UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
INGENIEROS DE SISTEMAS

TEMA:
DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DEL PLAN GENERAL DE INTERVENCIÓN DE LA INSPECTORÍA SALESIANA SAGRADO CORAZÓN DE JESÚS.

AUTORES:
DIEGO ANDRÉS ARAUJO VANEGAS
JOSUÉ DANIEL CAMPAÑA RAMÍREZ

TUTOR:
FRANKLIN EDMUNDO HURTADO LARREA

Quito, septiembre del 2016

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN

Nosotros DIEGO ANDRÉS ARAUJO VANEGAS y JOSUÉ DANIEL CAMPAÑA RAMÍREZ, con documento de identificación N° 1721480067 y 1721411906 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: "DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DEL PLAN GENERAL DE INTERVENCIÓN DE LA INSPECTORÍA SALESIANA SAGRADO CORAZÓN DE JESÚS", mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Diego Andrés Araujo Vanegas	Josué Daniel Campaña Ramírez
CI: 1721480067	CI: 1721411906

Fecha: 05-sep-2016

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN SUSCRITO POR EL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, "Desarrollo de una aplicación web para la gestión del plan general de intervención de la inspectoría salesiana sagrado corazón de Jesús", realizado por los estudiantes Diego Andrés Araujo Vanegas y Josué Daniel Campaña Ramírez, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, septiembre del 2016

Ing. Franklin Edmundo Hurtado Larrea

CI: 1713382016

Dedicatoria

Dedico este trabajo en primer lugar a Dios por haberme guiado durante toda esta etapa de mi vida y darme toda la fuerza para continuar en los momentos más difíciles que se presentaron.

A mi familia que ha sido un soporte en todo momento.

A mis Padres por toda la dedicación, amor, comprensión y apoyo incondicional en cada momento de mi vida y por ayudarme con todos los recursos para poder estudiar. Han sembrado en mí todos mis valores, mis principios, mi perseverancia y mi coraje para llegar a todas las metas que me proponga.

A mis hermanos por sus consejos y por su compañía que sin duda alguna son una fuente de motivación e inspiración.

Diego Andrés Araujo Vanegas

Este logro se lo dedico a Jehová Dios porque como menciona Revelación 4:11 “Digno eres tu Jehová, nuestro Dios de recibir la honra y el poder porque tu creaste todas las cosas y a causa existieron y fueron creadas”; también por darme a los mejores padres a quienes también les dedico este trabajo por su apoyo incondicional, inculcarme principios, valores, y por estar a mi lado en las buenas y en las malas.

A mis hermanos por estar a mi lado en todo momento dándome ánimo para seguir adelante.

Josué Daniel Campaña Ramírez

Agradecimiento

Agradecemos a la Universidad Politécnica Salesiana que ha contribuido en nuestra formación profesional y personal, a nuestro tutor de proyecto de titulación el Ingeniero Franklin Hurtado por habernos orientado y motivado para poder realizar nuestro trabajo.

Diego Andrés Araujo Vanegas

Josué Daniel Campaña Ramírez

Resumen

El siguiente proyecto trata de aplicar todos los conocimientos adquiridos en la carrera desarrollando una aplicación web que le sirva a la Inspectoría Salesiana Sagrado Corazón de Jesús como una herramienta de apoyo para realizar el seguimiento de la planificación que se lleva cada seis años con el objetivo de controlar la información generada por cada una de las instituciones que están asociadas a la Inspectoría, permitiéndoles contar con una plataforma en la que se pueda acceder desde cualquier lugar y en cualquier momento, facilitando así el cumplimiento de la planificación desarrollada. Por otro lado, la Inspectoría contará con valiosa información que le permita tomar decisiones basadas en reportes generados por la aplicación.

Abstract

The following project is about applying all the knowledges, acquired in the career developing a web application that serves the Salesian Province Sacred Heart of Jesus as a support tool to track the planning that takes every six years with the objective of controlling the information generated by each of the institutions that are associated with the Province, allowing to have them a platform which can be accessed from any where and at any time, facilitating the compliance of developed planning. On the other hand, the Province will have valuable information that allows to take decisions based on reports generated by this application.

Índice general

1 Introducción 1
1.1 Problema 1
1.2 Justificación del tema 1
1.3 Objetivo general – específicos 2
1.3.1 Objetivo general 2
1.3.2 Objetivos específicos 2
1.4 Alcance del proyecto 3
1.4.1 PBS 3
1.4.1.1 Identificación 4
1.4.1.2 Administración 4
1.4.1.3 Administración de obra 6
1.4.1.4 Usuario 7
1.4.1.5 Notificaciones 7
1.4.1.6 Reportería 8
2 Fundamentos teóricos 9
2.1 La comunidad Salesiana en Ecuador 9
2.2 Teoría de planificación y seguimiento 10
2.2.1 Planificación 10
2.2.2 Seguimiento 11
2.2.3 Planificación de la Inspectoría Salesiana Sagrados Corazón de Jesús 11
2.3 Especificación de requerimientos 12
2.3.1 Historias de usuario 13
2.3.2 Prototipado 13
2.4 Metodología 14
2.4.1 Scrum 14
2.4.2 Sprints 14
2.4.3 Reuniones diarias 14
2.5 UML 15
2.6 Scrum y UML 15
2.7 Arquitectura 15
2.7.1 Modelo Vista Controlador (MVC) 16
2.8 Diseño web adaptativo o responsivo 17
2.8.1 Ventajas del diseño adaptativo 18
3 Marco metodológico 19

3.1 Uso de SCRUM 19
3.1.1 Roles SCRUM utilizados 19
3.1.1.1 ScrumMaster y el equipo SCRUM 19
3.1.1.2 Propietario del producto 20
3.1.2 Sprints realizados 20
3.1.3 Planificación de reuniones 21
3.2 Uso UML 22
3.3 Uso de la PBS 22
3.4 Uso de la técnica del prototipo 22
4 Análisis y Diseño 24
4.1 Casos de uso 25
4.1.1 Caso de uso perfil administrador 26
4.1.2 Caso de uso perfil administrador obra 29
4.1.3 Caso de uso perfil usuario 30
4.2 Diagrama de base de datos 31
4.3 Arquitectura inicial del sistema 33
5 Construcción y pruebas 35
5.1 Arquitectura 35
5.1.1 Diagrama de componentes 37
5.2 Estándares de programación 40
5.2.1 Programación en servidor 40
5.2.2 Clases 40
5.2.3 Base de datos 41
5.3 Código relevante 41
5.3.1 Clase conexión 41
5.3.2 Planificación del sistema 43
5.3.3 Redirección desde correo electrónico 45
5.4 Configuración de aplicación en ambiente de pruebas 48
5.5 Pruebas 48
5.5.1 Pruebas funcionales 49
5.5.2 Pruebas de rendimiento 51
5.5.3 Pruebas de aceptación 56
5.6 Breve descripción de la aplicación 59
6 Conclusiones 63
7 Recomendaciones 64

8 Referencias bibliográficas 65
9 Anexos 67

Índice de tablas

Tabla 1. Especificación de Requerimientos. 25
Tabla 2. Métodos clase conexión 43
Tabla 3. Resumen de pruebas funcionales 49
Tabla 4. Resultado prueba de rendimiento caso 1 53
Tabla 5. Resultado prueba de rendimiento caso 2 54
Tabla 6. Diccionario de datos de base de datos. 71
Tabla 7. Prueba funcional 1. 78
Tabla 8. Prueba funcional 2. 79
Tabla 9. Prueba funcional 3. 79
Tabla 10. Prueba funcional 4. 80
Tabla 11. Prueba funcional 5. 80
Tabla 12. Prueba funcional 6. 81
Tabla 13. Prueba funcional 7. 81
Tabla 14. Prueba funcional 8. 82
Tabla 15. Prueba funcional 9. 82
Tabla 16. Prueba funcional 10. 83
Tabla 17. Prueba funcional 11. 84
Tabla 18. Prueba funcional 12. 84
Tabla 19. Prueba funcional 13. 85
Tabla 20. Prueba funcional 14. 86
Tabla 21. Prueba funcional 15. 86
Tabla 22. Prueba funcional 16. 87
Tabla 23. Prueba funcional 17. 88
Tabla 24. Prueba funcional 18. 88
Tabla 25. Prueba funcional 19. 89
Tabla 26. Prueba funcional 20. 90
Tabla 27. Prueba funcional 21. 90
Tabla 28. Prueba funcional 22. 91
Tabla 29. Prueba funcional 23. 92
Tabla 30. Prueba funcional 24. 93
Tabla 31. Prueba funcional 25. 94
Tabla 32. Prueba funcional 26. 94
Tabla 33. Prueba funcional 27. 95
Tabla 34. Prueba funcional 28. 95
Tabla 35. Prueba funcional 29. 96
Tabla 36. Prueba funcional 30. 96

Índice de figuras

Figura 1. PBS que modela los componentes del proyecto. 3
Figura 2. Diagrama organizacional de obras Salesianas 10
Figura 3. Diagrama de organización de PEPS 12
Figura 4. Diagrama del ciclo de vida MVC. 17
Figura 5. Caso de uso que muestra el funcionamiento general del perfil administrador. 26
Figura 6. Caso de uso que muestra el funcionamiento escenario planificar. 28
Figura 7.Caso de uso que muestra el funcionamiento general del perfil administrador obra. 29
Figura 8. Caso de uso que muestra el funcionamiento general del perfil usuario. 30
Figura 9. Versión 3 del diagrama de base de datos. 32
Figura 10. Diagrama inicial de la arquitectura del sistema 34
Figura 11. Diagrama que muestra la arquitectura del sistema 35
Figura 12. Diagrama de componentes de la aplicación. 38
Figura 13. Gráfico que muestra la clase conexión con sus métodos. 42
Figura 14. Gráfico que muestra la función para imprimir en tablas. 44
Figura 15. Gráfico que muestra la función que busca por palabra clave. 45
Figura 16. Código para control de sesión en perfil de actividad. 46
Figura 17.Código que muestra el inicio de sesión 47
Figura 18. Diagrama que muestra la configuración para la conexión a la base de datos en Azure. 48
Figura 19. Gráfico que muestra los valores correspondientes al caso 1. 55
Figura 20. Gráfico que muestra los valores correspondientes al caso 2. 55
Figura 21. Gráfico de resultados obtenidos a la pregunta la aplicación facilitó su planificación. 56
Figura 22. Gráfico de resultados obtenidos a la pregunta el aspecto visual de la aplicación. 57
Figura 23. Gráfico de resultados obtenidos a la pregunta la usabilidad que presenta la aplicación 57
Figura 24. Gráfico de resultados obtenidos a la pregunta la experiencia al utilizar la aplicación 58
Figura 25. Gráfico de resultados generales prueba de aceptación. 58
Figura 26. Gráfico que muestra el Iniciar Sesión de la aplicación. 59
Figura 27. Ilustración que muestra las opciones del perfil administrador. 60
Figura 28. Diagrama que muestra las opciones del perfil administrador obra. 60
Figura 29. Ilustración que muestra las opciones del perfil Usuario. 61
Figura 30. Gráfico que muestra las opciones que tiene en común todos los perfiles. 61
Figura 31. Grafica que muestra el login del prototipo. 68
Figura 32. Grafica que muestra las opciones del perfil Administración del prototipo. 68
Figura 33. Grafica que muestra las opciones de Reportería del prototipo. 69
Figura 34. Grafica que muestra las opciones del perfil Administración Obra del prototipo. 69
Figura 35. Grafica que muestra cómo se visualizan las actividades del usuario del prototipo. 70

Figura 36. Grafica que muestra cómo se visualiza el detalle de la actividad del prototipo. 70
Figura 37. Matriz de plan de iteración. 76
Figura 38. Matriz de revisión reunión 26_05_2016..... 77
Figura 39. Matriz de revisión reunión 02_06_2016..... 77
Figura 40. Encuesta para la prueba de aceptación. 97
Figura 41. Capacitación de software a usuarios..... 98
Figura 42. Carta de aceptación del software. 99

Índice de anexos

Anexo 1. Glosario de términos 67
Anexo 2. Prototipo 68
Anexo 3. Diccionario de datos 71
Anexo 4. Plan de iteración 76
Anexo 5. Matriz de revisión de sprint 77
Anexo 6. Registro de pruebas funcionales 78
Anexo 7. Encuesta de prueba de aceptación 97
Anexo 8. Grabaciones de historias de usuario 97
Anexo 9. Fotografías de capacitación a usuarios 98
Anexo 10. Carta de aceptación de software 99

1 Introducción

En las siguientes hojas del documento se tratara de documentar las diferentes experiencias y fases que se desarrollaron a lo largo del proceso de realización de este trabajo de titulación, en el cual fueron atendidos los problemas que se le presentaban a la Inspectoría Sagrados Corazones de Jesús al momento de realizar el seguimiento de la planificación de actividades de todas las obras salesianas, haciendo uso de los conocimientos necesarios adquiridos durante el paso por la carrera de ingeniería de sistemas de la Universidad Politécnica Salesiana.

1.1 Problema

La Inspectoría Salesiana Sagrado Corazón de Jesús realiza una planificación cada 6 años de un PGI (Plan General de Intervención) Inspectorial, cada obra Salesiana que pertenece a dicha institución debe realizar un PGI sectorial en el que se debe desarrollar actividades que cumplan con las intervenciones propuestas en el PGI Inspectorial, la Inspectoría ha identificado la necesidad de medir el cumplimiento del plan establecido en tiempo real, sin tener que realizar esfuerzos de recolección de datos y procesamiento de los mismos con el objetivo de obtener la información precisa de cada una de las obras, optimizando así recursos como tiempo y dinero.

1.2 Justificación del tema

Como respuesta al problema planteado anteriormente se propone la construcción de una aplicación web que tenga la capacidad de unificar la planificación con el fin de alinear el cumplimiento de los objetivos establecidos en el PGI Inspectorial, se podrán obtener reportes en tiempo real, sobre el nivel de cumplimiento de cada una de las obras pertenecientes a la Inspectoría.

La Universidad Politécnica Salesiana al pertenecer a la ISSCJ ayudo a través de los estudiantes de la Carrera de Ingeniería de Sistemas a desarrollar un software que ayude a resolver la problemática que se presenta al momento de gestionar el cumplimiento del PGI.

Hoy por hoy, con el advenimiento del Internet, la Web 3.0, la necesidad de procesar grandes volúmenes de datos para la toma de decisiones en tiempo real, hace que las organizaciones hagan el uso de la tecnología para de esta manera agilizar, optimizar y perfeccionar algunos procesos que se ejecutan dentro de las mismas.

1.3 Objetivo general – específicos

1.3.1 Objetivo general

Analizar, diseñar y construir una aplicación web que facilite la planificación y que automatice el seguimiento del PGI que actualmente se está realizando para la ISSCJ.

1.3.2 Objetivos específicos

Identificar y analizar la metodología y la estructura de planificación utilizadas para el desarrollo del PGI de la ISSCJ para adquirir el conocimiento necesario de los procesos relacionados con planificación que utiliza dicha institución.

Especificar los requerimientos generales del software.

Analizar y diseñar la aplicación de tal forma que los requerimientos sean atendidos.

Codificar la aplicación utilizando el patrón de arquitectura de software MVC para garantizar la reutilización de código y posteriormente facilitar la tarea de mantenimiento de la misma.

Desplegar una versión beta de la aplicación en un ambiente de pruebas.

1.4 Alcance del proyecto

Para definir el alcance del proyecto fue de gran utilidad el uso de la Estructura de Descomposición del Producto conocida en inglés como *Product Breakdown Structure* (PBS).

1.4.1 PBS

La PBS se utilizó para descomponer de una manera exhaustiva y ordenada cada una las partes funcionales del software.

PBS del Proyecto

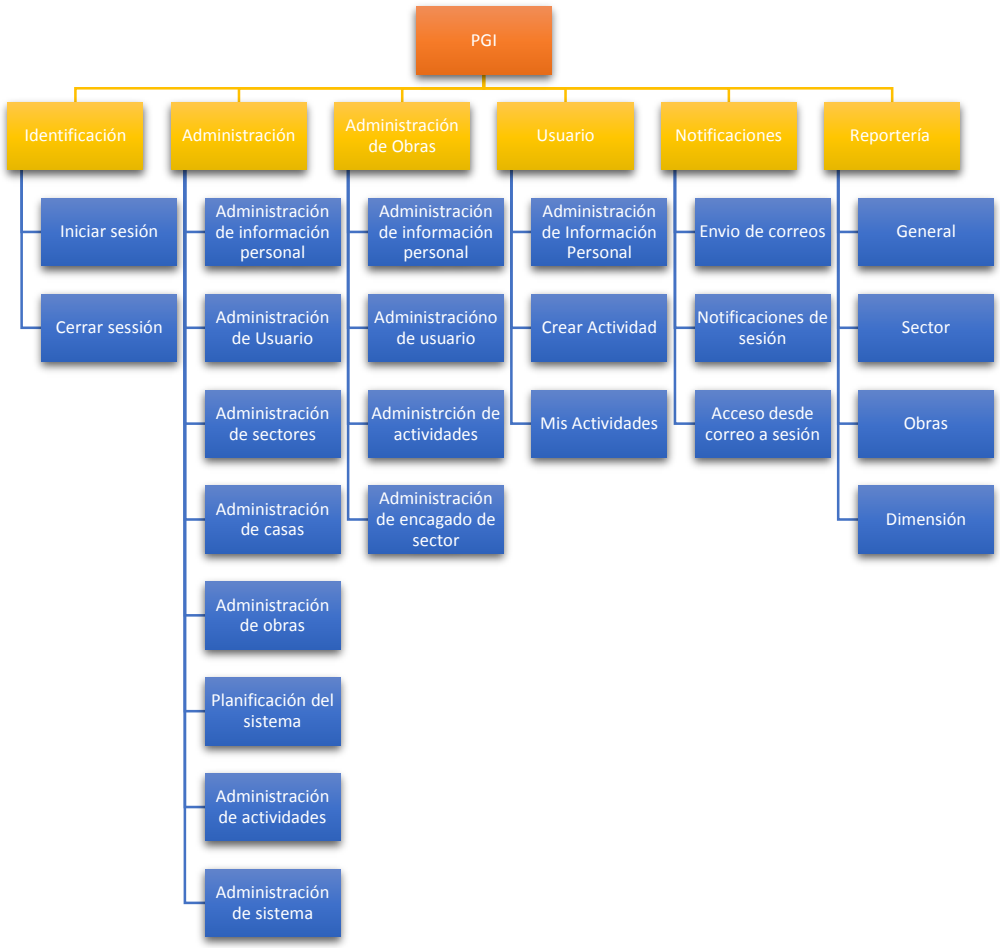


Figura 1. PBS que modela los componentes del proyecto.
Elaborado por: Campaña, J (2016).

