

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **DESARROLLO DE UN SISTEMA DE ADMINISTRACIÓN DE LAS ACTIVIDADES ECLESIASTICAS DE LA IGLESIA CATÓLICA PARROQUIAL SANTIAGO APÓSTOL USANDO HERRAMIENTAS WEB**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN  
ANÁLISIS DE SISTEMAS INFORMÁTICOS.**

**DIANA CRISTINA PILA VALDIVIEZO**

[chivi\\_diany@hotmail.com](mailto:chivi_diany@hotmail.com)

**DIRECTOR: ING. DANIEL MANANGÓN**

[daniel.manangon@epn.edu.ec](mailto:daniel.manangon@epn.edu.ec)

**Quito, Noviembre 2013**

## DECLARACIÓN

Yo, Diana Cristina Pila Valdiviezo declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Diana Cristina Pila Valdiviezo**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Diana Cristina Pila Valdiviezo, bajo mi supervisión.

---

**Ing. Daniel Manangón**

**DIRECTOR DE PROYECTO**

## AGRADECIMIENTO

A Dios, por cada día de vida que me regala en este mundo, por protegerme y cuidarme siempre y por la bendición de cumplir este objetivo tan anhelado.

A mis padres Zoila y José, que con esfuerzo y dedicación me han apoyado en el camino de la vida. Por enseñarme a valorar las pequeñas cosas. Con amor supieron guiarme pero a la vez dejarme tomar mis propias decisiones, y aunque muchas veces me equivoqué y tropecé, me ayudaron a levantarme y a seguir adelante. Gracias papis los adoro.

A mis hermanitas, compañeras de vida y de travesuras. Me siento afortunada y bendecida por tenerlas en mi vida y por estar a mi lado apoyándome.

A mi abuelita querida, por facilitarme un rinconcito en el que puedo refugiarme y por preocuparse por mí.

A mis amigos queridos, gracias por su amistad y consejos. Pero por sobre todo por apoyarme y quererme tal y como soy. Son un tesoro para mí y siempre los llevaré en mi corazón. Los quiero mucho. Y en especial, a mi mejor amigo y confidente Paúl, con quien he compartido los mejores y peores momentos de mi vida. Me motiva a ser mejor cada día y a darlo todo sin pedir nada a cambio. Gracias por tu apoyo incondicional.

A mi gran amigo D. por ser mi compañero inseparable y por hacerme sonreír.

A mi director de tesis, Ing. Daniel Manangón por las enseñanzas, tiempo y apoyo brindados durante mi vida estudiantil y en el desarrollo de mi proyecto de titulación.

A todos aquellos que creyeron en mí y que de alguna manera me inspiraron a continuar y a luchar por mis sueños. Este es uno de ellos.

*Diana Cristina*

## DEDICATORIA

A mi madre querida, consejera sabia y mujer excepcional. Eres mi ejemplo de lucha, amor y superación. No lo hubiera logrado sin ti. Te amo mamita hermosa.

A mis hermanas Erika y Yadira, son mi alegría y mi inspiración. Llenan mi vida de felicidad y ternura. Las quiero mucho.

*Diana Cristina*

## CONTENIDO

### ÍNDICE GENERAL

<b>CONTENIDO .....</b>	<b>VI</b>
<b>RESUMEN .....</b>	<b>X</b>
<b>PRESENTACIÓN .....</b>	<b>XI</b>
<b>CAPÍTULO 1. ASPECTOS GENERALES .....</b>	<b>1</b>
1.1 ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA IGLESIA CATÓLICA PARROQUIAL SANTIAGO APÓSTOL .....	1
1.1.1 DESCRIPCIÓN GENERAL DE LA INSTITUCIÓN .....	1
1.1.2 ORGANIGRAMA DE LA INSTITUCIÓN .....	1
1.1.3 SERVICIOS DE LA INSTITUCIÓN .....	2
1.2 PLANTEAMIENTO DEL PROBLEMA .....	4
1.3 FORMULACIÓN Y SISTEMATIZACIÓN .....	4
1.3.1 FORMULACIÓN .....	4
1.3.2 SISTEMATIZACIÓN .....	4
1.4 OBJETIVOS DE LA INVESTIGACIÓN .....	5
1.4.1 OBJETIVO GENERAL .....	5
1.4.2 OBJETIVOS ESPECÍFICOS .....	5
1.5 JUSTIFICACIÓN DEL PROYECTO .....	5
<b>CAPÍTULO 2. ASPECTOS TEÓRICOS. ....</b>	<b>7</b>
2.1 METODOLOGÍAS DE DESARROLLO CONSIDERADAS .....	7
2.1.1 METODOLOGÍA DE DESARROLLO SCRUM .....	7
2.1.2 METODOLOGÍA DE DESARROLLO EXTREME PROGRAMMING (XP) .....	8
2.2 JUSTIFICACIÓN DE LA METODOLOGÍA DE DESARROLLO SELECCIONADA – SCRUM .....	8
2.3 METODOLOGÍA DE DESARROLLO SCRUM .....	9
2.3.1 VISIÓN GENERAL DE LA METODOLOGÍA .....	11
2.3.2 ELEMENTOS DE LA METODOLOGÍA SCRUM .....	12
2.3.3 GRÁFICO DE SEGUIMIENTO BURNDOWN .....	14
2.3.4 BENEFICIOS DE LA METODOLOGÍA SCRUM .....	15
2.4 JUSTIFICACIÓN DEL USO DE SOFTWARE LIBRE .....	16
2.5 HERRAMIENTAS DE DESARROLLO .....	16
2.5.1 PLATAFORMAS DE DESARROLLO CONSIDERADAS .....	16

2.5.2	JUSTIFICACIÓN DE LA PLATAFORMA DE DESARROLLO SELECCIONADA – JAVA EE .....	17
2.5.3	LENGUAJES DE PROGRAMACIÓN CONSIDERADOS .....	19
2.5.4	JUSTIFICACIÓN DEL LENGUAJE DE PROGRAMACIÓN SELECCIONADO – JAVA .....	20
2.5.5	ENTORNOS DE DESARROLLO INTEGRADO (IDE) CONSIDERADOS.....	21
2.5.6	JUSTIFICACIÓN DEL IDE SELECCIONADO – ECLIPSE.....	22
2.5.7	SERVIDORES DE APLICACIONES CONSIDERADOS.....	23
2.5.8	JUSTIFICACIÓN DEL SERVIDOR DE APLICACIONES SELECCIONADO - JBOSS .....	24
2.5.9	PLATAFORMA DE CONSTRUCCIÓN DE PÁGINAS WEB CONSIDERADAS .....	25
2.5.10	JUSTIFICACIÓN DE LA PLATAFORMA DE CONSTRUCCIÓN DE PÁGINAS WEB SELECCIONADA – JAVA SERVER FACES CON RICHFACES.....	25
2.5.11	GESTORES DE BASE DE DATOS CONSIDERADOS.....	27
2.5.12	JUSTIFICACIÓN DEL GESTOR DE BASE DE DATOS SELECCIONADO – MySQL.....	28
2.5.13	LENGUAJE DE MODELAMIENTO UML.....	28
<b>CAPÍTULO 3. DESARROLLO DEL SISTEMA.....</b>		<b>33</b>
3.1	PLANIFICACIÓN DE LA ITERACIÓN .....	33
3.1.1	PRESENTACIÓN DEL EQUIPO DE TRABAJO Y ASIGNACIÓN DE ROLES .....	33
3.1.2	ELABORACIÓN DEL PRODUCT BACKLOG .....	33
3.1.3	DEFINICIÓN Y OBJETIVOS DE LOS SPRINTS .....	37
3.1.4	PLANIFICACIÓN DE CADA SPRINT.....	38
3.2	EJECUCIÓN DE LA ITERACIÓN .....	45
3.2.1	DISEÑO.....	45
3.2.2	CONSTRUCCIÓN.....	55
3.2.3	PRUEBAS .....	73
3.3	INSPECCIÓN Y ADAPTACIÓN.....	85
3.3.1	SEGUIMIENTO DEL PROGRESO DEL SPRINT.....	85
<b>CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>88</b>
4.1	CONCLUSIONES .....	88
4.2	RECOMENDACIONES.....	89
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>90</b>
<b>GLOSARIO.....</b>		<b>93</b>
<b>ANEXOS.....</b>		<b>97</b>

## ÍNDICE DE FIGURAS

Figura 1.1 Organigrama de la Iglesia. ....	2
Figura 2.1 Visión General de la Metodología. ....	11
Figura 2.2 Componentes de la Metodología Scrum ....	13
Figura 2.3 Gráfico Burndown ....	14
Figura 2.4 Arquitectura JAVA EE ....	19
Figura 2.5 Divisiones de una Clase.....	30
Figura 3.1 Tabla Modelo para elaborar el Product Backlog.....	34
Figura 3.2 Tabla modelo para elaborar el Sprint Backlog.....	38
Figura 3.3 Diseño de la plantilla para la interfaz gráfica. ....	47
Figura 3.4 Mapa de la Aplicación Web.....	51
Figura 3.5 Diagrama de Clases.....	53
Figura 3.6 Estructura del Proyecto.....	55
Figura 3.7 Estructura del Proyecto dentro del IDE. ....	57
Figura 3.8 Interfaz gráfica para el registro de Personas. ....	65
Figura 3.9 Gráfica Burndown para el Sprint 1 ....	86
Figura 3.10 Gráfica Burndown para el Sprint 2.....	87



## ÍNDICE DE TABLAS

Tabla 2.1 Cuadro comparativo entre las metodologías ágiles Scrum y XP.....	9
Tabla 3.1 Asignación de Roles. ....	33
Tabla 3.2 Product Backlog .....	36
Tabla 3.3 Definición y objetivos de cada Sprint .....	37
Tabla 3.4 Sprint Backlog para el Sprint 1 .....	41
Tabla 3.5 Sprint Backlog para el Sprint 2 .....	44
Tabla 3.6 Prueba de Aceptación Registro de Persona .....	73
Tabla 3.7 Prueba de Aceptación Búsqueda de Personas .....	74
Tabla 3.8 Prueba de Aceptación Registro de Eventos .....	74
Tabla 3.9 Prueba de Aceptación Listado de Eventos .....	75
Tabla 3.10 Prueba de Aceptación Registro de Celebración .....	75
Tabla 3.11 Prueba de Aceptación Búsqueda de Celebraciones .....	76
Tabla 3.12 Prueba de Aceptación Emisión de Certificados .....	76
Tabla 3.13 Prueba de Aceptación Creación de Usuarios. ....	77
Tabla 3.14 Prueba de Aceptación Búsqueda de Usuarios. ....	77
Tabla 3.15 Prueba de Aceptación Asignación de Personas como Tutores.....	78
Tabla 3.16 Prueba de Aceptación Asignación de Personas como Celebrantes. ....	78
Tabla 3.17 Prueba de Aceptación Registro de Lugares. ....	79
Tabla 3.18 Prueba de Aceptación Modificación de Parámetros. ....	79
Tabla 3.19 Prueba de Aceptación Registro de Grupos.....	80
Tabla 3.20 Prueba de Aceptación Búsqueda de Grupos.....	80
Tabla 3.21 Prueba de Aceptación Registro de Inscripciones.....	81
Tabla 3.22 Prueba de Aceptación Búsqueda de Inscripciones.....	81
Tabla 3.23 Prueba de Aceptación Listado de Mis Grupos. ....	82
Tabla 3.24 Prueba de Aceptación Registro de Asistencia. ....	82
Tabla 3.25 Prueba de Aceptación Registro de Notas. ....	83
Tabla 3.26 Prueba de Aceptación Autenticación de Usuarios. ....	83
Tabla 3.27 Prueba de Aceptación Generación de Reportes.....	84

## RESUMEN

El presente proyecto de titulación describe el desarrollo de un sistema para la administración de las actividades eclesiales de la Iglesia Católica Parroquial Santiago Apóstol usando herramientas Web. Se divide en cuatro capítulos que se describen a continuación:

El Capítulo 1 se titula Aspectos Generales y detalla la situación actual de la Iglesia Católica Parroquial Santiago Apóstol, el planteamiento del problema, la formulación y sistematización y se justifica la realización del proyecto.

El Capítulo 2 se titula Aspectos Teóricos y describe las metodologías de desarrollo y las herramientas consideradas, se justifican aquellas que fueron seleccionadas para el desarrollo del sistema, así como también se justifica el uso de software libre y lenguaje de modelamiento UML.

El Capítulo 3 se enfoca en el desarrollo del sistema, en la planificación, ejecución, inspección y adaptación de cada uno de las iteraciones en las que se divide el proyecto.

El Capítulo 4 especifica las Conclusiones y Recomendaciones obtenidas como resultado de la realización del presente proyecto.

## **PRESENTACIÓN**

En la actualidad gracias al avance de la tecnología es cada vez más fácil acceder a soluciones informáticas que permitan mejorar tanto la administración de procesos como la calidad del servicio que ofrece una determinada institución.

Por este motivo, el presente proyecto de titulación se enfoca en el desarrollo de un sistema para la administración de las actividades eclesíásticas de la Iglesia Católica Parroquial Santiago Apóstol usando herramientas Web.

