SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE PROCESOS Y CONTROL DE ACTIVIDADES EN LA PARROQUIA SAN MARTÍN DE PORRES DE TULUÁ VALLE DEL CAUCA

JULIA GRACIELA MARTÍNEZ ARENAS

Cód. 200955982

UNIVERSIDAD DEL VALLE SEDE TULUÁ

PROGRAMA DE INGENIERÍA DE SISTEMAS Y CIENCIAS DE LA COMPUTACIÓN

TULUÁ - VALLE

JUNIO, 2016

SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE PROCESOS Y CONTROL DE ACTIVIDADES EN LA PARROQUIA SAN MARTÍN DE PORRES DE TULUÁ VALLE DEL CAUCA

JULIA GRACIELA MARTÍNEZ ARENAS
Cód. 200955982

Trabajo de grado para optar al título de
Ingeniero de Sistemas

Director: Mg. Julián Andrés Rodas Laverde

UNIVERSIDAD DEL VALLE SEDE TULUÁ
PROGRAMA DE INGENIERÍA DE SISTEMAS Y CIENCIAS DE LA COMPUTACIÓN
TULUÁ - VALLE
JUNIO, 2016

Nota de Aceptación

Presidente del jurado

Jurado 1

Jurado 2

Tuluá, Junio de 2016

Resumen

En este trabajo se presenta Ecclesia, un sistema de información web para la gestión de procesos parroquiales que combina funcionalidades de administración de la información, eventos, seguimiento de estudiantes y determinación de ingresos. Ecclesia se fundamenta en un estudio del trabajo realizado diariamente en la Parroquia San Martín de Porres de Tuluá Valle del Cauca, a fin de analizar y determinar los motivos que generaban retardos en las actividades a desarrollar e ineficiencia en los canales de comunicación entre el personal. Para la recolección de datos fueron aplicadas tres técnicas, (I) la observación directa, (II) la entrevista no estructurada y (III) revisión documental. Los resultados obtenidos del estudio hicieron concientizar a la Parroquia de la necesidad de adoptar las nuevas tecnologías de la información como herramientas que permiten realizar un uso adecuado de los recursos, el tiempo, la información, entre otros.

La solución tecnológica desarrollada permite la reducción en los tiempos de respuesta de los procesos parroquiales, enfatizando especialmente la actividad sacramental orientada a los feligreses. Además asiste en la toma de decisiones presentando la información contenida de manera oportuna, es decir que permite la extracción de información real en el momento que se requiera.

Palabras clave: framework, gestión de procesos, parroquia, sistema de información.

Abstract

In this work Ecclesia is presented, an information system for parochial processes management which combines managing, events, student tracking and income determination functions. Ecclesia is founded in a work study performed daily at San MartÃn de Porres Parish in Tuluá â Valle, so the delay-generating motives for the task to be done could be analyzed and determined as well as the inefficiency in the personnel communication channels. Three techniques were applied to data collecting: Direct observation, non-structured interview and document reviewing. Obtained data in the study made clear the need to embrace new information technologies for the parish, such as tools that allow an adequate use of resources, time, information among others.

The technological solution developed allows for response time reduction in parochial processes, focusing specially on sacramental activity towards the congregation. It also assists on decision making displaying content information promptly, so it allows for real information extraction at the required time.

Keywords: framework, process management, parish, information system.

Dedicatoria

Este trabajo de grado lo dedico a DIOS como alfa y omega, por regalarme cada día un soplo de vida para seguir avanzando y hacer que mi vida sea llena de bendiciones, al igual va dirigido a esas personas que amo representadas por el majestuoso título de FAMILIA, de manera especial a mi abuela Graciela Cardona G., por ser la roca que me sustenta en aquellos momentos de flaqueza, por estar en los triunfos y en los tropiezos, simplemente por ser el reflejo del amor de DIOS hacia mí; finalmente lo dedico a aquellas almas que se llevaron consigo el título de PADRES, mi madre Ana R. Arenas C. y mi padre Isaías Martínez V., quienes dejaron en mí el ejemplo de lucha y perseverancia durante el día a día de la vida.

Julia Graciela Martínez Arenas

Agradecimientos

A la Parroquia San Martín de Porres de Tuluá Valle representada por el Pbro. Harold Londoño M., que contribuyeron de una u otra forma a la realización de este proyecto.

A mi director de trabajo de grado, el Ingeniero Julián A. Rodas L., por su dedicación y pujanza, siendo el componente necesario para la culminación de este trabajo.

A aquellas valerosas personas que vivieron conmigo este proceso de aprendizaje, mis amigos y compañeros, que compartieron sus conocimientos y experiencias para ayudarme en mi formación profesional y crecimiento personal.

A los docentes y toda la familia Univalle que con su esfuerzo me brindaron una educación de calidad e infraestructura idónea para el conocimiento.

Tabla de Contenido

1 Introducción **1**
1.1 Descripción general 1
1.2 Planteamiento del Problema 2
1.2.1 Descripción del Problema 2
1.2.2 Formulación del Problema 5
1.3 Objetivos 6
1.3.1 Objetivo General 6
1.3.2 Objetivos Específicos 6
1.4 Estructura del documento 6

2 Marco Referencial **7**
2.1 Marco Teórico 7
2.1.1 *Laravel* 9
2.1.2 *Cakephp* 10
2.1.3 *Yiiframework* 11
2.1.4 *Codeigniter* 12
2.1.5 *Simfony 2* 12
2.2 Plataforma de desarrollo 13
2.2.1 PHP (*Hypertext Preprocessor*) 13
2.2.2 MySQL 14
2.2.3 HTML (*HyperText Markup Language*) 14
2.2.4 CSS (*Cascading Style Sheets*) 15
2.2.5 JavaScript 15
2.2.6 Bootstrap 15
2.2.7 jQuery 16
2.2.8 Las TIC en la estrategia 16
2.3 Antecedentes 16

3 Aspectos del desarrollo de software **19**
3.1 Inicio 21

3.1.1 Análisis de Requerimientos 22
3.1.2 Diagrama Caso de Uso (Negocio) 23
3.1.3 Diagrama de Actividades 23
3.2 Elaboración 24
3.2.1 Diagrama de casos de uso 25
3.2.2 Especificación de casos de uso 26
3.2.3 Modelo ER 26
3.2.4 Diagrama de clases 27
3.2.5 Diagrama de secuencia 28
3.2.6 Prototipos de interfaz de usuario 29
3.2.7 Arquitectura Diagrama de componentes 30
3.3 Construcción 31
3.3.1 Descripción general del aplicativo web 32

4 Pruebas **38**
4.1 Desarrollo por pruebas unitarias 38
4.2 Metodología de pruebas de usabilidad 40

5 Conclusiones y trabajos futuros **43**
5.1 Conclusiones 43
5.2 Trabajos futuros 44

Referencias **45**

Listado de Tablas

2.1 Cuadro comparativo entre los frameworks 13
2.2 Características de los sistemas de información parroquiales del mercado 17

3.1 Resultado Fase Inicial 21
3.2 Requerimientos, módulo Actas Sacramentales 22
3.3 Información almacenada del feligrés 23
3.4 Especificación de caso de uso, Convirtiendo Boleta 26

4.1 Pasos, prueba de usabilidad 41

5.1 Requerimientos, módulo Catequesis 48
5.2 Requerimientos, módulo Agendamiento 48
5.3 Requerimientos, módulo Plan de acción 49
5.4 Requerimientos, módulo Presupuesto 49
5.5 Requerimientos, módulo Gestión de Usuarios 49
5.6 Información almacenada de las boletas 50
5.7 Información almacenada en la conversión de boleta a partida 50
5.8 Información almacenada de las partidas (parte 1) 51
5.9 Información almacenada de las partidas (parte 2) 51
5.10 Diagrama de actividades, proceso de Catequesis 52
5.11 Diagrama de actividades, Administración de eventos 53
5.12 Diagrama de actividades, proceso de Gestión de usuarios 54
5.13 Diagrama de actividades, proceso de Presupuesto 54
5.14 Diagrama casos de uso, módulo de Catequesis 55
5.15 Diagrama casos de uso, módulo de Agendamiento 55
5.16 Diagrama casos de uso, módulo de Plan de acción 56
5.17 Diagrama casos de uso, módulo de Presupuesto 56
5.18 Diagrama casos de uso, módulo de Usuarios 56
5.19 Diagrama casos de uso, módulo de Perfil 57
5.20 Diagrama casos de uso, módulo de Validación de usuarios 57
5.21 Especificaciones casos de uso, creando boleta 58
5.22 Especificaciones casos de uso, consultando boleta 59
5.23 Especificaciones casos de uso, imprimiendo boleta 59

5.24 Especificaciones casos de uso, modificando boleta 60
5.25 Especificaciones casos de uso, eliminando boleta 60

Lista de Figuras

1.1 Diagrama de procesos de la Parroquia San Martin de Porres 4
1.2 Relación objetivos específicos con resultados esperados 6

2.1 Arquitectura MVC aplicada a un framework web 9
2.2 Esquema del funcionamiento de PHP 14

3.1 Fases, iteraciones y disciplinas de la metodología RUP 20
3.2 Diagrama caso de uso (Negocio) Parroquia San Martín de Porres 23
3.3 Requerimientos, módulo Actas Sacramentales 24
3.4 Diagrama de casos de uso, módulo Actas Sacramentales 25
3.5 Modelo ER, módulo de Actas Sacramentales 27
3.6 Diagrama de clases, Ecclesia 28
3.7 Diagrama de secuencia, Convirtiendo Boleta 29
3.8 Prototipo de interfaz de usuario, Interfaz Principal 30
3.9 Requerimientos, módulo Actas Sacramentales 31
3.10 Actas sacramentales: Feligreses 33
3.11 Tipo de Boletas y Partidas 33
3.12 Actas Sacramentales: Boletas $\gg Crear$ 34
3.13 Actas Sacramentales: Partidas $\gg Crear$ 34
3.14 Agendamiento $\gg Crear$ 35
3.15 Plan de Acción $\gg Crear$ 35
3.16 Catequesis: Estudiantes $\gg Crear$ 36
3.17 Catequesis: Catequistas $\gg Crear$ 36
3.18 Catequesis: Notas 37
3.19 Presupuesto $\gg Consultar$ 37

4.1 Prueba PHPUnit, método Visit 38
4.2 Prueba PHPUnit, método GET 39
4.3 Prueba PHPUnit, llamados 39
4.4 Prueba entorno PHPUnit 40
4.5 Prueba entorno PHPUnit 40

5.1 Diagrama Entidad Relación Ecclesia 61

Lista de Anexos

Anexo A: Especificación de Requerimientos
Anexo B: Tablas de información
Anexo C: Diagramas de Actividades
Anexo D: Diagramas Casos de uso
Anexo E: Especificaciones casos de uso
Anexo F: Diagrama Entidad Relación

Capítulo 1

Introducción

1.1 Descripción general

En la actualidad la tecnología ha pasado de ser una herramienta más en la vida del ser humano a penetrar las diversas esferas del hombre; hablar de la era tecnológica y la revolución informática, como aspectos inherentes en la globalización de una sociedad cada vez más capacitad, ha dejado de ser una utopía, ya que a cada segundo es posible descubrir la necesidad de informar, gestionar y digitalizar todos los procesos y/o actividades de un mundo en desarrollo. En relación a lo anterior se hace realmente importante entender que la religión y sus diferentes procesos no deben ser lejanos a esta realidad.

Dado lo anterior, el objetivo del trabajo de grado que aquí se presenta, está orientado al desarrollo e implementación de un sistema de información en la Parroquia San Martin de Porres del municipio de Tuluá, que facilite la gestión, planificación y control de los procesos de actas, agendamiento, presupuesto, cronograma y plan de acción parroquial, y catequesis, así como la comunicación con su recurso humano, que son de gran importancia para la realización de las actividades de la Iglesia. En la actualidad, la gestión de estos procesos, no cuenta con el apoyo de las herramientas que ofrecen las Tecnologías de la Información, lo que impide el desarrollo de manera articulada y eficiente de los procesos, por ende no se puede medir el impacto de las actividades en la sociedad.

1.2 Planteamiento del Problema

1.2.1 Descripción del Problema

La Parroquia San Martín de Porres desde su creación hasta la fecha realiza todas sus funciones de manera manual, lo cual entorpece el mejoramiento y evaluación continua de sus procesos tales como agendamiento sacerdotal y litúrgico, actas, catequesis, presupuesto y cronograma general, el cual encierra las actividades planteadas del plan de acción y el cronograma parroquial; de igual manera no se cuenta con un canal de información efectivo entre las personas involucradas (Párroco, secretaria y líderes) en la realización de los diferentes procesos, lo que impide una retroalimentación adecuada que permita determinar el impacto social del servicio ofrecido a los feligreses [4].

Dado lo anterior, es importante agregar que según la aplicación, ejecución y planeación de los procesos se ven involucrados recursos económicos y humanos; en estos últimos se debe puntualizar que además del Párroco y la secretaria, la Parroquia cuenta con un grupo de líderes (laicos comprometidos), los cuales prestan su servicio sin remuneración alguna, estos se encargan de formar a través de la evangelización personas integras y coadyuvar en las actividades del plan de acción y el cronograma parroquial, además deben realizar una evaluación que refleje el resultado de cada tarea realizada y exponerla al Sacerdote para que esté aprecie aspectos a mejorar en próximas actividades, ya que de esto depende el mejoramiento de la calidad y pertinencia de los servicios ofrecidos a los feligreses. A continuación se detallan las funciones del servicio de la Iglesia:

Agendamiento sacerdotal, se elabora teniendo en cuenta los eventos religiosos y culturales que se presentan en la Parroquia y la Diócesis (celebraciones eucarísticas, visitas del Obispo, ordenación de Sacerdotes, entre otras), la demanda de la comunidad (visita a enfermos, confesiones, entre otras) y las actividades personales del Párroco.

Agendamiento litúrgico, es programado según la solemnidad (festividad eclesiástica) establecida para cada día del año, las reservaciones hechas por los feligreses (misas individuales y comunitarias) y elementos litúrgicos (vestimenta, libros, piezas sagradas, entre otras); esta agenda debe ajustarse a las actividades propuestas en el cronograma general debido a que no puede interferir en el buen desarrollo del proceso mencionado.

Actas (*partidas*), son documentos asentados en libros que certifican la celebración de los sacramentos (bautizo, confirmación y matrimonio) y exequias; es de aclarar que para la realización de las actas se debe haber recolectado una documentación básica pertinente a cada sacramento, tal es el caso del matrimonio, en el cual, los novios se acercan al despacho parroquial para suministrar la información que debe llevar el formato de expediente matrimonial, al cual deben anexar las actas de bautismo de cada uno y el comprobante del cursillo prematrimonial, posteriormente la secretaria les asigna

fecha y hora de entrevista con el Sacerdote y de celebración del sacramento, después de realizada la eucaristía se lleva el formato de partida de matrimonio a tipografía, donde digitan la información respectiva de los novios en un archivo digital que se imprime para finalmente asentarlo en el libro de matrimonios de la Parroquia y en el expediente de cada acta de bautismo. Este proceso se complementa haciendo un reporte anual donde se aprecie la cantidad de actas realizadas por cada sacramento.

Presupuesto, expresa la forma en que se van a aplicar los recursos disponibles en el futuro, además debe proporcionar toda la información concreta de sus recursos (monetarios, materiales, humanos, entre otros) y todo aquello que de soporte a su cálculo [11]. En el contexto de la Parroquia se manejan tres constancias diferentes, donde se tienen en cuenta los siguientes aspectos:

- Ingresos por motivo de actas, ofrendas, donaciones y diezmos, para los cuales se deben hacer entrega de certificados.
- Pago a trabajadores, servicios públicos, curia y seminario.
- Colectas.
- Registro del flujo de dinero, el cual se archiva en un libro contable y es para presentar rentas a la DIAN y al Obispo.

Cronograma parroquial, comprende las celebraciones fundamentales de la Iglesia (fiestas patronales, cuaresma, semana santa, adviento y navidad), las cuales tienen fecha única de realización pero varían en la hora, lugar y recursos tanto económicos como humanos.

Plan de acción parroquial, se elabora por actividades sociales que buscan responder a las diferentes necesidades de los barrios que conforman la comunidad parroquial, el conocimiento de estas necesidades se logra gracias a un estudio realizado por los líderes; es de aclarar que la ejecución de las componentes estipuladas en este proceso se pueden realizar paralelamente. Para su planeación se determina una fecha (uno o varios días), hora, lugar y recursos (humanos, económicos, materiales, entre otros).

En los últimos dos procesos mencionados con anterioridad, se hace indispensable la función que desempeñan de los líderes, mientras que la presencia del Sacerdote se debe asegurar en el primer caso por medio del agendamiento sacerdotal; dada la importancia que tiene el recurso humano con que cuenta la Parroquia, los procesos descritos no pueden ser realizados de manera paralela. En ambos casos cuando se termina cualquier

actividad esta pasa por una evaluación, en la cual se determina si lo programado fue ejecutado además del cumplimiento del objetivo por el que se realizó.

Catequesis, este proceso de enseñanza es impartido para tres sacramentos bautismo, primera comunión y confirmación, los cuales son planificados en un conjunto de temas según sea el caso; los encargados de esta formación son los lideres, estos deben llevar un registro de las personas que tienen a cargo con su respectiva asistencia al curso y calificaciones de desempeño si se requieren, a su vez si se presenta alguna anomalía con los feligreses esta debe ser reportada al Sacerdote.

En la Figura 1.1 se condensa el flujo de información de los procesos descritos con anterioridad en la Parroquia San Martin de Porres.