Como se observa en el gráfico anterior el PGI está compuesto por 6 módulos los cuales son Identificación, Administración, Administración de Obras, Usuario, Notificaciones y Reportaría.

1.4.1.1 Identificación

Está conformado por iniciar sesión y cerrar sesión.

Iniciar sesión. Se refiere a que el usuario use sus credenciales en este caso correo electrónico y contraseña para comenzar una sesión.

Cerrar sesión. El usuario finaliza el uso del sistema.

1.4.1.2 Administración

Conformado por la administración total del sistema, está constituido por administración de información personal, usuario, sectores, casas, obras, planificación, actividades y sistema.

Administración de información personal. El usuario podrá modificar su información personal como sus nombres, apellidos, correo electrónico e contraseña.

Administración de usuario. Permite crear, consultar, modificar, desactivar usuarios con privilegios del mismo nivel e inferiores pertenecientes a diferentes obras salesianas.

Administración de sectores. Se puede crear, consultar, modificar, descargar POA (Plan Operativo Anual) y desactivar sectores.

Al deshabilitar el sector todas las obras salesianas pertenecientes a dicho sector son reasignados a otro.

Administración de casas. Facilita crear, consultar, modificar y desactivar casas.

Suprimir una casa, las obras salesianas pertenecientes se asignan a otra.

Administración de obras. Proporciona la creación, consulta por casa, modificación y eliminación de obras salesianas.

El desactivar una obra salesiana, todas las actividades ligadas podrán ser asignadas a otra obra o desactivadas.

Planificación del sistema. Estructurada por dimensiones, objetivos, procesos e intervenciones.

El administrador tiene la posibilidad de crear, seleccionar, editar, activar o desactivar dimensión.

Al seleccionar una dimensión, el usuario podrá crear, elegir, modificar, activar o desactivar objetivos ligados a dicha dimensión.

El elegir un objetivo facilita crear, escoger, actualizar, activar o desactivar procesos enlazados al objetivo elegido.

Escoger un proceso, permite crear, editar, asignar sectores, activar o desactivar intervenciones acoplados al proceso seleccionado.

Desactivar o activar un componente de la planificación, todos los elementos ligados serán activados o desactivados según sea el caso, como por ejemplo al deshabilitar una

dimensión los objetivos, procesos, intervenciones y actividades enlazados también serán deshabilitados.

Administración de actividades. Posibilita crear, consultar, modificar, gestionar y desactivar actividades para las diferentes obras Salesianas.

Configuración del sistema. Se puede editar la fecha de inicio y duración del periodo de planificación.

1.4.1.3 Administración de obra

Conformado por administración de información personal, administración de usuarios, administración de actividades y administración de sectores.

Administración de usuarios. Permite crear, consultar, modificar y desactivar usuarios con privilegios de nivel inferior que pertenezca solo a la obra que corresponde el administrador de obra, por ejemplo un usuario administrador de obra que pertenece al Oratorio Magone puede crear, buscar, editar usuarios con perfil de usuario que estén aliados a este establecimiento.

Administración de actividades. Facilita crear, consultar, modificar, gestionar y desactivar actividades que estén relacionadas a la obra que pertenece el usuario administrador de obra, por ejemplo el administrador de obra de Oratorio Magone solo puede crear y gestionar actividades que pertenezcan a dicha institución.

Administración de encargado sector. Se habilita cuando el usuario Administrador de Obra fue asignado como responsable del sector, puede editar datos básicos y descargar POA.

1.4.1.4 Usuario

Formado por administración de información personal, crear actividad, mis actividades.

Crear actividad. Permite la creación de actividades relacionadas a la obra que pertenece el usuario.

Mis actividades. Proporciona la visualización de todas las actividades que pertenece al usuario, permite editar datos básicos de la actividad como por ejemplo nombre, detalle y añadir observación, mientras que el estado solo lo pueden editar el administrador de obra o el administrador general del sistema.

1.4.1.5 Notificaciones

Estructurada por envió de correos electrónicos, notificaciones de sesión y acceso desde correo a sesión.

Envió de correos electrónicos. El usuario recibirá un correo cuando se le adjudique alguna responsabilidad, por ejemplo cuando se le ponga como responsable de una actividad.

Notificaciones de sesión. Se podrá visualizar y gestionar la notificación desde sesión como Facebook.

Acceso desde correo a sesión. El usuario puede acceder desde el correo al detalle de la notificación que se ha generado.

1.4.1.6 Reportería

La reportería está presente en los perfiles de Administración y Administración de Obra.

Los usuarios tendrán reportes en tiempo real en modo gráfico y tabla. Administradores de obra pueden tener reportes a nivel de la obra a la que pertenece, mientras que administradores del sistema los obtiene a nivel general, sector u obra.

2 Fundamentos teóricos

2.1 La comunidad Salesiana en Ecuador

La Comunidad Salesiana con 128 años en el Ecuador se encuentra regada por todo el país en diversos ambientes y actividades (misiones andinas, misiones amazónicas, oratorios, centros juveniles, parroquias, proyecto salesiano ecuador y ups) en las que se desarrolla la misión salesiana según se específica en la propuesta educativo-pastoral que actúan con una idéntica finalidad: la promoción integral de los jóvenes y su mundo (Molina, 2016, pág. 5).

Los Salesianos con el fin de cumplir su misión basada en el Sistema Preventivo e inspirada en los valores del Evangelio, para formar "honrados ciudadanos y buenos cristianos"; han desarrollado una gran variedad de obras y servicios entre estos: escuelas, colegios, talleres, oratorios, centros, movimientos juveniles, parroquias, misiones amazónicas, misiones andinas, universidad, comunicación, centros de espiritualidad, fundaciones y casas de formación que se encuentran organizados de la siguiente manera (Molina, 2016, pág. 8).

Diagrama organizacional obras Salesianas

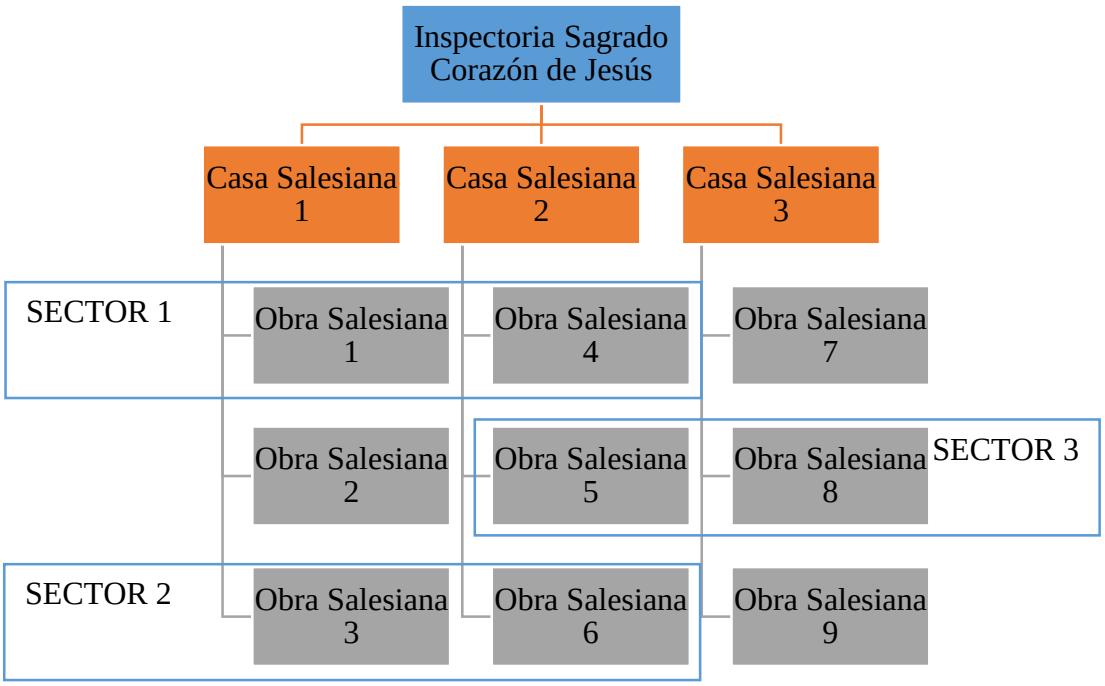


Figura 2. Diagrama organizacional de obras Salesianas
Elaborado por: Araujo, D (2016).

2.2 Teoría de planificación y seguimiento

2.2.1 Planificación

La planificación es un instrumento eficaz que a las organizaciones e instituciones ayuda a analizar y tomar decisiones sobre qué rumbo debe tomar en el futuro para que de esa manera pueda adaptase a los cambios y demandas del medio (Burgwal & Cuellar, 1999, págs. 25,26).

Según Robbins y Coulter (2005) “la planificación consiste en definir las metas de la organización, establecer una estrategia general para alcanzarlas y trazar planes exhaustivos para integrar y coordinar el trabajo de la organización. La planificación se ocupa tanto de los fines (que hay que hacer) como de los medios (como hay que hacerlo)” (pág. 157).

La planificación son un conjunto de acciones que utiliza organizaciones e instituciones para alcanzar los objetivos propuestos al futuro.

2.2.2 Seguimiento

Es un proceso sistemático del cual se recopila información con la visión de analizar y medir los avances logrados de las metas propuestas en la planificación, lo cual les permite tomar decisiones antes de que sea demasiado tarde y asegurar el cumplimiento de objetivos propuestos a futuro (Suazion, 2009, pág. 8).

2.2.3 Planificación de la Inspectoría Salesiana Sagrados Corazón de Jesús

La Inspectoría Salesiana Sagrados Corazón de Jesús con el fin de cumplir su objetivo educativo – pastoral cada sexenio elabora el Proyecto Educativo Pastoral Salesiano Inspectoría (PEPSI) que es el Plan General de Intervención que se convertirá en un instrumento que oriente todas las iniciativas y los recursos hacia una adecuada y eficaz realización de la misión salesiana en el país.

El objetivo principal del PEPSI es llevar a la Inspectoría y a las comunidades locales a obrar con una mentalidad compartida, clarificada en objetivos y criterios, delimitando las prioridades y dimensiones con los que los PEPS de los sectores de la pastoral están constituidos (Molina, 2016, pág. 22).

A continuación, se muestra un diagrama de cómo está constituido el PEPS.

Diagrama de PEPS

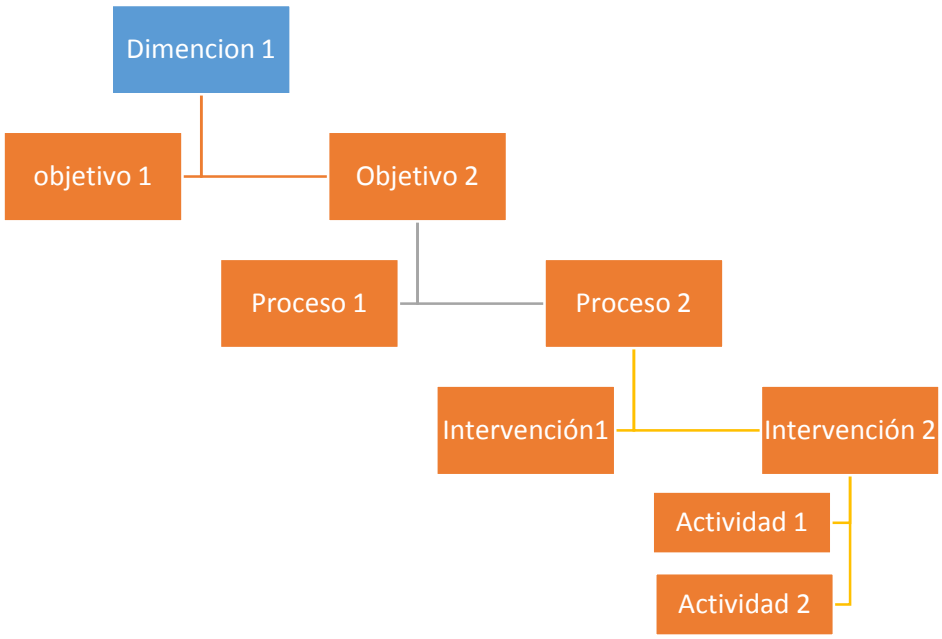


Figura 3. Diagrama de organización de PEPS
Elaborado por: Araujo, D (2016).

Como se puede observar en el diagrama el PEPS está conformado por dimensiones a su vez estas tienen varios objetivos, consecuentemente cada uno de estos están conformado por algunos procesos, consecutivamente estos están ligados a diferentes intervenciones y estas tiene asociadas distintas actividades que se realizan en cada obra Salesiana.

2.3 Especificación de requerimientos

Procedimiento donde se anotan los requerimientos de diferentes maneras como puede ser en lenguaje natural, lenguaje formal, representaciones graficas o representaciones simbólicas (Tuffley, 2005).

A los requerimientos no solo se los puede especificar de una forma escrita, se está utilizando técnicas como la construcción de modelos o prototipos que brindan una representación más clara de los requisitos (Borland, 2005).

2.3.1 Historias de usuario

Son especificaciones de requerimientos de manera breve y sencilla desde la visión de un usuario o cliente del sistema. (Cohn, 2013).

Las historias pueden cubrir varios requerimientos, las cuales se les llama épicas, al no poder finalizarlas en un sprint, se las fragmenta en diversas historias de usuario (Cohn, 2009).

Los propietarios del producto no son los únicos que pueden escribir las historias de usuario pero son responsables de que existan los requerimientos para formar la pila del producto, la cual es compuesta por múltiples historias de usuario. *Scrum* al ser una metodología incremental interactiva permite que las historias de usuario se escrita a lo largo de todo el proyecto (Cohn, 2013).

2.3.2 Prototipado

Los prototipos permiten recolectar requerimientos que no se tienen en claro ya que brinda una perspectiva tolerable de lo que será la aplicación (Fu, Hao, Yen & Bastani, 2008).

2.4 Metodología

2.4.1 Scrum

Más que una metodología es un marco de trabajo que está fundamentado en un modelo incremental e iterativo del desarrollo de software, que trata de lograr resultados papables en un corto tiempo dando así al proyecto evolución continúa. Scrum es idóneo para practicarlo en proyectos donde los requerimientos son cambiantes o por lo general no están completamente definidos por el cliente (Alaimo, 2013, pág. 21).

Según Palacio (2014) "Scrum se puede adoptar de forma técnica, aplicando reglas definidas, o pragmática, adoptando los valores originales Scrum con reglas personalizadas" (pag.18). De esta manera permite realizar pequeños ajustes para adaptar la metodología al proyecto con el que se encuentre.

2.4.2 Sprints

Se refiere a un plazo de tiempo que va desde una a seis semanas de trabajo en la cual se define el entregable en el que se va a realizar del proyecto, este contiene un conjunto de tareas que van a ser atendidos para lograr cierta funcionalidad así como también la fecha límite para la presentación (Kniberg, 2007, pág. 22).

2.4.3 Reuniones diarias

Scrum plantea que se deber realizar reuniones diarias en las que participan todos los integrantes del equipo las cuales deben realizarse en un mismo lugar con una duración máxima de 15 minutos, sin embargo hoy en día gracias a la tecnología se cuenta con herramientas que pueden lograr simular dicho ejercicio dando como resultado la introducción de un nuevo concepto, el Scrum distribuido el cual define que no siempre

se puedan realizar reuniones presenciales debido a la localización y disponibilidad de cada uno de los integrantes del equipo, dando la alternativa de realizar dicha actividad usando medios remotos de forma completa o parcialmente durante todo el proyecto (Raúl Herranz, 2011, pág. 17).

2.5 UML

Según Rumbaught, Jacobson y Booch (2000) “el Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software” (pag.27).

2.6 Scrum y UML

Scrum es una metodología que atenido una gran acogida debido a que se preocupa en el desarrollo del software como tal, mas no en la documentación pero tarde o temprano se puede llegar a la conclusión de que es necesario contar con documentación que ayude a llevar a cabo cada una de las tareas que se van presentando en el proceso de desarrollo es por esta razón que el lenguaje unificado de modelo puede colaborar con diagramas que ayuden al equipo a establecer detalles sobre el cómo se va a llevar a cabo dichas actividades. Por otro lado los diagramas de UML dan mayor valor agregado a cada uno de los entregables que se vallan realizando garantizando así la satisfacción del cliente (Wei et al., 2014, pág. 979).

2.7 Arquitectura

Según Arias y Durango (2016). “La Arquitectura es la organización fundamental de un sistema incorporada en sus componentes, sus relaciones con el entorno y los principios que conducen su diseño y evolución” (pág.207).

Según Sommerville (2011) la Arquitectura “identifica la estructura global del sistema, los principales componentes, sus relaciones y como se distribuyen” (pág.39).

La Arquitectura está compuesta de tres componentes que son:

Datos: información a ser procesada.

Proceso: transforman los datos.

Conexión: comunica entre los Datos y el proceso (Perry & Wolf, 1992, pág. 44).

2.7.1 Modelo Vista Controlador (MVC)

El MVC es un patrón de diseño de Arquitectura de Software, es muy utilizado en aplicaciones que manejan gran cantidad de datos, por este motivo este patrón sugiere separar el software en tres capas:

El modelo: son los datos del sistema y su lógica del negocio.

Vista: es la interfaz gráfica del usuario.

Controlador: es el intermediario entre el modelo y la vista (López, 2008, págs. 11,12).

A continuación se presenta el ciclo de vida del MVC.

Ciclo de Vida MVC

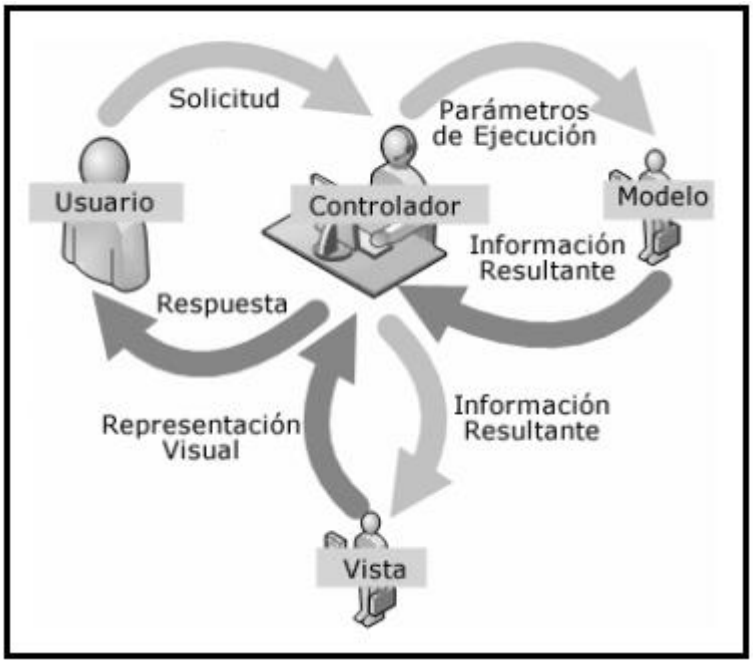


Figura 4. Diagrama del ciclo de vida MVC.
Elaborado por: López, A (2008).