Como resultado del proyecto se obtiene la aplicación Web SISAPE que permite la administración y automatización de procesos y emisión de certificados de bautizo y matrimonio.

Se ha seleccionado como metodología de desarrollo a Scrum debido a que incentiva a la participación activa del cliente y que es la que mejor se adapta a las necesidades del proyecto.

## **CAPÍTULO 1. ASPECTOS GENERALES**

### **1.1 ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA IGLESIA CATÓLICA PARROQUIAL SANTIAGO APÓSTOL**

#### **1.1.1 DESCRIPCIÓN GENERAL DE LA INSTITUCIÓN**

La iglesia Católica Parroquial Santiago Apóstol se encuentra ubicada en la Urbanización La Santiago, en las calles Temuco S12-22 y el Canelo. Fue fundada el 21 de diciembre de 1979 por el Padre José Carolo como parte de la creación del barrio La Santiago.

Cuenta con un salón comunal, la Casa Parroquial, 8 aulas en las que se imparten cursos de catequesis y de uso general, el despacho Parroquial y el templo en donde se celebran los distintos sacramentos y la eucaristía. Los horarios de la celebración Eucarística son de martes a sábado a las 18h00 y el domingo a las 07h00 – 09h00 – 19h00. El Despacho Parroquial brinda atención de martes a domingo en el horario de 15h00 a 18h00.

#### **1.1.2 ORGANIGRAMA DE LA INSTITUCIÓN**

En el organigrama presentado en la Figura 1.1 se describe la estructura organizacional de la iglesia, que permite distinguir las funciones de cada área de la institución, lo que facilita identificar los procesos de cada una.

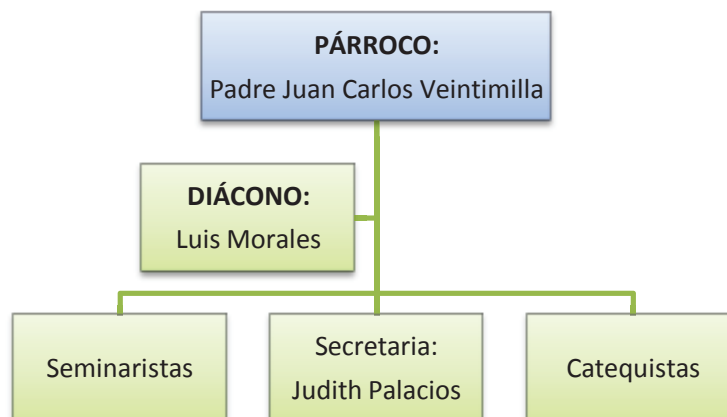


Figura 1.1 Organigrama de la Iglesia.<sup>1</sup>

### 1.1.3 SERVICIOS DE LA INSTITUCIÓN

La iglesia Católica Parroquial Santiago Apóstol ofrece los siguientes servicios:

**Emisión de certificados:** se entregan certificados de Bautismo, Matrimonio, Primera Comunión y Confirmación, y en ocasiones se entregan certificados o informes de actividad Pastoral.

**Celebración de Sacramentos:** Bautismos y Matrimonios (se realizan los días sábados en horarios a escoger), Penitencia y Confesión (martes a domingo bajo disponibilidad del párroco), Asistencia a los enfermos (de acuerdo a solicitud de la parroquia), Primera Comunión y Confirmación (una vez al año).

**Cursos de Catequesis Pre-Sacramental:** se realizan los domingos de 9 a 12h30.

**Servicios Funerarios:** Ofrece servicios de velación y funerales cuando los feligreses soliciten.

<sup>1</sup> Figura propuesta por la autora de la tesis ya que la iglesia Católica Parroquial Santiago Apóstol no cuenta con un organigrama propio

Existen diferentes grupos que realizan sus actividades dentro de la iglesia Católica Parroquial Santiago Apóstol. Entre estos grupos están:

**Catequesis Pre-Sacramental:** que consiste en la instrucción y preparación de niños, jóvenes y sus familias para recibir los sacramentos de Bautismo, Confirmación y Eucaristía. Consta de 2 grupos:

- Grupo de catequesis Familiar.
- Catecismo de niños y jóvenes.

**Movimientos de apostolado:** consisten en las distintas actividades que promueven la ayuda al prójimo y a los más necesitados. Cuenta con los grupos:

- *Juan XXIII:* llama a la conversión a las personas alejadas o al margen de la Iglesia así como la perseverancia y participación en actividades pastorales.
- *Grupo de oración del Padre Carolo:* realiza el estudio de la Biblia y ofrece oraciones por aquellos que lo necesitan.
- *Catequistas:* preparan los temas semanales e imparten los distintos cursos de catequesis Pre-Sacramental.
- *Grupos Misioneros:* dedicados a las actividades y formación misionera.
- *Grupos de intercepción:* interceden mediante oración por las personas del grupo que atraviesan dificultades.

**Monaguillos:** Asisten al sacerdote durante la eucaristía mediante el ejercicio del ministerio del altar.

**Grupos de coros y acompañamiento musical:** 4 grupos que participan en las celebraciones de la Eucaristía en diferentes horarios.

**Pastoral Social:** ayuda solidaria a los más necesitados y servicio de Comedor Popular.

## **1.2 PLANTEAMIENTO DEL PROBLEMA**

Actualmente en la Iglesia Católica Parroquial Santiago Apóstol el registro de eventos católicos, cursos, grupos de catequesis, feligreses; así como los procesos de emisión de certificados de bautizo y de matrimonio, gestión de cursos y eventos se realizan de forma manual y sin una organización específica, lo cual provoca un desperdicio tanto de tiempo como de recursos económicos.

## **1.3 FORMULACIÓN Y SISTEMATIZACIÓN**

### **1.3.1 FORMULACIÓN**

¿Cómo administrar eficientemente la información y actividades que se realizan en la Iglesia Católica Parroquial Santiago Apóstol?

### **1.3.2 SISTEMATIZACIÓN**

- ¿Cómo mejorar la entrega de certificados que emite la Iglesia Católica Parroquial Santiago Apóstol?
- ¿Cómo registrar en forma correcta la información relacionada a actividades y eventos de la Iglesia Católica Parroquial Santiago Apóstol?
- ¿Con cuántos módulos e interfaces debe contar el sistema?
- ¿Cómo comprobar que el sistema cumple con los requerimientos de los usuarios?

## **1.4 OBJETIVOS DE LA INVESTIGACIÓN**

### **1.4.1 OBJETIVO GENERAL**

Desarrollar un sistema de administración de las actividades eclesiales de la Iglesia Católica Parroquial Santiago Apóstol usando herramientas Web.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Mejorar el proceso de entrega de certificados que emite la parroquia.
- Registrar la información correspondiente a la realización de eventos y cursos, registro de miembros, familias, personas y demás actividades que se realizan en la Iglesia Católica Parroquial Santiago Apóstol.
- Programar las interfaces y módulos del sistema.
- Probar el funcionamiento del sitio Web mediante la evaluación de su funcionalidad por parte de los usuarios.

## **1.5 JUSTIFICACIÓN DEL PROYECTO**

La realización del presente proyecto permite que la Iglesia Católica Parroquial Santiago Apóstol brinde un mejor servicio a la comunidad mediante la emisión rápida y oportuna de certificados de bautizo y matrimonio, aprovechando la ventaja de que existe una red inalámbrica que cubre todas las áreas que integran la iglesia. Por ello se plantea la implementación de un sistema web para facilitar su ingreso desde cualquier dispositivo conectado a la red a través de un navegador.

La orientación a la Web permite que en un futuro la iglesia disponga de más servicios tales como la visualización de notas, asistencia a los grupos de catequesis y horarios de eventos.



Se administra la información personal de los integrantes de la parroquia, así como de los grupos y eventos que se realizan, para mantener una organización coherente evitando problemas de logística, cruces de horarios, tutores, horarios, etc.

## **CAPÍTULO 2. ASPECTOS TEÓRICOS.**

### **2.1 METODOLOGÍAS DE DESARROLLO CONSIDERADAS.**

La gestión y desarrollo de un producto de software es una tarea complicada que exige esfuerzo y concentración, por lo que debe ser realizada con la mejor calidad posible. Por tal motivo se realiza una investigación detallada de las metodologías ágiles de desarrollo de software para determinar aquella que se adapta mejor a las características del proyecto. Como resultado de esta investigación se consideran dos metodologías para el desarrollo del proyecto. A continuación se describen brevemente estas metodologías y se realiza un cuadro comparativo para determinar la metodología a seguir para el presente proyecto.

#### **2.1.1 METODOLOGÍA DE DESARROLLO SCRUM**

Scrum es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software dentro del cual las personas pueden afrontar complejos problemas adaptativos, a la vez que entregan productos del máximo valor posible de forma productiva y creativa. Es ligero y fácil de entender.

Scrum es un marco de trabajo de procesos que ha sido utilizado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos:

- El desarrollo de software se realiza mediante iteraciones, denominada Sprints, con una duración recomendada de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
- La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

### **2.1.2 METODOLOGÍA DE DESARROLLO EXTREME PROGRAMMING (XP)**

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los valores y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Los valores de la programación extrema son: simplicidad, comunicación, retroalimentación, coraje y respeto.

## **2.2 JUSTIFICACIÓN DE LA METODOLOGÍA DE DESARROLLO SELECCIONADA – SCRUM**

En la Tabla 2.1 se muestran las principales características por las que Scrum es la que mejor se adapta al desarrollo del proyecto sobre XP, por lo que se selecciona Scrum como metodología de desarrollo.

SCRUM	eXTREME PROGRAMMING - XP
Es una metodología de desarrollo ágil basada en la administración del proyecto.	Es una metodología de desarrollo ágil que está más centrada en la programación y creación del proyecto.
Programación Individual.	Programación en parejas.
Trata de seguir el orden de prioridades que marca el Product Owner en el Sprint Backlog, pero puede cambiarlo si es mejor para el desarrollo de las tareas.	El equipo de desarrollo sigue estrictamente el orden de prioridad de las tareas definido por el cliente.

Tabla 2.1 Cuadro comparativo entre las metodologías ágiles Scrum y XP<sup>2</sup>

Debido a que Scrum se caracteriza por ser iterativa e incremental, permite realizar un seguimiento del avance de cada una de las tareas así como de los problemas que van surgiendo durante el proyecto. Además Scrum cuenta con otras características como adaptabilidad, flexibilidad, revisión continua, etc., por lo que al aplicarlo a la elaboración de un sistema dará como resultado la creación de un proyecto de mejor calidad que satisfaga las necesidades y expectativas del cliente.

Con esta metodología el cliente se siente entusiasmado y comprometido con el proyecto ya que lo ve crecer iteración a iteración. Asimismo le permite re alinear el software con los objetivos del negocio, ya que puede solicitar cambios funcionales o de prioridad al inicio de una nueva iteración sin inconvenientes.

## 2.3 METODOLOGÍA DE DESARROLLO SCRUM

Es una metodología de desarrollo ágil y flexible para gestionar el desarrollo de software basado en un proceso iterativo e incremental. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación en el mínimo tiempo. Permite tener el control continuo sobre el estado actual del software desarrollado.

<sup>2</sup> [http://www.islavisual.com/articulos/desarrollo\\_web/diferencias-entre-scrum-y-xp.php](http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php)

Scrum es una metodología de desarrollo muy simple, que requiere esfuerzo y compromiso del equipo que forma parte del proyecto, ya que no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias que se presenten durante la evolución del proyecto.

Al ser una metodología ágil:

- Es un modelo de desarrollo que se caracteriza por ser más adaptable que predictivo.
- Está orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo incremental basada en iteraciones y revisiones periódicas.
- Permite tener un control continuo sobre el estado actual del producto de software.
- Cada iteración termina con un producto de software entregable que se puede publicar o bien seguir mejorando.

El grado de funcionamiento de Scrum depende directamente de tres condiciones:

1. Características del entorno (organización y proyecto), adecuadas para desarrollo ágil.
2. Conocimiento de la metodología de trabajo por parte del equipo de desarrollo y las personas implicadas por el lado del cliente.
3. Asignación de responsabilidades:
  - Sobre el producto.
  - Sobre el desarrollo.
  - Sobre el funcionamiento de Scrum.

### 2.3.1 VISIÓN GENERAL DE LA METODOLOGÍA

Scrum denomina “Sprint” a cada iteración de desarrollo. Es el período en el cual se lleva a cabo el trabajo en sí. Se recomienda que el sprint tenga una duración constante y sea definida en base a la experiencia del equipo de desarrollo. Se puede comenzar con una duración de sprint de 2 a 3 semanas e ir ajustándolo con base en el ritmo del equipo.

Al final de cada sprint el equipo presentará los avances realizados y el resultado obtenido es un producto potencialmente entregable al cliente.