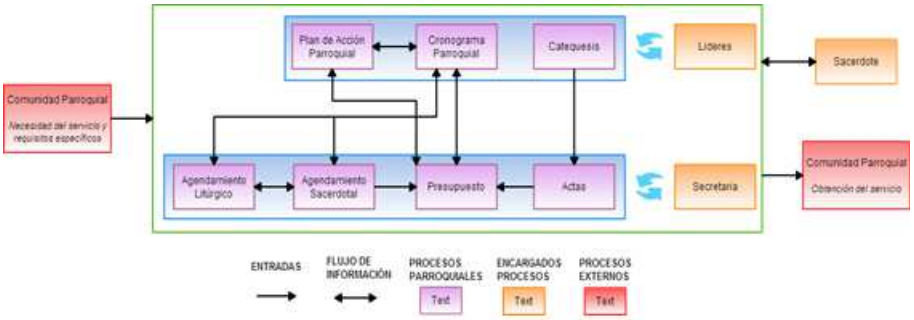

Figura 1.1: Diagrama de procesos de la Parroquia San Martin de Porres

Teniendo en cuenta lo anterior, se evidencia la necesidad que el Párroco tiene de controlar y evaluar cada una de las funciones desarrolladas por la Parroquia San Martin de Porres, para lo cual, él expresa que no se hace de la mejor manera por factores de tiempo, información generalizada que no permite medir cual es el verdadero impacto de las actividades en la sociedad, dificultad y poca calidad de la información. Es significativo tener en cuenta que cuando se habla de calidad de información, se busca que esta sea oportuna, actual, frecuente, completa, exacta, importante, ordenada y accesible. Dadas las necesidades particulares que presenta la Parroquia, se debe encontrar una herramienta que permita una comunicación eficiente entre su recurso humano (Párroco, secretaria y líderes), que facilite información con calidad y que a su vez permita analizar y evaluar las tareas que se vienen realizando manualmente, con el fin de lograr mayor productividad y eficiencia. Lo que en consecuencia la Parroquia San Martin de Porres debe considerar la creación de un sistema computacional que permita apoyar la gestión de procesos y controlar las actividades propias de su servicio.

1.2.2 Formulación del Problema

¿Cómo mejorar la planificación, control y evaluación de los procesos realizados en la Parroquia San Martin de Porres en el Municipio de Tuluá?

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema de información que permita planificar, controlar y evaluar los procesos eclesiales (actas, agendamiento, presupuesto, cronograma, plan de acción parroquial y catequesis) de la Parroquia San Martín de Porres de Tuluá Valle del Cauca.

1.3.2 Objetivos Específicos

#	Objetivo Específico	Sección	Apartado
1	Crear un módulo que permita gestionar las actas sacramentales de los feligreses.	3.3.1	A
2	Crear un módulo que permita al Sacerdote observar, modificar y evaluar a través de la web las diferentes actividades encaminadas al cumplimiento de los procesos de agendamiento, plan de acción parroquial y catequesis.		B
3	Crear un módulo de presupuesto que permita el seguimiento del flujo de caja a través del tiempo.		C
4	Implementar el sistema de información en la Parroquia San Martín de Porres.	3.3	

Figura 1.2: Relación objetivos específicos con resultados esperados

1.4 Estructura del documento

El presente documento se divide en cuatro capítulos, que se describen a continuación:

En el capítulo 2 se presentan el marco referencial, donde se encuentra la teoría que va a fundamentar el proyecto y los antecedentes sobre herramientas tecnológicas e informáticas orientadas a generar control de los procesos parroquiales.

En el capítulo 3 se presenta el proceso de ingeniería de software llevado a cabo para cumplir con los objetivos propuestos y adquirir la obtención de resultados esperados.

En el capítulo 4 se presentan los diferentes escenarios de pruebas para medir la efectividad y pertinencia del sistema frente a los objetivos y requerimientos propuestos en el proceso de desarrollo.

Este trabajo finaliza con el Capítulo 5, conclusiones y trabajos futuros.

Capítulo 2

Marco Referencial

2.1 Marco Teórico

La parroquia es una determinada comunidad de fieles constituida de modo estable en la Iglesia particular, cuya cura pastoral, bajo la autoridad del Obispo diocesano, se encomienda a un Párroco, como su pastor propio, para que cumpla las funciones de enseñar, santificar y regir, con la cooperación también de otros presbíteros o diáconos, y con la ayuda de fieles laicos [5]. En Colombia las parroquias son centros de cuarto nivel en la jerarquización de la Iglesia Católica, el primer nivel está constituido por provincias eclesiasticas (agrupación de varias diócesis vecinas), el segundo nivel por arquidiócesis (diócesis con un rango superior a las convencionales) y diócesis (territorio en que tiene y ejerce jurisdicción espiritual un Obispo), y en el tercer nivel se instauran los arciprestazgos (territorio con jurisdicción).

Las parroquias en correlación con sus derechos y deberes fundamentales del Derecho Canónico y las regulaciones del Estado Colombiano, refleja una amplia presencia en la sociedad a través de diversos servicios como evangelización, educación, atención de enfermos, personas encarceladas, víctimas del conflicto, niños, jóvenes y familias, asistencia social y promoción humana. Es decir que estás adelantan diversos programas, de acuerdo con las necesidades de su realidad particular, mediante la colaboración de laicos e instituciones reconocidas por las comunidades [6]. La planeación, desarrollo y resultados de estos trabajos, al igual que los procesos administrativos propios del servicio religioso, son fuentes de información que permiten la toma de decisiones basadas en pro del mejoramiento de los mismos. Es por esto que la Iglesia desde una perspectiva religiosa enmarca los beneficios y ventajas que brindan las tecnologías de la información y la comunicación, con el fin de promover el uso de estas en los diversos servicios mencionados anteriormente, para fomentar su correcto desarrollo con

vista al progreso humano, la justicia y la paz, y para la construcción de la sociedad en los ámbitos local, nacional y comunitario a la luz del bien común y con espíritu de solidaridad [14].

De acuerdo a lo anterior se hace importante mencionar a las tecnologías informáticas, entre estas, a los sistemas de información (SI) como conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización [12]; por lo general los sistemas de información son divididos en función de la principal finalidad a la cual están destinados, surgiendo así dos tipos de SI genéricos: (I) SI Transaccionales, se encargan de manera específica de procesar las transacciones de información provocadas por las interacciones formales entre el entorno y la organización; y (II) SI Decisorios, se dedican a dar apoyo a los diferentes tipos de procesos de toma de decisiones llevados a cabo por parte de los directivos, gestores y otros profesionales de la organización [12].

Debido a la inclusión de nuevas tecnologías en los navegadores web, se ha permitido el abandono de los SI tradicionales construidos con aplicaciones de escritorio para pasar a SI basados en aplicaciones de Internet que se ejecutan y visualizan en un servidor, sin embargo, el desarrollo de un sistema de información con esta naturaleza requiere el conocimiento y control de un número elevado de técnicas y tecnologías, desde tecnologías para la construcción rápida de sistemas de información en web hasta las técnicas que permiten diseñar interfaces de usuario ágiles, flexibles y accesibles. En respuesta a los requerimientos mencionados, las tecnologías informáticas recomiendan la utilización de frameworks de desarrollo, ya que éstos ofrecen una estructura conceptual y tecnológica con un conjunto de bloques predefinidos de software los cuales se pueden extender o personalizar de acuerdo a las necesidades de un proyecto. En la ingeniería de software la palabra framework es usada para describir una diversidad de modelos y ambientes que tienen por común una característica mínima: establecen un marco de trabajo para desarrollar un producto de un género determinado; o también se define como la abstracción del esqueleto de una solución a una serie de problemas relacionados [8]. Un framework por tanto puede ser definido como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación específica.

Comúnmente los frameworks web están basados en el patrón de arquitectura MVC (Model View Controller). En la figura 2.1, se muestra como esté patrón organiza la aplicación en tres modelos separados, 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.

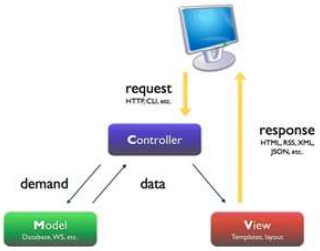

Figura 2.1: Arquitectura MVC aplicada a un framework web

Las siguientes son algunas características de los frameworks web.

- Utilizan un solo servlet que tiene la función de controlador, para toda la aplicación o gran parte de ella.
- Se configura el servlet controlador a través de propiedades, para indicarle a quien delegar la responsabilidad de atender peticiones entrantes. En algunas ocasiones esas propiedades están indicadas de acuerdo a los URLs y el URL entrante.
- Las vistas pueden tener nombres claves, sin necesidad que exista una relación con el nombre del archivo de la vista. El framework se encarga de realizar la conversión para obtener el nombre de la vista que se debe cargar para ser desplegada.

En la actualidad existe una gran variedad de proyectos de frameworks web para los diferentes lenguajes de programación .NET, Ruby, Java, entre otros. Esto debido a la facilidad que ofrecen a la hora de crear sistemas desde lo más simple, hasta sistemas cada vez más robustos. Como una de las plataformas de desarrollo web más conocidas se encuentra PHP, por su flexibilidad, potencia, alto rendimiento, entre otras características. Por lo que este lenguaje posee un gran abanico de frameworks que son importantes de analizar, entre ellos se encuentran los siguientes [7]:

2.1.1 *Laravel*

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Laravel se basa en arquitectura MVC y tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla permitiendo multitud de funcionalidades.

Principales características:

- Cuenta con un código modular y extensible por medio de un administrador de paquetes.

- Posee un sistema de enrutamiento rápido y eficiente, HTTP Routing, que permite relacionar las partes de la aplicación con las rutas que ingresa el usuario en el navegador.
- Ofrece el uso de Middleware para el análisis y filtro de llamadas HTTP en el servidor, con el fin de evitar problemas de tipo Cross-Site-Scripting (XSS) y otras medidas de seguridad.
- Provee un robusto sistema de caché configurable.
- Provee la autenticación de usuarios de forma nativa e incluye la opción recordar al usuario. Además permite incluir parámetros adicionales.
- Provee una API que permite la automatización de tareas con la tecnología Gulp, definiendo el uso pre-procesadores para comprimir CSS y JavaScript.
- Cuenta con la encriptación de datos, OpenSSL y cifrado AES-256-CBC.
- Facilita la definición, registro y escucha de eventos con la propiedad listen de EventServiceProvider, la cual contiene una lista de los eventos registrados en la aplicación.
- Incluye una capa para manejo de bases de datos que cuenta con un ORM (Object-Relational-Map) llamado Eloquent.
- Ofrece un framework transversal para realizar Unit Testing, PHPUnit.
- Cuenta con Micro-servicios y APIs para proyectos de menor envergadura.
- Se puede ejecutar procesos largos y complejos en segundo plano usando listas de tareas.
- Facilita la integración del desarrollo con el sistema de cobros Stripe.

2.1.2 *Cakephp*

Es un framework de código abierto para el desarrollo aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails. Utiliza el patrón de diseño MVC y su objetivo principal es que el trabajo sea de forma estructurada y rápida, sin pérdida de flexibilidad.

Principales características:

- Es compatible con PHP4 y PHP5

- Posee licencia MIT.
- Provee un CRUD integrado para la interacción con la base de datos.
- Soporta scaffolding, una técnica para especificar el uso de la base de datos en la aplicación.
- Provee un despachador de peticiones que permite acceder a la aplicación a través de URLs amigables y configurables
- Permite la generación de plantillas de manera rápida y flexible usando la sintaxis de PHP y asistentes o helpers.
- Incorpora asistentes de construcción de vistas para la automatización de la generación de código en AJAX (Asynchronous JavaScript and XML), JavaScript, formularios HTML, entre otros.
- Ofrece componentes de email, cookie, seguridad, sesión y manejo de solicitudes
- Implementa Data Sanitization para determinar el ingreso y formato de los datos según las reglas de validación.

2.1.3 *Yiiframework*

Es un framework MVC de alto rendimiento basado en componentes, PHP y framework de aplicaciones web. Yii es un acrónimo para Yes, it is (en español: ¡Sí lo es!).

Principales características:

- Incluye Database Access Objects (DAO), query builder, Active Record y migración de BD.
- Incorpora formularios de entrada, validación y soporte para AJAX.
- Provee Gii, una herramienta de generación de código para acelerar el desarrollo de aplicaciones.
- Permite la integración con otros frameworks, como Zend o PEAR.
- Ofrece variedad de plugins y widgets gratuitos de código abierto.
- Ofrece herramientas para realizar pruebas unitarias y funcionales, basadas en PHPUnit y Selenium.
- Soporta el esquema de caching por capas y permite el cambio del almacenamiento del cache.

- Permite la generación automática de WSDL, especificaciones y administración de peticiones Web Service.

2.1.4 *Codeigniter*

Es un framework pequeño de código abierto para la creación rápida de aplicaciones web. Contiene una serie de librerías estructuradas que sirven para el desarrollo de soluciones web, implementa el paradigma MVC.

Principales características:

- Es compatible con PHP4 y PHP5
- Compatible con un gran número de entornos o servidores, incluyendo sistemas de alojamiento compartido.
- Ofrece documentación fácil de seguir y asimilar, porque está escrita en modo tutorial.
- Tiene una curva de aprendizaje baja, lo que hace que sea menos rígido que otros frameworks.
- Permite la aplicación de módulos o clases de manera opcional, así su núcleo se vuelve bastante ligero, lo que permite que el servidor no se sobrecargue.

2.1.5 *Simfony 2*

Es un framework de código abierto diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón MVC. Simfony trabaja separando la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.

Principales características:

- Posee licencia MIT.
- Tiene independencia del sistema gestor de base de datos. Su capa de abstracción y el uso de ORM, permite cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

- Permite la integración de bibliotecas de otros fabricantes.
- Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.
- Ofrece el uso de templates y layouts que pueden ser construidos por diseñadores de HTML que no posean conocimientos del framework.
- Posee un sistema de enrutamiento y URLs inteligentes.
- Soporta validación automática de formularios, lo cual asegura mejor calidad de los datos en las base de datos.

En la tabla 2.1, se muestra un resumen de características tenidas en cuenta para el desarrollo de la aplicación que se presenta en este trabajo de grado.

PHP Framework	ORM	Templates	Caching	Auth Module	Modules	Testing
Laravel	✓	✓	✓	✓	✓	✓
Cakephp	✓	-	✓	✓	✓	✓
Yii	✓	✓	✓	✓	✓	✓
Codeigniter	-	✓	✓	-	-	✓
Simfony 2	✓	✓	✓	✓	✓	✓

Tabla 2.1: Cuadro comparativo entre los frameworks

En el ejercicio de conocer los diferentes framework PHP, se pudo concluir que Laravel es el apropiado para el desarrollo del software Ecclesia, ya que cumple con los criterios descritos en la tabla 2.1 y conceptos cómo: desarrollo activo, comunidad activa y buena documentación.

2.2 Plataforma de desarrollo

2.2.1 PHP (*Hypertext Preprocessor*)

PHP es un lenguaje de código abierto muy popular y especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML . Esta tecnología permite realizar páginas web dinámicas cuyo contenido puede ser completa o parcialmente generado en el momento de invocar la página, gracias a la información obtenida de un formulario o extraída de una base de datos. La característica que distingue a PHP de otros lenguajes de programación es su ejecución del lado del servidor, es decir, el código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página resultante para ser enviada al explorador [10]. Lo descrito anteriormente se expone en la figura 2.2.

Figura 2.2: Esquema del funcionamiento de PHP

2.2.2 MySQL

MySQL es un sistema administrador de bases de datos relacionales (RDBMS, Relational Data Base Management System) de código abierto, desarrollado en C, C++ e independiente del sistema operativo. Este gestor a su vez puede verse como un sistema cliente/servidor, ya que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, es decir, cada vez que se establece una conexión con el servidor este crea un subproceso para manejar la solicitud del cliente, controlando el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso [10].

El conjunto de características de MySQL han logrado que se convierta en uno de los sistemas gestores de bases de datos más utilizados en la actualidad, no solo por pequeñas empresas sino también por algunas grandes corporaciones, como: Wikipedia, Google, Facebook, Twitter, Flickr y YouTube.

2.2.3 HTML (*HyperText Markup Language*)

HTML es el lenguaje utilizado en Internet para definir las páginas del World Wide Web, básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que pueden componer una página web, como: imágenes, listas, videos, entre otros . Cabe destacar que HTML no es un lenguaje de programación ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de dichos entornos, por lo que HTML genera únicamente páginas web estáticas, sin embargo, esté se puede usar en conjunto con diversos lenguajes para la creación de sitios dinámicos.

2.2.4 CSS (*Cascading Style Sheets*)

CSS, en español Hojas de Estilo en Cascada, es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación , es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille.

CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, entre otros, modificando así la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

2.2.5 JavaScript

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para crear páginas web dinámicas. Este lenguaje posee varias características, entre ellas se pueden mencionar que es basado en acciones, posee menos restricciones, independiente de la plataforma, ejecutable por el lado del cliente Navigator JavaScript, y el lado del servidor LiveWire Javascript [2].

La Web 2.0 se basa en el uso de Javascript para implementar aplicaciones enriquecidas que son capaces de realizar todo tipo de efectos, interfaces de usuario y comunicación asíncrona con el servidor por medio de Ajax. Entre las aplicaciones que usan este lenguaje se encuentran: Google, Facebook, Twitter, Outlook, entre otros.