El diagrama anterior muestra el ciclo de vida del MVC, el usuario hace la petición al Controlador. Entonces el Controlador envía la petición al Modelo. El modelo responde al Controlador con los datos solicitados, en esta etapa el Controlador entrega los datos a la Vista la cual se encarga de transfórmalos en forma visual. Finalmente la Vista devuelve los datos al Controlador quien se encarga de enviar al usuario (López, 2008, págs. 12,13).

2.8 Diseño web adaptativo o responsivo

Conocido también como RWD de las inglés Responsive Web Desing se define como una técnica de diseño y desarrollo web que tiene por objetivo adaptar al sitio web a los diferentes ambientes que el usuario este haciendo uso utilizando estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS (Cascading Style Sheets) (Labrada Martínez & Salgado Ceballo, 2013, págs. 3-9).

2.8.1 Ventajas del diseño adaptativo

Optimización de la visualización de contenido del sitio web tanto de escritorio como de dispositivos móviles a través una sola versión de HTML y CSS.

Reducción de costos en creación y mantenimiento en el diseño de las diferentes pantallas que se puedan presentar.

Evita el desarrollo de versiones diferentes según las marcas de los dispositivos es decir una versión para Android y otra versión para iPhone.

3 Marco metodológico

Las siguientes líneas del documento detallan como se utilizó la metodología Scrum a lo largo de todo el proyecto.

3.1 Uso de SCRUM

Hoy en día la mayoría de proyectos de desarrollo de software son llevados a cabo usando metodologías adaptativas, debido a que la naturaleza de los proyectos de desarrollo de software es altamente cambiante.

Dado que el cliente deseaba llevar un seguimiento continuo a cada incremento para de esta manera regular el cumplimiento de sus expectativas y proporcionar una retroalimentación en cada uno de estos, se tomó la decisión de usar *SCRUM* como metodología de desarrollo de software, debido a que permite al dueño del producto ser una pieza fundamental para el proyecto facilitando la oportunidad que desde el inicio del proyecto, se gestione las expectativas que se van generando al paso de cada incremento.

3.1.1 Roles SCRUM utilizados

3.1.1.1 ScrumMaster y el equipo SCRUM

Durante la ejecución del proyecto en cada sprint el papel de ScrumMaster fue rotado entre las personas que conformaron el equipo de trabajo, dando así la oportunidad al equipo, de planificar actividades y entregables, generando una valiosa experiencia al llevar a cabo las funciones de dicho papel.

3.1.1.2 Propietario del producto

Desde el inicio del proyecto se identificó a la persona que desempeñaría dicho papel, la cual facilitó toda la información necesaria para definir cada uno de los requerimientos funcionales del software y a su vez aprobar cada uno de los sprints realizados.

3.1.2 Sprints realizados

Dada la disponibilidad de trabajo de cada uno de los integrantes del equipo la jornada de labor diario se definió en 4 horas de desarrollo de lunes a viernes y 8 horas diarias los fines de semana, dando como resultado un total de seis sprints, los cuales fueron planificados de acuerdo a la complejidad que representaba cada una de las tareas que contenían cada incremento previsto.

Cada sprint tuvo una duración de una semana a continuación se detalla el contenido de cada uno de estos:

Sprint 1. Se realizó un prototipo general de toda la aplicación con la finalidad de tener una idea clara sobre la interfaz que se desarrolló para los diferentes usuarios.

Sprint 2. Se trabajó en la arquitectura del software, así como el diagrama físico de base de datos, tomando en cuenta las interrelaciones que tienen los diferentes módulos del software. También se trabajó en el primer elemento del *Product Backloc* que consistió en todas las funcionalidades de administración del software, teniendo así una primera versión del sistema.

Sprint 3. Se construyó el siguiente ítem del *Product Backloc* que consistió en implementar las funciones de administración de cada obra, obteniendo una segunda versión del sistema.

Sprint 4. El consecuente componente del *Product Backloc* consistió en la implantación de las funcionalidades que tiene un usuario final dentro del sistema, lo cual permitió lograr una nueva versión del sistema.

Sprint 5. La consecutiva pieza del *Product Backloc* fue la creación de las funciones de notificaciones que tiene el sistema.

Sprint 6. Finalmente el último eslabón del *Product Backloc* fue la construcción de la funcionalidad de reportaría, de esta forma se tuvo como resultado la versión final del sistema.

3.1.3 Planificación de reuniones

Con el objetivo de dar el seguimiento al trabajo realizado día a día, se sostuvo reuniones diarias con duración de quince minutos en las primeras horas del día, en las que se trataban los avances realizados del sprint vigente.

Estas reuniones fueron realizadas vía video conferencia utilizando herramientas de trabajo adecuadas, tal es el caso de *Hangouts*, debido a la jornada laboral de los integrantes del equipo, se notó una gran dificultad de realizar reuniones presenciales.

Al terminar cada sprint se planificaba una reunión con el dueño del producto en la cual se trataba los siguientes puntos, una revisión de todo lo presentado evaluando el cumplimiento de todos los requerimientos que fueron establecidos al inicio, así mismo

como recibiendo una retroalimentación sobre lo presentado, el segundo punto de la reunión se trataba de clarificar los requerimientos para el siguiente incremento.

3.2 Uso UML

Dado que el sistema cuenta con varios tipos de perfiles de usuario, se tomó la decisión de utilizar diagramas de casos de uso para facilitar el análisis y explicar las funcionalidades que el sistema permite realizar a cada uno.

Este diagrama facilitó la comprensión del flujo con el que el usuario debe operar el sistema.

Por otro lado también se utilizó un diagrama físico de base de datos para determinar la estructura de datos que se desplegó en *Mysql*. A pesar de haber iniciado los desarrollos con una estructura defina de base de datos, se vio la necesidad de ir modificando a medida que se fue avanzando a través de cada sprint, teniendo así dos versiones previas a la tercera y definitiva.

3.3 Uso de la PBS

Se utilizó para desglosar cada una de las funciones que tiene el sistema de una forma gráfica, con la finalidad de disponer un esquema global del sistema que permitiera definir los distintos módulos que alimentaron el *Product Backloc* y concretar una estrategia para la reutilización de código.

3.4 Uso de la técnica del prototipo

Se desarrolló un prototipo del sistema global, este fue presentando en el primer sprint del proyecto, el objetivo que se tenía con este, era darle al dueño del producto una visión sobre la aplicación que se tenía en mente dando así la oportunidad a que el

propietario aterrice sus ideas y apruebe las diferentes interfaces que se pensó desde el inicio del proyecto. Este prototipo solamente se usó al inicio del proyecto debido a que después de cada sprint se tenía una parte funcional de la aplicación, la cual en la mayoría de todos los incrementos terminaba completamente diferente al prototipo ya que los requerimientos fueron cambiando a medida que el sistema se iba construyendo.

4 Análisis y Diseño

Uno de los desafíos más relevantes que se tuvo que enfrentar fue el entender la lógica del negocio respecto a la planificación, para eso se utilizaron algunas técnicas como el prototipado e historias de usuarios.

La metodología fue de gran ayuda para resolver el desafío, dada que al ser incremental interactiva permitió que se realizaran diferentes reuniones con el propietario del producto claramente descrito en el marco metodológico, de esta manera en cada reunión se entendía cada vez más la lógica del negocio respecto a la planificación de la ISSCJ.

La especificación de requerimientos se llevó a cabo de una manera repetitiva, esto se logró con historias de usuario, estas historias de usuario no fueron escritas si no grabadas con la previa autorización del propietario del producto.

A continuación, se muestra ejemplos de requerimientos con su respectiva especificación.

Tabla 1.
Especificación de Requerimientos

Requerimiento	**Especificación**
Sistema debe tener perfiles.	Los perfiles que debe manejar el sistema son: Administrador, Administrador Obra y Usuario.
Usuario puede pertenecer a varias obras y tener varios perfiles.	En administración de usuarios del perfil Administrador implementar que un usuario puede pertenecer a varias obras u tener varios perfiles
El sistema debe generar notificaciones	Las notificaciones que debe generar el sistema son: envió de correos y notificaciones de sesión como Facebook. El usuario puede acceder desde el correo directamente a gestionar la notificación.
Generar reportes.	El perfil Administrador puede generar reporte general, sector u obra. El perfil Administrador Obra genera reportes a nivel general pero solo de la obra a la que pertenece.
Implementar administración por sector.	Los sectores deben tener nombre, descripción y un responsable. El responsable del sector es un administrador de obra que pertenezca a una obra que este asociado a ese sector
Implementar administración por casa	Las casas deben tener nombre, director de casa y teléfono.
Las obras deben estar asociadas a un sector y a una casa	En administración de obras implementar que una obra debe estar asociada a un sector y a una casa.
Intervenciones deben estar asociadas a un sector o varios.	Implementar en planificación del sistema que una intervención debe estar asociada un sector.
Cambiar perspectivas por dimensiones.	En todo el sistema cambiar los textos que mencionan perspectivas por dimensiones.
Los usuarios deben ser filtrados por la casa que pertenecen.	En administración de sectores generar filtro de usuarios por casa seleccionada
Ordenar por seleccionar, editar y desactivar.	En planificación ordenar los botones de las tablas por seleccionar, editar y desactivar.
Subir evidencia.	La evidencia de cumplimento de las actividades se la puede subir en un archivo PDF o doc, el límite para subir la evidencia es de 7 días.

Nota: Esta tabla contiene ejemplos de requerimientos con su respectiva especificación.

Fueron varios los entregables a continuación se exponen los más relevantes.

4.1 Casos de uso

Los casos de uso fueron de gran utilidad ya que permitieron identificar los actores de la aplicación y las funciones que cumple cada uno.

4.1.1 Caso de uso perfil administrador

Caso de uso perfil administrador

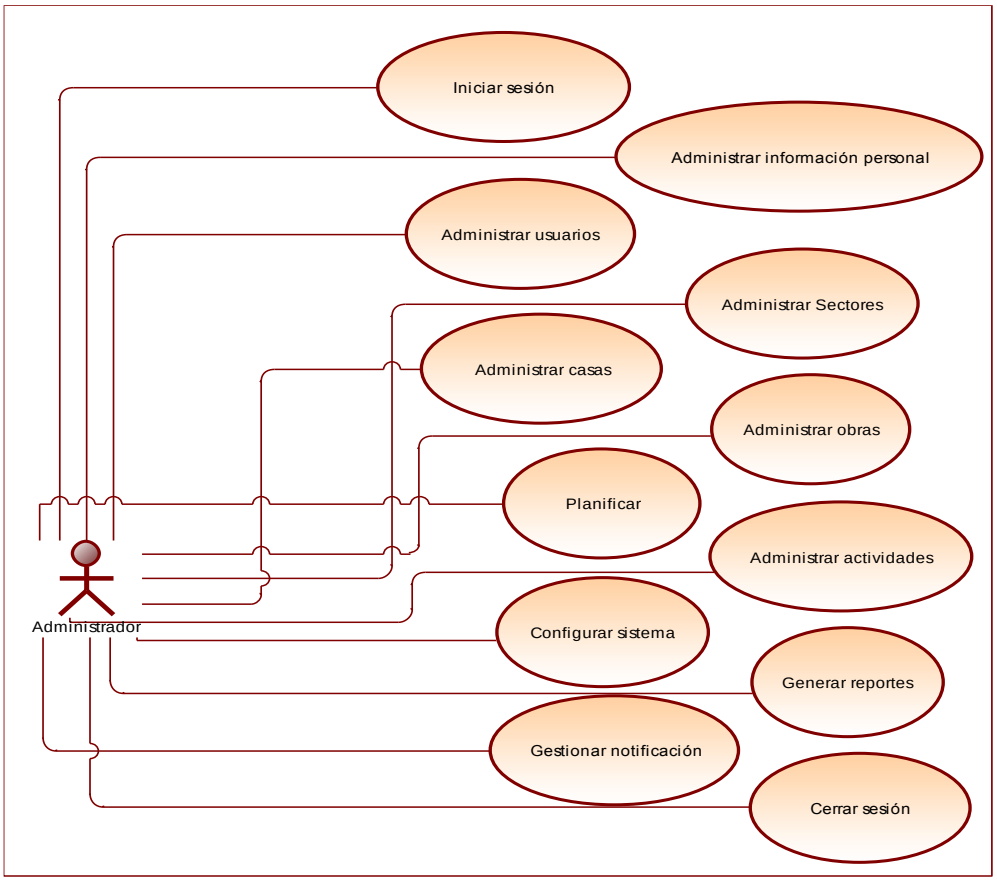


Figura 5. Caso de uso que muestra el funcionamiento general del perfil administrador.
Elaborado por: Campaña, J (2016).

En la figura anterior se puede apreciar como el perfil Administrador va interactuando con la aplicación, el primer escenario al que se enfrenta es iniciar sesión en donde el usuario utiliza sus credenciales para autentificarse al sistema, una vez realizado el requisito anterior tendrá las siguientes opciones: administrar información personal, administrar usuarios, administrar sectores, administrar casas, administrar obras, planificar, administrar actividades, configurar sistema el cual permite reiniciar el periodo de planificación, generar reportes, gestionar notificaciones y cerrar sesión donde el usuario sale del sistema.

Para administrar usuarios, sectores, casas, obras, actividades, el escenario es parecido entre sí, ya que el usuario puede crear, editar y desactivar los registros de cada uno. En cuanto al escenario de planificar varia, eso se lo aprecia en el siguiente caso de uso.

Caso de uso perfil administrador escenario planificar

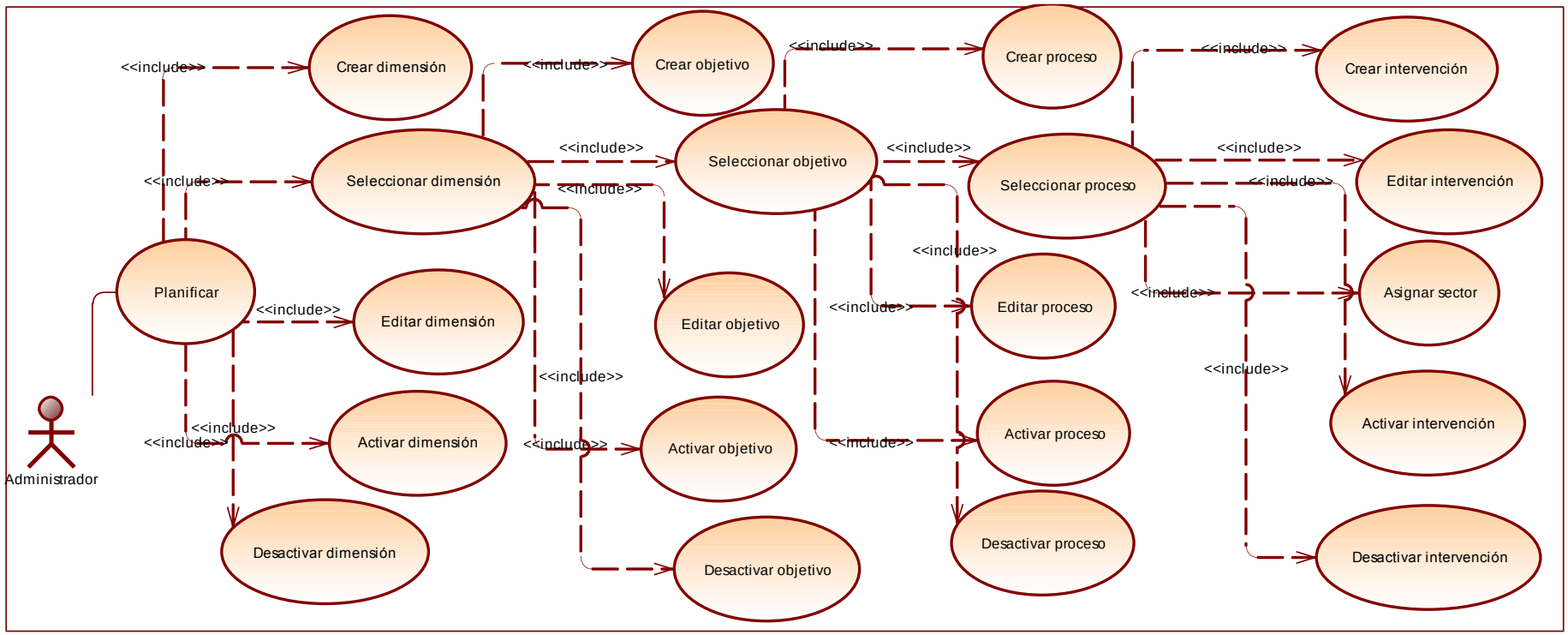


Figura 6. Caso de uso que muestra el funcionamiento escenario planificar.
Elaborado por: Campaña, J (2016).

El perfil Administrador en el escenario de planificar puede crear, seleccionar, modificar, activar o desactivar dimensiones. Al seleccionar una dimensión podrá crear, elegir, editar, deshabilitar o habilitar objetivos. Elegir un objetivo permite crear, escoger, editar, activar o desactivar procesos. Escoger un proceso permite crear, asignar sector, modificar, deshabilitar o habilitar intervenciones.

Como se pudo apreciar el caso de uso anterior ayuda a entender la manera de planificación de la Inspectoría Salesiana Sagrado Corazón de Jesús.