El sprint es considerado como el núcleo central que proporciona la base de desarrollo iterativo e incremental, como se observa en la Figura 2.1

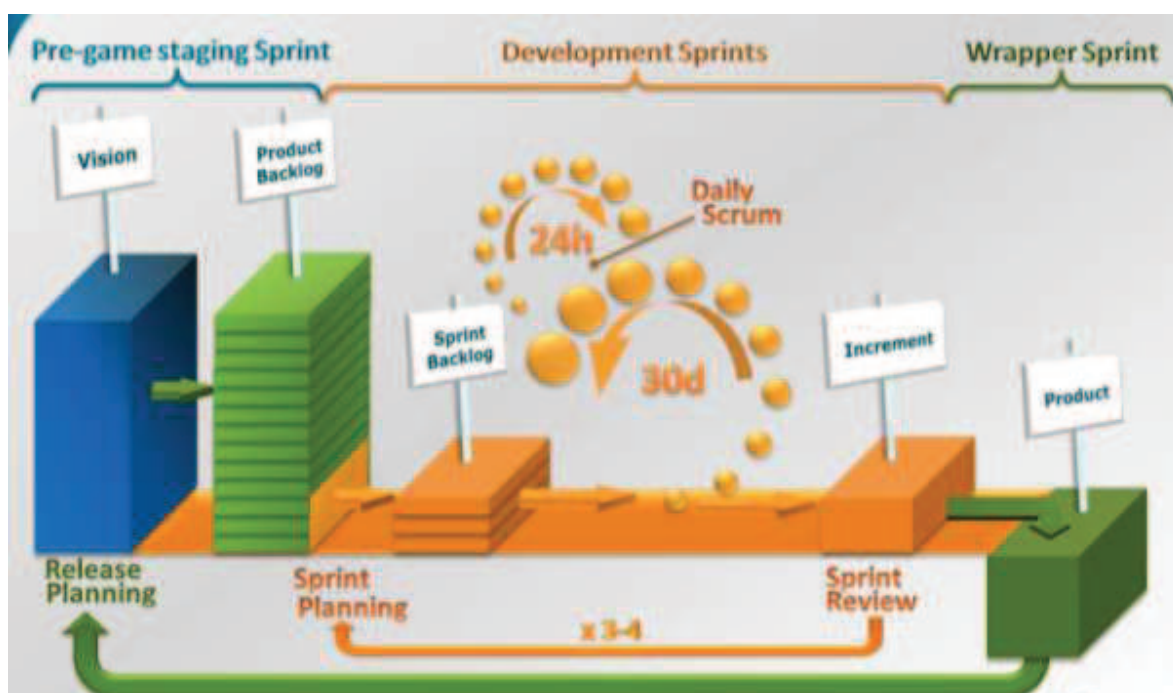


Figura 2.1 Visión General de la Metodología.<sup>3</sup>

<sup>3</sup> VERHEYEN, Gunther. Introduction to the Professional Scrum Developer program.

<http://mscop.be.capgemini.com/tag/scrum/>

### 2.3.2 ELEMENTOS DE LA METODOLOGÍA SCRUM

El desarrollo de Scrum está conformado por:

#### Las reuniones

- *Planificación de sprint (Sprint Planning Meeting)*: se realiza previamente al inicio de cada ciclo sprint, en la que se determina cuál va a ser el equipo de trabajo y los objetivos que se deben cumplir durante la actual iteración. En esta reunión se genera el respectivo “Sprint Backlog”.
- *Reunión diaria (Daily Scrum)*: se realiza una revisión del trabajo efectuado por el equipo hasta la fecha y la previsión para el día siguiente. La reunión se realiza todos los días a la misma hora y en la misma ubicación. Deben acudir todos los miembros del equipo de desarrollo.
- *Revisión de sprint (Sprint Review Meeting)*: Se efectúa al finalizar cada sprint, en la que se realiza el análisis y revisión del incremento generado. Se presenta al cliente el incremento desarrollado. Esta reunión tiene como propósito la mejora continua del proceso.
- *Retrospectiva del Sprint (Sprint Retrospective)*: Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso.

#### Los componentes

- *Pila del producto*: Conocida como **Product Backlog**. Es una lista que contiene los requerimientos y características desde el punto de vista del usuario que se

genera con una visión inicial del producto y va creciendo y evolucionando durante el desarrollo. Es un documento de alto nivel para el proyecto que está en constante evolución.

- *Pila del sprint*: Conocida como **Sprint Backlog**. Es una lista que determina las tareas que debe realizar el equipo de desarrollo durante el sprint para generar el incremento previsto. El equipo asume el compromiso de la ejecución.
- *Incremento*: Es el resultado de cada sprint. Se trata de un resultado completamente probado, terminado y en condiciones de ser utilizado

## Los roles

Se clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto.

- *Propietario del producto*: Conocido como **Product Owner**. Representa la voz del cliente. Es responsable de obtener el mayor valor de producto para los clientes, usuarios y demás implicados.
- *Equipo de desarrollo*: Lo conforma el grupo o grupos de trabajo que desarrollan el producto. Tiene la responsabilidad de entregar el producto.
- *Scrum Master*: conocido también como **Scrum Manager**. Es el responsable de garantizar que se aplica la metodología Scrum de manera correcta y de la productividad del equipo de desarrollo.

En la Figura 2.2 se muestran los componentes de la metodología Scrum descritos anteriormente.



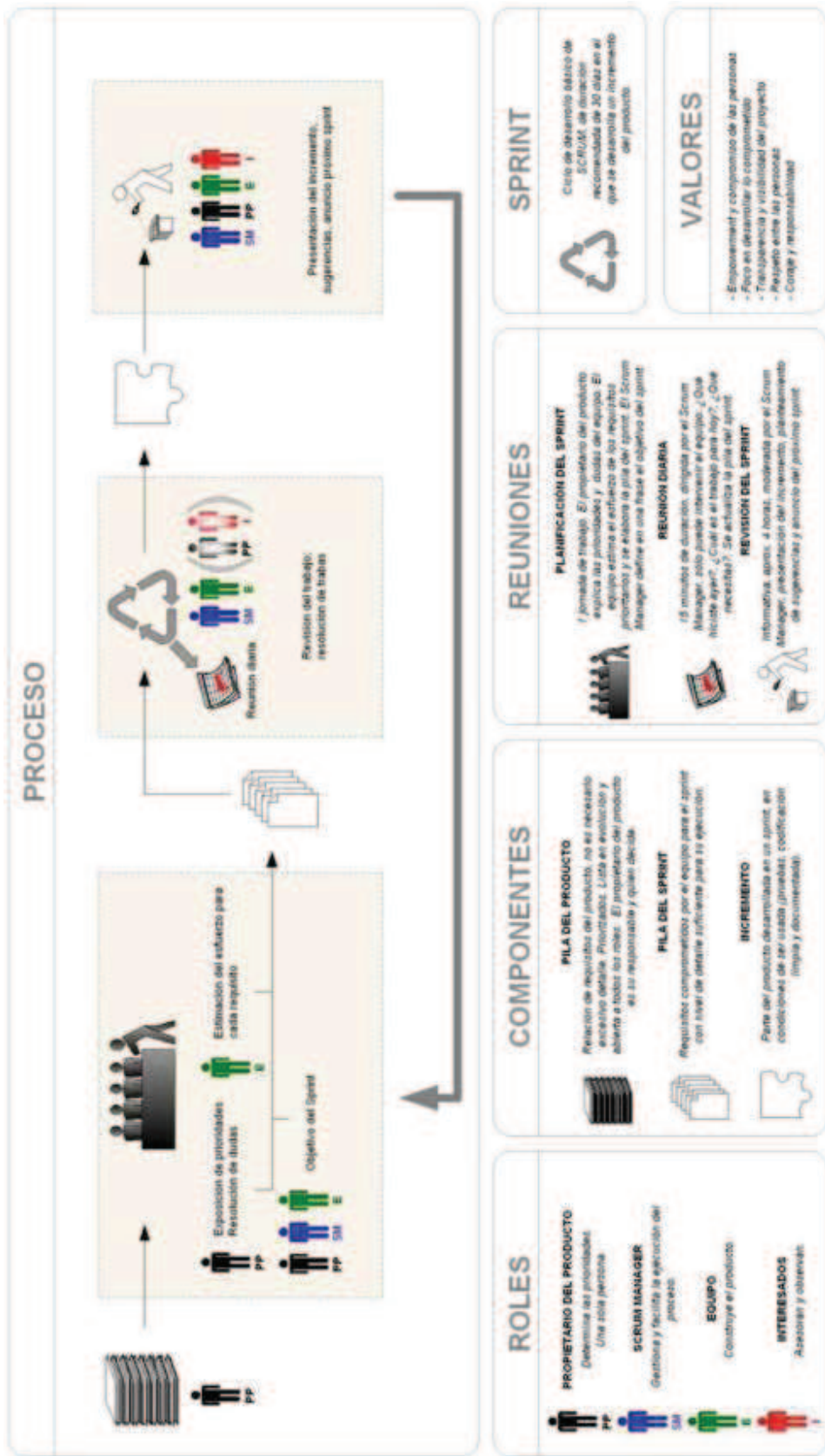


Figura 2.2 Componentes de la Metodología Scrum<sup>4</sup>

<sup>4</sup> IS-T. METODOLOGÍA SCRUM. <http://is-t.wikispaces.com/metodologia+scrum>

### 2.3.3 GRÁFICO DE SEGUIMIENTO BURNDOWN

Es un gráfico que muestra la velocidad a la que se están cumpliendo los requerimientos. Este gráfico se muestra abiertamente a todos los que participan en el proyecto y refleja la cantidad de requisitos del proyecto que están pendientes para el sprint en curso. Este diagrama es útil para determinar cuándo se completará el trabajo y tomar decisiones para cumplir con los objetivos en el tiempo establecido.

Sobre el eje horizontal se ubican los días hábiles que tiene el Sprint, y sobre el eje vertical la cantidad de puntos a realizar durante ese Sprint.

La línea de color naranja que se muestra en la Figura 2.3 representa la línea de "avance ideal". Esta "línea ideal" se traza el primer día, y al equipo le sirve como referencia para saber su situación respecto al sprint.

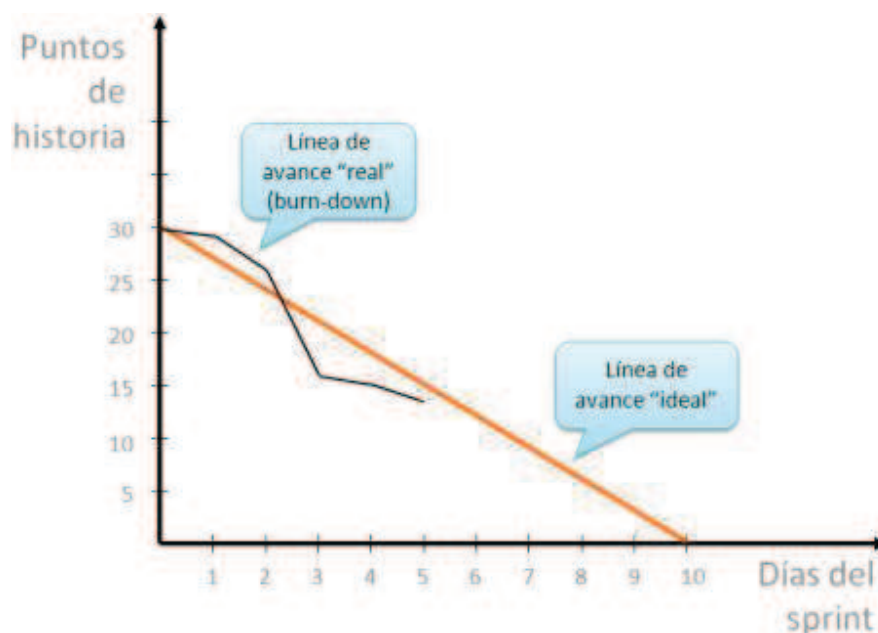


Figura 2.3 Gráfico Burndown<sup>5</sup>

<sup>5</sup> <http://www.dosideas.com/noticias/metodologias/757-los-graficos-para-quemar-puntos-en-scrum.html>

El gráfico se traza para cada sprint y se actualiza diariamente. Todos los días, durante la Reunión Diaria, el equipo actualiza las estimaciones de las tareas, y así va contando cuántos puntos le quedan por terminar en el Sprint. Se apunta en el gráfico el trabajo realizado el día anterior y según la dirección de la línea de avance real se toma las decisiones y se planifica el trabajo del día.

#### 2.3.4 BENEFICIOS DE LA METODOLOGÍA SCRUM

- **Cumplimiento de expectativas:** El cliente establece sus expectativas indicando el valor que le aporta a cada requerimiento del proyecto y comprueba que se han cumplido.
- **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por parte del cliente.
- **Reducción del Time to Market:** El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- **Mayor calidad del software:** la metodología de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a generar software de calidad superior.
- **Mayor productividad:** Mayor motivación del equipo debido al hecho de que son autónomos para organizarse y asignar prioridades.
- **Maximiza la satisfacción del cliente:** Se produce software únicamente con las prestaciones que aportan mayor valor para el cliente.
- **Predicción de tiempos:** mediante esta metodología se conoce la velocidad media del equipo por sprint, con lo que consecuentemente, es posible estimar fácilmente la disponibilidad de una determinada funcionalidad.
- **Reducción de riesgos:** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con la que el equipo avanza en el proyecto, permite disminuir riesgos eficazmente de manera anticipada.