2.2.6 Bootstrap

interfaces web con CSS y JavaScript, que ayudan a adaptar la interfaz dependiendo del tamaño del dispositivo en el que se visualice. A la hora de crear interfaces Bootstrap permite crear diseños simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos .

El framework contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz así como extensiones opcionales de JavaScript.

2.2.7 jQuery

La estrategia empresarial, a veces también llamada gestión estratégica de empresas, es la búsqueda deliberada de un plan de acción que desarrolle la ventaja competitiva de una empresa y la acentúe, de forma que ésta logre crecer y expandir su mercado reduciendo la competencia. La estrategia articula todas las potencialidades de la empresa, de forma que la acción coordinada y complementaria de todos sus componentes contribuya al logro de objetivos definidos y alcanzables. La planeación estratégica es el proceso mediante el cual quienes toman decisiones en una organización, obtienen, procesan y analizan información pertinente, interna y externa, para evaluar la situación presente y nivel de competitividad, para anticipar y decidir sobre el direccionamiento de la institución a futuro [13].

2.2.8 Las TIC en la estrategia

jQuery es una librería de JavaScript, rápida, pequeña y con muchas funcionalidades, que permite la manipulación de los objetos DOM, eventos, animaciones y la llamada a funciones AJAX, a través de un API, de una manera mucho más simple y sencilla .

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

2.3 Antecedentes

En la actualidad son pocas las herramientas que se han diseñado y desarrollado para dar apoyo a los procesos sacramentales, pastorales y administrativos de la Iglesia Católica, por ende la inclusión de las tecnologías de la información en las parroquias es poca. De esta forma se hace indispensable conocer las características más relevantes en cuanto a casos de uso y servicios ofrecidos, para en base a estos identificar las funcionalidades imprescindibles para brindar solución al problema parroquial.

A continuación se encuentran diferentes sistemas de escritorio con ciertas características en común, resumidas en la tabla 2.2.

En base a la Tabla 2.2, serán expuestos los tres software con sus respectivas características. A nivel nacional se encuentra el Sistema de Información Parroquial o más conocido como Sistematización SIP ; el cual responde en un alto porcentaje a las necesidades de las parroquias en el territorio nacional, siendo la herramienta más usada en

Software	Open Source	Actas Sacramentales	Catequesis	Agenda	Presupuesto	Plan de Acción	Directorio	Celebraciones	Cementerio y Osarios
SIP	-	✓	-	-	✓	-	-	✓	✓
AsaaDioc	✓	✓	-	✓	✓	-	✓	✓	-
Office Eclesial	-	✓	✓	✓	✓	✓	✓	✓	-

Tabla 2.2: Características de los sistemas de información parroquiales del mercado

Colombia. En este se destacan 4 módulos, denominados de la siguiente manera: (I) Sistema de partidas, cumple su función a través de la generación de datos estadísticos por sacramento e impresión de partidas y certificados; (II) Sistema de contabilidad, se encuentra implementado bajo las normas internacionales NIIF (Normas internacionales de información financiera) y permite generación de comprobantes de pago; (III) Sistema de misas, permite el registro de intenciones individuales y comunitarias, consulta e impresión de cartelera; (IV) Sistema de osarios y cementerios, controla la administración de bóvedas y osarios e informes de exhumados.

Como referente internacional para este trabajo se describe el Sistema de Administración Parroquial ASaaDioc , el cual ha sido desarrollado en Monterrey, México; dicho sistema consta de tres áreas: (I) área sacramental, aborda bautismos, confirmaciones, primeras comuniones, presentaciones y matrimonios; (II) área administrativa, contiene servicios, ingresos y egresos, agenda, catálogos y reportes; (III) área pastoral, administra los grupos parroquiales, evangelización y censos. AASADIOC es una herramienta de uso gratuito, pero la dificultad de acceder a está radica en su disponibilidad sólo para las Parroquias de Monterrey.

Finalmente se describe el sistema parroquial Office Eclesial del centro de formación y desarrollo de la RIIAL Nuestra Señora de Guadalupe; este sistema consta de ocho módulos: (I) Directorio, permite cargar la información de las parroquias que componen la diócesis; (II) Personas, guarda la información relacionada con los feligreses y/o laicos de la parroquia; (III) Catequesis, registra cursos, catequistas, aulas y catequizándoos; además de permitir agendar el año completo de la catequesis; (IV) Económico, contabiliza los ingresos y egresos, automatiza los libros contables y balances, y gestiona el pago de los colaboradores parroquiales; (V) Agenda, permite el manejo de una o múltiples agendas; (VI) Familias, registra la información del grupo familiar basándose en las personas registradas del tercer módulo; (VII) Celebraciones, agenda celebraciones sacramentales y las intenciones de las mismas; (VIII) Sacramental, incorpora, gestiona e imprime actas sacramentales. Este software está desarrollado bajo Fox Pro. Aunque este programa es de distribución gratuita en todas las diócesis de América Latina, no es aplicable a Colombia, ya que no cumple con algunos requerimientos del Episcopado Colombiano. Puede utilizarse sobre Windows 3.11, 95, 98, 2000, NT y XP, con un equipo mínimo de tipo 486 DX con 16 Mb de memoria RAM.

Teniendo en cuenta las soluciones descritas que existen en el mercado, se puede decir que, el acceso de estas aplicaciones son de elevado costo, aun las open source ya que no se ajustan a la necesidad que tiene la Parroquia. Este proyecto busca la consolidación de nuevas tecnologías como un sistema de información basado en una aplicación web para la Parroquia San Martín de Porres, con la que se permitirá la planificación, control y evaluación de los procesos eclesiales (actas, agendamiento, presupuesto, plan de acción parroquial y catequesis).

Capítulo 3

Aspectos del desarrollo de software

En el presente capítulo se muestra el proceso de desarrollo de software del sistema de información con la documentación obtenida; además se exponen las diferentes tecnologías usadas para la implementación del mismo y se evidencia la obtención de resultados esperados.

La metodología de desarrollo de software seleccionada para llevar a cabo el proceso de elaboración del presente trabajo de grado fue RUP (*Rational Unified Process*). Esta tiene como objetivo ordenar y estructurar el desarrollo de software mediante un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema "Software". El proceso de desarrollo de RUP se describe desde tres perspectivas: perspectiva dinámica, contiene las fases (inicio, elaboración, construcción y transición) del modelo sobre el tiempo; perspectiva estática, definición del quién hace qué, cómo y cuándo; y perspectiva práctica, aplicación de las buenas prácticas de Ingeniería del Software [1].

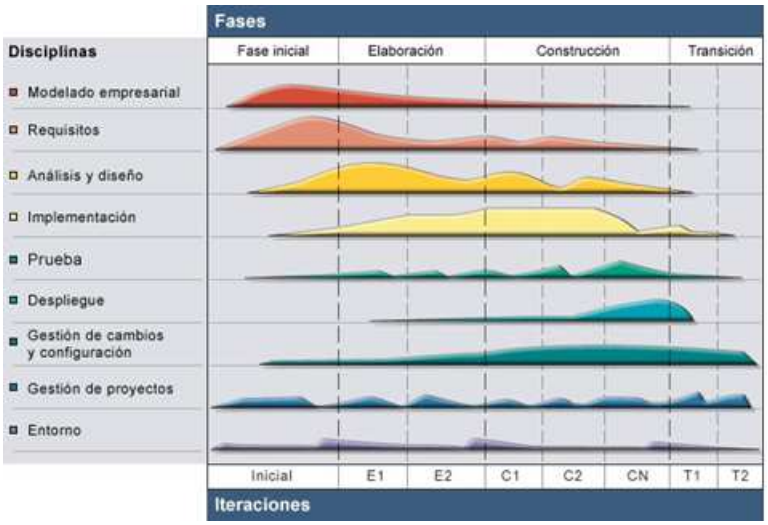

Figura 3.1: Fases, iteraciones y disciplinas de la metodología RUP

La metodología RUP se repite a lo largo de una serie de iteraciones que constituyen la vida de un sistema desde su nacimiento hasta su muerte en cuatro fases. La cantidad de iteraciones realizadas en cada fase depende principalmente del proyecto, en donde el trabajo se divide en partes pequeñas o mini proyectos de los que se obtiene un incremento que produce un crecimiento en el producto. El proceso iterativo e incremental de la metodología permite analizar cada iteración después de su finalización, con el fin de reajustar los objetivos para las siguientes iteraciones y así mitigar los riesgos que aún queden en el desarrollo del proyecto.

La fase inicial tiene como objetivo la comunicación con el cliente y las actividades de planeación, donde se establece el caso del negocio para el sistema, así como la identificación de todas las entidades externas que interactúan con el sistema y sus respectivas iteraciones. La fase de elaboración tiene como fin desarrollar un entendimiento del dominio del problema, crear un marco de trabajo arquitectónico para el sistema, desarrollar el plan del proyecto e identificar los riesgos claves. Al finalizar esta fase se debe tener el modelo de requerimientos del sistema (UML), una arquitectura y un plan de desarrollo. La fase de construcción es el diseño del sistema, la programación, las pruebas y la integración de todas las partes del sistema software. Finalmente en la fase de transición, el sistema software se entrega a los usuarios finales para sus respectivas pruebas en un entorno real.

Características:

- Guiado/Manejado por casos de uso
- Centrado en la arquitectura

- Iterativo e incremental
- Desarrollo basado en componentes
- Disciplinada asignación de tareas
- Control de cambios
- Utilización de un único lenguaje de modelo (UML)

3.1 Inicio

En esta fase se analizó las tareas que deben ser realizadas en pro del cumplimiento satisfactorio de los procesos realizados en la Parroquia San Martín de Porres de Tuluá â Valle, para la posterior construcción de los requerimientos funcionales y no funcionales del sistema, los cuales fueron clasificados por medio de módulos, donde cada uno de estos representa el cumplimiento de un objetivo específico del proyecto. A partir de esto se realizaron los diagramas de: (I) Diagrama de casos de uso del negocio, describe la funcionalidad de la Parroquia (sistema) independiente de la implementación, disponible en la sección 3.1.2, y (II) Diagrama de actividades, describen la secuencia de las diferentes actividades que hacen parte del negocio, disponible en la sección 3.1.3. En la tabla 3.1 se presenta los módulos establecidos para el sistema, junto con la cantidad de requerimientos y usuarios asociados.

Resultado Fase Inicial	Cantidad	Detalles	
Módulos	5	Módulo de Usuarios	
		Módulo de Perfiles	
		Módulo de Actas Sacramentales	
		Módulo de Catequesis	
		Módulo de Agendamiento	
		Módulo de Presupuesto	
		Módulo de Plan de Acción	
Requerimientos	60	Módulo de Actas Sacramentales	16
		Módulo de Catequesis	15
		Módulo de Agendamiento	5
		Módulo de Plan de Acción	7
		Módulo de Presupuesto	6
		Módulo de Gestión de Usuarios	11
Usuarios	4	Párroco	
		Secretaría	
		Catequista	
		Líder	

Tabla 3.1: Resultado Fase Inicial

3.1.1 Análisis de Requerimientos

El proceso de análisis de requerimientos se lleva a cabo para la comprensión y definición de los servicios que se quieren del sistema. En la tabla 3.2 se muestran los requerimientos funcionales del módulo de actas sacramentales, los demás requerimientos se encuentran en el apartado de anexos A.

	Documento: Requerimientos	**Fecha:** 08/03/2015 **Revisión:** 001 **Página:** 1
	Módulo: Actas Sacramentales **Objetivo 1**: Crear un módulo que permita gestionar las actas sacramentales de los feligreses	
#	**Requerimiento**	**Categoría**
Módulo de Feligreses		
1	El sistema debe permitir al usuario registrar feligreses con información específica Anexo C (*feligrés*).	F
2	El sistema debe permitir al usuario consultar los diferentes feligreses registrados.	F
3	El sistema debe permitir al usuario modificar la información de los feligreses.	F
4	El sistema debe permitir al usuario eliminar feligreses.	F
Módulo de Boletas		
5	El sistema debe permitir al usuario crear boletas con datos específicos según su tipo Anexo C (*boleta*).	F
6	El sistema debe permitir al usuario asignar un feligrés registrado a la boleta.	F
7	El sistema debe permitir al usuario consultar las boletas sacramentales registradas.	F
8	El sistema debe permitir al usuario modificar boletas sacramentales.	F
9	El sistema debe permitir al usuario eliminar boletas sacramentales.	F
10	El sistema debe permitir al usuario imprimir boletas sacramentales.	F
11	El sistema debe permitir al usuario generar partidas sacramentales.	F
Módulo de Partidas		
12	El sistema debe permitir al usuario crear partidas sacramentales Anexo C (*partida*).	F
13	El sistema debe permitir al usuario consultar las partidas sacramentales.	F
14	El sistema debe permitir al usuario modificar partidas sacramentales.	F
15	El sistema debe permitir al usuario imprimir las partidas sacramentales.	F
16	El sistema debe permitir generar reportes de los sacramentos celebrados en la Parroquia.	F

Tabla 3.2: Requerimientos, módulo Actas Sacramentales

El documento de requerimientos se compone de una tabla con su fecha de creación, número de revisión y paginación, nombre del módulo y el objetivo específico al que hace referencia junto con la lista de requerimientos que poseen un identificador, descripción y categoría que los clasifica como requerimientos funcionales o no funcionales. Para apoyar la recolección de información de requerimientos y la posterior elaboración del diagrama relacional de la base de datos, se organizó un documento que está compuesto de diferentes tablas correspondientes a cada módulo o sub módulo a desarrollar. Cada tabla tiene los datos que serán almacenados en el sistema con su tipo y condiciones. En la tabla 3.3 se presentan la información correspondiente al usuario feligrés, las demás tablas se encuentran en el apartado de anexos B.

ANEXO C (*Feligrés*)		
Dato	**Tipo de Dato**	**Condición**
Nombre	Texto	Not Null
Tipo de documento	Seleccionable	-
Nro. del documento	Numérico	Not Null
Teléfono	Numérico	-
Celular	Numérico	Not Null
Fecha de nacimiento	Date	Not Null
Lugar de nacimiento	Texto	Not Null
Nombre del padre	Texto	-
Nombre de la madre	Texto	-

Tabla 3.3: Información almacenada del feligrés

3.1.2 Diagrama Caso de Uso (Negocio)

Para la comprensión de los procesos de la Parroquia en términos de actores, se representa cada proceso como un caso de uso y estos a su vez son relacionados con los actores influyentes. Los actores son la abstracción de una persona o sistema que en el negocio realizan una o varias actividades e interactúan con otros actores. En la Figura 3.2 se muestra el diagrama de caso de uso obtenido para los procesos ejecutados en la Parroquia.

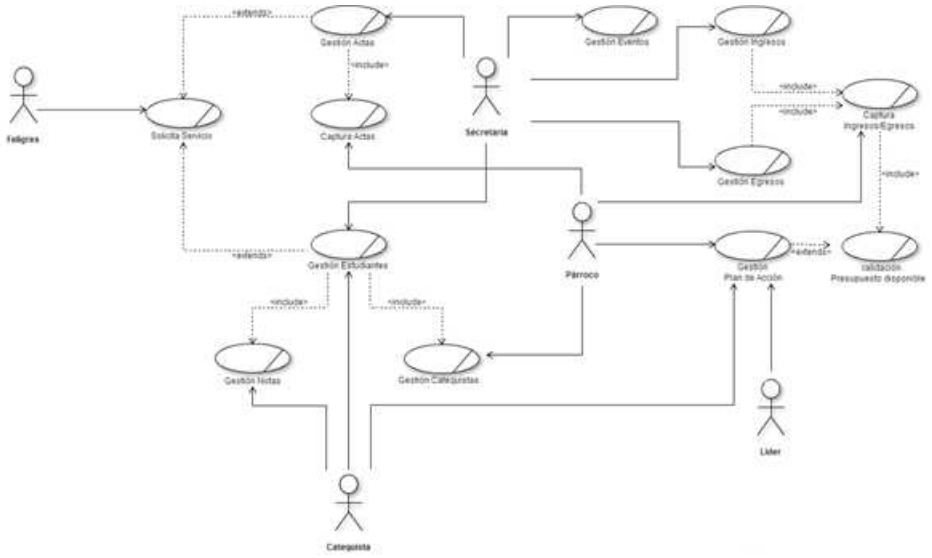

Figura 3.2: Diagrama caso de uso (Negocio) Parroquia San Martín de Porres

3.1.3 Diagrama de Actividades

Permite mostrar el flujo o la secuencia de procesos con las diferentes rutas de decisión, siendo esto un punto clave para la determinación de escenarios y casos de prueba del

sistema. En la Figura 3.3 se presenta el diagrama de actividades para el proceso de actas sacramentales.

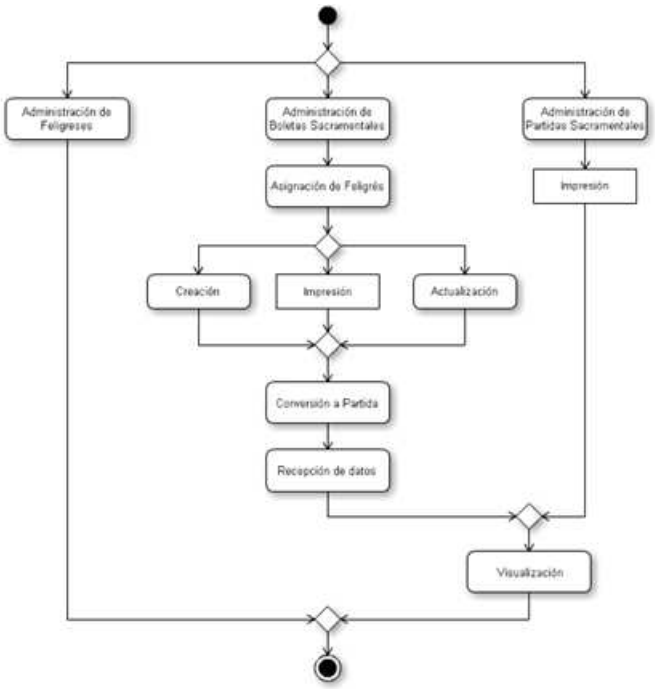

Figura 3.3: Requerimientos, módulo Actas Sacramentales

Los diagramas de actividades referentes a los otros procesos desarrollados en la Parroquia se encuentran en el apartado de anexos C.