4.1.2 Caso de uso perfil administrador obra

Caso de uso perfil administrador obra

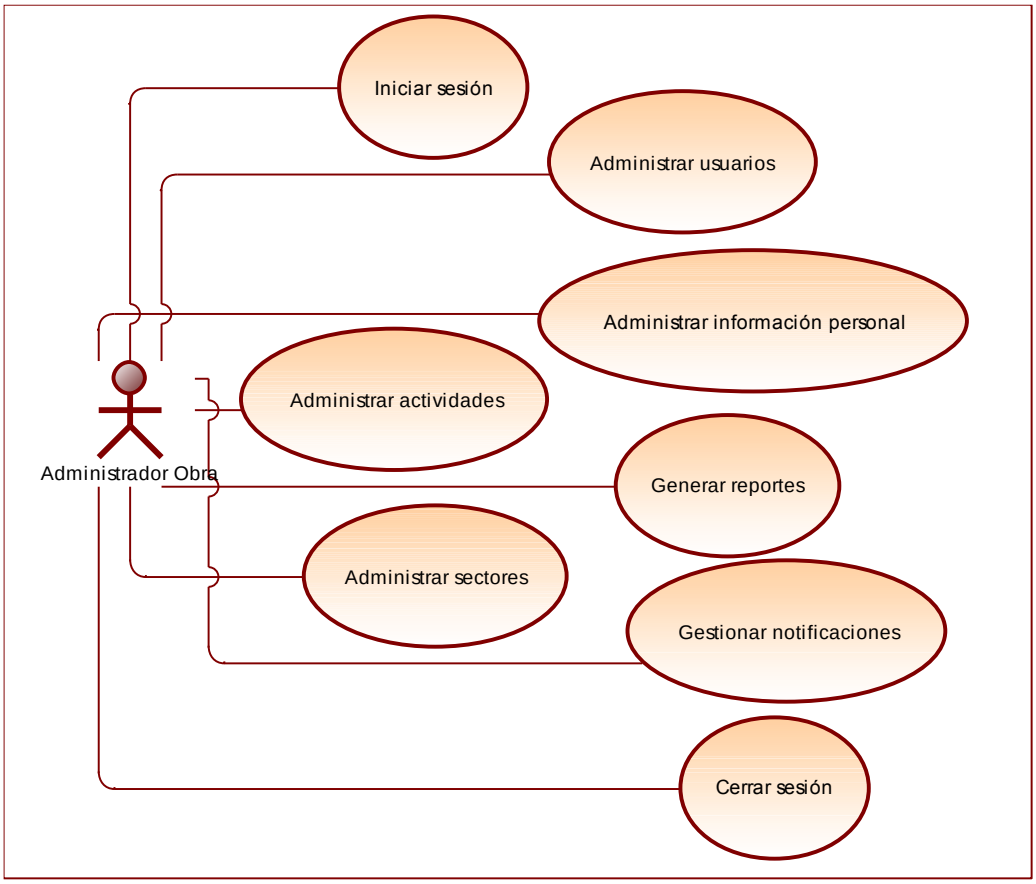


Figura 7.Caso de uso que muestra el funcionamiento general del perfil administrador obra.
Elaborado por: Campaña, J (2016).

En el gráfico anterior se puede apreciar como el perfil Administrador Obra interactúa con el sistema, su primer contacto con la aplicación es iniciar sesión, en esta instancia utiliza sus credenciales para autentificarse, una vez cubierto este requisito, se le despliegas las siguientes opciones: administrar información personal, administrar usuarios a nivel de perfiles inferiores, administrar actividades, generar reportes, gestionar notificaciones, administrar sectores y cerrar sesión.

Todas las opciones que presenta este perfil las tiene a nivel de obra, en cuanto a administrar de sectores, el administrador de obra la tendrá siempre y cuando haya sido nombrado representante del sector y podrá descargar POA y editar datos básicos del sector.

4.1.3 Caso de uso perfil usuario

Caso de uso perfil usuario

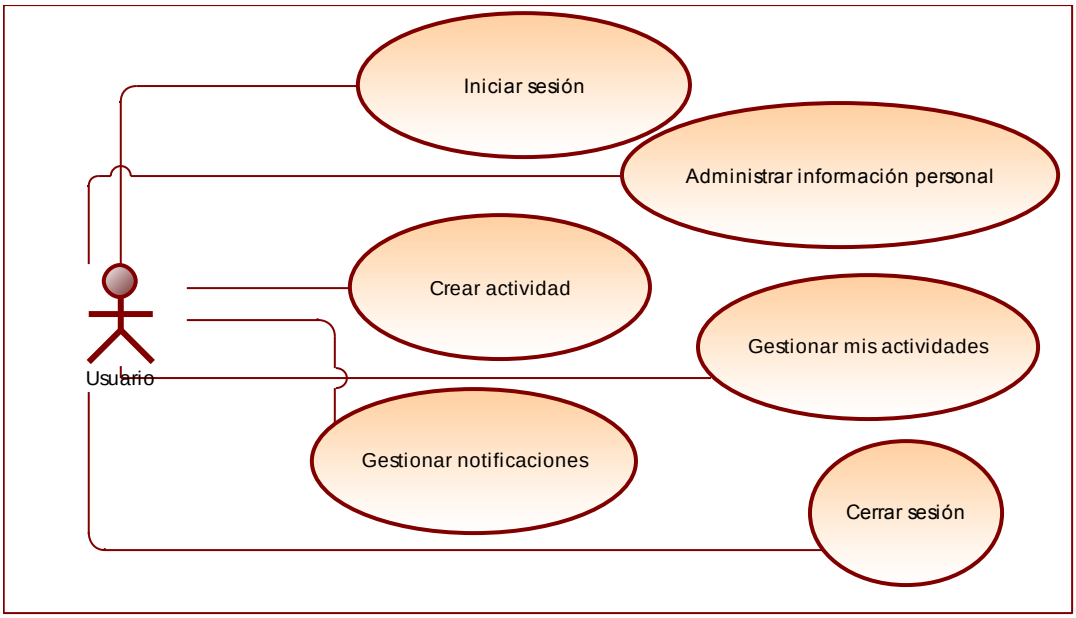


Figura 8. Caso de uso que muestra el funcionamiento general del perfil usuario.
Elaborado por: Campaña, J (2016).

La ilustración anterior muestra como el perfil *Usuario* se relaciona con el software, su primer requisito que debe cubrir es autentificarse, una vez autentificado el usuario podrá: administrar su información personal, crear actividad a nivel de obra, gestionar sus actividades, gestionar notificaciones y cerrar sesión.

4.2 Diagrama de base de datos

Para definir el diagrama de Base de datos que utilizaría el sistema se decidió utilizar el modelo entidad relación.

Diagrama de base de datos

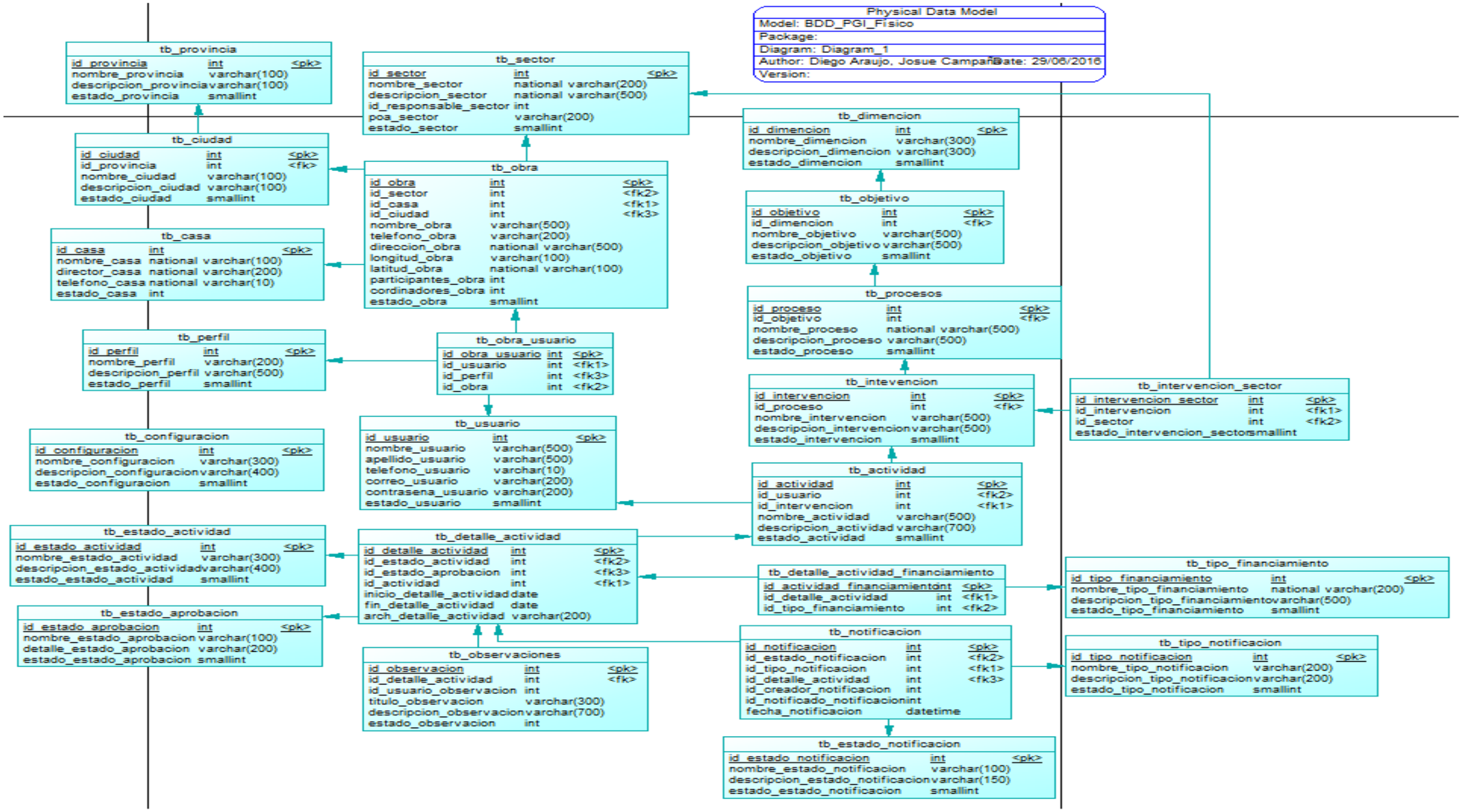


Figura 9. Versión 3 del diagrama de base de datos.
Elaborado por: Araujo, D (2016).

Después de dos versiones previas, se llegó al tercer y último diagrama de base de datos mostrado en la figura.

En la segunda versión del diagrama de base de datos se agregó la tabla de tb_usuarioobra para de esta manera se pueda agregar la funcionalidad al sistema de que un usuario pueda pertenecer a varias obras con diferentes perfiles.

En la tercera versión de diagrama de base de datos se agregó las tablas de tb_provincia, tb_ciudad y se implementó una relación de estas, con la tabla tb_obra con la finalidad de filtrar cada una de las obras Salesianas por ciudad y provincia, así mismo se agregó la tabla de tb_intervencion_sector con la finalidad de agregar los diferentes sectores que agrupan cada una de las intervenciones.

A pesar de haber realizado los cambios detallados anteriormente en el diagrama de base de datos, incremento tras incremento la base de datos se iba refinando de esta manera se llegó a tener una estructura óptima para el sistema.

4.3 Arquitectura inicial del sistema

Durante el análisis para definir la arquitectura que llevaría el sistema, se llegó a la conclusión de que se debía utilizar una arquitectura amigable para el equipo de desarrollo para de esta manera optimizar la reutilización de código y tiempos llegando a definir el uso del patrón de arquitectura MVC que se ata a todas las características mencionadas anteriormente y además brinda otras ventajas como la fragmentación de la lógica del negocio para lograr que la aplicación sea escalable, sea entendible por todo el equipo de trabajo, facilita la realización de pruebas a cada uno de los componentes que conforman la aplicación.

Debido a que el patrón MVC divide a la aplicación en tres principales componentes como es el modelo en el que se definió albergar todas las clases que procesan los datos ingresados por el usuario, así como las solicitudes a la base de datos, en lo que se refiere a vista se definió ubicar todos los estilos y códigos que sean necesarios para la presentación de la información obtenida por la aplicación y finalmente en el controlador se centralizara todos los conductores de solicitudes de datos a las diferentes clases que realice el usuario.

Diagrama Modelo Vista Controlador

Figura 10. Diagrama inicial de la arquitectura del sistema
Elaborado por: Araujo, D (2016).

5 Construcción y pruebas

La Inspectoría Salesiana Sagrado Corazón de Jesús posee usuarios de todo nivel en conocimientos tecnológicos, por este motivo se requería que la aplicación sea amigable con el usuario y responsive para los usuarios que utilicen dispositivos móviles para esto se hizo uso de frameworks que permiten crear interfaces de gran usabilidad y que se adaptan a cualquier dispositivo como es Bootstrap en su versión 3.

Por otro lado se definieron estándares y patrones de programación como es MVC con el propósito de llevar una codificación en donde todo el equipo pueda entender y llevar de una manera ordenada cada uno de los componentes que conforman la aplicación.

5.1 Arquitectura

Se aplicó el patrón de arquitectura MVC en el software construyendo cada uno de los elementos de dicho patrón, los cuales son: Modelo, Vista y Controlador de la siguiente manera.

MVC del sistema

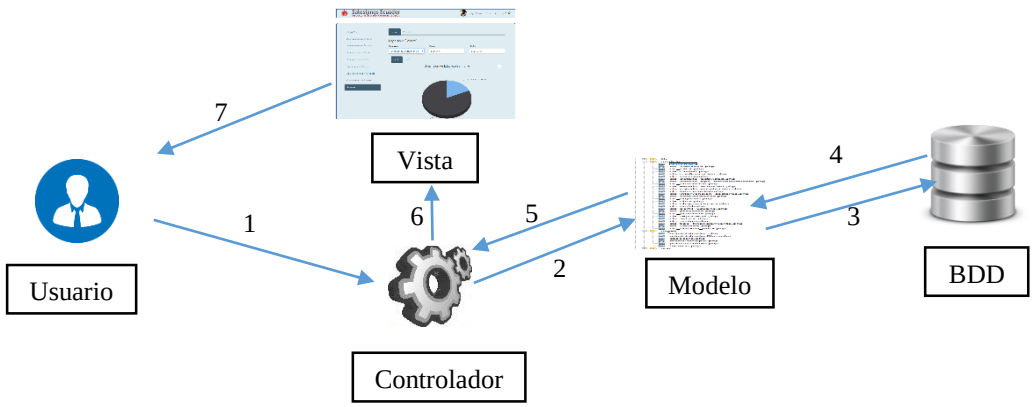


Figura 11. Diagrama que muestra la arquitectura del sistema
Elaborado por: Araujo, D (2016).

En el diagrama mostrado anteriormente se muestra la interacción de cada una de los componentes de la arquitectura del sistema:

1. El usuario a través del navegador web realiza una solicitud al controlador, ya sea de tipo *request*, *get* o *post*.

2. El controlador recibe diferentes parámetros para buscar la parte del modelo que se necesita y envía una solicitud al modelo.

3. El modelo que está conformado de clases que describen la lógica de negocio recibe parámetros para procesar los datos ingresados por el usuario y de ser necesario realiza una solicitud de datos al gestor de base de datos.

4. El gestor de base de datos responde a la solicitud con un resultado.

5. El modelo recibe la información del gestor de base de datos la procesa, la transforma en datos en formato de JSON y contesta a la solicitud que realizo el controlador.

6. El controlador recibe la información y la reenvía a la vista para ser procesada.

7. La vista a través de frameworks de presentación como son bootsrap y jquery del lado del cliente dan formato a los resultados obtenidos de acuerdo a las solicitudes realizadas por el usuario.

5.1.1 Diagrama de componentes

En el siguiente diagrama de componentes se tratará de mostrar de una manera gráfica cada uno de los elementos que conforman la aplicación teniendo en cuenta cada una de las librerías a nivel de todos los elementos que conformen la vista, el controlador y modelo.

En dicho diagrama se encontrará de derecha a izquierda varios niveles que se explicará a continuación.

Diagrama de componentes

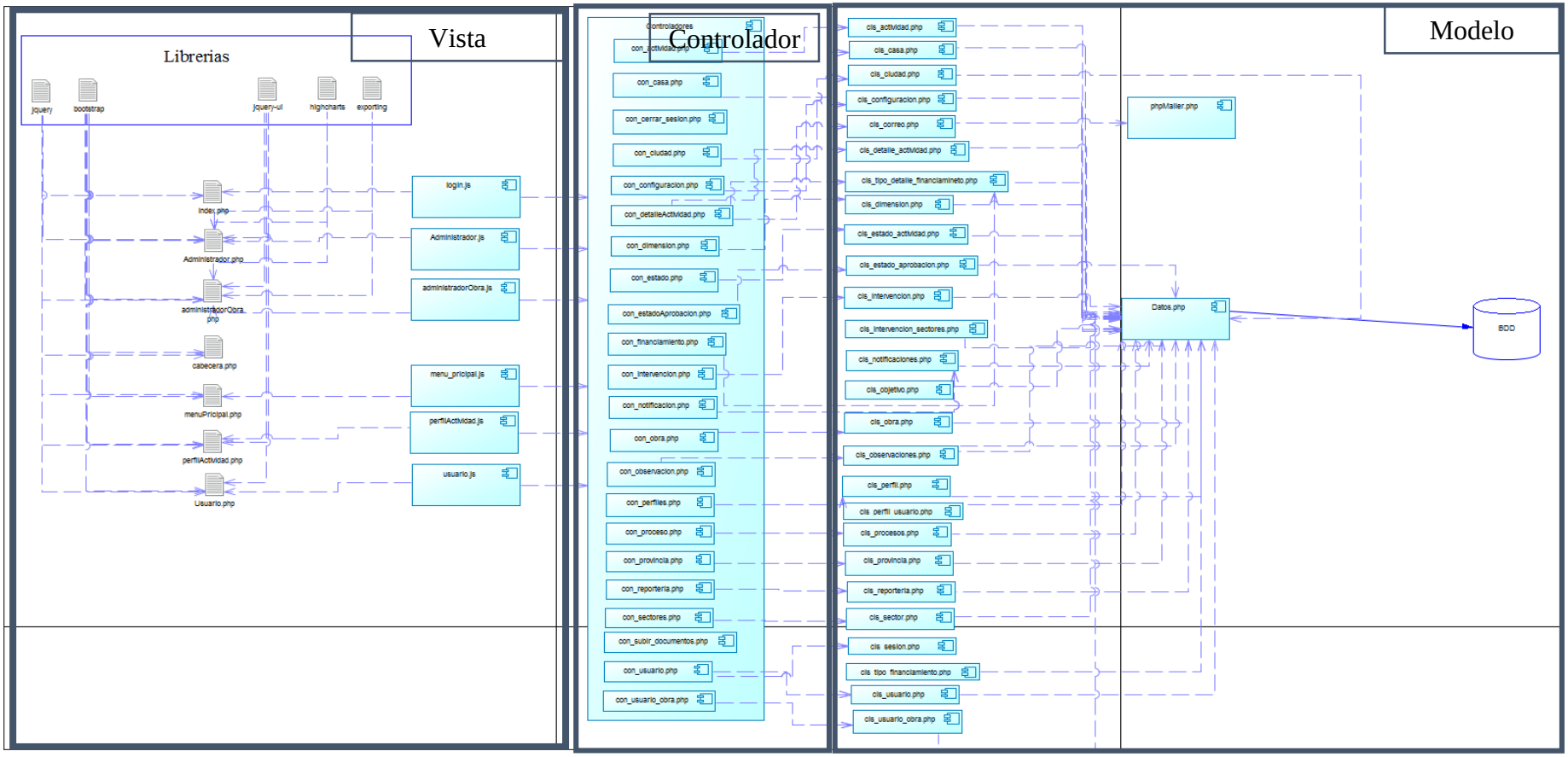


Figura 12. Diagrama de componentes de la aplicación
Elaborado por: Araujo, D (2016).