## **2.4 JUSTIFICACIÓN DEL USO DE SOFTWARE LIBRE**

El término Software libre hace referencia a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Existen cuatro clases de libertad para los usuarios de este tipo de software:

- Libertad 0: Libertad para ejecutar el programa sin importar el propósito.
- Libertad 1: Libertad para estudiar el funcionamiento del programa y adaptarlo según la necesidad (es indispensable permitir el acceso al código fuente).
- Libertad 2: Libertad para redistribuir copias y que otros se beneficien de las mismas.
- Libertad 3: Libertad para mejorar el programa y posteriormente publicarlo para el bien de toda la comunidad (es indispensable permitir el acceso al código fuente).

Para el desarrollo del sistema de administración de las actividades eclesiales de la Iglesia Católica Parroquial Santiago Apóstol se utilizan herramientas de libre difusión (software libre) ya que ofrecen la ventaja de manejarlas libremente, facilitando la adaptación del sistema a los requerimientos de la iglesia.

A continuación se describen las herramientas utilizadas en el desarrollo del sistema.

## **2.5 HERRAMIENTAS DE DESARROLLO**

### **2.5.1 PLATAFORMAS DE DESARROLLO CONSIDERADAS**

#### **2.5.1.1 .Net**

.NET es una plataforma para el desarrollo de aplicaciones, que integra múltiples tecnologías que han ido apareciendo en los últimos años como ASP.NET,

ADO.NET, LINQ, WPF, etc., junto con el potente entorno integrado de desarrollo Visual Studio, que permite desarrollar múltiples tipos de aplicaciones.

Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma.

### **2.5.1.2 Java EE**

La plataforma Java Enterprise Edition (Java EE) es un conjunto de especificaciones que facilitan el desarrollo y despliegue de aplicaciones empresariales multi-capa. Java EE ofrece un conjunto de especificaciones y técnicas que proporcionan soluciones completas, seguras, estables y escalables para el desarrollo, despliegue y gestión de aplicaciones de múltiples niveles de funcionalidad basadas en servidores. Se reduce el costo y complejidad de desarrollo, lo cual resulta en servicios que se pueden desplegar y extender fácilmente.

## **2.5.2 JUSTIFICACIÓN DE LA PLATAFORMA DE DESARROLLO SELECCIONADA – JAVA EE**

Se selecciona Java EE como plataforma de desarrollo ya que ofrece las siguientes ventajas:

- Soporte para múltiples plataformas y sistemas operativos.
- Es orientado a objetos.
- Avalado por múltiples empresas (SUN, IBM, ORACLE, etc.).
- Sencillo.
- Robusto.
- Seguro.
- Portable.

*Java Enterprise Edition* o *Java EE* es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

En esta plataforma se especifica la siguiente arquitectura, donde se contemplan tres capas en función del tipo de servicio y contenedores:

- **Capa de cliente.**

Contiene los componentes Java (applets o aplicaciones) y no-Java (páginas HTML, JavaScript, etc.).

- **Capa de Servidor JAVA EE.**

Se identifican las siguientes capas:

- **Capa Web.** Contiene el código HTML, JSP o JSF.
- **Capa del Negocio.** Contiene los Enterprise Java Beans (EJB).

- **Capa de Base de Datos.**

Contiene el sistema de Base de Datos.

En la Figura 2.4 se muestra las capas de la arquitectura JAVA EE descritas.

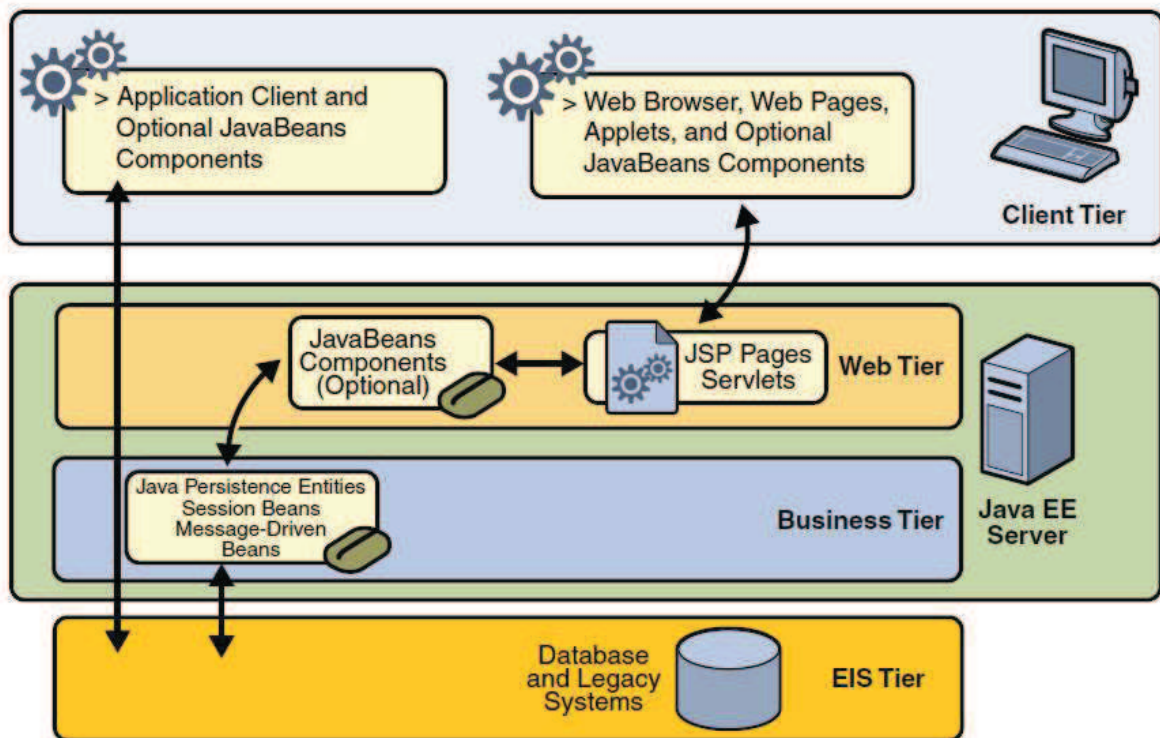


Figura 2.4 Arquitectura JAVA EE<sup>6</sup>

## 2.5.3 LENGUAJES DE PROGRAMACIÓN CONSIDERADOS

### 2.5.3.1 Java

Java es un lenguaje de programación por objetos creado por Sun Microsystems que permite crear programas que funcionan en cualquier tipo de ordenador y sistema operativo. Se usa para crear programas especiales denominados applets, que pueden ser incorporados en páginas web para hacerlas interactivas.

Los pilares en los que se sustenta Java son: la programación orientada a objetos, la posibilidad de ejecutar un mismo programa en diversos sistemas operativos, la opción de ejecutar el código en sistemas remotos de manera segura y la facilidad de uso.

<sup>6</sup> Sun Microsystems. The Java EE 5 Tutorial.

### 2.5.3.2 Visual Basic .NET

Visual Basic .NET es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre la plataforma .NET. Su introducción resultó muy controvertida, ya que debido a cambios significativos en el lenguaje VB.NET no es compatible hacia atrás con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas.

Tiene una gran cantidad de características nuevas y mejoradas como la herencia, las interfaces y la sobrecarga que la convierten en un lenguaje eficaz de programación orientado a objetos. Otras características nuevas del lenguaje son el subprocesamiento libre y el control de excepciones estructurado.

### 2.5.4 JUSTIFICACIÓN DEL LENGUAJE DE PROGRAMACIÓN SELECCIONADO – JAVA

Se elige Java como lenguaje de desarrollo porque la desarrolladora del proyecto cuenta con mayor conocimiento y experiencia en este lenguaje.

Es un lenguaje de programación idóneo para diseñar programas que funcionen en Internet. Es un lenguaje de plataformas cruzadas, lo que significa que puede ser diseñado para que se ejecute tanto en Windows, así como en las diferentes versiones de UNIX y Macintosh.

Sus principales características y ventajas son:

1. **Reutilización:** La reutilización de código es la esencia de la programación. Dividir los requerimientos de una aplicación en componentes es una forma de Reutilización. Java al ser un lenguaje orientado a objetos, provee mecanismos que permiten reutilizar código.



2. **Modularidad:** Permite desarrollar una aplicación de tipo Web dividiéndola en pequeñas partes que realicen una tarea en particular, lo que se conoce como módulos. De esta manera la aplicación se vuelve más fácil de entender y mantener. Los servlets, JSF y los Enterprise Java Beans proveen formas de desglosar las aplicaciones en capas y tareas.
3. **Orientado a objetos:** Java maneja los datos en forma de objetos y permite proveer interfaces a esos objetos. Soporta las características propias del paradigma orientado a objetos: herencia, polimorfismo y encapsulación.
4. **Distribuido:** Java contiene diferentes librerías que permiten acceder e interactuar con protocolos como HTTP y FTP, lo que facilita el acceso a la información a través de la red con tanta facilidad como a los archivos locales.
5. **Robusto:** Java realiza comprobaciones y verificaciones para detectar errores y solucionar problemas tanto en tiempo de compilación como en tiempo de ejecución. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. De igual manera, administra la memoria de forma automática para que el desarrollador no se preocupe de esto.
6. **Arquitectura neutral:** Para que Java se establezca como parte integral de la red, el compilador compila su código a un archivo objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier computadora que tenga el sistema de ejecución (runtime) puede ejecutar ese código objeto, independientemente de la máquina en que ha sido generado.

## 2.5.5 ENTORNOS DE DESARROLLO INTEGRADO (IDE) CONSIDERADOS

### 2.5.5.1 Eclipse

Eclipse es un entorno de desarrollo integrado que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Aunque Eclipse pretende ser un entorno versátil soportando varios lenguajes de programación, es con el lenguaje Java con el que mejor se integra y con el que ha ganado su popularidad. Es una aplicación gratuita y de código abierto.

Eclipse emplea módulos (plugins) para proporcionar toda su funcionalidad. Este mecanismo de módulos es una plataforma ligera para componentes de software.

La arquitectura de plugins de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías, etc.

### **2.5.5.2 NetBeans**

NetBeans IDE es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas que está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Modular y normalizado.

NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de packs.

Existe además un número importante de módulos para extender NetBeans IDE. Es un producto libre y gratuito sin restricciones de uso.

## **2.5.6 JUSTIFICACIÓN DEL IDE SELECCIONADO – ECLIPSE**

Se selecciona Eclipse como IDE de desarrollo porque la desarrolladora cuenta con mayor conocimiento y experiencia en el uso de esta herramienta.

Eclipse cuenta con numerosas características como son:

- Dispone de un editor de texto con resaltado de sintaxis.
- La compilación es en tiempo real.

- Tiene pruebas unitarias con JUnit.
- Control de versiones con CVS.
- Asistentes para creación de proyectos, clases, etc.
- Refactorización.
- A través de plugins libremente disponibles es posible añadir control de versiones con Subversion.4 e integración con Hibernate.5

En cuanto a la utilización de eclipse para la creación de aplicaciones clientes se puede decir que:

- Eclipse provee al programador con Frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de Software, Aplicaciones web, etc.
- El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código.
- El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadata en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

## **2.5.7 SERVIDORES DE APLICACIONES CONSIDERADOS**

### **2.5.7.1 JBoss**

JBoss es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java. Puede usarse libremente sin costo alguno en cualquier aplicación comercial.

Permite alojar y servir aplicaciones e información en una interfaz Web de portal, publicar y gestionar su contenido y personalizar la experiencia del usuario.

JBoss en particular ha sido desde sus inicios un servidor de aplicaciones extensible y modular.

#### **2.5.7.2 GlassFish**

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito, de código libre. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server).

Entre las características más notables de GlassFish cabe destacar su velocidad, alta escalabilidad, manejo centralizado de clusters e instancias, bajo consumo de memoria, interoperabilidad con .NET 3 y un excelente panel de administración.

La instalación de GlassFish bajo Windows puede ser un tanto compleja, pero hay abundante documentación disponible para resolver cualquier posible incidencia.

#### **2.5.8 JUSTIFICACIÓN DEL SERVIDOR DE APLICACIONES SELECCIONADO - JBOSS**

Se utiliza el servidor de aplicaciones JBoss en este proyecto ya que simplifica el acceso a las aplicaciones e información, tiene la ventaja de ser código abierto, permite aumentar la productividad y reducir la posibilidad de errores. Tiene soporte completo con la plataforma Java EE que también se utiliza en el desarrollo del sistema. Además al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. Cumple los estándares por lo que es confiable a nivel de empresa.

## **2.5.9 PLATAFORMA DE CONSTRUCCIÓN DE PÁGINAS WEB CONSIDERADAS**

### **2.5.9.1 Java Server Faces con RichFaces**

JSF es una tecnología y marco de trabajo para aplicaciones Java basadas en Web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. Está orientado a la interfaz gráfica de usuario (GUI), facilitando el desarrollo de las mismas, y que sin embargo, realiza una separación entre comportamiento y presentación, además de proporcionar su propia servlet como controlador, lo que permite un desarrollo más simple y una aplicación mejor estructurada.