3.2 Elaboración

Culminado el análisis de los procesos parroquiales en la fase de iniciación, se procede a estudiar los artefactos generados para establecer la arquitectura del sistema. En las iteraciones de esta fase se logra la obtención de (I) Diagramas de casos de uso correspondientes a los requerimientos descritos en la primera fase, disponible en la sección 3.2.1; (II) Especificación de casos de uso, descripción detallada de cada caso de uso definido, disponible en la sección 3.2.2; (III) Modelo ER, representación de las entidades del sistema con sus atributos e interrelaciones, disponible en la sección 3.2.3; (IV) Diagrama de clases, pilar básico del modelado UML para mostrar que puede hacer el sistema y su posible construcción, disponible en la sección 3.2.4; (V)Diagramas de secuencia, describen gráficamente la interacción usuario-sistema, disponible en la sección 3.2.5; y (VI) Prototipos de usuario, diseño de interfaz de usuario que será implementado, disponible en la sección 3.2.6. La aceptación y aprobación del prototipo de la

arquitectura del sistema marca el cierre de esta fase en la sección 3.2.7.

3.2.1 Diagrama de casos de uso

Describen la secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Con la ayuda de este diagrama, se puede analizar y comunicar: los escenarios en los que el sistema o aplicación interactúa con personas, organizaciones o sistemas externos; los objetivos que el sistema o aplicación contribuye a lograr y el ámbito del sistema. Estos diagramas son utilizados para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo.

En la figura 3.4 se presenta el diagrama de caso de uso para el módulo de actas sacramentales, los diagramas referentes a los otros módulos se encuentran en el apartado de anexos D.

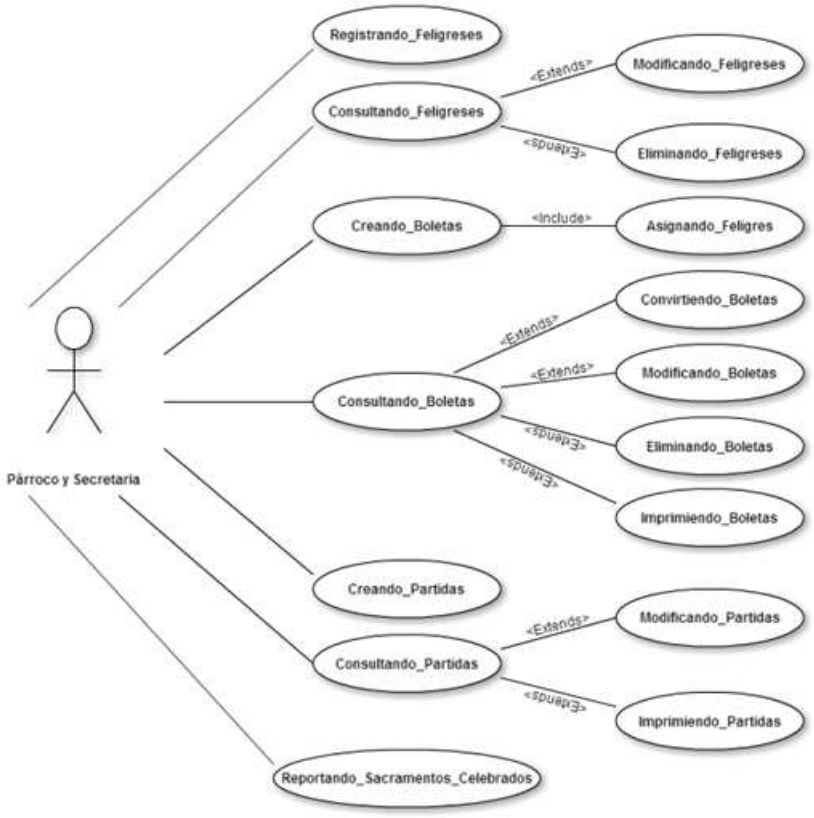

Figura 3.4: Diagrama de casos de uso, módulo Actas Sacramentales

3.2.2 Especificación de casos de uso

Para los casos de uso descritos en el diagrama anterior y demás diagramas de casos de uso, se realiza una descripción detallada utilizando una plantilla de especificación de casos de uso, donde se incluyen: actores, resumen, precondición, post-condición, flujo de eventos y cursos alternos. En la tabla 3.4 se presenta la especificación de caso de uso para el caso de uso convirtiendo boletas del módulo actas sacramentales.

	Especificación de Caso de Uso	Revisión: 001	Página: 1/1 Fecha: 08/04/2015
Convirtiendo Boleta			
Actores: Párroco y Secretaria			
Resumen: Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe convertir las boletas registradas a partidas.			
Precondición: El usuario debe estar registrado y validado en el sistema.			
Post-condición: Con la conversión de una boleta el usuario podrá realizar reportes acorde a los sacramentos celebrados en la Parroquia.			
Paso	**Acción de los Actores**	**Respuesta del Sistema**	
1	Ingresa al módulo de Actas Sacramentales.		
2		Despliega los sub módulos Feligrés, Boletas y Partidas.	
3	Ingresa al su módulo de Boletas.		
4		Despliega las opciones crear y consultar.	
5	Selecciona la opción Consultar.		
6		Mostrará un listado de catalogación por tipo de las boletas registrados en el sistema con algunas opciones (descargar, convertir editar y eliminar)	
7	Selecciona el tipo de boleta que desea consultar (*bautismo, confirmación, matrimonio y defunción*)		
8		Despliega un listado de feligreses que cumplan con el criterio de catalogación dado por el usuario.	
9	Selecciona la opción Convertir.		
10		Despliega en ventana modal un formulario con los datos del Anexo C (*Boleta – Partida*), y la opción de convertir boleta.	
11	Ingresa los datos solicitados por el sistema y selecciona la opción Convertir Boleta.		
12		Mostrará un mensaje como el siguiente *"La boleta se ha convertido a partida"*, confirmando que los datos se han guardado.	
13		Regresa al paso 8, actualizando el listado.	

Tabla 3.4: Especificación de caso de uso, Convirtiendo Boleta

Las especificaciones de casos de uso referente a los demás diagramas y módulos se encuentran en el apartado de anexos E.

3.2.3 Modelo ER

Para el sistema de información Ecclesia se trabajó con una base de datos relacional MySQL, que se presenta en el anexo F. En la figura 3.5 se muestra el segmento del modelo ER perteneciente al módulo de actas sacramentales, en esta se puede evidenciar la relación de las diferentes entidades que intervienen en este proceso y los atributos de cada entidad, resaltando el atributo único o llave primaria llamada ïd.

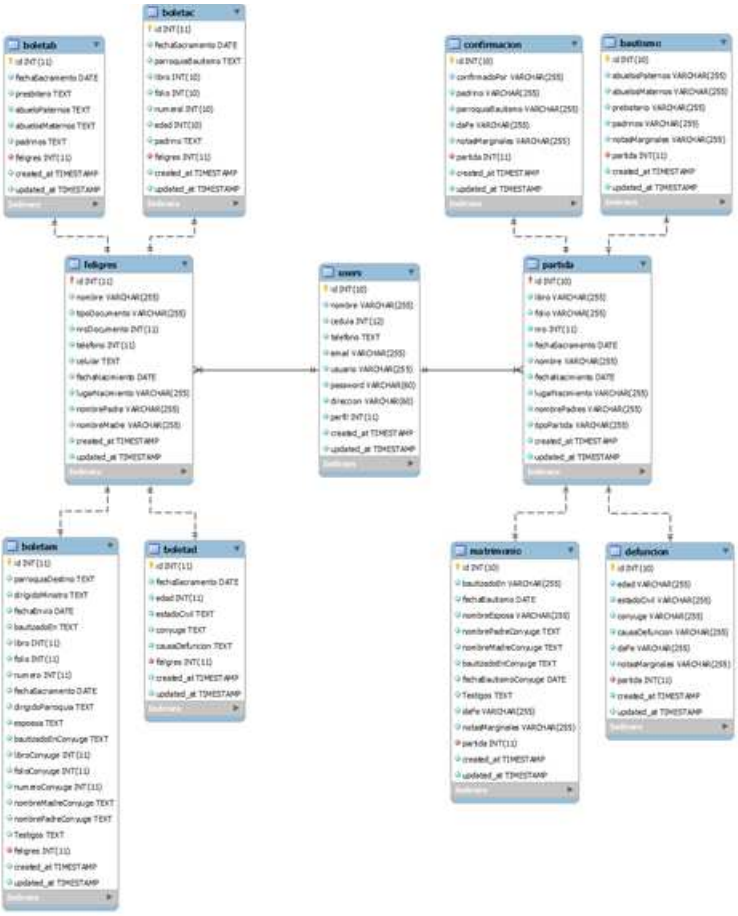

Figura 3.5: Modelo ER, módulo de Actas Sacramentales

3.2.4 Diagrama de clases

En el desarrollo de software la creación y aprobación de este diagrama es indispensable para la construcción del código fuente de cualquier proyecto, ya que permite visualizar las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos. El cumpliendo de este artefacto se presenta en la figura 3.6.

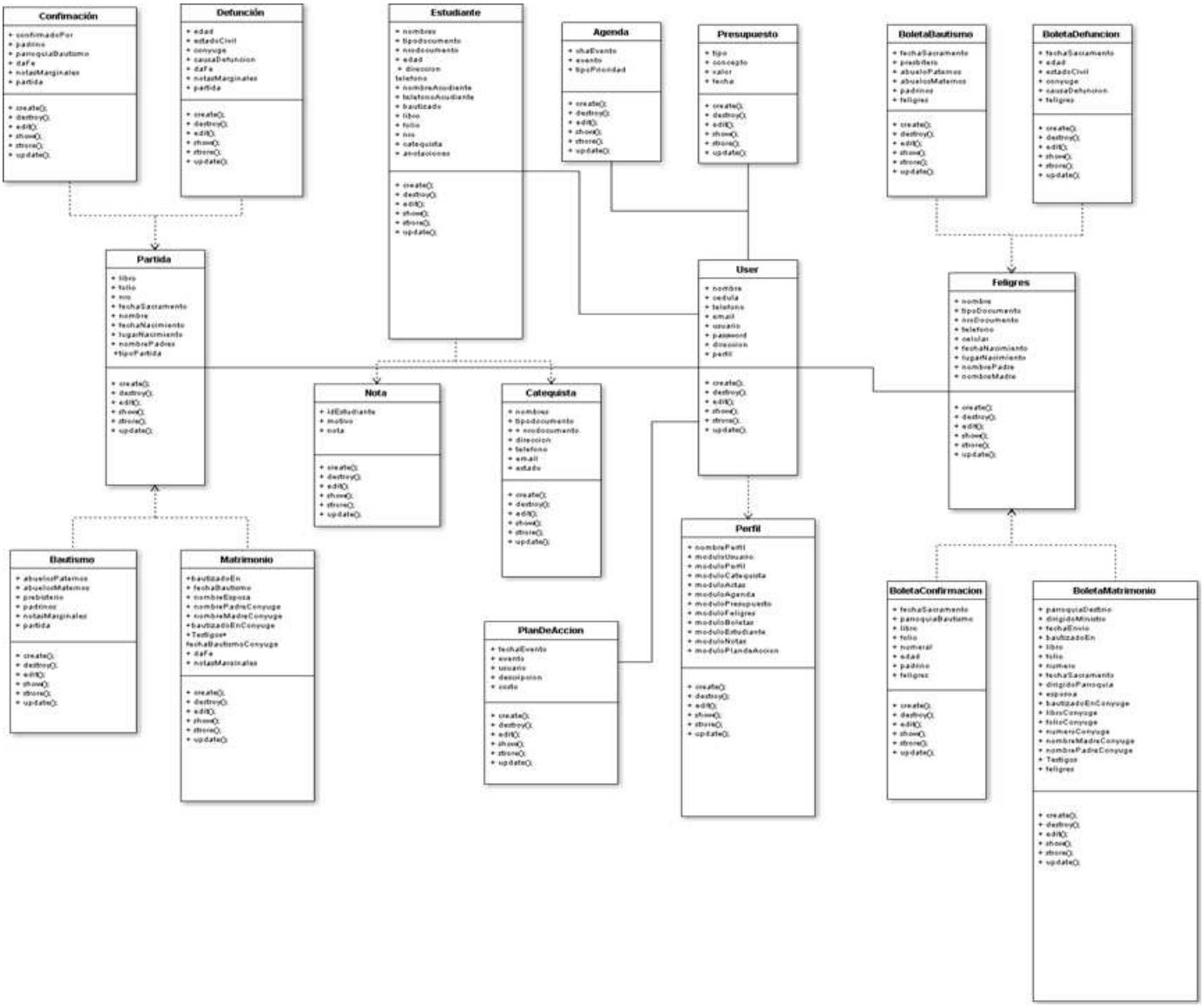

Figura 3.6: Diagrama de clases, Ecclesia

3.2.5 Diagrama de secuencia

Muestra gráficamente los eventos que originan los actores dentro de un sistema y cómo se comunican (interactúan) entre sí a lo largo del tiempo. En la figura 3.7 se presenta el modo en que los principales componentes del sistema (servidor, vista, controlador, modelo, request y base de datos) interactúan para lograr convertir una boleta a partida.

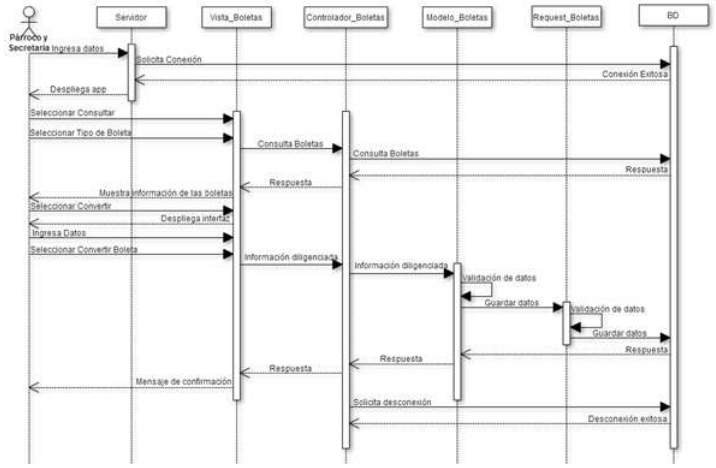

Figura 3.7: Diagrama de secuencia, Convirtiendo Boleta

Los diagramas de secuencias referentes a los otros casos de uso desarrollados en la elaboración del proyecto se encuentran en el apartado de anexos G.

3.2.6 Prototipos de interfaz de usuario

La interfaz de usuario (IU) es uno de los componentes más importantes de cualquier sistema computacional, pues funciona como el vínculo entre el humano y la máquina. Frecuentemente el éxito de un programa depende de la IU, pues esta debe responder en gran medida a qué tan rápido el usuario puede aprender a emplear el software y a su vez que alcance sus objetivos con el programa de la manera más sencilla. Para lograr el objetivo de la IU, en el presente proyecto se han realizado dos prototipos, uno inicial o de baja fidelidad donde se esboza una interfaz preliminar teniendo en cuenta los requisitos previos del usuario, y otra más detallada o de alta fidelidad donde se expone cómo será la interfaz final de Ecclesia.

En la iteración de prototipos de interfaz de usuario, se buscó la consistencia entre todos los módulos que dispone la aplicación, es decir, la agrupación de menús, iconos, tablas de información, entre otros, permitiendo que el usuario tenga mayor grado de asimilación, aportando reconocimiento antes que recuerdo [9].

La figura 3.8 muestra el prototipo de interfaz principal para todos los usuarios en Ecclesia, esta se compone de tres secciones: (I) área de control, (II) acceso a las diferentes funcionalidades del sistema y (III) área contenedora de formularios y/o información. Los prototipos de IU referentes a los módulos de la aplicación se encuentran en el apartado de anexos H, los cuales fueron levemente modificados por el uso de la plantilla SB Admin.

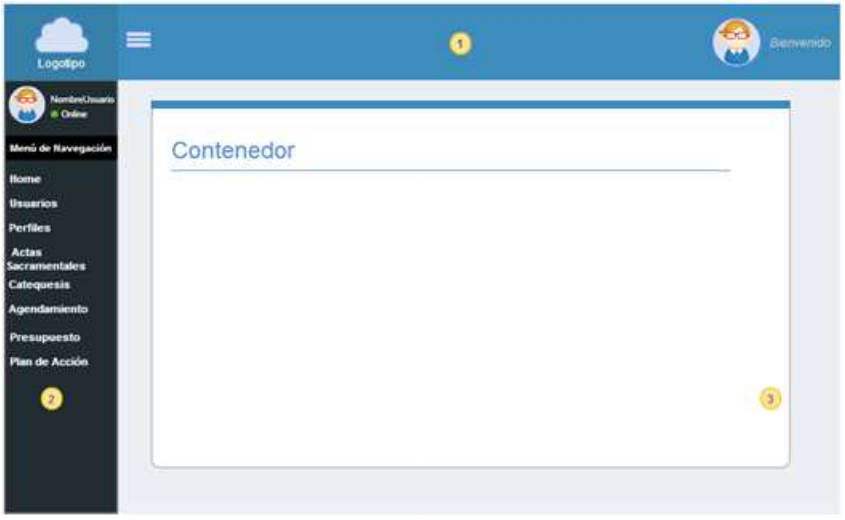

Figura 3.8: Prototipo de interfaz de usuario, Interfaz Principal

3.2.7 Arquitectura Diagrama de componentes

La aplicación Web Ecclesia se basa en una arquitectura modelo-vista-controlador que se conoce como Arquitectura MVC. La Figura 3.9 muestra la arquitectura sobre la cual se soporta Ecclesia mediante un diagrama de componentes.