Partiendo de izquierda a derecha se encuentran las páginas web que conforman todo el sistema, las mismas que utilizan las librerías de *Bootstrap*, *Jquery*, *Jquery*-ui, *Exporting* y *highcharts* con el objetivo de garantizar una experiencia agradable y clara al momento de interactuar con el sistema.

En el siguiente nivel se tiene un componente extenso que es el de los controladores cuya función será de enviar solicitudes a las respectivas clases de acuerdo a las necesidades que se presenten debido a la interacción que esté realizando el usuario, a su vez los diferentes controladores también tendrán la función de recibir la respuesta de la solicitud enviada anteriormente y enviarla al componente que realizo dicha solicitud.

A continuación, se encuentran las clases, estas tienen la función de recibir las solicitudes que vienen desde los diferentes controladores así mismo como procesarlas, realizar solicitudes a la base de datos de ser necesario y responder a dichas solicitudes.

En este eslabón también se localiza la librería de *Phpmailer*, la misma que ayuda a la clase cls_correo.php a enviar correos electrónicos a través de la aplicación.

Consecutivamente se tiene la clase Datos.php la misma que tiene la función de comunicar con *Mysql* y retornar las respuestas a las solicitudes realizadas por cada una de las clases que se encuentran en el anterior nivel.

5.2 Estándares de programación

5.2.1 Programación en servidor

- Se omitió el uso de la letra ñ por n.
- Se omitió el uso de tilde en todas las palabras usadas dentro de la programación.
- Los archivos de clase solo contienen una y solo una clase y los nombres de los archivos inician con el prefijo “cls_” seguido del nombre de la clase por ejemplo: cls_actividad.php, cls_casa, etc.
- Los nombres de los archivos de controlador inician con el prefijo “con_” seguido del nombre del controlador por ejemplo: con_actividad.php, con_casa.php , etc.

5.2.2 Clases

- Todos los nombres de las clases están definidos en singular.
- Todas las clases están declaradas usando el prefijo “cls_” por ejemplo: class cls_usuario { }, class cls_perfil{} , etc.
- Todas las variables declaradas dentro de una clase llevan el nombre de dicha clase después del nombre de la variable usando “_” ejm: nombre_usuario, apellido_usuario, etc.
- Todas las clases cuentan con métodos *get* y *set* que permiten la lectura y la modificación correspondientemente de todas las variables declaradas dentro de la clase.

- Todos los nombres de las funciones dentro de una clase llevan escritura tipo camelcase por ejemplo: actualizaUsuario(), desactivaUsuario(), etc.

5.2.3 Base de datos

- Todos los nombres de las tablas de base de datos tienen el “tb_” por ejemplo: tb_usuario , tb_perfil, etc

- Todas las claves foráneas de cada una de las tablas están declaradas con el prefijo “fk_r_” seguido por el nombre de la relación por ejemplo: fk_r_intervencion_actividad, fk_r_usuario_actividad, etc.

- Todos los nombres de variables de las diferentes tablas de la base de datos están definidas por el nombre en singular de la variable seguida de “_” y del nombre de la tabla que pertenecen por ejemplo: id_usuario, nombre_usuario, etc.

5.3 Código relevante

A pesar de que todo el código de la aplicación es sumamente importante destacan ciertos fragmentos de código para de esta manera dar a conocer como está estructurada la codificación de toda la aplicación.

5.3.1 Clase conexión

Se definió la creación de esta clase con el objetivo de adaptarse al patrón de arquitectura utilizado centralizando el flujo de datos y la reutilización de métodos que fueron desarrollados desde el inicio de la construcción de la aplicación.

La clase conexión tiene un papel de enlace con la base de datos debido a que es el conducto por el cual se trasmiten todas las solicitudes que se van generando a través de la interacción de los usuarios con la aplicación.

Clase conexión

```php
<?php
class Datos {
    private $conexion;
    public function __construct() {
        $prueba = true;
        if ($prueba) {
            $servidor = "localhost";
            $usuario = "root";
            $clave = "";
            $base = "pgi3";
        } else {...6 lines }
        $this->conexion = mysql_connect($servidor, $usuario, $clave);
        mysql_select_db($base, $this->conexion);
        mysql_query("SET NAMES 'utf8'", $this->conexion);
    }
    public function __destruct() {
    }
    public function filtro($sql) {
        $result = mysql_query($sql, $this->conexion);
        return $result;
    }
    public function cerrarfiltro($datos) {
        mysql_free_result($datos);
    }
    public function proximo($datos) {
        $arreglo = mysql_fetch_array($datos);
        return $arreglo;
    }
    public function ejecutar($sql) {
        mysql_query($sql, $this->conexion);
        return mysql_insert_id();
    }
    public function actualiza($sql) {
        mysql_query($sql, $this->conexion);
        return mysql_affected_rows();
    }
    public function ejecutarFoto($sql) {
        mysql_query($sql, $this->conexion);
    }
    public function cerrarConexion() {
        mysql_close($this->conexion);
    }}
```

Figura 13. Gráfico que muestra la clase conexión con sus métodos.
Elaborado por: Araujo, D (2016).

Tabla 2.
Métodos clase conexión

Método	Función
__construct()	El constructor se encarga de instanciar una conexión con la base de datos y el sistema.
__destruct()	El destructor se ejecuta cuando el objeto es destruido.
filtro($sql)	Esta función devuelve un resourse si la consulta se ha realizado con éxito y un valor de false si la consulta no devolvió nada.
cerrarfiltro($datos)	Esta función cumple la función de liberar la memoria del resultado.
proximo($datos)	Esta función devuelve un arreglo de datos que provienen desde el resourse generado anteriormente.
ejecutar($sql)	Esta función sirve específicamente ingresar datos en la base de datos, de modo que retorna el valor de la nueva primary ken generado o el valor de -1 si no se pudo ingresar los datos.
actualiza($sql)	Esta función sirve específicamente para actualizar registros de modo que retorne el valor de registros modificados
cerrarConexion()	Esta función se encarga de cerrar la conexión con la base de datos.

Nota: Esta tabla contiene los métodos de la clase conexión.

5.3.2 Planificación del sistema

En planificación del sistema, los nombres de las dimensiones, objetivo, procesos e intervenciones son demasiado grandes para mostrar en un combobox, por esta razón se buscó una mejor manera para mostrar dicha información, llegando a la conclusión de mostrar los nombres haciendo el uso de tablas.

A continuación en la gráfica siguiente se visualiza la función que imprime en la tabla.

Función que imprime en tabla

```
function actualizartablaobjetivoss_m(num) {
    var idPerspectiva = idperspetiva_m;
$.ajax({
        data: "op=obtenerTablaObjetivossObra&id_obra=" + idObra + "&idDimension=" + idPerspectiva + "&num=" + num,
        url: "../controladores/con_objetivo.php",
        type: 'POST',
        beforeSend: function() {        },
        success: function(response) {

            var a = JSON.parse(response);
            var aux = "";
            if (a.str != 0) {
                aux = "<thead><tr><td>Objetivo</td><td>Seleccionar</td></tr></thead>";
                for (var i = 0; i < a.str; i++) {
                    aux += "<tr><td class='oculto' id='idpersob_m" + a[i].ID_OBJETIVO + "'>" +
                            a[i].ID_DIMENSION + "</td><td id='nombreobjetivo_m" + a[i].ID_OBJETIVO + "'>" +
                            a[i].NOMBRE_OBJETIVO + "</td><td class='oculto' id='descripcionobjetivo_m" +
                            a[i].ID_OBJETIVO + a[i].DESCRIPCION_OBJETIVO
                            + "</td><td><button onclick='seleccionarObjetivo_m ("
                            + a[i].ID_OBJETIVO +
                            ")' type='button' class='btn btn-default'>\n\
                              <span class='glyphicon glyphicon-ok'></span></button></td></tr>";
                }
            }

            $("#tablaObjetivoss_m").html(aux);
        }
    });
```

Objetivos

Dimensión

Dimensión de Educación en la Fe

Crear Objetivo Atrás

Objetivo	Descripción	Seleccionar	Editar	Desactivar
Renovar la educación en la Fe de corresponsables y destinatarios vivenciando la primacía de Dios.	Renovar la educación en la Fe de corresponsables y destinatarios vivenciando la primacía de Dios.	✔	✎	⊘
Fortalecer la primicia de la Palabra y su inculturación, en el mundo indígena acorde a las distintas situaciones que viven en la actualidad	Fortalecer la primicia de la Palabra y su inculturación, en el mundo indígena acorde a las distintas situaciones que viven en la actualidad	✔	✎	⊘

Figura 14. Gráfico que muestra la función para imprimir en tablas.
Elaborado por: Campaña, J (2016).

En la figura anterior se puede apreciar que para imprimir en la tabla se extrae la información con *Ajax*, en *Javascript* se arma en una variable la estructura de la tabla y con el nombre de la tabla punto *Html* se imprime en la tabla.

Para crear una actividad se necesita seleccionar una intervención, se encontró que una manera más eficiente para llegar a la intervención que se desea seleccionar es buscando por palabra clave.

En la siguiente figura se observa la función que realiza la búsqueda por palabra clave.

Función que busca por palabra clave

```
function buscarcon() {
    var a = jsonInterveciones;

        intervencionesconsulta = new Array();
    var ni = ($("#busquedaPC").val()).toLowerCase();
    if(ni.trim()!=""){
        var resu = (ni.trim()).split(" ");
    for (var i = 0; i < a.str; i++) {

        for (var j = 0; j < resu.length; j++) {
            if (resu[j].trim == "") {

            } else {
                var n = ((a[i].NOMBRE_INTERVENCION).toLowerCase()).search(resu[j]);

                if (n > -1) {
                    intervencionesconsulta.push(i);
                    break;
                }
            }
        }
    }
}
}
```

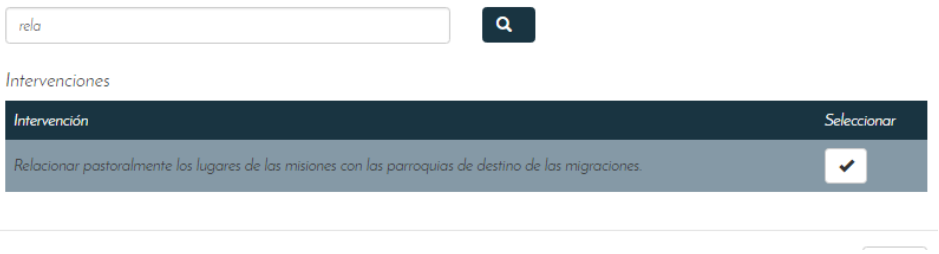


Figura 15. Gráfico que muestra la función que busca por palabra clave.
Elaborado por: Campaña, J (2016).

En la ilustración anterior se observa la función que realiza la busque por palabra clave, para ello se realiza una sola carga de todas las intervenciones que pertenecen al sector al que pertenece la obra y se las almacena en una variable global, desde el lado del cliente cada vez que el cliente realiza una búsqueda se recorre la variable global buscando coincidencias, se almacena en un *Array* y finalmente se imprime el *Array*.

5.3.3 Redirección desde correo electrónico

Debido a que el sistema envía diferentes notificaciones a los diferentes correos electrónicos de cada usuario con el objetivo de dar la posibilidad de un acceso directo a la actividad donde se generó alguna notificación, se presentó el reto de controlar el acceso de tal manera que solamente se pueda ver el perfil de dicha actividad siempre

y cuando el usuario haya iniciado sesión. A continuación se muestra las líneas de código que llevan a cabo dicha funcionalidad.

Control de sesión en actividad

```php
<?php
require_once '../modelo/cls_sesion.php';
$sesion = new cls_sesion();
if (isset($_REQUEST["idObra"]) == 1 && $sesion->getLogin() == false) {
    header("Location:../index.php?id=" . $_GET["id"] . "&idObra=" . $_GET["idObra"]);
} else {

    if ($sesion->getLogin() == false) {
        header("Location:../index.php");
    }
}
?>
```

Figura 16. Código para control de sesión en perfil de actividad.
Elaborado por: Araujo, D (2016).

El código mostrado anteriormente verifica si el usuario ha iniciado sesión de tal manera que si no lo ha hecho, está le redireccionará a la página de inicio de sesión, una vez que el usuario ingresa sus credenciales para acceder a la aplicación, se le enviará a la página de perfil de actividad a continuación se mostrará las líneas de código que realizan dicha funcionalidad.

Código de inicio de sesión

```
function onLogin() {
    var idObra = $("#obr").text();
    var act = $("#act").text();
    var correo = $("#correo").val();
    var pass = $("#pass").val();
    if (pass.trim() && validaCorreo(correo)) {
        $("#error").hide("slow");
        $("#formulario").removeClass("has-error");
        $("#login").append("<div class='loader'></div>")
        $.ajax({
            data: "op=logear&pass=" + pass + "&correo=" + correo + "&act=" + act + "&obr=" + idObra,
            url: "controladores/con_usuario.php",
            type: 'POST',
            beforeSend: function() {
            },
            success: function(a) {
                if (a != 0) {
                    location.href = a;
                } else {
                    $("#formulario").addClass("has-error");
                    $("#error").show("slow");
                    $("#login").html("Ingresar  ");
                }
            }
        });
    } else {
        $("#formulario").addClass("has-error");
        $("#error").show("slow");
    }
}
```

Figura 17.Código que muestra el inicio de sesión
Elaborado por: Araujo, D (2016).

En la figura anterior se muestra el código que se ejecuta al dar clic en iniciar sesión de tal manera que a través de *Ajax* se consulta al controlador la función de *logear* la cual responde como resultado la dirección de página a la cual se debe redireccionar al usuario siempre y cuando el mismo se autentifique con su correo electrónico y contraseña correcta. De esta manera si el usuario proviene desde el perfil de actividad se le enviará a dicha página terminando así el ciclo, caso contrario se le retornara a la página que le corresponde ver de acuerdo al perfil que maneja dicho usuario.

5.4 Configuración de aplicación en ambiente de pruebas

El servidor en el que se desplego la aplicación fue en un espacio otorgado por Microsoft Azure, el cual se localiza en Brasil, tiene como capacidad 10GB de almacenamiento para la aplicación y 0,20 GB para base de datos.

Para cargar una nueva versión del software al servidor se utilizó Visual Studio Online.

La clase Datos.php es la que contiene la conexión a la base de datos en donde se tuvo que realizar las siguientes configuraciones.

Configuración en clase Datos.php

```
$base = "pgi";

$servidor = "br-cdbr-azure-south-b.cloudapp.net";
$usuario = "b729cc0727f267";
$clave = "9f4361a8d560346";
$base = "planificacion";
```

Figura 18. Diagrama que muestra la configuración para la conexión a la base de datos en Azure. Elaborado por: Campaña, J (2016).

En cuanto a la configuración del servidor de base de datos se la realizo a través de MysqlWorkbench 6.3 el cual permitió agregar todos los datos necesarios para el despliegue correcto del sistema.

5.5 Pruebas

Las pruebas realizadas al sistema ayudaron a garantizar un producto de calidad. Para esta aplicación se optó realizar pruebas funcionales para darle al cliente la satisfacción de que todos los requerimientos fueron atendidos, pruebas de rendimiento para que el dueño del producto tenga una idea de cuán rápido la aplicación procesa los datos

ingresados y finalmente las pruebas de aceptación con el fin de medir las ventajas que justifican el uso de la aplicación.

5.5.1 Pruebas funcionales

El objetivo que se tuvo al realizar estas pruebas fue para garantizar que los requerimientos fueron entendidos claramente y atendidos de tal forma que el sistema esté completamente funcional.

En los anexos se puede encontrar detallas cada una de las pruebas que fueron realizadas, dichas pruebas corresponden a una prueba final de todo el sistema mas no a pruebas que se desarrollaban al finalizar cada uno de los incrementos del software.

A continuación, se muestra un resumen de las pruebas que fueron realizadas.

Tabla 3.
Resumen de pruebas funcionales

Código Prueba	**Satisfactoria**		**Observación**
	Si	No	
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Alinear columnas de reporte de acuerdo a los campos que se muestran

	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna

	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Adjuntar logo de la Inspectoría Sagrado Corazones de Jesús.
	X		Ninguna
	X		Ninguna
	X		Ninguna
	X		Ninguna

Nota: Tabla de resultados de pruebas funcionales.

Como se puede observar se tiene un 93% de satisfacción en cuando a la funcionalidad del software por parte del dueño del producto.

5.5.2 Pruebas de rendimiento

Se realizó las pruebas de rendimiento utilizando la aplicación *Jmeter* la cual permitió analizar el comportamiento del sistema.

Para el análisis de resultados en *Jmeter* se necesitó tener en claro lo siguiente:

Muestras: cantidad de hilos utilizados para la prueba.

Media: tiempo promedio para un conjunto de resultados en milisegundos.

Min: tiempo mínimo que demora un hilo en acceder a una página.

Max: tiempo máximo que demora un hilo en acceder a una página.

Rendimiento: rendimiento medido en los requerimiento por segundo/minuto / hora.

Kb/sec: rendimiento medido en Kbytes por segundo.

Media en bytes: tamaño medio de respuesta del servidor (en bytes).

El estudio de resultados se lo realizó en un intervalo de confianza del 95% equivalente a 1,96 para una distribución normal y la fórmula para el cálculo es la siguiente:

$$\frac{\text{Media} \pm \frac{1{,}96 * \text{Desv. Estándar}}{\sqrt{\#\text{Muestras}}}}{1000}$$

A continuación, se presentan los resultados obtenidos en las pruebas de rendimiento, las cuales se realizaron en dos maneras distintas.