RichFaces es una librería de componentes JSF original de Exadel y adquirida por JBoss. Posee un avanzado framework Ajax que permite la creación de componentes UI dinámicos, y un kit de desarrollo de componentes. Corre en cualquier contenedor Servlet.

### **2.5.9.2 IceFaces**

IceFaces es un Framework que permite desarrollar Aplicaciones Ricas de Internet RIA de una manera fácil y rápida, utilizando el lenguaje Java. Éste está basado en una serie de estándares que permiten continuar trabajando con la clásica forma de desarrollo con Java. De esa manera, se puede trabajar con herramientas existentes como NetBeans, Eclipse; y Servidores de Aplicaciones como Sun GlassFish, Apache Tomcat, IBM WebSphere, JBoss, entre otros.

## **2.5.10 JUSTIFICACIÓN DE LA PLATAFORMA DE CONSTRUCCIÓN DE PÁGINAS WEB SELECCIONADA – JAVA SERVER FACES CON RICHFACES**

Se selecciona Java Server Faces junto con RichFaces por la factibilidad de aprendizaje de la desarrolladora. Adicionalmente se considera como ventaja que ya

está incorporada dentro del servidor de Aplicaciones JBoss previamente seleccionado.

Además estas tecnologías incluyen:

- Un conjunto de API (interfaz de programación de aplicaciones) para representar componentes de una interfaz de usuario, lo que permite administrar su estado, manejar eventos, validar entradas, definir el esquema de navegación entre páginas.
- Un conjunto de componentes para la interfaz de usuario establecidos por defecto.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- JavaBeans administrados.
- JSF es extensible, por lo que se pueden crear nuevos elementos de la interfaz o modificar los existentes.
- A pesar de que HTML es el lenguaje de marcado por defecto, no está limitado a éste, ya que tiene la capacidad de utilizar diferentes *renderers* para los componentes GUI y obtener así diferentes salidas para mandar al cliente.
- Definición simple de controladores y JavaBeans.
- Posee un sistema de validaciones intrínseco.
- Es fácil de usar, su arquitectura define claramente una separación entre las capas de presentación y lógica permitiendo que la conexión entre ambas sea sencilla. Este diseño posibilita a cada miembro del equipo de desarrollo centrarse en la parte de la aplicación que le corresponde desarrollar, proporcionando además un modelo de programación sencilla que simplifica la posterior unión de todos los módulos.

## **2.5.11 GESTORES DE BASE DE DATOS CONSIDERADOS**

### **2.5.11.1 PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo.

PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. El soporte a orientación a objetos es una simple extensión que ofrece prestaciones como la herencia, no un soporte completo.

### **2.5.11.2 MySQL**

MySQL es un sistema de gestión de base de datos relacionales multiusuario desarrollado por Sun Microsystems que fue adquirido recientemente por Oracle. Se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

Las características de este gestor de base de datos son:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.

- Herramientas gráficas de diseño y administración de base de datos.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Replicación.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.
- Es altamente confiable en cuanto a estabilidad se refiere.
- Funciona en diferentes plataformas.
- Soporte completo para operadores y funciones en las cláusulas de consultas

`SELECT` y `WHERE`.

### **2.5.12 JUSTIFICACIÓN DEL GESTOR DE BASE DE DATOS SELECCIONADO – MySQL.**

El motor de base datos escogido para el desarrollo de este proyecto es MySQL, ya que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. Cuenta con conectividad segura y alta velocidad en la búsqueda de datos e información.

### **2.5.13 LENGUAJE DE MODELAMIENTO UML**

Se utiliza el lenguaje UML en el desarrollo del proyecto para apoyar las tareas de planificación y diseño del sistema web.

El Lenguaje de Modelado Unificado (UML por sus siglas en inglés *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software que permite visualizar, especificar, construir y documentar un sistema. Es independiente de los métodos de análisis y diseño. Se puede aplicar en el desarrollo de software de diferentes formas para dar soporte a una metodología de desarrollo de software.



Los beneficios que conlleva el uso de UML en el desarrollo de software son:

- Mejores tiempos totales de desarrollo.
- Modelar sistemas utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Se basa en una notación gráfica con una semántica bien definida.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.
- Permite lograr un equilibrio entre expresividad y simplicidad.
- Facilita la comunicación entre el equipo al existir un lenguaje común.

UML posibilita la captura, comunicación y nivelación de conocimiento estratégico, táctico y operacional para facilitar el incremento de valor, aumentando la calidad, reduciendo costos y reduciendo el tiempo de presentación al mercado; manejando riesgos y siendo proactivo para el posible aumento de complejidad o cambio.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

#### **2.5.13.1 Modelamiento de Clases**

Un diagrama de clases es un diagrama que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño de un sistema. Permite visualizar las relaciones entre las clases que involucran el sistema.

**Elementos:** Un diagrama de clases está compuesto por:

- **Clase:** atributos, métodos y visibilidad.
- **Relaciones:** Herencia, Composición, agregación, asociación y uso.

## Clase

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio.

En UML una clase es representada por un rectángulo con tres divisiones como lo muestra la Figura 2.5:

- División superior: nombre de la clase
- División intermedia: atributos de la clase
- División inferior: métodos de la clase

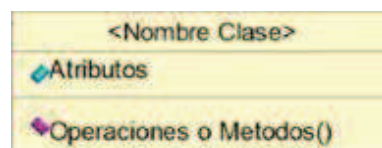


Figura 2.5 Divisiones de una Clase

### a) Atributos de la clase:

Es una propiedad o característica de una clase. Describe un rango de valores que puede tomar esa propiedad en las instancias.

### b) Métodos de la clase:

Los métodos u operaciones de una clase son la forma como ésta interactúa con su entorno, éstos pueden tener las características de visibilidad: public, private o protected.

**c) Visibilidad:**

Puede ser de 3 tipos, los que definen el grado de comunicación y visibilidad tanto de los métodos como de los atributos con el entorno. Son:

- **public:** Indica que el atributo o método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
- **private:** Indica que el atributo o método sólo será accesible desde dentro de la clase (sólo sus métodos pueden acceder a él).
- **protected:** Indica que el atributo o método no será accesible desde fuera de la clase, pero si podrán acceder a él métodos de la clase además de las subclases que se deriven.

**Relaciones entre clases:****1. Herencia (Especialización/Generalización):**

Indica que una clase (clase derivada) hereda los métodos y atributos especificados por una clase (clase base), por lo cual una clase derivada además de tener sus propios métodos y atributos, podrá acceder a los métodos y atributos visibles de su clase base.

**2. Agregación:**

La agregación es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye (el objeto base utiliza al incluido para su funcionamiento).

**3. Composición:**

La composición es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye (el objeto base se construye a partir del objeto incluido, es decir, es parte/todo).

#### **4. Dependencia o instanciación (uso):**

Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase). Se denota por una flecha punteada. El uso más particular de este tipo de relación es para denotar la dependencia que tiene una clase de otra.

#### **5. Asociación:**

Permite asociar objetos que colaboran entre sí. No es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

En una relación de este tipo existe un elemento adicional:

**Multiplicidad:** La multiplicidad de una relación determina cuantos objetos de cada tipo intervienen en la relación. Cada asociación tiene dos multiplicidades (una para cada extremo de la relación).

## CAPÍTULO 3. DESARROLLO DEL SISTEMA

### 3.1 PLANIFICACIÓN DE LA ITERACIÓN

Para que la planificación de los diferentes sprints se realice de forma adecuada, Scrum nos brinda diferentes componentes que nos ayudarán a realizar las tareas y asignar responsabilidades para el proyecto.

#### 3.1.1 PRESENTACIÓN DEL EQUIPO DE TRABAJO Y ASIGNACIÓN DE ROLES

El equipo de trabajo y los roles asignados se presentan en la Tabla 3.1:

<b>ROLES</b>	<b>PERSONA ASIGNADA</b>
<i>Product Owner</i>	<i>Padre Juan Carlos Veintimilla</i>
<i>Equipo de desarrollo</i>	<i>Diana Pila</i>
<i>Scrum Master</i>	<i>Ing. Daniel Manangón</i>

Tabla 3.1 Asignación de Roles.<sup>7</sup>

La metodología Scrum permite al cliente participar activamente en el proyecto y lo anima a brindar sus ideas permitiendo obtener un producto de software que tenga el mayor nivel de satisfacción posible.

#### 3.1.2 ELABORACIÓN DEL PRODUCT BACKLOG

Para desarrollar el sistema de administración de las actividades eclesiales de la Iglesia Católica Parroquial Santiago Apóstol es necesario conocer los requerimientos y características del sistema desde el punto de vista del cliente, por lo que se procede a elaborar el Product Backlog, en el cual cada ítem corresponde a un

<sup>7</sup> Tabla de Asignación de Roles elaborada por la autora.

requerimiento del cliente. Se establece la prioridad y la estimación de tiempo en la que se planea entregar cada requerimiento; y en base a esta estimación se define el sprint en el que se realizara cada ítem.

La prioridad la establece el cliente y se basa en una escala del 1 al 5, siendo 5 la menor prioridad y 1 la prioridad más alta.

El modelo utilizado para generar el Product Backlog se muestra en la Figura 3.1.

ID	TÍTULO	DESCRIPCIÓN	SPRINT	PRIORIDAD	ESTIMACIÓN (HORAS)

Figura 3.1 Tabla Modelo para elaborar el Product Backlog.<sup>8</sup>

En la Tabla 3.2 se muestra la versión final del Product Backlog elaborado durante la realización del proyecto.

---

<sup>8</sup> Tabla Modelo elaborada por la autora

ID	TÍTULO	DESCRIPCIÓN	SPRINT	PRIORIDAD	ESTIMACIÓN (HORAS)
1	Registrar Persona	Se registra la información de una persona: nombres, apellidos, dirección, cédula, teléfono, etc.	1	1	12
2	Buscar Personas	Muestra una lista de personas como resultado de una búsqueda. Permite editar y eliminar personas.	1	2	9
3	Crear Usuario	Se crea un usuario para el sistema con los parámetros contraseña, username y se asigna perfiles de usuario según se requiera.	1	2	15
4	Buscar Usuarios	Muestra una lista de usuarios como resultado de una búsqueda por criterio. Permite editar y eliminar usuarios.	1	2	11
5	Editar la información de la iglesia	Permite visualizar, modificar y guardar la información de la Iglesia Católica Parroquial Santiago Apóstol.	1	2	8
6	Asignar a una persona como tutor	Permite agregar o eliminar una persona a la lista de tutores.	2	3	6
7	Asignar a una persona como celebrante	Permite agregar o eliminar una persona a la lista de celebrantes.	2	3	6
8	Registrar Lugares	Permite agregar o eliminar un lugar a la lista de lugares	2	3	6
9	Modificar parámetros	Permite establecer los parámetros destinados a control de aprobación de grupos, asistencia y notas: número de faltas, número de atrasos y mínima nota de aprobación.	2	3	9
10	Registrar grupo	Se registra un grupo seleccionando el tutor, horarios, lugar, fechas.	2	3	15
11	Buscar Grupos	Muestra una lista de los grupos como resultado de una búsqueda por criterio, Permite editar y cancelar grupos.	2	4	11
12	Registrar Inscripción	Se registra la inscripción de una persona a uno de los grupos.	2	3	17
13	Buscar Inscripciones	Muestra una lista de las inscripciones como resultado de una búsqueda por criterio, permite editar o eliminar inscripciones.	2	4	11

14	<b>Listar Mis Grupos</b>	Muestra una lista de los grupos correspondientes al usuario que ingresa con perfil de instructor, que permite ingresar al registro de asistencia y al registro de notas.	2	4	10
15	<b>Registrar Asistencia</b>	Permite registrar la asistencia a los grupos.	2	4	13
16	<b>Registrar Notas</b>	Permite registrar las notas por cada grupo y alumno.	2	4	13
17	<b>Registrar Evento</b>	Se registra un evento seleccionando el lugar, celebrante y horario del evento.	1	1	15
18	<b>Listar Eventos</b>	Muestra una lista de todos los eventos existentes. Permite editar y eliminarlos.	1	2	11
19	<b>Registrar Celebración</b>	Se registra la reservación de la celebración de un determinado evento seleccionando la fecha, el lugar, el horario y los participantes. Al finalizar el evento se ingresa la información del registro civil y del registro parroquial correspondiente al evento.	1	1	17
20	<b>Buscar Celebraciones</b>	Muestra una lista de celebraciones como resultado de una búsqueda por criterio. Permite editar o eliminar celebraciones.	1	2	11
21	<b>Emitir certificados</b>	Permite emitir certificados de bautizo o matrimonio según se solicita.	1	1	13
22	<b>Autenticación de usuario</b>	Permite gestionar el acceso a los diferentes recursos del sistema web por medio de los perfiles asignados, nombre de usuario y contraseña.	2	3	17
23	<b>Generar Reportes</b>	Permite generar reportes de personas, grupos y eventos según el criterio seleccionado.	2	5	26

Tabla 3.2 Product Backlog<sup>9</sup><sup>9</sup> Elaborado por la autora.