La capa de presentación (vista) fue elaborada con la plantilla de administración SB Admin [3], en esta se usa la librería Bootstrap y diferentes plugins JQuery que permiten generar interfaces de calidad, responsivas o con la posibilidad de ajustarse automáticamente a las dimensiones de la pantalla donde estén mostrándose, siendo esto un punto clave para las tendencias actuales del uso de dispositivos móviles o tablets para acceder a internet. La comunicación con el servicio Web se hace a través de HTTP con la técnica AJAX. La capa de lógica (controlador) la constituye los diferentes componentes de Actas Sacramentales, Catequesis, Agendamiento, Presupuesto, Plan de Acción, Usuarios y Perfiles, que constituyen las funcionalidades principales de la aplicación y conforman los diferentes modelos y controladores, que a su vez se ven apoyados por el componente controlador de peticiones que tiene por función el gestionar las peticiones provenientes de la capa de presentación y direccionarlas al siguiente componente de servicios o controladores, en el cual se procesan las peticiones y se determinan los servicios web que se deben suplir, para acceder a los Modelos o funcionalidades requeridas. En relación a los modelos se encuentra el componente Model, en esté se gestionan las consultas que se requieran al motor de información y la carga dinámica de componentes e información a las interfaces de usuario de acuerdo a la petición que haya llegado a la capa lógica. La capa de datos por su parte, tiene como responsabilidad interactuar con la capa lógica y mantener la información de la aplicación, como es la gestión de usuarios, actas sacramentales, estudiantes, entre otros.

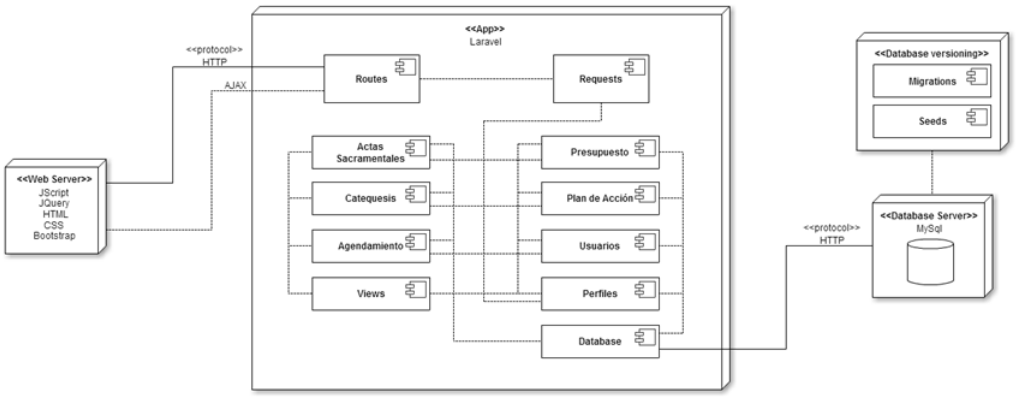

Figura 3.9: Requerimientos, módulo Actas Sacramentales

3.3 Construcción

La codificación se llevó a cabo por medio de un proceso de prototipado constante que genero una versión, módulo, sub módulo o componente de software por cada iteración del proceso de desarrollo, bajo la metodología en RUP.

Para la elaboración del sistema de información Ecclesia se usó el lenguaje de programación PHP7, integrado con JavaScript y JQuery, facilitando de esta manera el desarrollo web de contenido dinámico y la integración con servidores para probar el correcto funcionamiento de los prototipos. Se dispuso de HostGator como contenedor para probar y desplegar la aplicación. Como fue mencionado en la sección anterior, para generar la interfaz de comunicación entre el usuario y Ecclesia se utilizó la plantilla SB Admin, la cual unifica herramientas como HTML y CSS por medio de Bootstrap, además ofrece variados componentes y herramientas graficas prediseñadas, que aceleran el diseño de la interfaz.

La administración de los datos almacenados en el sistema, se realiza a través del gestor de base de datos web phpMyAdmin en su versión 4.3, aprovechando así la capacidad que tiene para ejecutar las características de MySQL (CRUD de datos, tablas, campos e índices, gestión de usuarios y privilegios, entre otros), su compatibilidad con múltiples servidores y la posibilidad de exportación de datos a diferentes formatos.

Es importante denotar que al inicio de esta fase fue necesario investigar sobre los beneficios y dificultades de usar los diferentes lenguajes de programación en la web, con el objetivo de determinar cuál sería el más beneficioso para el desarrollo de este proyecto, consecuentemente se debió hacer el mismo proceso en la elección del framework, para finalmente realizar la curva de aprendizaje de los entornos seleccionados, en

este caso PHP7 y Laravel.

La aplicación se encuentra disponible en el subdominio: http://p.paimportar.com/Login, mientras es posible su instalación en un servidor dedicado sólo para la misma, esto se hizo con el fin de probar su correcto funcionamiento en ambiente.

3.3.1 Descripción general del aplicativo web

En esta sección se presenta de manera general el sistema de información Ecclesia, como herramienta desarrollada para facilitar los procesos llevados a cabo por la Parroquia San Martín de Porres de Tuluá â Valle. Esta aplicación contempla la facilidad de accederse desde diferentes dispositivos, al contar con interfaces de usuarios responsivas. Dentro del sistema intervienen cuatro perfiles párroco, secretaria, catequista y líder, que comparten la misma interfaz principal compuesta de tres secciones: (I) área de control, (II) acceso a las diferentes funcionalidades del sistema y (III) área contenedora de formularios y/o información, pero con diferentes funcionalidades de acuerdo a los privilegios de los perfiles anteriormente mencionados.

En los apartados siguientes de esta sección se abordan los resultados de la aplicación en función del cumplimiento de los objetivos del proyecto.

A. Mecanismo para gestión de actas sacramentales

En esta sección se presenta uno de los procesos más importantes que permite llevar a cabo Ecclesia, donde se hace referencia al proceso de actas sacramentales, entendiéndose como la gestión de boletas y partidas de los feligreses de la Parroquia, en este se ven involucrados los perfiles párroco y secretaria.

La gestión de actas sacramentales es un proceso dividido en tres sub módulos: (I) feligreses, (II) boletas y (III) partidas, que se relacionan entre sí. La gestión de feligreses va desde la creación, al proceso de selección en la creación de una boleta, permitiendo visualizar la relación feligreses â boletas, hasta que se realiza la conversión de la boleta a una partida sacramental, donde la relación se transforma feligreses â partidas. Dentro de los sub módulos boletas y partidas, el usuario puede visualizar y seleccionar el tipo de archivo (bautismo, confirmación, matrimonio y defunción) que desea registrar. Ecclesia permite la creación de partidas sin la necesidad de registrar con anterioridad el feligrés, acogiendo los documentos que reposan en los libros parroquiales, impresión de la información relacionada en archivos pdf, y visualizar estadísticas de las boletas y partidas registradas.

Para la creación de las diferentes boletas y partidas, Ecclesia despliega inicialmente una interfaz modal que permite al usuario seleccionar el tipo (bautismo, confirmación,

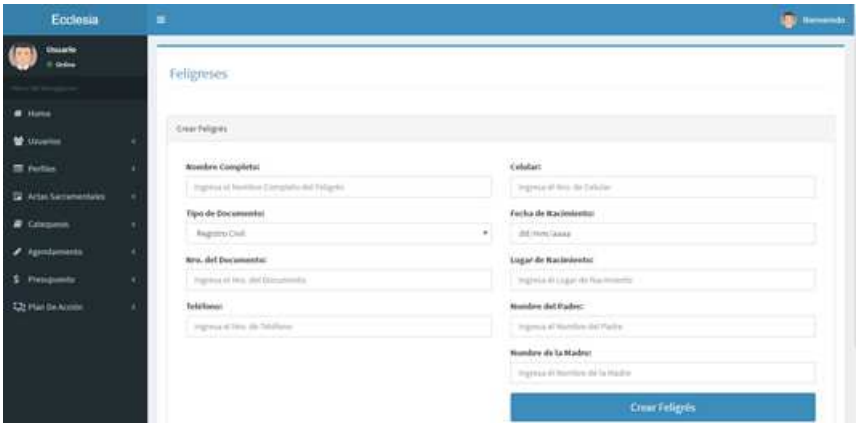

Figura 3.10: Actas sacramentales: Feligreses

matrimonio y defunción) para posteriormente desplegar la interfaz principal el formulario correspondiente.

Figura 3.11: Tipo de Boletas y Partidas

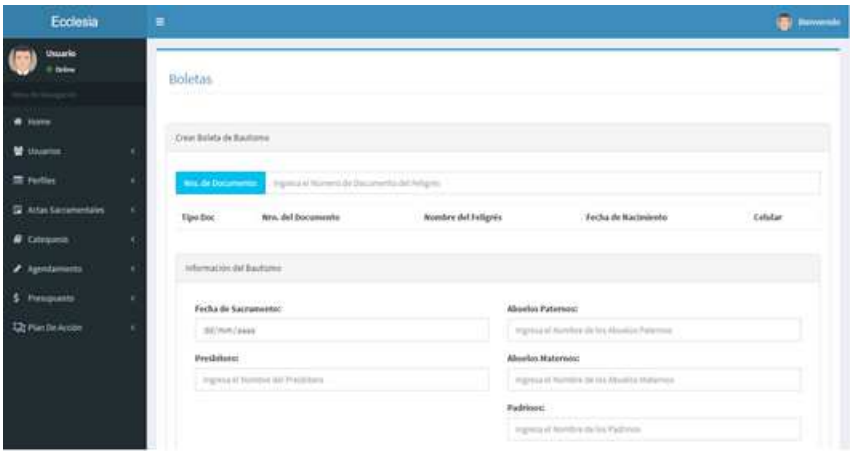

Figura 3.12: Actas Sacramentales: Boletas $\gg Crear$

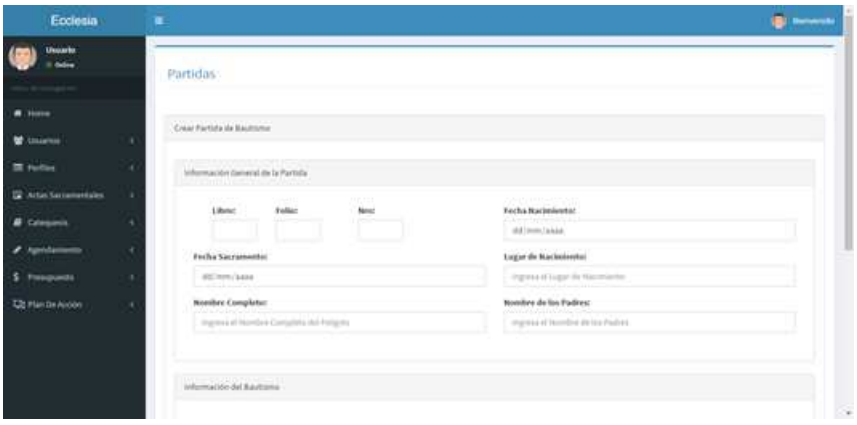

Figura 3.13: Actas Sacramentales: Partidas $\gg Crear$

B. Agendamiento, plan de acción y catequesis

La gestión de agendamiento y plan de acción proporciona un calendario con posibles visualizaciones de mes, semana o día, a los diferentes perfiles del sistema, en el que pueden registrar un evento con diferentes detalles como hora, prioridad, descripción, responsable y costo según sea el caso, buscando así usar adecuadamente el recurso tiempo. La información recolectada del dato costo es base para los egresos generador en la parroquia y la creación de reportes en el módulo presupuesto. En la figura 3.14 y 3.15 se presentan los módulos de agendamiento y plan de acción respectivamente.

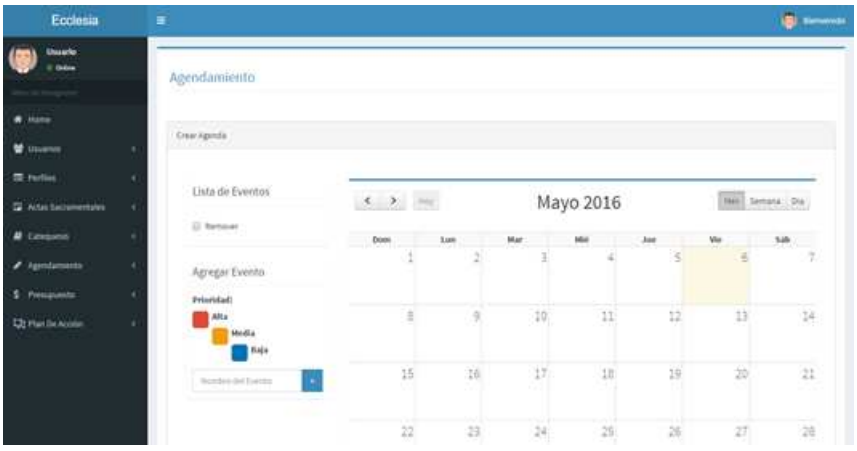

Figura 3.14: Agendamiento $\gg Crear$

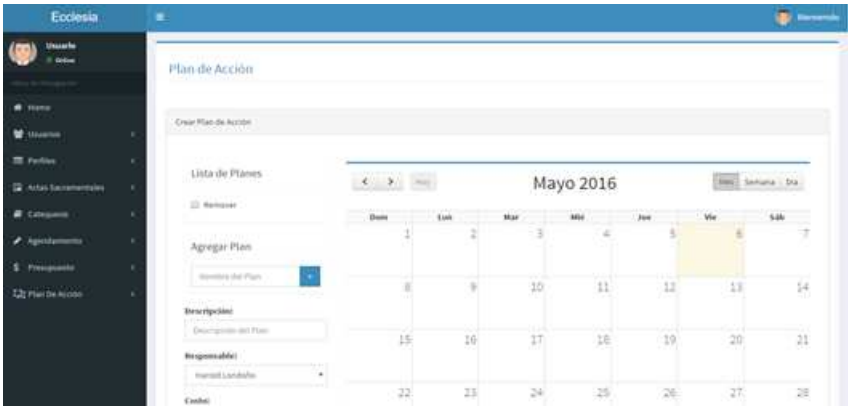

Figura 3.15: Plan de Acción $\gg Crear$

Respecto a la gestión del proceso de catequesis, Ecclesia divide el proceso en tres sub módulos: (I) estudiantes, (II) catequistas y (III) notas, en los que intervienen según sean sus privilegios los usuarios con perfil párroco, secretaria y catequista. La relación entre estos sub módulos va desde la creación de un estudiante, al proceso de selección de catequista, hasta que se realiza la asignación de notas.

El sub módulo de estudiante permite la creación, consulta, edición y eliminación de los mismos. El contenido del formulario para crear un estudiante además de solicitar la información básica, busca complementarla a través del área de descripción en la que es posible detallar si esté tiene alguna dificultad de aprendizaje u otro problema que pueda atrasar el proceso de enseñanza, esto con el fin de asegurar al alumno un catequista idóneo. Además de las tareas principales para la gestión de información, el sub módulo de catequista facilita la operación de asignación de profesores a los estudiantes por medio de la habilitación o inhabilitación de los mismos.

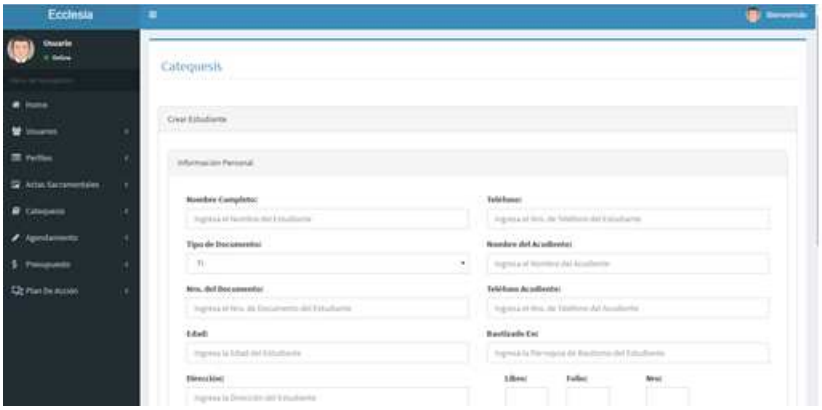

Figura 3.16: Catequesis: Estudiantes $\gg Crear$

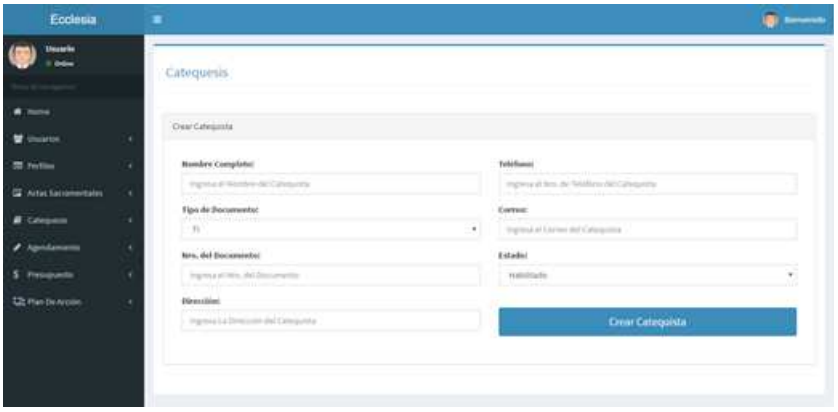

Figura 3.17: Catequesis: Catequistas $\gg Crear$

En las figuras 3.16 y 3.17 expuestas anteriormente se pueden apreciar la interfaz para la creación de estudiantes y catequistas respectivamente.