Caso 1. Con 100 usuarios en un periodo de 7 segundos de subida en 1 bucle.

Caso 2. Con 200 usuarios en un periodo de 7 segundos de subida en 1 bucle.

Tabla 4.
Resultado prueba de rendimiento caso 1

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento (/sec)	Kb/sec	Media de Bytes
Login	100	2806	454	13346	2694,42	0,00%	5,4	18,53	3538
Administrador	100	30409	4894	103263	16175,93	2,00%	1	136,22	144427,3
Administrador Obra	100	9440	1369	180501	17947,52	1,00%	0,5	24,59	46654
Usuario	100	4373	530	21022	3385,89	1,00%	0,5	17,61	32895
Total	400	11757	454	180501	16510,6	1,00%	2,1	113,89	56878,6

Nota: Tabla de resultados de prueba de rendimiento con 100 usuarios.

Tabla 5.
Resultado prueba de rendimiento caso 2

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento (/sec)	Kb/sec	Media de Bytes
Login	200	7145	588	336907	23790,33	3,00%	0,6	2,01	3494,2
Administrador	200	93890	3212	306505	74055,93	14,50%	0,6	72	125650,2
Administrador Obra	200	43590	1340	5562248	392178,41	2,00%	0	1,62	46639,2
Usuario	200	9120	536	119878	11700,14	0,00%	0,6	17,91	33205
Total	800	38473	536	5562248	203182,63	4,88%	0,1	7,24	52271

Nota: Tabla de resultado de prueba de rendimiento con 200 usuarios.

Valores correspondientes al caso 1

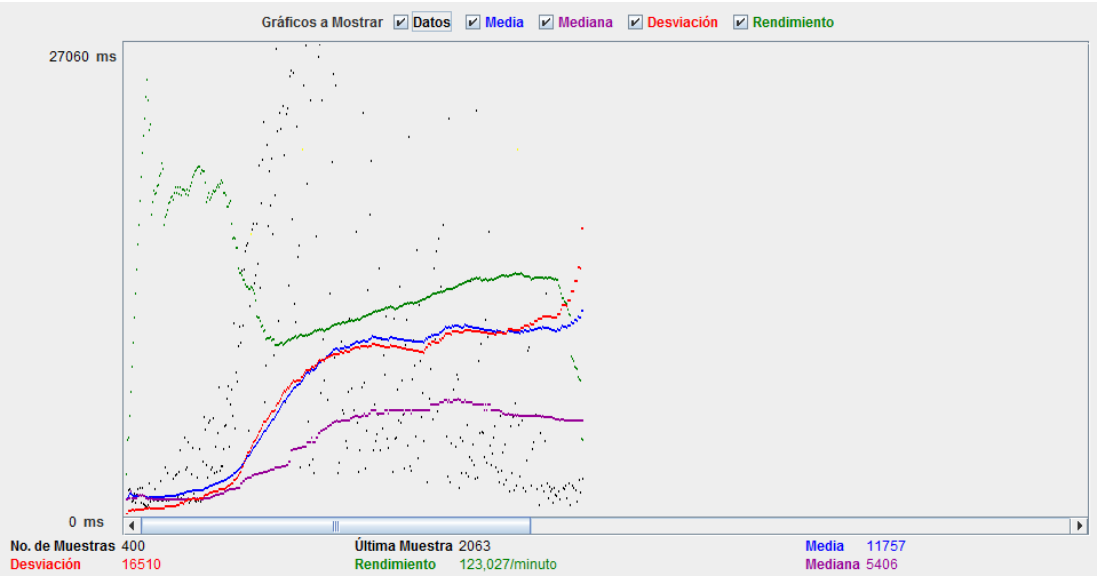


Figura 19. Gráfico que muestra los valores correspondientes al caso 1.
Elaborado por: Campaña, J (2016).

Valores correspondientes al caso 2

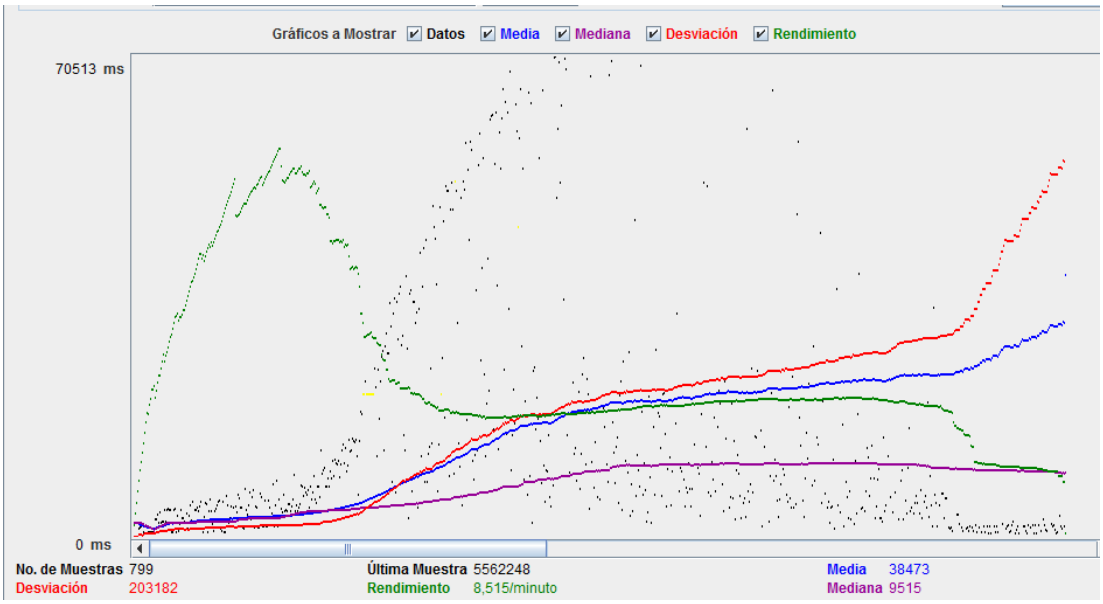


Figura 20. Gráfico que muestra los valores correspondientes al caso 2.
Elaborado por: Campaña, J (2016).

Para el caso 1 se tiene un tiempo de respuesta entre 10,13 y 13,38 segundos para 100 usuarios realizando 400 peticiones, mientras que para el caso 2 está entre 24,40 y 52,55 segundos para 200 usuarios realizando 800 peticiones.

Para esta aplicación que se encuentra en un servidor de pruebas en Microsoft Azure el tiempo de respuesta es bueno.

5.5.3 Pruebas de aceptación

Se realizó una prueba a veinte usuarios que usaran de manera permanente el software con el fin de evaluar la aceptabilidad del mismo, para la cual una vez que los usuarios se familiarizaron con el software se procedió a realizar una encuesta la misma que arrojó los siguientes resultados:

La aplicación facilito su planificación

Figura 21. Gráfico de resultados obtenidos a la pregunta la aplicación facilitó su planificación. Elaborado por: Campaña, J (2016).

En el gráfico anterior se puede apreciar que un 62 % de los encuestados respondió excelente, mientras que el 38% dijo muy bueno, lo que permite concluir que la aplicación es de gran ayuda para la planificación.

El aspecto visual de la aplicación

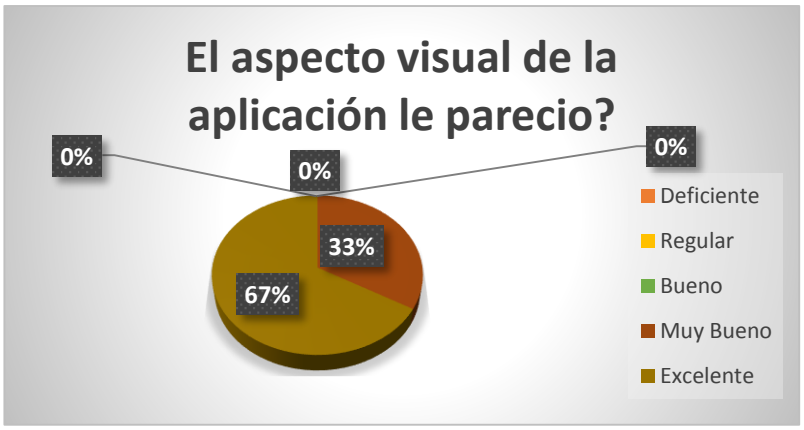


Figura 22. Gráfico de resultados obtenidos a la pregunta el aspecto visual de la aplicación.
Elaborado por: Campaña, J (2016).

En la ilustración anterior se observa que el 67% respondió excelente y el 33% mencionó muy bueno, entonces se puede concluir que la aplicación tiene un aspecto visual muy aceptable.

La usabilidad que presenta la aplicación

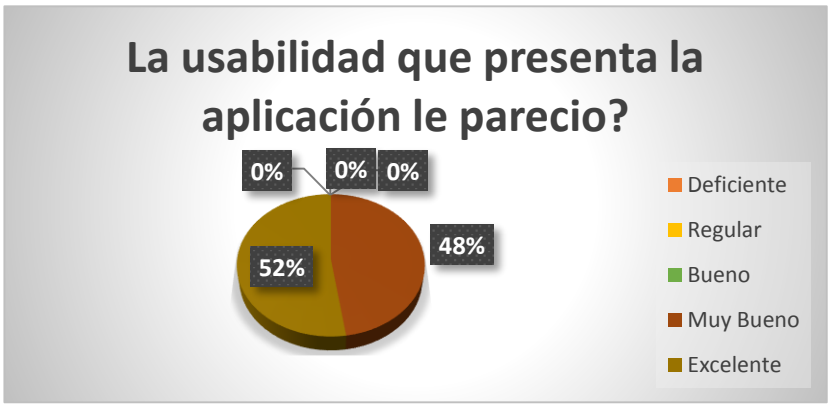


Figura 23. Gráfico de resultados obtenidos a la pregunta la usabilidad que presenta la aplicación
Elaborado por: Campaña, J (2016).

En la imagen anterior se visualiza que el 52% señalo excelente, mientras que el 48% mencionó muy bueno, lo cual indica que la aplicación está bastante usable.

La experiencia al utilizar la aplicación

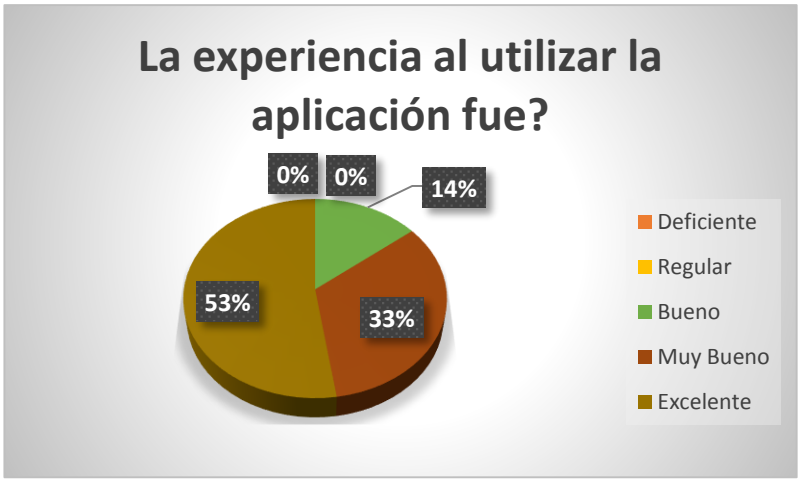


Figura 24. Gráfico de resultados obtenidos a la pregunta la experiencia al utilizar la aplicación
Elaborado por: Campaña, J (2016).

La figura anterior presenta que el 53% de encuestados señalo excelente, 33% muy bueno y un 14% bueno, lo que permite concluir que la aplicación es amigable para el usuario.

A continuación se presenta el gráfico global con los resultados de la encuesta realizada.

Resultados generales prueba de aceptación

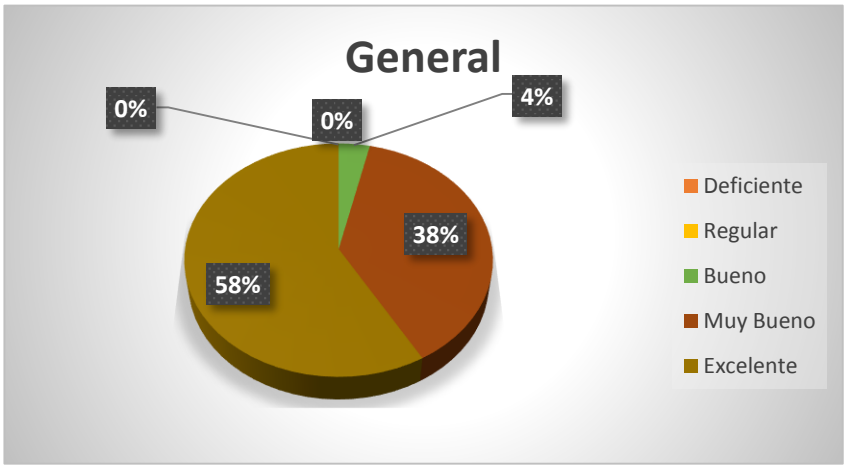


Figura 25. Gráfico de resultados generales prueba de aceptación.
Elaborado por: Campaña, J (2016).

En la ilustración anterior se observa que el 58% contesto excelente, el 38% muy bueno y solo un 4% contesto bueno, los resultados obtenidos al realizar la encuesta son favorables lo quiere decir es que la aplicación es aceptable para los usuarios.

5.6 Breve descripción de la aplicación

A continuación se describen las pantallas más relevantes de la aplicación:

Iniciar Sesión

Figura 26. Gráfico que muestra el Iniciar Sesión de la aplicación.
Elaborado por: Campaña, J (2016).

En el gráfico anterior se visualiza el Iniciar Sesión de la aplicación, los usuarios en esta instancia se autentifican con el correo electrónico y contraseña.

La aplicación presenta tres perfiles los cuales son Administrador, Administrador Obra y Usuario.

Perfil administrador

Figura 27. Ilustración que muestra las opciones del perfil administrador.
Elaborado por: Campaña, J (2016).

La ilustración anterior muestra una visualización general del perfil *Administrador*. En lado izquierdo se puede apreciar las opciones que presenta este perfil las cuales son: administración de usuarios, administración de sectores, administración de casas, administración de obras, planificación del sistema, administración de actividades.

Perfil administrador obra

Figura 28. Diagrama que muestra las opciones del perfil administrador obra.
Elaborado por: Campaña, J (2016).

En diagrama anterior se puede observar la pantalla del perfil Administrador Obra, las opciones que presenta este perfil son: administración de usuarios, administración de actividades, administración sector y reportería.

Perfil usuario

Figura 29. Ilustración que muestra las opciones del perfil Usuario.
Elaborado por: Campaña, J (2016).

La ilustración anterior permite contemplar la pantalla del perfil Usuario, este usuario puede crear actividad, gestionar sus actividades.

Opciones en común en todos los perfiles

Figura 30. Gráfico que muestra las opciones que tiene en común todos los perfiles.
Elaborado por: Campaña, J (2016).

El gráfico anterior destaca todas las opciones que tienen en común todos los perfiles, las cuales son administrar su información personal la cual está señalada con color

verde, gestionar notificaciones que se encuentra subrayada con color anaranjado y cerrar sesión la que está marcada con color amarrillo.

6 Conclusiones

Se investigó todo el conocimiento relacionado con la metodología, la estructura y los procesos de planificación que lleva a cabo la Inspectoría Sagrado Corazón de Jesús, lo cual fue clave para identificar y analizar las funcionalidades requeridas para la aplicación que se construyó.

Haciendo uso de los diferentes instrumentos que ofrece cada una de las técnicas aprendidas a lo largo de la carrera se logró analizar, diseñar e implementar una aplicación que satisface cada uno de los requerimientos tomados en cada inicio de los incrementos que se tuvieron que realizar a lo largo del desarrollo de la aplicación.

El uso de técnicas no establecidas en la metodología Scrum como es el prototipado y los diferentes diagramas que ofrece UML permitió especificar cada requerimiento que se fue presentando según el avance de la construcción del software.

Se implementó el patrón de arquitectura MVC con el fin de garantizar el fácil entendimiento del código por parte de los desarrolladores que conformaron el equipo y de los técnicos que a futuro realizaran mantenimientos a la aplicación, así como también permitió la reutilización de código disminuyendo el tiempo de construcción de la aplicación.

Debido a que al finalizar cada incremento se obtenía una parte tangible y funcional de la aplicación se optó por utilizar a Microsoft Azure como centralizador de todo el software, teniendo un despliegue continuo de cada una de las partes que conformaban la aplicación evidenciando la evolución que se iba llevando a cabo durante todo el proceso de construcción teniendo así al finalizar la aplicación en su versión final completamente funcional en un ambiente de pruebas.

7 Recomendaciones

Se recomienda el uso de la técnica del prototipo debido a que da una amplia visión del software al inicio del proyecto, garantizando desde el primer incremento la satisfacción del cliente.

Utilizar librerías que se encuentren arduamente documentadas con la finalidad de facilitar la búsqueda de información sobre las mismas, garantizando de esta manera el uso adecuado, así como también se optimiza el tiempo de investigación que se debe realizar.

El uso de artefactos que no estén definidos en la metodología agrega mayor valor a los entregables que se proponen en el proyecto.

Se recomienda realizar pruebas exhaustivas con el fin de proporcionar un software que no genere inconvenientes al momento de ejecución, entregando un producto de alta calidad.

Realizar capacitaciones a todos los usuarios de la aplicación de tal modo que se sientan favorecidos con el software, de esta manera alcanzar el menor grado posible de resistencia.

8 Referencias bibliográficas

Palacio, J. (2014). *Gestiónde proyectos Scrum Manager.* Scrum Manager.

Alaimo, D. M. (2013). *Proyectos ágiles con Scrum.* Buenos Aires: Kleer.

Arias , A., & Durango, A. (2016). *Ingeniería y Arquitectura del Software* (2 ed.). IT Campus Academy.

Borland. (2005). *Mitigating Risk with Effective Requirements Engineering.*

Burgwal, G., & Cuellar, C. J. (1999). *Planificación Estratégica y Operativa.* Quito: Abya Yala.

Cohn, M. (2009). *User Stories Applied for Agile Software Development.* Indiana: Addisson - Wesley.