### 3.1.3 DEFINICIÓN Y OBJETIVOS DE LOS SPRINTS

Se realiza la primera reunión de Planificación del proyecto, en la que se define la duración de cada sprint en 20 días laborables (sin contar los fines de semana), y se obtiene un estimado de 8 horas diarias de trabajo en las que el equipo de desarrollo se compromete a realizar las tareas asignadas; como resultado se obtienen 2 sprints.

En la tabla 3.4 se muestra la definición y objetivos de cada sprint:

SPRINT	OBJETIVO SPRINT	DURACIÓN ESTIMADA (DÍAS)
1	Desarrollar la gestión de Personas, Eventos, Celebraciones y emisión de Certificados	20
2	Desarrollar la gestión de Grupos, Inscripciones, Configuración General del sistema y generación de reportes	20

Tabla 3.3 Definición y objetivos de cada Sprint<sup>10</sup>

---

<sup>10</sup> Elaborado por la autora.

### 3.1.4 PLANIFICACIÓN DE CADA SPRINT

La reunión de planificación de cada Sprint se la realiza al inicio de los mismos. Se elabora el Sprint Backlog correspondiente, donde se especifica qué ítems del Product Backlog se desarrollarán y las tareas que se realizarán para cumplir con cada uno de éstos. Se utiliza para saber el estado y avance de cada una de las tareas, lo que permite tomar decisiones para conseguir que los objetivos de cada sprint se cumplan en los plazos estimados.

Este documento está compuesto por el Id del ítem del Product Backlog, las tareas de cada ítem, el Id de cada tarea, la duración estimada en horas y el estado.

El modelo utilizado para generar el Product Backlog se muestra en la Figura 3.2.

ID - PB	ITEM	ID	TAREAS	ESTIMACIÓN (HORAS)	ESTADO

Figura 3.2 Tabla Modelo para elaborar el Sprint Backlog

El Sprint Backlog obtenido para el sprint 1 se muestra en la Tabla 3.4.

El Sprint Backlog obtenido para el sprint 2 se muestra en la Tabla 3.5.

ID - PB	ITEM	ID	TAREAS	ESTIMACIÓN (HORAS)	ESTADO
	Definición de la arquitectura del sistema	1	Definición de la arquitectura del sistema.	1	Terminada
	Preparación del ambiente de BDD para el desarrollo	2	Descarga e instalación del gestor de base de datos.	1	Terminada
		3	Diseño del modelo conceptual.	8	Terminada
		4	Implementación del diseño conceptual.	1	Terminada
		5	Descarga e instalación del IDE Eclipse y el marco de trabajo JSF.	1	Terminada
	Preparación IDE	6	Configuración del IDE para el uso de la arquitectura JAVA EE	1	Terminada
		7	Instalación y configuración del servidor de aplicaciones JBoss.	1	Terminada
	Diseño Interfaz Gráfica	8	Diseño de la plantilla de la interfaz gráfica de usuario.	12	Terminada
	Diseño del Diagrama de Clases	9	Diseño del diagrama de clases del modelo de negocio.	12	Terminada
1	Registrar Persona	10	Análisis del modelo conceptual correspondiente a personas.	1	Terminada
		11	Diseño de la interfaz gráfica del registro de una nueva persona.	1	Terminada
		12	Programación de la lógica de negocio correspondiente a personas.	8	Terminada
		13	Pruebas de aceptación.	2	Terminada
2	Buscar Personas	14	Diseño de la interfaz gráfica para la búsqueda y edición de personas.	2	Terminada
		15	Programación de la lógica de negocio correspondiente a la búsqueda y edición de personas.	6	Terminada
		16	Pruebas de aceptación.	1	Terminada

			17	Análisis del modelo conceptual correspondiente a Eventos.	2	Terminada
20	<b>Registrar Evento</b>		18	Diseño de la interfaz gráfica del registro de un nuevo Evento.	2	Terminada
			19	Programación de la lógica de negocio correspondiente a Eventos.	10	Terminada
			20	Pruebas de aceptación.	1	Terminada
21	<b>Listar Eventos</b>		21	Diseño de la interfaz gráfica del listado y edición de eventos.	2	Terminada
			22	Programación de la lógica de negocio correspondiente al listado y edición de eventos.	8	Terminada
			23	Pruebas de aceptación.	1	Terminada
22	<b>Registrar Celebración</b>		24	Análisis del modelo conceptual correspondiente a celebraciones de eventos.	2	Terminada
			25	Diseño de la interfaz gráfica del registro de una nueva celebración.	2	Terminada
			26	Programación de la lógica de negocio correspondiente a Celebraciones.	12	Terminada
			27	Pruebas de aceptación.	1	Terminada
23	<b>Buscar Celebraciones</b>		28	Diseño de la interfaz gráfica para la búsqueda y edición de celebraciones.	2	Terminada
			29	Programación de la lógica de negocio correspondiente a la búsqueda y edición de celebraciones.	8	Terminada
			30	Pruebas de aceptación.	1	Terminada
25	<b>Emitir certificados</b>		31	Análisis del modelo conceptual correspondiente a la emisión de certificados.	2	Terminada
			32	Diseño de la interfaz gráfica correspondiente a la emisión de certificados.	2	Terminada

		33	Programación de la lógica de negocio para la emisión de certificados.	8	Terminada
		34	Pruebas de aceptación.	1	Terminada
		35	Análisis del modelo conceptual correspondiente a Usuarios.	2	Terminada
4	<b>Crear Usuario</b>	36	Diseño de la interfaz gráfica de la creación de un nuevo usuario.	2	Terminada
		37	Programación de la lógica de negocio correspondiente a Usuarios.	10	Terminada
		38	Pruebas de aceptación.	1	Terminada
		39	Diseño de la interfaz gráfica de la búsqueda y edición de usuarios.	2	Terminada
5	<b>Buscar Usuarios</b>	40	Programación de la lógica de negocio correspondiente a la búsqueda y edición de usuarios.	8	Terminada
		41	Pruebas de aceptación.	1	Terminada
		42	Diseño de la interfaz gráfica para el registro y modificación de la información de la iglesia.	2	Terminada
6	<b>Editar la información de la iglesia</b>	43	Programación de la lógica de negocio correspondiente al registro y modificación de la información de la iglesia.	6	Terminada

Tabla 3.4 Sprint Backlog para el Sprint 1 <sup>11</sup>

ID - PB	ITEM	ID	TAREAS	ESTIMACIÓN (HORAS)	ESTADO
7	Asignar a una persona como tutor	1	Diseño de la interfaz gráfica del registro y modificación de la lista de Tutores.	1	Terminada

---

<sup>11</sup> Sprint Backlog elaborado por la autora.

			2	Programación de la lógica de negocio correspondiente al registro y modificación de la información de la lista de Tutores.	4	Terminada
			3	Pruebas de aceptación.	1	Terminada
			4	Diseño de la interfaz gráfica del registro y modificación de la lista de Celebrantes.	1	Terminada
8	<b>Asignar a una persona como celebrante</b>		5	Programación de la lógica de negocio correspondiente al registro y modificación de la información de la lista de Celebrantes.	4	Terminada
			6	Pruebas de aceptación.	1	Terminada
			7	Diseño de la interfaz gráfica del registro y modificación de la lista de Lugares.	1	Terminada
9	<b>Registrar Lugares</b>		8	Programación de la lógica de negocio correspondiente al registro y modificación de la información de la lista de Lugares.	4	Terminada
			9	Pruebas de aceptación.	1	Terminada
			13	Análisis del modelo conceptual correspondiente a los parámetros del sistema.	1	Terminada
10	<b>Modificar parámetros</b>		14	Diseño de la interfaz gráfica para la modificación de los parámetros del sistema.	1	Terminada
			15	Programación de la lógica de negocio correspondiente a parámetros.	6	Terminada
			16	Pruebas de aceptación.	1	Terminada
			17	Análisis del modelo conceptual correspondiente al registro de un grupo.	2	Terminada
11	<b>Registrar grupo</b>		18	Diseño de la interfaz gráfica para el registro de un nuevo grupo.	2	Terminada
			19	Programación de la lógica de negocio del registro de un nuevo grupo.	10	Terminada

		20	Pruebas de aceptación.	1	Terminada
		21	Diseño de la interfaz gráfica de la búsqueda y edición de grupos.	2	Terminada
12	<b>Buscar Grupos</b>	22	Programación de la lógica de negocio correspondiente a la búsqueda y edición de grupos.	8	Terminada
		23	Pruebas de aceptación.	1	Terminada
		24	Análisis del modelo conceptual correspondiente a la inscripción de una persona a un grupo.	2	Terminada
13	<b>Registrar Inscripción</b>	25	Diseño de la interfaz gráfica para el registro de una nueva inscripción.	2	Terminada
		26	Programación de la lógica de negocio del registro de una nueva inscripción.	12	Terminada
		27	Pruebas de aceptación.	1	Terminada
		28	Diseño de la interfaz gráfica de la búsqueda y edición de inscripciones.	2	Terminada
14	<b>Buscar Inscripciones</b>	29	Programación de la lógica de negocio correspondiente a la búsqueda y edición de inscripciones.	8	Terminada
		30	Pruebas de aceptación.	1	Terminada
		31	Diseño de la interfaz gráfica que muestra la lista de grupos del usuario que ingresó al sistema con perfil de Instructor.	1	Terminada
15	<b>Listar Mis Grupos</b>	32	Programación de la lógica de negocio que permite mostrar la lista Mis Grupos.	8	Terminada
		33	Pruebas de aceptación.	1	Terminada
		34	Análisis del modelo conceptual correspondiente al registro de asistencia de un grupo.	2	Terminada
16	<b>Registrar Asistencia</b>	35	Diseño de la interfaz gráfica para el registro de asistencia de un grupo.	2	Terminada
		36	Programación de la lógica de negocio al registro de asistencia de un grupo.	8	Terminada

		37	Pruebas de aceptación.	1	Terminada
		38	Análisis del modelo conceptual correspondiente al registro de notas de un grupo.	2	Terminada
17	<b>Registrar Notas</b>	39	Diseño de la interfaz gráfica para el registro de notas de un grupo.	2	Terminada
		40	Programación de la lógica de negocio del registro de notas de un grupo.	8	Terminada
		41	Pruebas de aceptación.	1	Terminada
		42	Análisis del modelo conceptual para la autenticación de usuarios.	2	Terminada
22	<b>Autenticación de usuario</b>	43	Diseño de la interfaz gráfica para la autenticación de usuarios.	2	Terminada
		44	Programación de la lógica de negocio para la autenticación de usuarios.	12	Terminada
		45	Pruebas de aceptación.	1	Terminada
		46	Análisis del modelo conceptual para la generación de reportes.	3	Terminada
23	<b>Generar Reportes</b>	47	Diseño de la interfaz gráfica para la generación de reportes de Personas, Grupos, Eventos.	6	Terminada
		48	Programación de la lógica de negocio para la generación de reportes.	14	Terminada
		49	Pruebas de aceptación.	3	Terminada

Tabla 3.5 Sprint Backlog para el Sprint 2 <sup>12</sup>

<sup>12</sup> Sprint Backlog elaborado por la autora.



Para dar seguimiento al cumplimiento de las tareas que se encuentran en cada uno de los Sprint Backlogs se realizan revisiones diarias (*Daily Scrum*) en las que se verifica el estado de cada una de las tareas y se planifica el trabajo del equipo para ese día. En cada una de estas reuniones no se presentan novedades de alto impacto y se pone especial énfasis en la cantidad y calidad del trabajo que queda, por lo que las tareas se realizan con normalidad y sin mayores inconvenientes dentro del tiempo establecido.

## **3.2 EJECUCIÓN DE LA ITERACIÓN**

### **3.2.1 DISEÑO**

Esta etapa del desarrollo del proyecto permite tener una visión clara de la implementación del sistema. El diseño se realiza en dos partes, la primera al inicio del proyecto para tener un entendimiento general de lo que se va a implementar y la segunda parte se la realiza durante la ejecución de cada ítem de cada uno de los Sprint Backlog como una revisión al diseño general, por lo que se la considera como una tarea específica.

En las siguientes secciones se describen los resultados del diseño obtenidos al final de la implementación del sistema.

#### **3.2.1.1 Diseño de la Interfaz Gráfica**

Para el diseño y elaboración de la plantilla base que se utiliza en el sistema se tomaron en cuenta aspectos tales como la facilidad del usuario para acceder a cada una de las tareas que realiza el sistema, la navegación entre páginas, que refleje las necesidades y solicitudes del cliente y que se presente en una forma clara y sencilla de usar.

Para esto se define que la plantilla base esté compuesta de la siguiente forma:

La zona denominada *Encabezado* está dividida en dos partes:

- La parte superior, en la que se ubica el logo de la iglesia junto con el nombre del sistema.
- El *Menú Principal* está formado por los enlaces a los módulos en los que se divide el sistema, los mismos que son:
  1. Inicio
  2. Personas
  3. Grupos
  4. Eventos
  5. Certificados
  6. Administración

Estos módulos serán detallados más adelante.

La zona denominada *Menú Lateral* la misma que se divide en dos secciones: la sección *Menú* contiene los enlaces para acceder a las diferentes tareas que se realizan en cada uno de los módulos, y la sección *Ayuda* que muestra consejos para el uso adecuado del sistema.