El sub módulo de notas permite al párroco y catequista crear, consultar y editar notas de un estudiante, al momento de esté ser agregado el sistema le asigna una nota de â0â, que puede ser modificada en cualquier momento, tras el registro de varias notas se va obteniendo el promedio del rendimiento académico por estudiante, permitiendo a los usuarios que intervienen en este subproceso tomar acciones frente al método de enseñanza que se está impartiendo. En la figura 3.18 se presenta la interfaz de notas de los estudiantes.

Figura 3.18: Catequesis: Notas

C. Presupuesto

Ecclesia en sus funcionalidades de presupuesto permite el registro de ingresos y egresos de la Parroquia con su respectivo motivo y fecha, esto con el fin que la Iglesia no cambie el método de trabajo en esta área, además genera recibos en formato pdf para la constancia de pago de los laicos, gráfico estadístico y consultas donde se pueden evidenciar el flujo de caja obtenido. En la figura 3.19 se muestran los ingresos y egresos obtenidos en el año 2016 por la Parroquia y el dinero que posee en el momento.

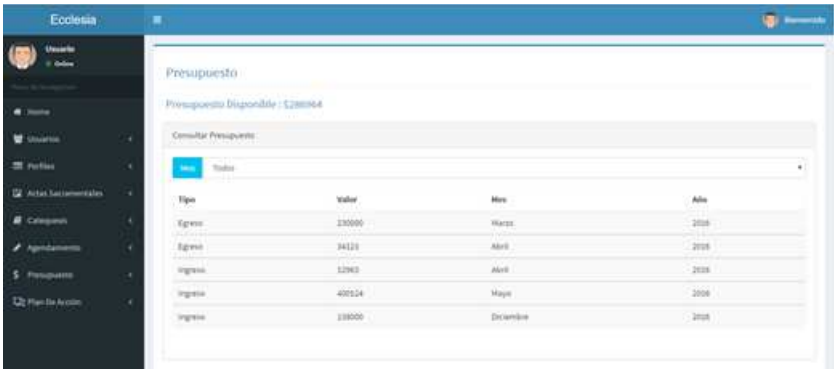

Figura 3.19: Presupuesto $\gg$ *Consultar*

Capítulo 4

Pruebas

Este capítulo muestra el proceso de pruebas realizado para la aplicación Eclessia, las cuales se centran en la técnica de comprobación a nivel de módulos individuales para cerciorar el correcto funcionamiento por separado de los mismos donde se empleó el framework PHPUnit y la metodología de pruebas de usabilidad, estos mecanismos estuvieron dirigidos a verificar la interacción de componentes, implementación correcta de requerimientos y corrección oportuna de defectos encontrados.

4.1 Desarrollo por pruebas unitarias

El framework Laravel ofrece el directorio tests para albergar las pruebas de la aplicación Ecclesia, soportándose en el software para testing PHPUnit, siendo esté el más extendido en el desarrollo de aplicaciones PHP. Las pruebas unitarias se realizaron bajo las características que deben cumplir estos tests para que se puedan considerar unit tests: automatizables, completas, repetibles, independientes y profesionales [14]. Estas se realizaron para la comprobación de las funcionalidades principales como enrutamiento, envió de datos y manejo de formularios. La ejecución de las pruebas con PHPUnit inicia con la verificación de datos retornados, para ello se usa el método Visit, como se expone en la figura 4.1, la función de este método es probar que un controlador al momento de direccionar hacía alguna vista, lo haga de manera correcta.

```
// Test Login
public function testRutaLogin()
{
    $this->visit('/Login')->see('login');
}
```

Figura 4.1: Prueba PHPUnit, método Visit

Finalizadas las pruebas para el retorno de vistas de los diferentes módulos, se continuó con la comprobación de los métodos resource, Get y Post pertenecientes a los controladores, el objetivo de esta prueba es demostrar que estos métodos se encuentren retornando la información pedida por la ruta, de igual manera se busca evidenciar que no exista perdida de datos en el transcurso de la información. En la figura 4.2 se aprecia la implementación de esta prueba.

```
// Test Crear Usuario
public function testRutaUsuarioCrear()
{
    $this->call('GET', '/usuario/create');
    $this->assertResponseOk();
}
```

Figura 4.2: Prueba PHPUnit, método GET

Después de realizadas las pruebas para el retorno correcto de información de los diferentes métodos CRUD, se realiza el último tipo de prueba que ofrece PHPUnit, utilizado para verificar que los datos enviados desde los diferentes formularios de la aplicación se encuentren perfectamente enrutados con sus respectivos controladores teniendo en cuenta su intermediario llamado Request, el correcto funcionamiento de esta unidad debe realizar primero un llamado al respectivo controlador y esté, a su vez llama su respectivo Request. En la figura 4.3 se evidencia la aplicación de esta prueba al proyecto.

```
// Enviar Datos  de Prueba
public function testRutaUsuarioCrearDatosFull()
{
    $datosUsuario = array(
        'nombre' => 'Julia ',
        'apellido' => 'Martinez',
        'cedula' => 1116269909,
        'telefono' => 2303676,
        'email' => 'jugamaa@hotmail.com',
        'usuario' => 'damin',
        'password' => 'admin',
        'direccion' => 'Calle 26E #11A-13',
        'perfil' => 1
    );

    $this->call('POST', '/usuario/store', $datosUsuario);
}
```

Figura 4.3: Prueba PHPUnit, llamados

Realizadas las pruebas unitarias, se ejecuta por medio de consola el entorno de PHPUnit para verificar que los tres tipos de pruebas, descritas anteriormente se ejecuten correctamente. En la figura 4.4 se evidencian los resultados de la corrida de PHPUnit.

```
#59 C:\xampp\htdocs\Parroquia\vendor\laravel\framework\src\Illuminate\Foundation\Testing\CrawlerTrait.php(102): Illumina
te\Foundation\Testing\TestCase->call('GET', 'http://localhos...', Array, Array, Array)
#60 C:\xampp\htdocs\Parroquia\vendor\laravel\framework\src\Illuminate\Foundation\Testing\CrawlerTrait.php(66): Illuminat
e\Foundation\Testing\TestCase->makeRequest('GET', '/perfiles/creat...')
#61 C:\xampp\htdocs\Parroquia\tests\PruebasTest.php(115): Illuminate\Foundation\Testing\TestCase->visit('/perfiles/creat
...')
#62 [internal function]: PruebasTest->testRutaPerfilesCrearDato()
#63 C:\xampp\htdocs\Parroquia\vendor\phpunit\phpunit\src\Framework\TestCase.php(881): ReflectionMethod->invokeArgs(Objec
t(PruebasTest), Array)
#64 C:\xampp\htdocs\Parroquia\vendor\phpunit\phpunit\src\Framework\TestCase.php(746): PHPUnit_Framework_TestCase->runTes
t()
#65 C:\xampp\htdocs\Parroquia\vendor\phpunit\phpunit\src\Framework\TestResult.php(601): PHPUnit_Framework_TestCase->runB
are()
#66 C:\xampp\htdocs\Parroquia\vendor\phpunit\phpunit\src\Framework\TestCase.php(702): PHPUnit_Framework_TestResult->run(
Object(PruebasTest))
#67 C:\xampp\htdocs\Parroquia\vendor\phpunit\phpunit\src\Framework\TestSuite.php(735): PHPUnit_Framework_TestCase->run(O
bject(PHPUnit_Framework_TestResult))
#68 C:\xampp\htdocs\Parroquia\vendor\phpunit\phpunit\src\Framework\TestSuite.php(735): PHPUnit_Framework_TestSuite->run(
Object(PHPUnit_Framework_TestResult))
#69 C:\xampp\htdocs\Parroquia\vendor\phpunit\phpunit\src\TextUI\TestRunner.php(432): PHPUnit_Framework_TestSuite->run(Ob
ject(PHPUnit_Framework_TestResult))
#70 C:\xampp\php\pear\PHPUnit\TextUI\Command.php(176): PHPUnit_TextUI_TestRunner->doRun(Object(PHPUnit_Framework_TestSui
te), Array)
#71 C:\xampp\php\pear\PHPUnit\TextUI\Command.php(129): PHPUnit_TextUI_Command->run(Array, true)
#72 C:\xampp\php\phpunit(46): PHPUnit_TextUI_Command::main()
#73 {main}
FAILURES!
Tests: 60, Assertions: 43, Failures: 5.

C:\xampp\htdocs\Parroquia>
```

Figura 4.4: Prueba entorno PHPUnit

En la figura anterior se puede apreciar la obtención de errores, por lo que fue necesario revisar los diferentes tests que lanzaron los errores para realizar las correcciones pertinentes al código fuente, finalizado el proceso de retroalimentación se ejecuta nuevamente el entorno de PHPUnit obteniendo cero errores, como lo demuestra la figura 4.5.

```
Microsoft Windows [Versión 10.0.10586]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Andres>cd ..

C:\Users>cd..

C:\>cd xampp

C:\xampp>cd htdocs

C:\xampp\htdocs>cd parroquia

C:\xampp\htdocs\Parroquia>phpunit
PHPUnit 4.8.9 by Sebastian Bergmann and contributors.

............................................................

Time: 2.96 seconds, Memory: 43.25Mb

OK (60 tests, 46 assertions)

C:\xampp\htdocs\Parroquia>
```

Figura 4.5: Prueba entorno PHPUnit

4.2 Metodología de pruebas de usabilidad

La prueba de usabilidad se basa en un caso de estudio con un flujo de actividades para el uso de Ecclesia, como se muestra en la Tabla 4.1.

Paso	Descripción
1	Ingrese a la aplicación e inicie sesión.
2	Diríjase a la opción de Actas Sacramentales – Feligreses.
3	Visualice los feligreses registrados.
4	Diríjase a la opción de Actas Sacramentales – Boletas.
5	Cree una boleta del tipo que desee.
6	Visualice las boletas registradas.
7	Convierta la boleta que ha creado a partida.
8	Diríjase a la opción de Actas Sacramentales – Partidas.
9	Visualice la partida que ha creado anteriormente.
10	Diríjase a la opción de Catequesis – Estudiantes.
11	Registre un estudiante.
12	Diríjase a la opción de Catequesis – Notas.
13	Registre una nota al estudiante que ha creado.
14	Diríjase a Agendamiento.
15	Visualice los eventos registrados.
16	Seleccione el motivo de consulta "Hoy"
17	Elimine un evento registrado.
18	Diríjase a Presupuesto – Añadir Egreso.
19	Registre un egreso.
20	Genere el recibo por motivo del egreso.
21	Diríjase a Plan de Acción.
22	Registre un evento.
23	Verifique la creación del evento.
24	Vuelva a la página principal.
25	Cierre la sesión.

Tabla 4.1: Pasos, prueba de usabilidad

Una vez aplicado el caso de estudio se lleva a cabo la siguiente encuesta de usabilidad:

1. En una escala de 1 a 5, donde 1 es Inútil y 5 Muy útil. ¿Qué concepto le queda del sistema de información Ecclesia?

(1) (2) (3) (4) (5)

2. ¿Encontró fácilmente las funcionalidades requeridas?

(a) La mayoría de los casos (b) En algunos casos (c) En ninguno de ellos

3. ¿Necesito algún tipo de orientación para la realización de los pasos?

(a) La mayoría de los pasos (b) En algunos pasos (c) En ninguno de ellos

4. ¿Cómo calificaría el tiempo de respuesta de la aplicación?

(a) Bueno (b) Regular (c) Malo

5. En una escala de 1 a 5, donde 1 es Muy difícil y 5 Muy fácil. ¿Qué tan difícil, le

pareció buscar un feligrés?

(1) (2) (3) (4) (5) 6. En una escala de 1 a 5, donde 1 es Muy difícil y 5 Muy fácil.

¿Qué tan difícil, le pareció el proceso de creación de boletas?

(1) (2) (3) (4) (5)

7. En una escala de 1 a 5, donde 1 es Muy difícil y 5 Muy fácil. ¿Qué tan difícil, le pareció el proceso de convertir una boleta a partida?

(1) (2) (3) (4) (5)

8. En una escala de 1 a 5, donde 1 es Muy difícil y 5 Muy fácil. ¿Qué tan difícil, le pareció registrar un egreso y generar el recibo correspondiente?

(1) (2) (3) (4) (5)

9. En una escala de 1 a 5, donde 1 es Muy difícil y 5 Muy fácil. ¿Qué tan difícil, le pareció registrar un estudiante y su nota?

(1) (2) (3) (4) (5)

10. ¿Encontró algún error en la aplicación?

(a) Si (b) No

11. ¿Tiene alguna observación o sugerencia?

La elaboración de esta prueba se realizó en el mes de mayo del 2016, en la Parroquia San Martín de Porres con los que finalmente serán los usuarios del sistema.

Capítulo 5

Conclusiones y trabajos futuros

5.1 Conclusiones

Después de realizado el proceso de ingeniería del presente proyecto y obtenida la experiencia basada en el trabajo del mismo, se concluye:

1. Se pudo identificar que la implementación de una metodología de desarrollo acarrea el estudio de esta y de sus componentes (fases, ciclos, artefactos, entre otros). Al seguir con el proceso de ingeniería se inició una serie de adaptaciones de la metodología al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema entre otros) para finalmente dar comienzo al ciclo de vida del proyecto. De esta manera se garantiza que con la implementación de una metodología es posible ejecutarse optima y eficientemente la repetitividad del proceso de desarrollo, facilitando de esta manera la mantenibilidad y escalabilidad del sistema.

2. Es de resaltar que durante el proceso de desarrollo surgieron varias dificultades y cambios que influyeron en el sistema, de esta experiencia se concluye la importancia de considerar en futuros proyectos la gestión del riesgo y los beneficios que ésta puede aportar al producto final como concepto de la Ingeniería del Software.

3. Para la construcción de un sistema de información que sirva de apoyo a la gestión de procesos de una organización/compañía, en este caso particular la Iglesia, se vuelve fundamental la comprensión de conceptos propios en la administración de la organización y el entendimiento pleno del funcionamiento de cada proceso, ya que esto facilitará al equipo de desarrollo la visión y asimilación de los objetivos que la organización busca alcanzar con el desarrollo de una herramienta tecnológica.

5.2 Trabajos futuros

En el desarrollo de la aplicación se identificaron líneas de trabajo que extienden o complementan Ecclesia. Como trabajos futuros se consideran:

- Generar un mecanismo que permita el versionamiento de toda la información del sistema.1. Generar un mecanismo que permita realizar una encuesta después de desarrollar un evento del plan de acción, para complementar el proceso de toma de decisiones.
- Permitir que la información recolectada en los diferentes módulos como base para la creación de gráficos estadísticos sean exportados en diferentes formatos como png, jpeg o pdf, junto con sus respectivas gráficas.
- Implementar un módulo que permita la gestión de mensajes, desde su creación, visualización, eliminación y notificación, para facilitar y maximizar la comunicación entre los usuarios del sistema.

Referencias

[1] Cuatro enfoques metodológicos para el desarrollo de software rup msf xp scrum. http://biblioteca.uniminuto.edu/ojs/index.php/Inventum/article/view/9. [Accesado 4-abril-2015].

[2] Qué es javascript? http://www.maestrosdelweb.com/que-es-javascript/. [Accesado 26-marzo-2015].

[3] Adquisición de template. http://startbootstrap.com/template-overviews/sb-admin/, 2015. [Accesado 4-junio-2015].

[4] Diagrama de procesos de la Parroquia San Martin de Porres. Entrevista personal con harold londoño, sacerdote encargado, Febrero de 2014.

[5] Framework. Febrero del 2015 [En línea]. Disponible En:. Código del derecho canónico. http://www.vatican.va/archive/ESL0020/$_P1T.HTML.[Accesado06-junio-2016]$.

[6] Observatorio Pastoral del CELAM.

[7] Comparativo del interés de framework a lo largo del tiempo. https://www.google.es/trends/exploreq=[Accesado 09-febrero-2015].

[8] Framework. http://www.bdigital.unal.edu.co/10588/1/71095012013.pdf. [Accesado 06-febrero-2015].

[9] María Paula. GONZÁLEZ. Evaluación heurística. http://interaccion2011.m.aipo.es/libro/pdf/15-Evaluacion-Heuristica.pdf. [Accesado 13-junio-2015].

[10] Oliver. HEURTEL. Php y mysql. Barcelona, Ed. Eri, 2014, pp. 255-272.

[11] Luis. España MUÑIZ. Control presupuestario, 2009.

[12] Joan Antoni. PASTORI COLLADO. Usos de los sistemas de información en la organización. pp. 9-19.

[13] Colegio Inglaterra Real. Estrategia empresarial. http://colegioinglaterrareal.edu.co/mecir/wp-content/uploads/2013/03/ESTRATEGIAS-EMPRESARIALES.pdf, 2008. [Accesado 20-marzo-2013].