Cohn, M. (20 de 10 de 2013). *Mountain Goat Software:Scrum*. Recuperado el 13 de 7 de 2016, de Mountain Goat Software: https://www.mountaingoatsoftware.com/agile/scrum

Cohn, M. (28 de 10 de 2013). *Mountain Goat Software:User Stories*. Recuperado el 12 de 7 de 2016, de Mountain Goat Software: https://www.mountaingoatsoftware.com/agile/user-stories

Fu, J., Hao, W., Yen, I.-L., & Bastani, F. B. (2008). *Model-Driven Prototyping Based Requirements.* Dallas, Texas.

Kniberg, H. (2007). *Scrum y XP desde las trincheras.* C4Media.

Labrada Martínez, E., & Salgado Ceballo, C. (2013). DISEÑO WEB ADAPTATIVO O RESPONSIVO. *Revista Digital Universitaria*, 3-9.

López, A. (2008). *Tesis profesional Sistema asistente para la generación de horarios de Cursos.* Recuperado el 13 de 7 de 2016, de Tesis profesional Sistema asistente para la generación de horarios de Cursos: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf

Molina, J. (2016). *Proyecto Educativo Pastoral Salesiano Inspectorial.* Quito: Salesianos Ecuador.

Perry, D. E., & Wolf, A. L. (10 de 1992). *Foundations for the Study of Software Architecture.* (V. 1. ACM Sigsoft - Software Engineering Notes, Ed.) Recuperado el 13 de 7 de 2016, de www.ics.uci.edu/~taylor/ICS221/papers/swa-sen.pdf

Raúl Herranz, N. M. (2011). *Scrum Distribuido.*

Robbins, S., & Coulter, M. (2005). *Administración* (8 ed.). Mexico: Pearson Educacion.

Rumbaught, J., Jacobson, I., & Booch, G. (2000). *El Lenguaje Unificado de Modelado. Manual de Referencia.* Madrid: Pearson.

Sommerville, I. (2011). *Ingenieria del Sofware* (9 ed.). Mexico: Pearson Edicación.

Suazion, I. (2009). *Manual de Planificación, Seguimiento y Evaluación de Resultados.* (A. Gerez, Trad.) New York: A.K.Office Supplies(NY).

Tuffley, A. (2005). *CIT3190 IT Project Course. Requirements Elicitation & Management.*

Wei, Q., Danwei , G., Yaohong, X., Jingtao, F., Cheng, H., & Zhengang. (2014). *Research on Software Development Process Conjunction of Scrum.*

9 Anexos

Anexo 1. Glosario de términos

ISSCJ.-Inspectoría Salesiana Sagrado Corazón de Jesús.

MVC.-Modelo Vista Controlador.

PBS.-Product Breakdown Structure traduciendo al español Estructura de descomposición del producto.

PEPSI.-Proyecto Educativo Pastoral Salesiano Inspectorial.

PGI.-Plan General de Intervención.

POA.-Plan Operativo Anual.

UML.-Lenguaje de Modelado Unificado.

RWD.- Del inglés Responsive Web Design.

Anexo 2. Prototipo

A continuación se presenta el prototipo que se realizó en el primer sprint, el cual sirvió para aclarar los requerimientos.

Iniciar sesión del prototipo

Figura 31. Grafica que muestra el login del prototipo.
Elaborado por: Campaña, J (2016).

Administración del prototipo

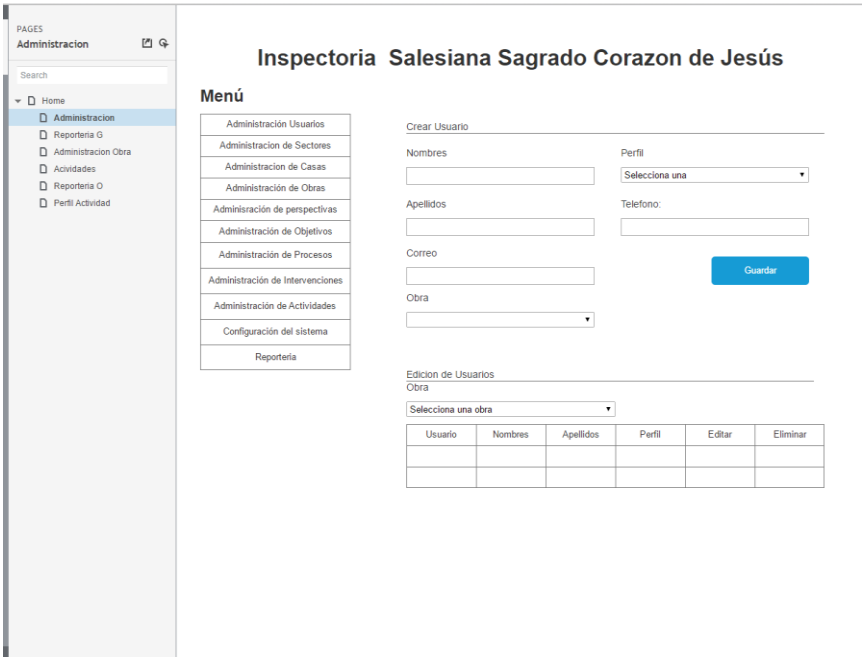


Figura 32. Grafica que muestra las opciones del perfil Administración del prototipo.
Elaborado por: Campaña, J (2016).

Reporteria del prototipo

Figura 33. Grafica que muestra las opciones de Reportería del prototipo.
Elaborado por: Campaña, J (2016).

Administración Obra del prototipo

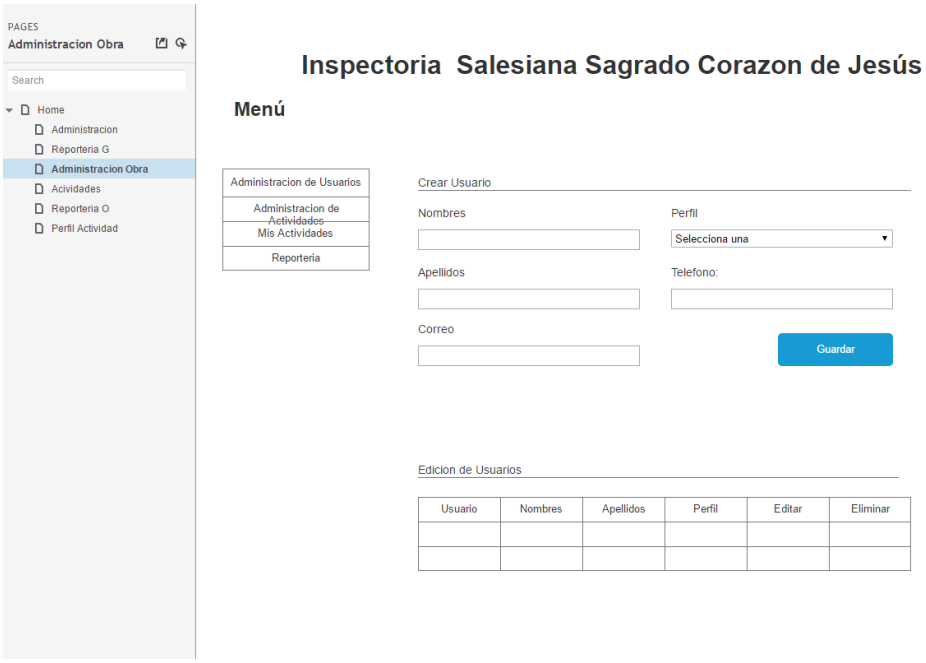


Figura 34. Grafica que muestra las opciones del perfil Administración Obra del prototipo.
Elaborado por: Campaña, J (2016).

Actividades del prototipo

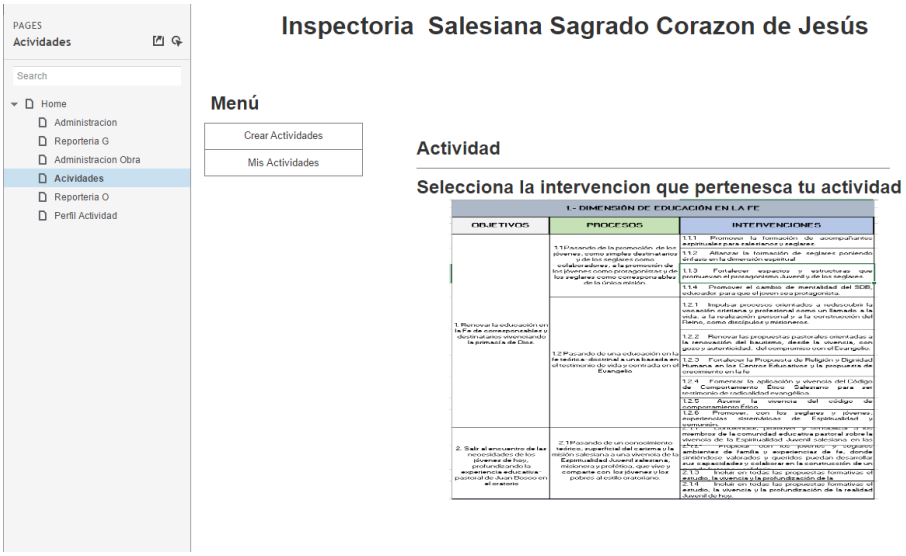


Figura 35. Grafica que muestra cómo se visualizan las actividades del usuario del prototipo. Elaborado por: Campaña, J (2016).

Perfil Actividad del prototipo

Figura 36. Grafica que muestra cómo se visualiza el detalle de la actividad del prototipo. Elaborado por: Campaña, J (2016).

Anexo 3. Diccionario de datos

En la siguiente tabla se detalla el tipo que se definió a cada columna utilizada en la base de datos diseñada para la aplicación.

Tabla 6.
Diccionario de datos de base de datos

Nombre	**Tabla**	**Tipo de dato**	**Tamaño**
descripcion_actividad	tb_actividad	varchar(700)	700
estado_actividad	tb_actividad	Smallint	
id_actividad	tb_actividad	Int	
id_intervencion	tb_actividad	Int	
id_usuario	tb_actividad	Int	
nombre_actividad	tb_actividad	varchar(500)	500
director_casa	tb_casa	national varchar(200)	200
estado_casa	tb_casa	Int	
id_casa	tb_casa	Int	
nombre_casa	tb_casa	national varchar(100)	100
telefono_casa	tb_casa	national varchar(10)	10
descripcion_ciudad	tb_ciudad	varchar(100)	100
estado_ciudad	tb_ciudad	smallint	
id_ciudad	tb_ciudad	Int	
id_provincia	tb_ciudad	Int	
nombre_ciudad	tb_ciudad	varchar(100)	100
descripcion_configuracion	tb_configuracion	varchar(400)	400
estado_configuracion	tb_configuracion	smallint	
id_configuracion	tb_configuracion	Int	

nombre_configuracion	tb_configuracion	varchar(300)	300
arch_detalle_actividad	tb_detalle_actividad	varchar(200)	200
fin_detalle_actividad	tb_detalle_actividad	date	
id_actividad	tb_detalle_actividad	Int	
id_detalle_actividad	tb_detalle_actividad	Int	
id_estado_actividad	tb_detalle_actividad	Int	
id_estado_aprobacion	tb_detalle_actividad	Int	
inicio_detalle_actividad	tb_detalle_actividad	date	
id_actividad_financiamiento	tb_detalle_actividad_financiamiento	Int	
id_detalle_actividad	tb_detalle_actividad_financiamiento	Int	
id_tipo_financiamiento	tb_detalle_actividad_financiamiento	Int	
descripcion_dimencion	tb_dimencion	varchar(300)	300
estado_dimencion	tb_dimencion	smallint	
id_dimencion	tb_dimencion	Int	
nombre_dimencion	tb_dimencion	varchar(300)	300
descripcion_estado_actividad	tb_estado_actividad	varchar(400)	400
estado_estado_actividad	tb_estado_actividad	smallint	
id_estado_actividad	tb_estado_actividad	Int	
nombre_estado_actividad	tb_estado_actividad	varchar(300)	300
detalle_estado_aprobacion	tb_estado_aprobacion	varchar(200)	200
estado_estado_aprobacion	tb_estado_aprobacion	smallint	
id_estado_aprobacion	tb_estado_aprobacion	Int	
nombre_estado_aprobacion	tb_estado_aprobacion	varchar(100)	100
descripcion_estado_notificacion	tb_estado_notificacion	varchar(150)	150
estado_estado_notificacion	tb_estado_notificacion	smallint	
id_estado_notificacion	tb_estado_notificacion	Int	

nombre_estado_notificacion	tb_estado_notificacion	varchar(100)	100
estado_intervencion_sector	tb_intervencion_sector	smallint	
id_intervencion	tb_intervencion_sector	Int	
id_intervencion_sector	tb_intervencion_sector	Int	
id_sector	tb_intervencion_sector	Int	
descripcion_intervencion	tb_intevencion	varchar(500)	500
estado_intervencion	tb_intevencion	smallint	
id_intervencion	tb_intevencion	Int	
id_proceso	tb_intevencion	Int	
nombre_intervencion	tb_intevencion	varchar(500)	500
fecha_notificacion	tb_notificacion	datetime	
id_creador_notificacion	tb_notificacion	Int	
id_detalle_actividad	tb_notificacion	Int	
id_estado_notificacion	tb_notificacion	Int	
id_notificacion	tb_notificacion	Int	
id_notificado_notificacion	tb_notificacion	Int	
id_tipo_notificacion	tb_notificacion	Int	
descripcion_objetivo	tb_objetivo	varchar(500)	500
estado_objetivo	tb_objetivo	smallint	
id_dimencion	tb_objetivo	Int	
id_objetivo	tb_objetivo	Int	
nombre_objetivo	tb_objetivo	varchar(500)	500
cordinadores_obra	tb_obra	Int	
direccion_obra	tb_obra	national varchar(500)	500
estado_obra	tb_obra	smallint	
id_casa	tb_obra	Int	
id_ciudad	tb_obra	Int	
id_obra	tb_obra	Int	

id_sector	tb_obra	Int	
latitud_obra	tb_obra	national varchar(100)	100
longitud_obra	tb_obra	varchar(100)	100
nombre_obra	tb_obra	varchar(500)	500
participantes_obra	tb_obra	Int	
telefono_obra	tb_obra	varchar(200)	200
id_obra	tb_obra_usuario	Int	
id_obra_usuario	tb_obra_usuario	Int	
id_perfil	tb_obra_usuario	Int	
id_usuario	tb_obra_usuario	Int	
descripcion_observacion	tb_observaciones	varchar(700)	700
estado_observacion	tb_observaciones	Int	
id_detalle_actividad	tb_observaciones	Int	
id_observacion	tb_observaciones	Int	
id_usuario_observacion	tb_observaciones	Int	
titulo_observacion	tb_observaciones	varchar(300)	300
descripcion_perfil	tb_perfil	varchar(500)	500
estado_perfil	tb_perfil	smallint	
id_perfil	tb_perfil	Int	
nombre_perfil	tb_perfil	varchar(200)	200
descripcion_proceso	tb_procesos	varchar(500)	500
estado_proceso	tb_procesos	smallint	
id_objetivo	tb_procesos	Int	
id_proceso	tb_procesos	Int	
nombre_proceso	tb_procesos	national varchar(500)	500
descripcion_provincia	tb_provincia	varchar(100)	100
estado_provincia	tb_provincia	smallint	

id_provincia	tb_provincia	Int	
nombre_provincia	tb_provincia	varchar(100)	100
descripcion_sector	tb_sector	national varchar(500)	500
estado_sector	tb_sector	smallint	
id_responsable_sector	tb_sector	Int	
id_sector	tb_sector	Int	
nombre_sector	tb_sector	national varchar(200)	200
poa_sector	tb_sector	varchar(200)	200
descripcion_tipo_financiamiento	tb_tipo_financiamiento	varchar(500)	500
estado_tipo_financiamiento	tb_tipo_financiamiento	smallint	
id_tipo_financiamiento	tb_tipo_financiamiento	Int	
nombre_tipo_financiamiento	tb_tipo_financiamiento	national varchar(200)	200
descripcion_tipo_notificacion	tb_tipo_notificacion	varchar(200)	200
estado_tipo_notificacion	tb_tipo_notificacion	smallint	
id_tipo_notificacion	tb_tipo_notificacion	Int	
nombre_tipo_notificacion	tb_tipo_notificacion	varchar(200)	200
apellido_usuario	tb_usuario	varchar(500)	500
contrasena_usuario	tb_usuario	varchar(200)	200
correo_usuario	tb_usuario	varchar(200)	200
estado_usuario	tb_usuario	smallint	
id_usuario	tb_usuario	Int	
nombre_usuario	tb_usuario	varchar(500)	500
telefono_usuario	tb_usuario	varchar(10)	10

Nota. Diccionario de datos.

Anexo 4. Plan de iteración

En la siguiente figura se muestra el plan de iteración que se propuso al iniciar el proyecto. Este ayudaba a dar seguimiento del cumplimiento de cada uno de los sprints propuestos.

Plan de iteración

PRODUCT BACKLOCK	SPRINT	RESPONSABLE	AVANCE
Administración	Login	Diego Araujo	100%
	Administración de Usuario	Diego Araujo	100%
	Administración de Sectores	Diego Araujo	100%
	Administración de Casas	Diego Araujo	100%
	Administración de Obras	Diego Araujo	100%
	Administración de perspectivas	Josue Campaña	100%
	Administración de objetivos	Josue Campaña	100%
	Administración de procesos	Josue Campaña	100%
	Administración de intervenciones	Josue Campaña	100%
	Administración de actividades	Diego Araujo	100%
	Administración de sistemas	Josue Campaña	100%
Administración de obras	Administración de Usuario	Josue Campaña	100%
	Administración de actividades	Josue Campaña	100%
	Mis Actividades	Diego Araujo	100%
	Admin Encargado Sector	Diego Araujo	100%
Perfil de Usuario	Administración de información personal	Diego Araujo	100%
Notificaciones	Envios de Correos	Diego Araujo	100%
	Notificaciones de sesión	Diego Araujo	100%
	Acceso desde correo a sesión	Diego Araujo	100%
Reporteria	General	Josue Campaña	100%
	Obras	Josue Campaña	100%
	Dimensión	Josue Campaña	100%

Figura 37. Matriz de plan de iteración.
Elaborado por: Araujo, D (2016).

Anexo 5. Matriz de revisión de sprint

A continuación se muestra algunas de las matrices que ayudaban a recoger las observaciones que se iban generando en la reunión de finalización de sprint.