La zona de *Mensajes* se utiliza para mostrar 3 tipos de mensajes: informativo, de error y de advertencia.

La zona denominada *Contenido* permite visualizar y ejecutar cada tarea que se especifica en el Menú Lateral: visualizar listas, generar reportes, ingresar información, etc.

Por último la zona denominada *Pie de Página* que contiene la información sobre las reglas de uso, presentación de la iglesia y presentación de la autora del sistema.

En la Figura 3.3 se muestra la organización de los elementos que conforman el diseño de la interfaz gráfica.

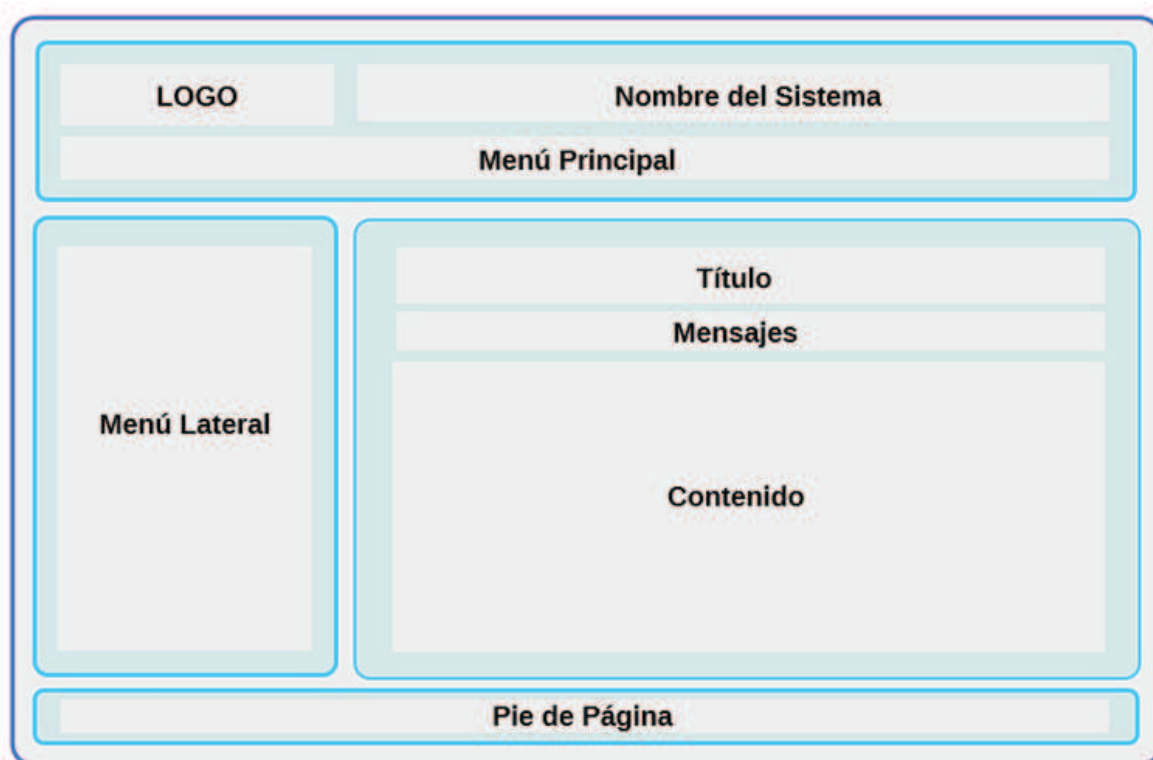


Figura 3.3 Diseño de la plantilla para la interfaz gráfica<sup>13</sup>.

### 3.2.1.2 Módulos del Sistema

A continuación se describen los módulos que conforman el sistema de administración de las actividades eclesiales de la Iglesia Católica Parroquial Santiago Apóstol:

---

<sup>13</sup> Elaborado por la autora.

## **MÓDULO INICIO**

En este módulo se muestra la información del usuario que ingresa al sistema: nombre, apellido, username, y permite cambiar la contraseña ingresando la contraseña actual y la nueva contraseña

## **MÓDULO PERSONAS**

En este módulo se realiza la gestión de Personas. Permite registrar, editar y eliminar personas. Muestra todas las personas registradas o listas de personas según el texto de búsqueda y el criterio seleccionado. Se generan reportes de personas ordenadas alfabéticamente y agrupadas según se requiera.

Está formado por las siguientes secciones:

- Nueva Persona.
- Lista de Personas.
- Reporte de Personas.

## **MÓDULO GRUPOS**

En este módulo se realiza la gestión de los grupos que existen en la iglesia, las inscripciones, registro de asistencia y notas sobre los grupos del usuario que ingresa al sistema con perfil de Instructor y la lista de grupos.

Está formado por las siguientes secciones:

- Nuevo Grupo.
- Lista de Grupos.
- Nueva Inscripción.

- Lista de Inscripciones.
- Mis Grupos.
- Registro de Asistencia.
- Registro de Notas.
- Reporte de Grupos.

## **MODULO EVENTOS**

En este módulo se realiza la gestión tanto de eventos como de celebraciones a determinados eventos. Permite registrar, editar, listar, eliminar eventos y celebraciones, así como generar reportes.

Las secciones de este módulo son:

- Nuevo Evento.
- Lista de Eventos.
- Nueva Celebración.
- Lista de Celebraciones.
- Reporte de Celebraciones.

## **MÓDULO CERTIFICADOS**

Este módulo permite la emisión de certificados de matrimonio y de bautizo.

Está formado por 2 secciones:

- Certificados de Bautizo.
- Certificados de Matrimonio.

## **MÓDULO ADMINISTRACIÓN**

Este módulo está destinado a la Administración del sistema. Permite registrar, modificar, listar y eliminar usuarios, establece los parámetros del sistema para el control de asistencia y notas, y la configuración General del Sistema.

Está formado por las secciones:

- Nuevo Usuario.
- Lista de Usuarios.
- Parámetros
- Configuración General.

La sección de Configuración General está formada por 5 subsecciones:

- Iglesia.
- Tutores.
- Celebrantes.
- Lugares.
- Roles.

### **3.2.1.3 Estructura Jerárquica de la Aplicación Web**

En la Figura 3.4 se muestra la estructura jerárquica de las páginas Web que conforman el sistema, organizadas en los módulos y secciones correspondientes, definidos en base a los ítems del Product Backlog.

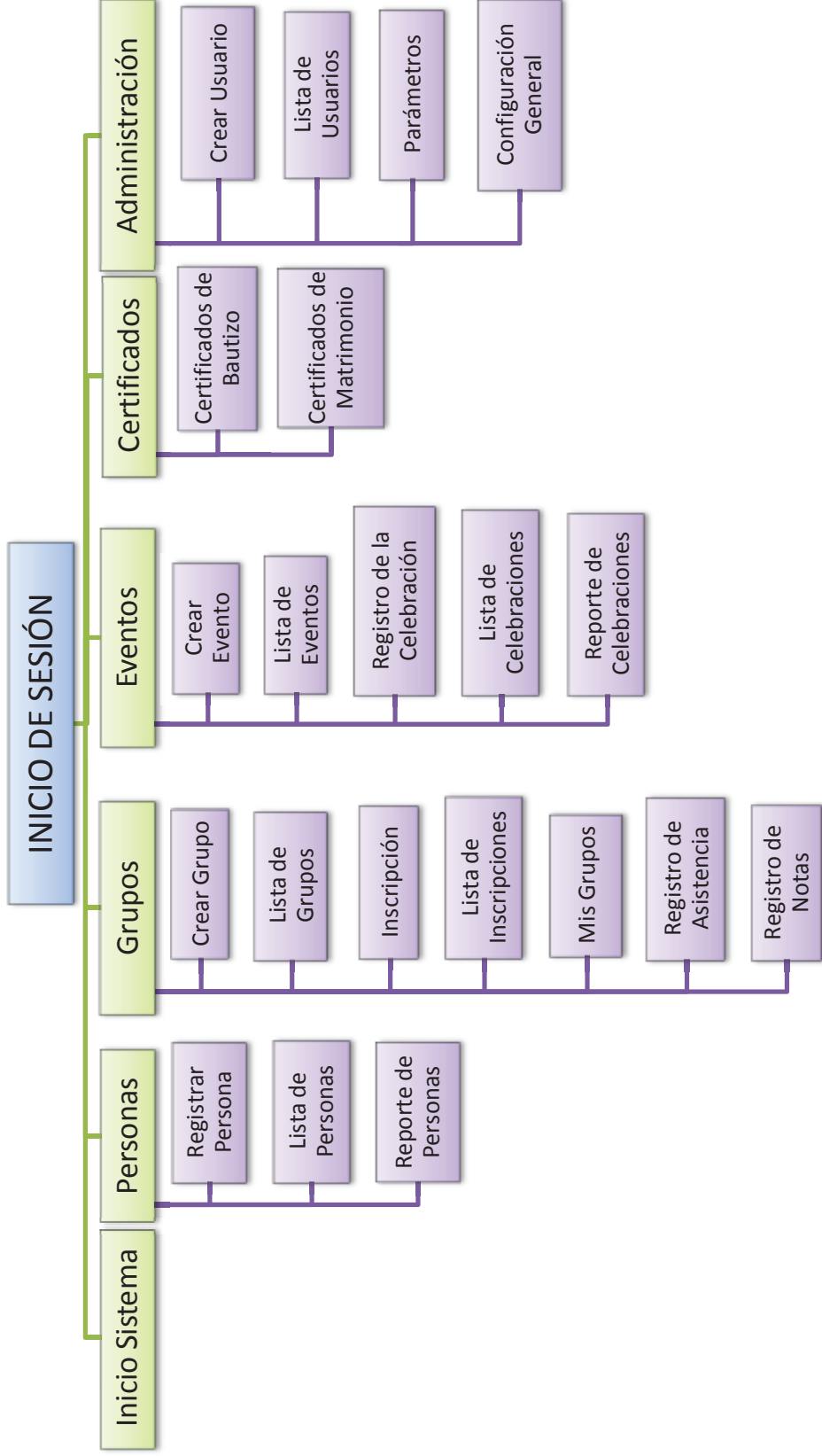


Figura 3.4 Mapa de la Aplicación Web<sup>14</sup>

<sup>14</sup> Elaborado por la autora.

#### **3.2.1.4 Diagrama de Clases**

El diseño del diagrama de clases permite mostrar de forma estática la estructura de la información del sistema, sus clases y relaciones entre clases.

Basado en los requerimientos del usuario que se encuentran en el Product Backlog del proyecto, se realiza el diseño del diagrama de clases y se obtiene la versión final.

Esta versión se muestra en la Figura 3.5.



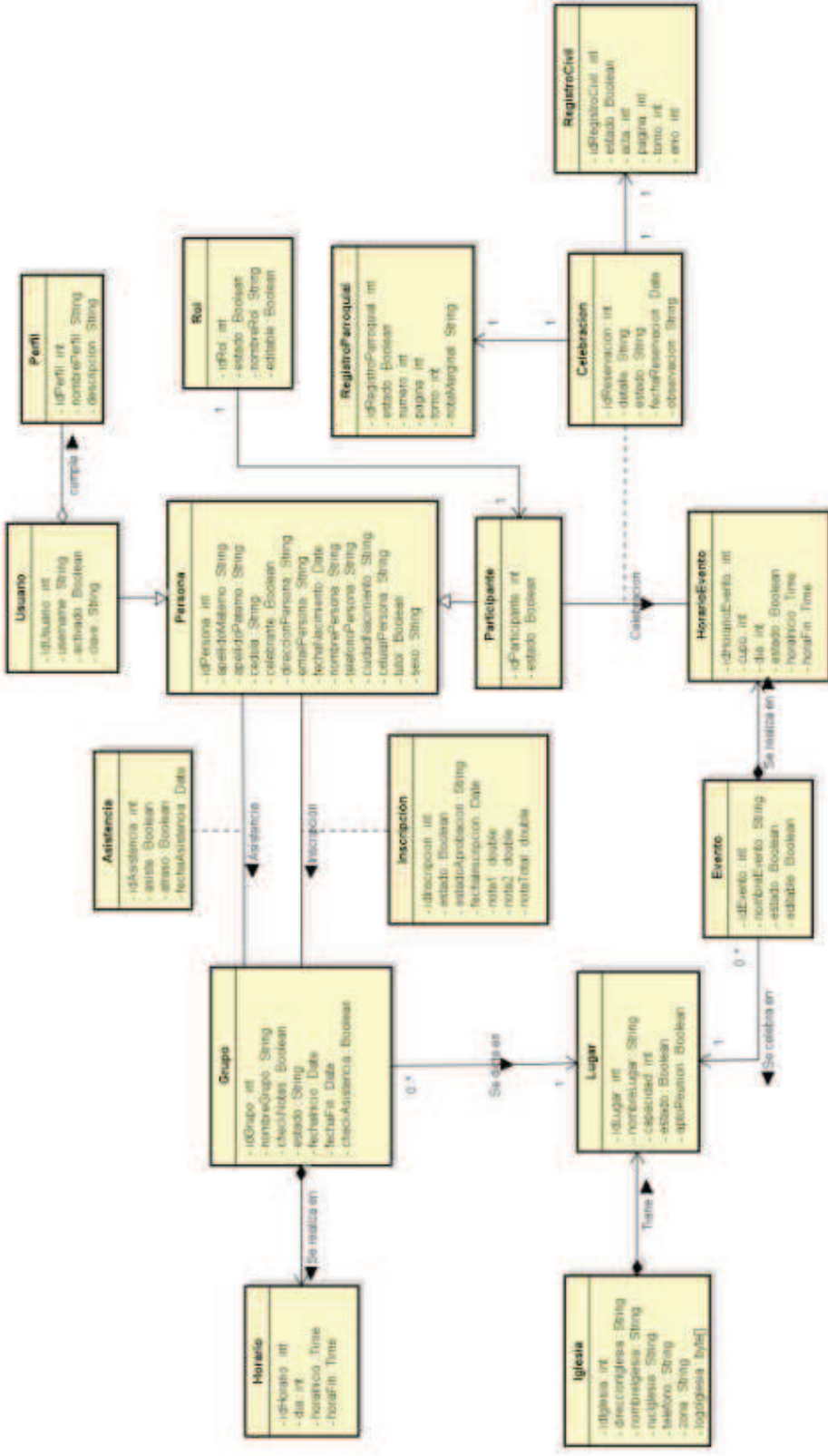


Figura 3.5 Diagrama de Clases<sup>15</sup>

<sup>15</sup> Elaborado por la autora.

