del funcionamiento de los módulos del sistema.
6. Se implementó el sistema en la Iglesia Bautista de Cobija.
7. Considerando como resultado con implantación del sistema se elaboró los reportes de asistencia y reportes financieros.

Después de su implementación del Sistema en la Iglesia Bautista de Cobija se logró registrar todos los informes de gestiones pasadas.

Los procesos se hacen de forma computarizada ya no llevan de forma manual.

4.2. RECOMENDACIONES

Al culminar el proyecto de grado se plantea las siguientes recomendaciones:

1. Se recomienda realizar la conexión de una red entre la central de la iglesia Bautista y sus dependencias.
2. Se recomienda desarrollar e implementar otros Módulos al Sistema por ejemplo de Egresos.
3. Se recomienda que el sistema de información sea utilizado en una red local en Cobija proponiendo un servidor en la iglesia central y conectando esta con una red LAN entre las sucursales afiliadas a la congregación.

BIBLIOGRAFIA

- ✓ Booch, G. Análisis y Diseño Orientado a Objetos con Aplicaciones. Addison-Wesley. (1996).
- ✓ Jacobson, I. Object Oriented Software Engineering. Addison-Wesley. (1992).
- ✓ Martin, J. y Odell, J. Análisis y Diseño Orientado a Objetos. Prentice Hall. (1994).
- ✓ Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. y Lorensen, W. Modelado y Diseño Orientado a Objetos. Prentice Hall. (1996).
- ✓ Jacobson, Ivar, Booch, Grady & Rumbaugh, James “El Proceso Unificado de Desarrollo de Software”, Pág. 5
- ✓ Pressman, Roger, Ingeniería del software, MC Graw Hill, 4ta edición, España, (1997)
- ✓ Pressmann, Roger S., Ingeniería del Software: un enfoque práctico, Pág. 526
- ✓ Kroenke, David M. "Procesamiento de Bases de Datos: Fundamentos, Diseño e Instrumentación", Pág. 485

Referencias Web

- ✓ Gaceta Oficial de Bolivia, Administración de Almacenes
- ✓ Pablo Castells, Programación Orientada a Objetos, Universidad Autónoma de Madrid
- ✓ Alberto Restrepo V. Hernán Darío Toro E., Programación Orientada A Objetos Conceptos Básicos <http://dis.eafit.edu.co/cursos/st-051/pooconc.html>
- ✓ Programación Orienta a Objetos (Migue Ángel Alvares, 2001) <http://www.desarrolloweb.com/articulos/499.php>
- ✓ Arquitectura del Software, www.vico.org visitado 20 de mayo del 2011.
- ✓ *Ajax. Pragmatic Ajax: A Web 2.0 Primer*, Justin Gehtland, Ben Galbraith, Dion Almaer (ISBN—13: 978-0976694083).
- ✓ Hojas de Estilo CSS, <http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es> visitado en 25 de mayo del 2011
- ✓ Lenguaje de programación PHP5, <http://www.opencontent.org/openpub/>, visitado el 28 de mayo del 2011.

- ✓ Metodo de Valuacion UEPS, <http://www.gerencie.com/metodo-ueps.html>, visitado el 30 de mayo del 2011.
- ✓ Metodo de Valuacion PEPS, <http://www.gerencie.com/metodo-peps.html>, visitado el 30 de mayo del 2011.
- ✓ Kardex, Microsoft Dynamics GP, kardex de inventario en julio (2008)