Matriz de revisión de sprint del 26_05_2016

	PRODUCT BACKLOCK	SPRINT	PROBLEMA	MEJORA	RESPONSABLE
1	Administración	Adm Actividades	cambiar perspetivas por dimensiones	implementación de cambio	Josue Campaña
2	Administración	Adm sector	implentar en intervención la clasificacion del sector	implementación de cambio	Josue Campaña
3	Administración	Adm usuarios	al crear el usuario solo debe aparecer las del sector	implementación de cambio	Diego Araujo
4	Administración	Adm usuarios	un usuario puede pertenecer a varios sectores	implementación de cambio	Diego Araujo
5	Administración	Adm Actividades	crear actividad tiene que aparecer las intervenciones y mostrarte a que sector	implementación de cambio	Diego Araujo y Josue Campaña
6	Administración	Adm Obras	las obras hay que clasificar por ciudad	clasificar las obras por ciudad	Diego Araujo
7	Administración	Adm Actividades	mostrar todas las intervenciones que estan asociadas ese sector que pertenece a esa obra	implementación de cambio	Diego Araujo y Josue Campaña
8	Administración	Adm Actividades	Tiempo limite para usbir actividad desconocido	Hay 7 dias para que pueda subir la evidencia	Josue Campaña
9	Administración	Reporteria	reporte por sector y tiempo	implementación de cambio	Josue Campaña
10	Reporteria	Reporteria	reporte por dimensión	implementación de cambio	Josue Campaña

Figura 38. Matriz de revisión reunión 26_05_2016
Elaborado por: Araujo, D (2016).

Matriz de revisión de sprint del 02_06_2016

	PRODUCT BACKLOCK	SPRINT	PROBLEMA	MEJORA	RESPONSABLE
1	Administración	Planificacion del Sistema	ordenar por seleccionar,editar, eliminar	Ordenar	Josue Campaña
2	Administración	Presentación	definir colores de aplicación	Realizara propuestas con colores que usa el libro Pgi	Diego Araujo
3	Administración	Administración de Sectores	El administrador general no puede subir el archivo del poa	Dar esta funcionalidad al Administrador General	Josue Campaña
4	Administración	Administración de Sectores	Los usuarios deben estar filtrados por la casa que pertenecen	Generar filtro de usuarios por casa seleccionada	Diego Araujo
5	Administración	Administración de planificación	Los nombres principales de las tablas aparecen como objetivos	Cambiar el nombre de las tablas principales por el nombre que perteneces	Josue Campaña
6	Administración	Administración de planificación	Cuando se genera error se queda cargando	Corregir Error	Josue Campaña

Figura 39. Matriz de revisión reunión 02_06_2016.
Elaborado por: Araujo, D (2016).

Anexo 6. Registro de pruebas funcionales

Tabla 7.
Prueba funcional 1

Prueba Funcional	
ID Prueba	P1
Módulo	Identificación
Precondiciones:	
• Usuario: Correcto e incorrecto • Contraseña: Correcta e incorrecta	
Descripción:	
• Comprobación de validación en autentificación de usuario y contraseña	
Resultado Obtenido:	
• Validación y autentificación correcta	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de identificación parte 1.

Tabla 8.
Prueba funcional 2

Prueba Funcional	
ID Prueba	P2
Módulo	Identificación
Precondiciones:	
• Validar redirección de página web de acuerdo a perfil de usuario	
Descripción:	
• Comprobar la redirección de acuerdo a cada uno de los perfiles que puede tener un usuario.	
Resultado Obtenido:	
• La redirección fue correcta de acuerdo a cada perfil de usuario	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de identificación parte 2.

Tabla 9.
Prueba funcional 3

Prueba Funcional	
ID Prueba	P3
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de usuario	
Descripción:	
• Crear un usuario • Modificar usuario creado anteriormente • Desactivar usuario creado anteriormente	
Resultado Obtenido:	
• Usuario creado satisfactoriamente • Usuario modificado satisfactoriamente • Usuario desactivado satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 1.

Tabla 10.
Prueba funcional 4

Prueba Funcional	
ID Prueba	P4
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de sector	
Descripción:	
• Crear un sector • Modificar sector creado anteriormente • Desactivar sector creado anteriormente	
Resultado Obtenido:	
• Sector creado satisfactoriamente • Sector modificado satisfactoriamente • Sector desactivado satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 2.

Tabla 11.
Prueba funcional 5

Prueba Funcional	
ID Prueba	P5
Módulo	Administración
Precondiciones:	
• Generación de reporte de actividades de un sector específico.	
Descripción:	
• Generar reporte desde una fecha hasta una fecha donde existan actividades.	
Resultado Obtenido:	
• Reporte generado satisfactoriamente con observaciones	
Estado de Prueba:	Satisfactoria
Observaciones:	Alinear columnas de reporte de acurdo a los campos que se muestran
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 3.

Tabla 12.
Prueba funcional 6

Prueba Funcional	
ID Prueba	P6
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de una casa Salesiana.	
Descripción:	
• Creación de una nueva casa Salesiana. • Modificar una casa Salesiana ya existente. • Desactivar una casa Salesiana.	
Resultado Obtenido:	
• Casa creada satisfactoriamente • Casa modificada satisfactoriamente • Casa desactivada satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 4.

Tabla 13.
Prueba funcional 7

Prueba Funcional	
ID Prueba	P7
Módulo	Administración
Precondiciones:	
• Visualización de casas existentes	
Descripción:	
• Realización de carga de casas guardadas en base de datos	
Resultado Obtenido:	
• Casas mostradas satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 5.

Tabla 14.
Prueba funcional 8

Prueba Funcional	
ID Prueba	P8
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de una Obra Salesiana.	
Descripción:	
• Creación de una nueva Obra Salesiana. • Modificar una Obra Salesiana ya existente. • Desactivar una Obra Salesiana.	
Resultado Obtenido:	
• Obra creada satisfactoriamente • Obra modificada satisfactoriamente • Obra desactivada satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 6.

Tabla 15.
Prueba funcional 9

Prueba Funcional	
ID Prueba	P9
Módulo	Administración
Precondiciones:	
• Visualización de Obras existentes	
Descripción:	
• Realización de carga de Obras guardadas en base de datos	
Resultado Obtenido:	
• Obras mostradas satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 7.

Tabla 16.
Prueba funcional 10

Prueba Funcional	
ID Prueba	P10
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de una Dimensión de planificación.	
Descripción:	
• Creación de una nueva Dimensión de planificación. • Modificar Dimensión de planificación ya existente. • Desactivar Dimensión de planificación.	
Resultado Obtenido:	
• Dimensión de planificación creada satisfactoriamente • Dimensión de planificación modificada satisfactoriamente • Dimensión de planificación desactivada satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 8.

Tabla 17.
Prueba funcional 11

Prueba Funcional	
ID Prueba	P11
Módulo	Administración
Precondiciones:	
• Visualización de dimensiones de planificación existentes	
Descripción:	
• Realización de carga de dimensiones de planificación guardadas en base de datos	
Resultado Obtenido:	
• Obras mostradas satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 9.

Tabla 18.
Prueba funcional 12

Prueba Funcional	
ID Prueba	P12
Módulo	Administración
Precondiciones:	
• Activación de dimensión de planificación desactivada.	
Descripción:	
• Activar una dimensión de planificación que fue desactivada previamente.	
Resultado Obtenido:	
• Dimensión de planificación activada nuevamente satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 10.

Tabla 19.
Prueba funcional 13

Prueba Funcional	
ID Prueba	P13
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de un objetivo de planificación.	
Descripción:	
• Creación de un nuevo objetivo de planificación. • Modificar objetivo de planificación ya existente. • Desactivar objetivo de planificación.	
Resultado Obtenido:	
• Objetivo de planificación creado satisfactoriamente • Objetivo de planificación modificado satisfactoriamente • Objetivo de planificación desactivado satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 11.

Tabla 20.
Prueba funcional 14

Prueba Funcional	
ID Prueba	P14
Módulo	Administración
Precondiciones:	
• Visualización de objetivos de planificación existentes.	
Descripción:	
• Realización de carga de objetivos de planificación guardadas en base de datos	
Resultado Obtenido:	
• Objetivos mostrados satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 12.

Tabla 21.
Prueba funcional 15

Prueba Funcional	
ID Prueba	P15
Módulo	Administración
Precondiciones:	
• Activación de objetivo de planificación desactivada.	
Descripción:	
• Activar un objetivo de planificación que fue desactivado previamente.	
Resultado Obtenido:	
• Objetivo de planificación activado nuevamente satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 13.

Tabla 22.
Prueba funcional 16

Prueba Funcional	
ID Prueba	P16
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de un proceso de planificación.	
Descripción:	
• Creación de un nuevo proceso de planificación. • Modificar proceso de planificación ya existente. • Desactivar proceso de planificación.	
Resultado Obtenido:	
• Proceso de planificación creado satisfactoriamente • Proceso de planificación modificado satisfactoriamente • Proceso de planificación desactivado satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 14.

Tabla 23.
Prueba funcional 17

Prueba Funcional	
ID Prueba	P17
Módulo	Administración
Precondiciones:	
• Visualización de procesos de planificación existentes.	
Descripción:	
• Realización de carga de procesos de planificación guardadas en base de datos	
Resultado Obtenido:	
• Procesos mostrados satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 15.

Tabla 24.
Prueba funcional 18

Prueba Funcional	
ID Prueba	P18
Módulo	Administración
Precondiciones:	
• Activación de un proceso de planificación desactivada.	
Descripción:	
• Activar un proceso de planificación que fue desactivado previamente.	
Resultado Obtenido:	
• Proceso de planificación activado nuevamente satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 16.

Tabla 25.
Prueba funcional 19

Prueba Funcional	
ID Prueba	P19
Módulo	Administración
Precondiciones:	
• Creación, modificación y desactivación de una intervención de planificación.	
Descripción:	
• Creación de una nueva intervención de planificación. • Modificar intervención de planificación ya existente. • Desactivar intervención de planificación.	
Resultado Obtenido:	
• Intervención de planificación creada satisfactoriamente • Intervención de planificación modificada satisfactoriamente • Intervención de planificación desactivada satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 17.

Tabla 26.
Prueba funcional 20

Prueba Funcional	
ID Prueba	P20
Módulo	Administración
Precondiciones:	
• Visualización de intervenciones de planificación existentes.	
Descripción:	
• Realización de carga de intervenciones de planificación guardadas en base de datos	
Resultado Obtenido:	
• Intervenciones mostrados satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 18.

Tabla 27.
Prueba funcional 21

Prueba Funcional	
ID Prueba	P21
Módulo	Administración
Precondiciones:	
• Activación de un Intervención de planificación desactivada.	
Descripción:	
• Activar una intervención de planificación que fue desactivado previamente.	
Resultado Obtenido:	
• Intervención de planificación activada nuevamente satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 19.

Tabla 28.
Prueba funcional 22

Prueba Funcional	
ID Prueba	P22
Módulo	Administración
Precondiciones:	
• Creación, modificación de una actividad.	
Descripción:	
• Creación de una actividad. • Modificar una actividad.	
Resultado Obtenido:	
• Actividad creada satisfactoriamente. • Actividad modificada satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 20.

Tabla 29.
Prueba funcional 23

Prueba Funcional	
ID Prueba	P23
Módulo	Administración
Precondiciones:	
• Modificación de configuración del sistema.	
Descripción:	
• Cambiar la configuración de la aplicación.	
Resultado Obtenido:	
• Configuración cambiada satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 21.

Tabla 30.
Prueba funcional 24

Prueba Funcional	
ID Prueba	P24
Módulo	Administración
Precondiciones:	
• Modificación de información personal de usuario.	
Descripción:	
• Cambiar nombre de usuario. • Cambio de Contraseña • Cambio de fotografía. • Cambio de Correo Electrónico	
Resultado Obtenido:	
• Nombre de usuario cambiado satisfactoriamente. • Cambio de contraseña cambiada. satisfactoriamente. • Fotografía cambiada satisfactoriamente. • Correo electrónico cambiado satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 22.

Tabla 31.
Prueba funcional 25

Prueba Funcional	
ID Prueba	P25
Módulo	Administración
Precondiciones:	
• Incremento de notificaciones al crear actividades.	
Descripción:	
• Creación de actividades para incrementar una notificación en el usuario de destino.	
Resultado Obtenido:	
• Notificación incrementada a usuario.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 23.

Tabla 32.
Prueba funcional 26

Prueba Funcional	
ID Prueba	P26
Módulo	Administración
Precondiciones:	
• Envió de correos de notificación al crear actividad.	
Descripción:	
• Crear actividades con el fin de que el sistema envié una notificación al dueño de la actividad.	
Resultado Obtenido:	
• Correos enviados satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Adjuntar logo de la Inspectoría Sagrado Corazones de Jesús.
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 24.

Tabla 33.
Prueba funcional 27

Prueba Funcional	
ID Prueba	P27
Módulo	Administración
Precondiciones:	
• Entrar a la actividad desde correo electrónico.	
Descripción:	
• Ingresar a la actividad generada desde el link desde el correo electrónico generado por la aplicación.	
Resultado Obtenido:	
• Se pudo acceder desde el correo electrónico a la actividad generada.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 25.

Tabla 34.
Prueba funcional 28

Prueba Funcional	
ID Prueba	P28
Módulo	Administración
Precondiciones:	
• Verificar observaciones nuevas generadas	
Descripción:	
• Ingresar nuevas observaciones a una actividad.	
Resultado Obtenido:	
• Se pudo generar nuevas observaciones.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 26.

Tabla 35.
Prueba funcional 29

Prueba Funcional	
ID Prueba	P29
Módulo	Administración
Precondiciones:	
• Verificar gráficos coherentes en reportaría.	
Descripción:	
• Ingresar nuevas actividades. • Cumplir algunas actividades. • Verificar reportes generales, por sector, por dimensión, por obras.	
Resultado Obtenido:	
• Se pudo verificar cada uno de los reportes generados por el sistema satisfactoriamente	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 27.

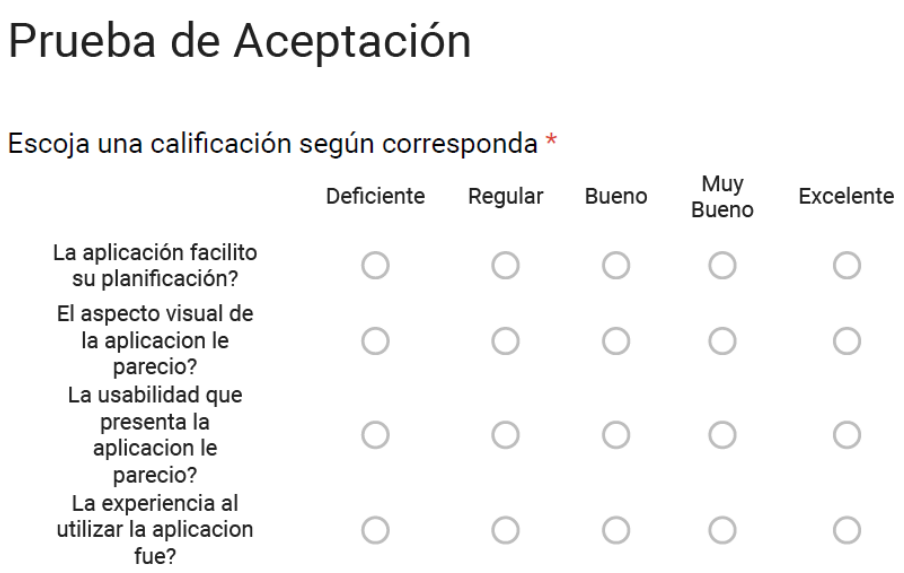
Tabla 36.
Prueba funcional 30

Prueba Funcional	
ID Prueba	P30
Módulo	Administración
Precondiciones:	
• Verificar datos de tablas coherentes en reportaría.	
Descripción:	
• Ingresar nuevas actividades. • Cumplir algunas actividades. • Verificar reportes generales, por sector, por dimensión, por obras.	
Resultado Obtenido:	
• Se pudo verificar cada uno de los reportes en tabla generados por el sistema satisfactoriamente.	
Estado de Prueba:	Satisfactoria
Observaciones:	Ninguna
Responsable:	Miriam Morales (Dueño del producto)

Nota: Tabla de prueba funcional al módulo de administración parte 28.

Anexo 7. Encuesta de prueba de aceptación

Encuesta para la Prueba de Aceptación



Prueba de Aceptación

Escoja una calificación según corresponda *

	Deficiente	Regular	Bueno	Muy Bueno	Excelente
La aplicación facilito su planificación?	○	○	○	○	○
El aspecto visual de la aplicacion le parecio?	○	○	○	○	○
La usabilidad que presenta la aplicacion le parecio?	○	○	○	○	○
La experiencia al utilizar la aplicacion fue?	○	○	○	○	○

Figura 40. Encuesta para la prueba de aceptación.
Elaborado por: Campaña, J (2016).

Anexo 8. Grabaciones de historias de usuario

En la carpeta historias de usuarios se encuentran todas las grabaciones que fueron tomadas en cada una de las reuniones con el dueño del producto.

Anexo 9. Fotografías de capacitación a usuarios

Capacitación Baños 22 jun 2016

Figura 41. Capacitación de software a usuarios.
Elaborado por: Araujo, D (2016).

Anexo 10. Carta de aceptación de software

Carta de aceptación del Software

SOCIEDAD SALESIANA EN EL ECUADOR

SALESIANOS DON BOSCO

Quito, 09 de Agosto del 2016

Ing. Patsy Prieto
Directora de carrera de Ingeniería en Sistemas
Universidad Politécnica Salesiana

Por medio de la presente comunico que el proyecto de titulación con el tema "Desarrollo de una aplicación Web para la gestión del plan general de intervención de la Inspectoría Sagrado Corazón de Jesús" que se llevó a cabo por los estudiantes Diego Andrés Araujo Vanegas con cédula de ciudadanía N° 1721480067 y Josué Daniel Campaña Ramírez con cédula de ciudadanía N° 1721411906, cumple con todas las funcionalidades y requerimientos que fueron solicitados por la Inspectoría "Sagrado Corazón de Jesús".

En Don Bosco y María Auxiliadora,

Pastoral Juvenil Salesiana

Padre Wladimir Acosta, sdb

Delegado Inspectorial
Pastoral Juvenil Salesiana

Madrid E12-68 y Andalucía **PBX:** 02 2566484 – 02 2566492 **Fax:** 02 2541436 **Ap.** 17.01.2303
www.salesianos.org.ec / inspector@salesianos.org.ec / vicario@salesianos.org.ec / economo@salesianos.org.ec / secretario@salesianos.org.ec

Figura 42. Carta de aceptación del software.
Elaborado por: Acosta, W (2016).