[14] vatican. La iglesa e internet. http://www.vatican.va/romansp.html. [Accesado 06-febrero-2015].

ANEXOS

ANEXO A: *Requerimientos*

	Documento: Requerimientos	**Fecha:** 08/03/2015 **Revisión:** 001 **Página:** 1

Módulo: Catequesis

Objetivo 2: Crear un módulo que permita al Sacerdote observar, modificar y evaluar a través de la web las diferentes actividades encaminadas al cumplimiento de los procesos de agendamiento, plan de acción parroquial y catequesis.

#	Requerimiento	Categoría
Módulo de Catequistas		
1	El sistema debe permitir al usuario registrar catequistas con información específica Anexo C (*catequista*).	F
2	El sistema debe permitir al usuario consultar los diferentes catequistas registrados.	F
3	El sistema debe permitir al usuario modificar la información de los catequistas.	F
4	El sistema debe permitir al usuario deshabilitar catequistas.	F
5	El sistema debe permitir al usuario eliminar catequistas.	F
Módulo de Estudiantes		
6	El sistema debe permitir al usuario registrar estudiantes con información específica Anexo C (*estudiante*).	F
7	El sistema debe permitir al usuario asignar un catequista registrado al estudiante.	F
8	El sistema debe permitir al usuario consultar los diferentes estudiantes registrados.	F
9	El sistema debe permitir al usuario modificar la información de los estudiantes.	F
10	El sistema debe permitir al usuario eliminar estudiantes.	F
Módulo de Notas		
11	El sistema automáticamente debe cargar los estudiantes registrados al módulo de notas, con el registro de una calificación inicial por defecto.	NF
12	El sistema debe garantizar que el usuario pueda visualizar el promedio de las calificaciones registradas de los diferentes estudiantes.	NF
13	El sistema debe permitir al usuario asignar nota(s) al estudiante con información específica Anexo C (*nota*).	F
14	El sistema debe permitir al usuario consultar las notas asignadas a los diferentes estudiantes.	F
15	El sistema debe permitir al usuario modificar la(s) nota(s) registrada(s) a un estudiante.	F

Tabla 5.1: Requerimientos, módulo Catequesis

	Documento: Requerimientos	**Fecha:** 08/03/2015 **Revisión:** 001 **Página:** 1

Módulo: Agendamiento

Objetivo 2: Crear un módulo que permita al Sacerdote observar, modificar y evaluar a través de la web las diferentes actividades encaminadas al cumplimiento de los procesos de agendamiento, plan de acción parroquial y catequesis.

#	Requerimiento	Categoría
1	El sistema debe permitir al usuario visualizar el calendario por día, semana y mes.	NF
2	El sistema debe permitir al usuario registrar los eventos asignados al Sacerdote con datos específicos Anexo C (*agenda*).	F
3	El sistema debe permitir al usuario consultar los diferentes eventos registrados.	F
4	El sistema debe permitir al usuario modificar los eventos registrados.	F
5	El sistema debe permitir al usuario eliminar los eventos registrados.	F

Tabla 5.2: Requerimientos, módulo Agendamiento

	Documento: Requerimientos	**Fecha:** 08/03/2015 **Revisión:** 001 **Página:** 1

Módulo: Plan de Acción

Objetivo 2: Crear un módulo que permita al Sacerdote observar, modificar y evaluar a través de la web las diferentes actividades encaminadas al cumplimiento de los procesos de agendamiento, plan de acción parroquial y catequesis.

#	Requerimiento	Categoría
1	El sistema debe permitir al usuario visualizar el calendario por día, semana y mes.	NF
2	El sistema debe permitir al usuario registrar los eventos dirigidos al plan de acción de la Parroquia con datos específicos Anexo C (*plan de acción*).	F
3	El sistema debe confirmar al usuario la disponibilidad de recursos para la realización de un evento dirigido al plan de acción.	NF
4	El sistema debe permitir al usuario asignar un responsable registrado al evento del plan de acción.	F
5	El sistema debe permitir al usuario consultar los diferentes eventos registrados.	F
6	El sistema debe permitir al usuario modificar los eventos registrados.	F
7	El sistema debe permitir al usuario eliminar los eventos registrados.	F

Tabla 5.3: Requerimientos, módulo Plan de acción

	Documento: Requerimientos	**Fecha:** 08/03/2015 **Revisión:** 001 **Página:** 1

Módulo: Presupuesto

Objetivo 3: Crear un módulo de presupuesto que permita el seguimiento del flujo de caja a través del tiempo.

#	Requerimiento	Categoría
1	El sistema debe permitir al usuario registrar los ingresos de la Parroquia con datos específicos Anexo C (*presupuesto*).	F
2	El sistema debe generar recibos por conceptos del flujo de caja de la Parroquia.	F
3	El sistema debe permitir al usuario registrar los egresos de la Parroquia con datos específicos Anexo C (*presupuesto*).	F
4	El sistema debe permitir al usuario consultar los diferentes movimientos del flujo de caja registrados.	F
5	El sistema automáticamente debe mostrar la relación resultante entre los motivos del flujo de caja.	NF
6	El sistema automáticamente debe mostrar el dinero disponible que posee la Parroquia.	NF

Tabla 5.4: Requerimientos, módulo Presupuesto

	Documento: Requerimientos	**Fecha:** 08/03/2015 **Revisión:** 001 **Página:** 1

Módulo: Gestión de Usuarios

#	Requerimiento	Categoría
Módulo de Validación de Usuarios		
1	El sistema debe identificar al usuario por medio de los datos usuario y contraseña, para permitirle el ingreso al sistema.	F
Módulo de Usuarios		
2	El sistema debe permitir al usuario administrador registrar nuevos usuarios con información específica Anexo C (*usuario*).	F
3	El sistema debe permitir al usuario administrador asignar un perfil registrado a un usuario.	F
4	El sistema debe permitir al usuario administrador consultar los diferentes usuarios registrados.	F
5	El sistema debe permitir al usuario administrador modificar la información de los usuarios.	F
6	El sistema debe permitir al usuario administrador eliminar usuarios.	F
Módulo de Perfiles		
7	El sistema debe permitir al usuario administrador crear perfiles, los cuales tendrán privilegios que proporcionan el acceso a los diferentes módulos.	F
8	El sistema debe permitir al usuario administrador consultar los diferentes perfiles registrados.	F
9	El sistema debe permitir al usuario administrador modificar los perfiles asignados a los demás usuarios.	F
10	El sistema debe permitir al usuario administrador eliminar perfiles.	F
11	El sistema automáticamente debe establecer un perfil predeterminado a los usuarios que hagan uso de un perfil eliminado.	NF

Tabla 5.5: Requerimientos, módulo Gestión de Usuarios

ANEXO B: *Tablas de información*

ANEXO C (*Boleta*)			
Tipo de Boleta	**Dato**	**Tipo de Dato**	**Condición**
Bautismo	Fecha sacramento	Date	Not Null
	Presbítero	Texto	Not Null
	Abuelos paternos	Texto	-
	Abuelos maternos	Texto	-
	Padrinos	Texto	Not Null
Confirmación	Fecha sacramento	Date	-
	Parroquia bautismo	Texto	-
	Libro	Numérico	Not Null
	Folio	Numérico	Not Null
	Numeral	Numérico	Not Null
	Edad	Numérico	-
	Padrino	Texto	-
Defunción	Fecha sacramento	Date	Not Null
	Edad	Numérico	Not Null
	Estado civil	Seleccionable	Not Null
	Cónyuge	Texto	-
	Causa defunción	Texto	Not Null
Matrimonio	Parroquia destino	Texto	Not Null
	Dirigido Ministro	Texto	Not Null
	Fecha envió	Date	-
	Bautizado en	Texto	Not Null
	Libro	Numérico	-
	Folio	Numérico	-
	Fecha sacramento	Date	Not Null
	Dirigido Parroquia	Texto	Not Null
	Esposo (a)	Texto	Not Null
	Bautizado en	Texto	Not Null
	Libro	Numérico	-
	Folio	Numérico	-
	Testigos	Texto	Not Null

Tabla 5.6: Información almacenada de las boletas

ANEXO C (*Boleta – Partida*)		
Dato	**Tipo de Dato**	**Condición**
Libro	Numérico	Not Null
Folio	Numérico	-
Numeral	Numérico	Not Null
Notas marginales	Texto	-
Fecha expedición partida	Date	Not Null

Tabla 5.7: Información almacenada en la conversión de boleta a partida

ANEXO C (*Partida 1*)			
Tipo de Partida	**Dato**	**Tipo de Dato**	**Condición**
Bautismo	Libro	Numérico	Not Null
	Folio	Numérico	Not Null
	Numeral	Numérico	-
	Fecha sacramento	Date	Not Null
	Nombre	Texto	Not Null
	Fecha nacimiento	Date	Not Null
	Lugar de nacimiento	Texto	Not Null
	Nombre padres	Texto	-
	Abuelos paternos	Texto	-
	Abuelos maternos	Texto	-
	Presbítero	Texto	Not Null
	Padrinos	Texto	Not Null
	Da fe	Texto	-
	Notas marginales	Texto	-
Confirmación	Libro	Numérico	Not Null
	Folio	Numérico	Not Null
	Numeral	Numérico	-
	Fecha sacramento	Date	Not Null
	Nombre	Texto	Not Null
	Fecha nacimiento	Date	Not Null
	Lugar de nacimiento	Texto	Not Null
	Nombre padres	Texto	-
	Confirmado por	Texto	Not Null
	Padrino	Texto	Not Null
	Parroquia bautismo	Texto	-
	Da fe	Texto	-
	Notas marginales	Texto	-

Tabla 5.8: Información almacenada de las partidas (parte 1)

ANEXO C (*Partida 2*)			
Tipo de Partida	**Dato**	**Tipo de Dato**	**Condición**
Defunción	Libro	Numérico	Not Null
	Folio	Numérico	Not Null
	Numeral	Numérico	-
	Fecha sacramento	Date	Not Null
	Nombre	Texto	Not Null
	Fecha nacimiento	Date	Not Null
	Lugar de nacimiento	Texto	Not Null
	Nombre padres	Texto	-
	Edad	Numérico	-
	Estado civil	Seleccionable	-
	Cónyuge	Texto	-
	Causa defunción	Texto	Not Null
	Da fe	Texto	-
	Notas marginales	Texto	-
Matrimonio	Libro	Numérico	Not Null
	Folio	Numérico	Not Null
	Numeral	Numérico	-
	Fecha sacramento	Date	Not Null
	Presbítero celebrante	Texto	Not Null
	Nombre esposo	Texto	Not Null
	Fecha nacimiento	Date	Not Null
	Lugar de nacimiento	Texto	Not Null
	Nombre padres	Texto	-
	Bautizado en	Texto	Not Null
	Fecha bautismo	Date	-
	Nombre esposa	Texto	Not Null
	Fecha nacimiento	Date	Not Null
	Lugar de nacimiento	Texto	Not Null
	Nombre padres	Texto	-
	Bautizado en	Texto	Not Null
	Fecha bautismo	Date	-
	Testigos	Texto	Not Null
	Da fe	Texto	-
	Notas marginales	Texto	-

Tabla 5.9: Información almacenada de las partidas (parte 2)

ANEXO C: *Diagramas de Actividades*

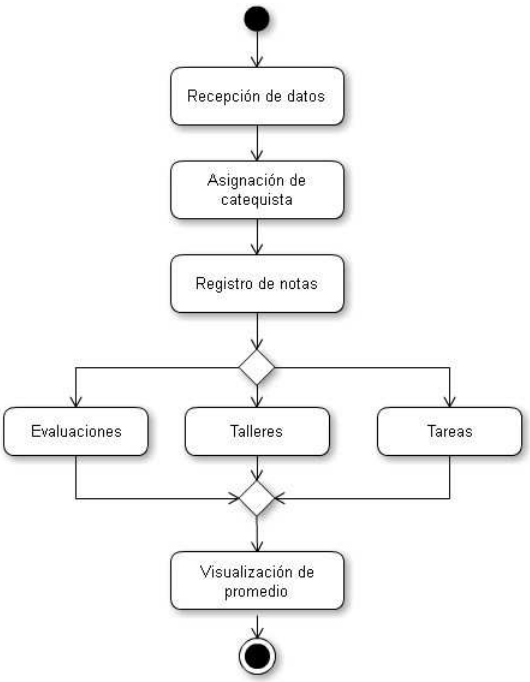

Tabla 5.10: Diagrama de actividades, proceso de Catequesis

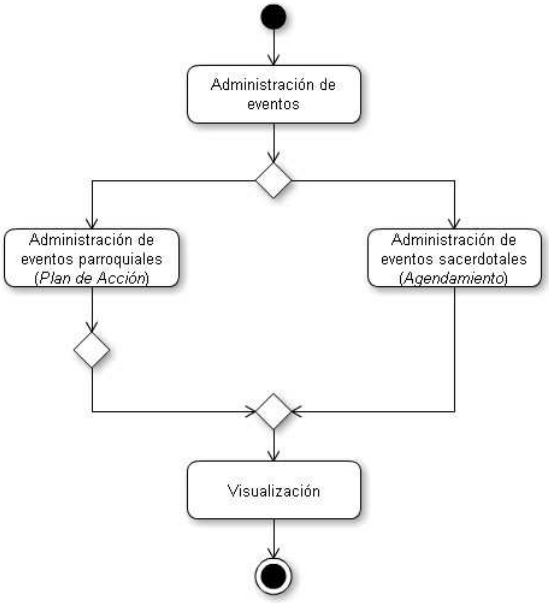

Tabla 5.11: Diagrama de actividades, Administración de eventos

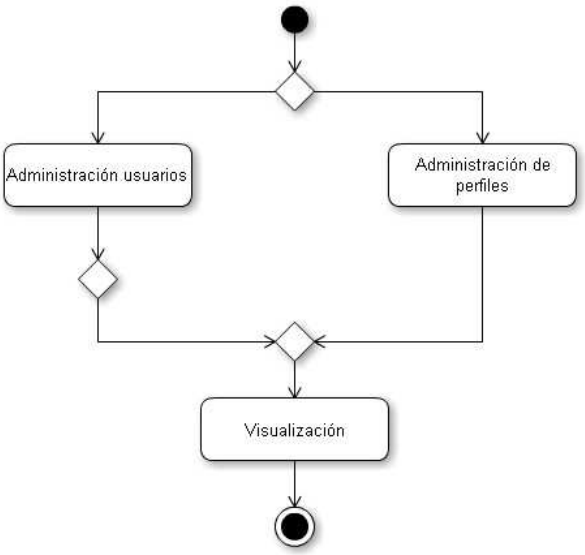

Tabla 5.12: Diagrama de actividades, proceso de Gestión de usuarios

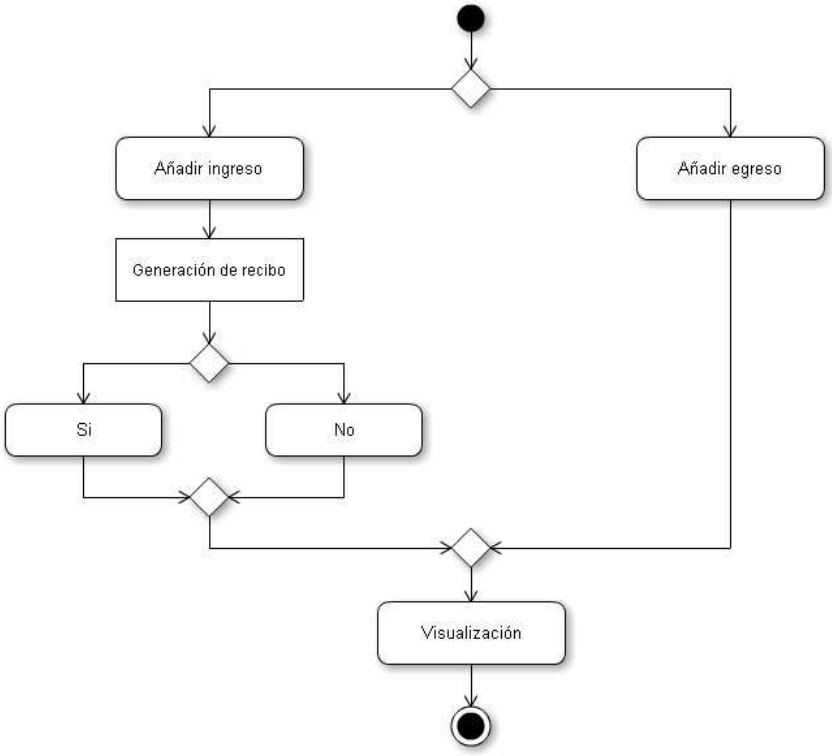

Tabla 5.13: Diagrama de actividades, proceso de Presupuesto

ANEXO D: *Diagramas Casos de uso*

Tabla 5.14: Diagrama casos de uso, módulo de Catequesis

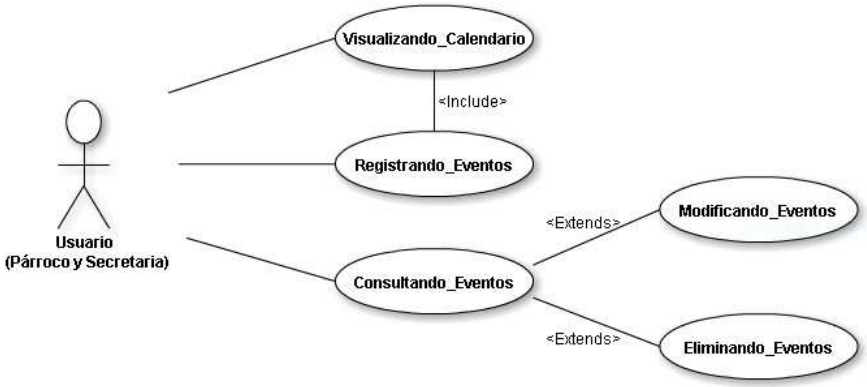

Tabla 5.15: Diagrama casos de uso, módulo de Agendamiento

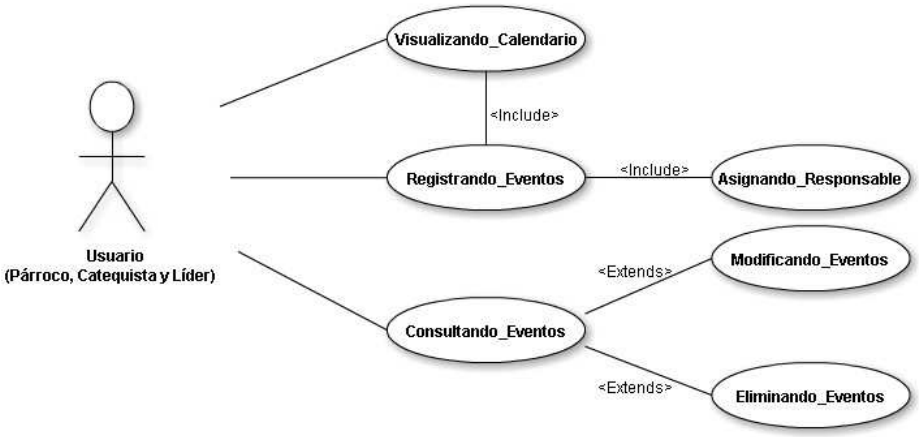

Tabla 5.16: Diagrama casos de uso, módulo de Plan de acción

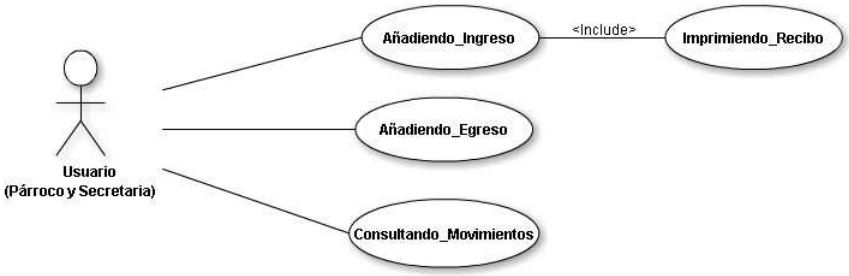

Tabla 5.17: Diagrama casos de uso, módulo de Presupuesto

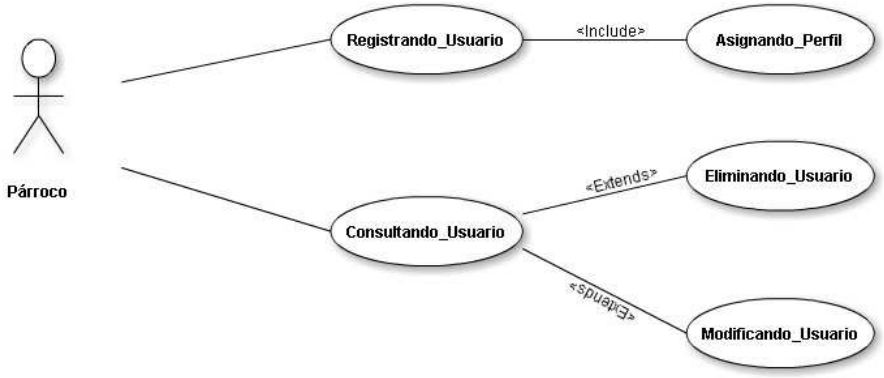

Tabla 5.18: Diagrama casos de uso, módulo de Usuarios

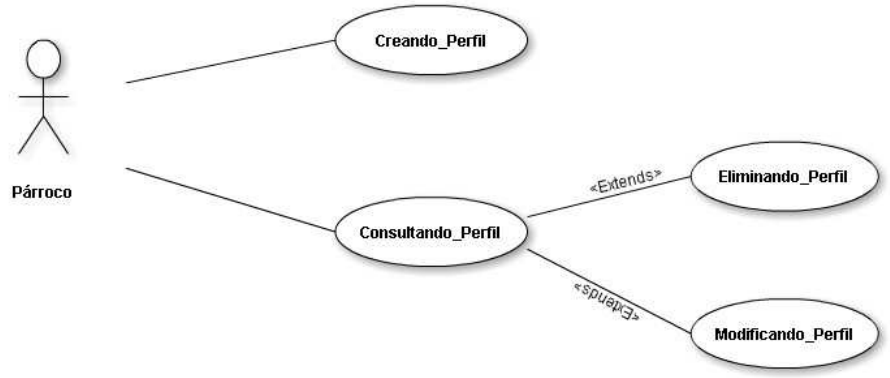

Tabla 5.19: Diagrama casos de uso, módulo de Perfil

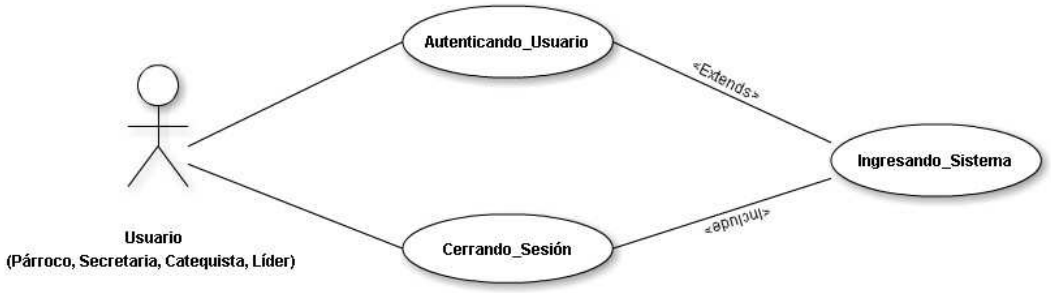

Tabla 5.20: Diagrama casos de uso, módulo de Validación de usuarios

ANEXO E: *Especificaciones casos de uso*

	Especificación de Caso de Uso	Revisión: 001	Página: 1/1 Fecha: 08/04/2015
Creando Boleta			
Actores: Párroco y Secretaria			
Resumen: Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe crear una boleta.			
Precondición: El usuario debe estar registrado y validado en el sistema.			
Post-condición: Con la creación de boletas se puede realizar posteriormente la conversión de estás a partidas.			
Paso	**Acción de los Actores**	**Respuesta del Sistema**	
1	Ingresa al módulo de Actas Sacramentales.		
2		Despliega los sub módulos Feligrés, Boletas y Partidas.	
3	Ingresa al su módulo de Boletas.		
4		Despliega las opciones crear y consultar.	
5	Selecciona la opción Crear.		
6		Despliega en ventana modal con las opciones bautismo, confirmación, matrimonio y defunción.	
7	Selecciona la opción que desee.		
8		Mostrará el formulario con los datos del Anexo C (*Boletas*) según su tipo, el campo de búsqueda por Nro. del documento y la opción de crear boleta de "tipo".	
9	Ingresa el Nro. del documento del feligrés que será relacionada la boleta y selecciona la opción Nro. del documento.		
10		Despliega la información del feligrés que cumpla con el criterio de búsqueda dado por el usuario.	
11	Ingresa los demás datos solicitados por el sistema y selecciona la opción Crear Boleta de "Tipo".		
12		Verifica que los datos ingresados sean válidos de acuerdo con el Anexo C (*Boletas*).	
13		Mostrará un mensaje como el siguiente *"Boleta de "tipo" creada"*, confirmando que los datos se han guardado.	
14		El formulario se limpia y queda listo para ser diligenciado nuevamente.	
Curso Alterno			
Feligrés no asignado			
Paso	**Descripción**		
12	Se muestra un mensaje como aparece a continuación "Por favor consulte o cree primero un feligrés". Llevándolo al paso 9.		
Campos vacíos			
Paso	**Descripción**		
12	Se muestra un mensaje como aparece a continuación "El campo "nombre del campo" es obligatorio". Llevándolo al paso 11.		

Tabla 5.21: Especificaciones casos de uso, creando boleta

	Especificación de Caso de Uso	**Revisión:** 001	**Página:** 1/1 **Fecha:** 08/04/2015

Consultando Boleta

Actores: Párroco y Secretaria

Resumen:
Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe consultar una boleta.

Precondición: El usuario debe estar registrado y validado en el sistema.

Post-condición:
Con las consultas realizadas se podrá visualizar la información relacionada con las boletas, para posteriormente descargarlas, convertirlas, modificarlas o eliminarlas.

Paso	Acción de los Actores	Respuesta del Sistema
1	Ingresa al módulo de Actas Sacramentales.	
2		Despliega los sub módulos Feligrés, Boletas y Partidas.
3	Ingresa al su módulo de Boletas.	
4		Despliega las opciones crear y consultar.
5	Selecciona la opción Consultar.	
6		Mostrará un listado de catalogación por tipo de las boletas registrados en el sistema con algunas opciones (descargar, convertir editar y eliminar)
7	Selecciona el tipo de boleta que desea consultar (*bautismo, confirmación, matrimonio y defunción*).	
8		Despliega un listado de feligreses que cumplan con el criterio de catalogación dado por el usuario.

Tabla 5.22: Especificaciones casos de uso, consultando boleta

	Especificación de Caso de Uso	**Revisión:** 001	**Página:** 1/1 **Fecha:** 08/04/2015

Imprimiendo Boleta

Actores: Párroco y Secretaria

Resumen:
Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe imprimir una boleta.

Precondición: El usuario debe estar registrado y validado en el sistema.

Post-condición:
Con la impresión de boletas los laicos pueden tener constancia del servicio que han adquirido.

Paso	Acción de los Actores	Respuesta del Sistema
1	Ingresa al módulo de Actas Sacramentales.	
2		Despliega los sub módulos Feligrés, Boletas y Partidas.
3	Ingresa al su módulo de Boletas.	
4		Despliega las opciones crear y consultar.
5	Selecciona la opción Consultar.	
6		Mostrará un listado de catalogación por tipo de las boletas registrados en el sistema con algunas opciones (descargar, convertir editar y eliminar)
7	Selecciona el tipo de boleta que desea consultar (*bautismo, confirmación, matrimonio y defunción*).	
8		Despliega un listado de feligreses que cumplan con el criterio de catalogación dado por el usuario.
9	Selecciona la opción Descargar.	
10		Descarga la boleta en formato *pdf*.

Tabla 5.23: Especificaciones casos de uso, imprimiendo boleta

	Especificación de Caso de Uso	Revisión: 001	Página: 1/1 Fecha: 08/04/2015

Modificando Boleta

Actores: Párroco y Secretaria

Resumen:
Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe modificar una boleta.

Precondición: El usuario debe estar registrado y validado en el sistema.

Post-condición:
El usuario podrá disponer de información actualizada de las boletas.

Paso	Acción de los Actores	Respuesta del Sistema
1	Ingresa al módulo de Actas Sacramentales.	
2		Despliega los sub módulos Feligrés, Boletas y Partidas.
3	Ingresa al su módulo de Boletas.	
4		Despliega las opciones crear y consultar.
5	Selecciona la opción Consultar.	
6		Mostrará un listado de catalogación por tipo de las boletas registrados en el sistema con algunas opciones (descargar, convertir editar y eliminar)
7	Selecciona el tipo de boleta que desea consultar (*bautismo, confirmación, matrimonio y defunción*).	
8		Despliega un listado de feligreses que cumplan con el criterio de catalogación dado por el usuario.
9	Selecciona la opción Editar.	
10		Mostrará los datos almacenados de la boleta.
11	Ingresa los datos que desea actualizar y selecciona la opción Editar Boleta "Tipo".	
12		Verifica que los datos ingresados sean válidos de acuerdo con el Anexo C (*Boleta*).
13		Mostrará un mensaje como el siguiente *"Boleta de "tipo" actualizada"*, confirmando que los datos se han guardado.
14		Regresa al paso 8, actualizando los campos modificados.

Curso Alterno

Campos vacíos

Paso	Descripción
12	Se muestra un mensaje como aparece a continuación "El campo "nombre del campo" es obligatorio". Llevándolo al paso 11.

Tabla 5.24: Especificaciones casos de uso, modificando boleta

	Especificación de Caso de Uso	Revisión: 001	Página: 1/1 Fecha: 08/04/2015

Eliminando Boleta

Actores: Párroco y Secretaria

Resumen:
Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe eliminar una boleta.

Precondición: El usuario debe estar registrado y validado en el sistema.

Post-condición:

Paso	Acción de los Actores	Respuesta del Sistema
1	Ingresa al módulo de Actas Sacramentales.	
2		Despliega los sub módulos Feligrés, Boletas y Partidas.
3	Ingresa al su módulo de Boletas.	
4		Despliega las opciones crear y consultar.
5	Selecciona la opción Consultar.	
6		Mostrará un listado de catalogación por tipo de las boletas registrados en el sistema con algunas opciones (descargar, convertir editar y eliminar)
7	Selecciona el tipo de boleta que desea consultar (*bautismo, confirmación, matrimonio y defunción*).	
8		Despliega un listado de feligreses que cumplan con el criterio de catalogación dado por el usuario.
9	Selecciona la opción Eliminar.	
10		Mostrará un mensaje como el siguiente *"Boleta de "tipo" eliminada"*, confirmando que los datos se han borrado del sistema.
11		Regresa al paso 8, actualizando el listado.

Tabla 5.25: Especificaciones casos de uso, eliminando boleta

ANEXO F: *Modelo ER, Ecclesia*

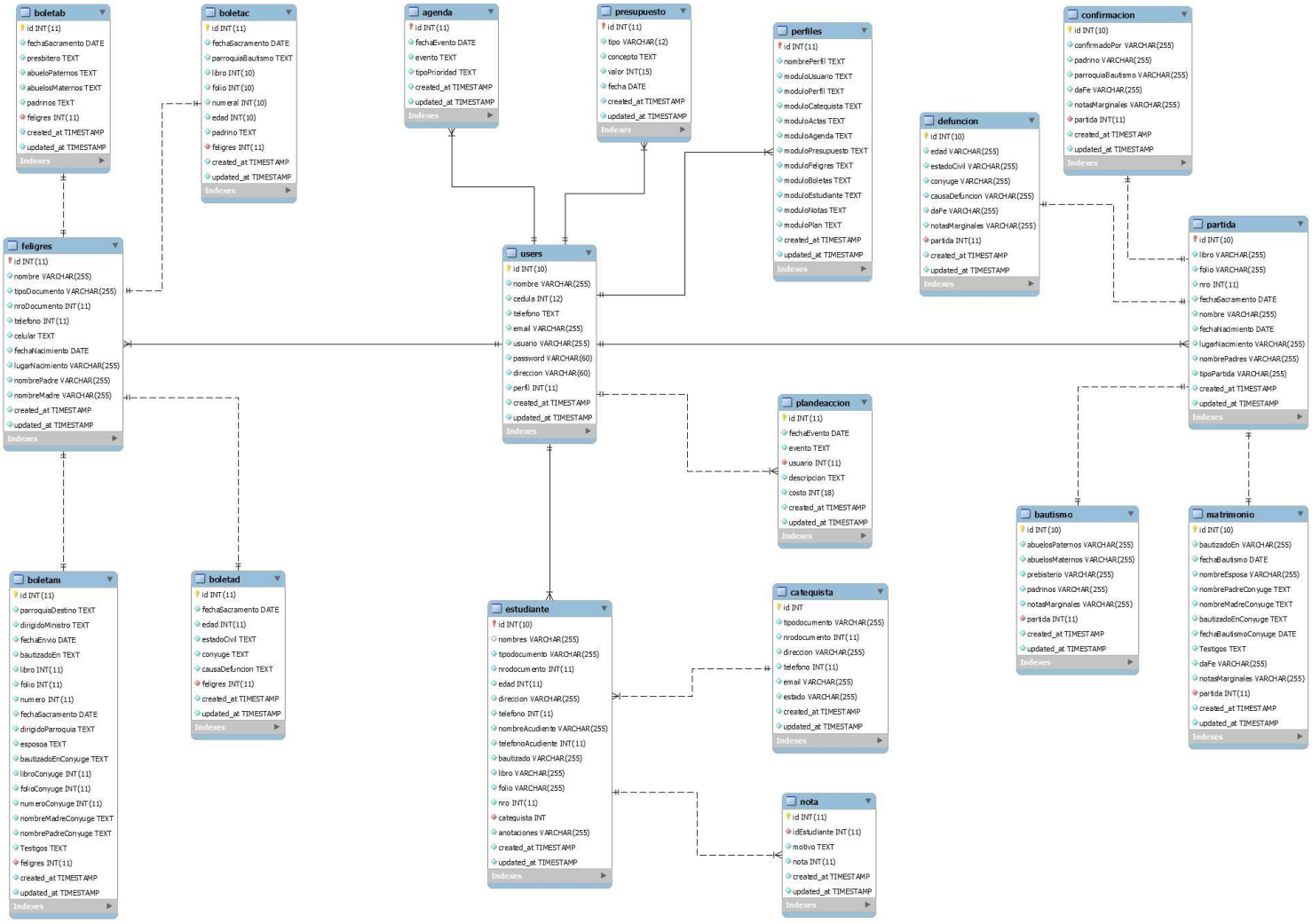

Figura 5.1: Diagrama Entidad Relación Ecclesia