### 3.2.1.5 Arquitectura del Sistema

El diseño de la arquitectura del sistema se basa en la arquitectura planteada por la plataforma de desarrollo Java Enterprise Edition (Java EE). Se establecen tres capas: Capa Cliente, Servidor Java EE y Datos.

- **Capa Cliente:** En esta capa se utiliza un navegador web como medio de acceso a la aplicación Web
- **Capa Servidor Java EE:** En esta capa se utiliza el Modelo Vista Controlador (MVC) como patrón de diseño para construir la aplicación Web.
- **Capa de Datos:** En esta capa se utiliza MySQL como motor de base de datos para almacenar la información que maneja el sistema.

## 3.2.2 CONSTRUCCIÓN

### 3.2.2.1 Estructura del Proyecto

Para desarrollar la aplicación se utiliza Eclipse como IDE de desarrollo. En base a la arquitectura planteada se define la estructura de los proyectos que se crean en el IDE. Estos son el proyecto empresarial, EJB y web dinámico.

En la Figura 3.7 se muestran los proyectos creados en el IDE, la organización entre ellos, sus paquetes y su contenido principal.



Figura 3.6 Estructura del Proyecto<sup>16</sup>

<sup>16</sup> Elaborado por la autora

El proyecto EJB contiene el paquete de Entidades y el paquete de Servicios:

- En el paquete de Entidades se encuentran las clases que representan como objetos las tablas de la base de datos.
- En el paquete de Servicios se encuentran las interfaces y las clases java que implementan los métodos que permiten la interacción con las entidades de la base de datos.

El proyecto Web dinámico contiene el paquete de Controladores y la carpeta Web Content.

- En el paquete de Controladores se encuentran las clases que permiten manejar los eventos que se realizan en la interfaz por medio de las etiquetas JSF.
- En la carpeta Web Content se encuentran las páginas XHTML que implementan las interfaces de usuario que componen el sistema.

El proyecto empresarial une los proyectos web y EJB. Como resultado se genera el archivo *sis\_ape\_project.ear*, el cual se utiliza para publicar la aplicación web en el Servidor JEE. Este archivo contiene los resultados generados por cada proyecto individual: *sis\_ape\_ejb.jar* y *sis\_ape.war*

En la Figura 3.7 se muestra la estructura del proyecto en el IDE de desarrollo.

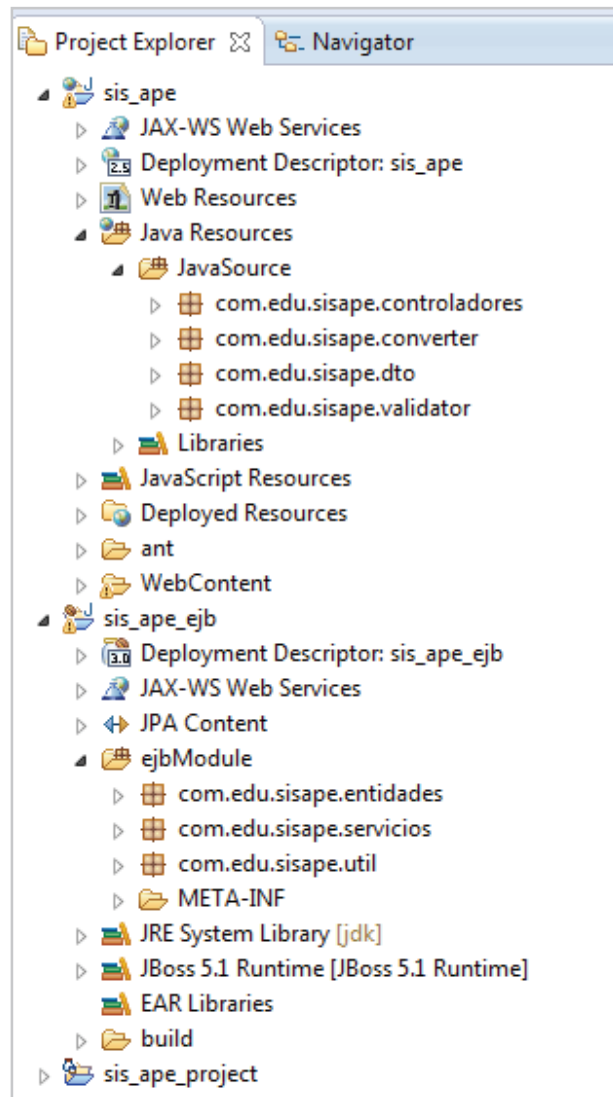


Figura 3.7 Estructura del Proyecto dentro del IDE.

### 3.2.2.2 Codificación

Definida la estructura del proyecto, el equipo de desarrollo puede utilizar estos recursos para codificar las soluciones de cada Sprint Backlog.

Como ejemplo de codificación se toma el ítem Registrar Persona del Sprint Backlog correspondiente al Sprint 1.

## Construcción de la plantilla base

Todas las páginas Web del sistema incluyen una plantilla base que se construye usando código html, estilos css y etiquetas jsf. El código de la plantilla base es el siguiente:

### *template1.xhtml*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:rich="http://richfaces.org/rich"
      xmlns:a4j="http://richfaces.org/a4j">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
  <title>Sistema de Administración Santiago Apóstol</title>
  <a4j:loadStyle src="/style.css"></a4j:loadStyle>
  <a4j:loadScript src="/script.js"></a4j:loadScript>
</head>
<body>
<div id="art-page-background-simple-gradient">
</div>
<div class="art-Sheet">

  <ui:include src="/templates/encabezado.xhtml" />
  <div class="art-contentLayout">
    <div class="art-sidebar1">
      <div class="art-Block">
        <div class="art-Block-tl"></div>
        <div class="art-Block-tr"></div>
        <div class="art-Block-bl"></div>
        <div class="art-Block-br"></div>
        <div class="art-Block-tc"></div>
        <div class="art-Block-bc"></div>
        <div class="art-Block-cl"></div>
        <div class="art-Block-cr"></div>
        <div class="art-Block-cc"></div>
        <div class="art-Block-body">
          <div class="art-BlockHeader">
            <div class="l"></div>
            <div class="r"></div>
            <div class="art-header-tag-icon">
              <div class="t">Menu</div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```



```

value="/images/Info1.png" />
<h:graphicImage
</f:facet>
<f:facet name="warnMarker">
<h:graphicImage
</f:facet>
</rich:messages>
</div>
</div>
<div class="art-PostContent" align="center">
<ui:insert name="contenido" />
</div>
</div>
</div>
</div>
<div class="cleared"></div>
</div>
</div>
</body>
</html>

```

### **encabezado.xhtml**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:fn="http://java.sun.com/jsp/jstl/functions">
<script type="text/javascript" src="/sis_ape/jquery/jquery-
1.4.2.min.js"></script>
<script type="text/javascript">
var $j = jQuery.noConflict();
</script>
<div id="header">
<div class="art-Sheet-tl"></div>
<div class="art-Sheet-tr"></div>
<div class="art-Sheet-bl"></div>
<div class="art-Sheet-br"></div>
<div class="art-Sheet-tc"></div>
<div class="art-Sheet-bc"></div>
<div class="art-Sheet-bc"></div>
<div class="art-Sheet-cl"></div>
<div class="art-Sheet-cr"></div>
<div class="art-Sheet-cc"></div>

```



```

<div class="art-Sheet-body">
  <div id="user">
    <h:form>
      <h:outputText
value="#{sessionControlador.usuario.persona.nombrePersona}"
("#{sessionControlador.usuario.username}")"
rendered="#{sessionControlador.usuario!=null}" />
      <h:commandLink action="#{sessionControlador.Logout}"
value="Salir" rendered="#{sessionControlador.usuario!=null}" />
    </h:form>
  </div>
  <div class="art-Header">
    <div class="art-Header-png"></div>
    <div class="art-Header-jpeg"></div>
    <div class="art-Logo">
      <h1 id="name-text" class="art-Logo-name"><strong>SISTEMA DE
ADMINISTRACIÓN</strong></h1>
      <div id="slogan-text" class="art-Logo-text"><strong><h:outputLabel
value="#{sessionControlador.iglesia.nombreIglesia}" /></strong><strong></strong>
</div>
    </div>
    <p><h:graphicImage value="/images/Logo4.png" width="204" height="143"
/></p>
  </div>
  <div class="art-nav">
    <div class="l"></div>
    <div class="r">

    </div>
    <ul class="art-menu">
      <li><a href="/sis_ape/inicio/inicioSistema.sisape"
class="#{fn:containsIgnoreCase(sessionControlador.path, '/inicio/') ? 'active' :
''}"><span class="l"></span><span class="r"></span><span
class="t">Inicio</span></a></li>
      <h:panelGroup rendered="#{sessionControlador.esAdministrador
|| sessionControlador.esSecretario}">
        <li><a href="/sis_ape/personas/inicioPersona.sisape"
class="#{fn:containsIgnoreCase(sessionControlador.path, 'personas') ? 'active' :
''}"><span class="l"></span><span class="r"></span><span
class="t">Personas</span></a></li>
      </h:panelGroup>
      <h:panelGroup rendered="#{sessionControlador.esInstructor ||
sessionControlador.esSecretario}">
        <li><a href="/sis_ape/grupos/inicioGrupos.sisape"
class="#{fn:containsIgnoreCase(sessionControlador.path, 'grupos') ? 'active' :
''}"><span class="l"></span><span class="r"></span><span
class="t">Grupos</span></a></li>
      </h:panelGroup>
      <h:panelGroup rendered="#{sessionControlador.esSecretario}">
        <li><a href="/sis_ape/eventos/inicioEventos.sisape"
class="#{fn:containsIgnoreCase(sessionControlador.path, 'eventos') ? 'active' :
''}"><span class="l"></span><span class="r"></span><span
class="t">Eventos</span></a></li>
      </h:panelGroup>
    </ul>
  </div>

```

```

                <h:panelGroup rendered="#{sessionControlador.esSecretario}">
                    <li><a
href="/sis_ape/certificados/inicioCertificados.sisape"
class="#{fn:containsIgnoreCase(sessionControlador.path, 'certificados') ? 'active'
: ''}"><span class="l"></span><span class="r"></span><span
class="t">Certificados</span></a></li>
                </h:panelGroup>
                <h:panelGroup
rendered="#{sessionControlador.esAdministrador}">
                    <li><a
href="/sis_ape/administracion/inicioAdministracion.sisape"
class="#{fn:containsIgnoreCase(sessionControlador.path, 'administracion') ?
'active' : ''}"><span class="l"></span><span class="r"></span><span
class="t">Administración</span></a></li>
                </h:panelGroup>
            </ul>
        </div>
    </div>
</div>
</ui:composition>

```

## Construcción de la interfaz gráfica para el registro de una nueva Persona

Se crea la página nuevaPersona.xhtml y se construye la interfaz gráfica utilizando html, y etiquetas JSF.

El código generado para esta página es el siguiente:

### *persona.xhtml*

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/templates/template1.xhtml">
    <ui:define name="titulo">Registrar Persona</ui:define>
    <ui:define name="menu">
        <ui:include src="/templates/menuPersona.xhtml" />
    </ui:define>
    <ui:define name="contenido">
        <ui:include src="/personas/nuevaPersona.xhtml" />
    </ui:define>

```

```

    <ui:define name="ayuda">
        <h:outputLabel value="#{ayudaBundle['ayuda.calendario']}" />
        <h:graphicImage value="/images/calendario.JPG" style="margin-bottom:
-3px;" />
        <h:outputLabel value="#{ayudaBundle['ayuda.camposObligatorios']}" />
    </ui:define>
</ui:composition>
</html>

```

### **menuPersona.xhtml**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:ui="http://java.sun.com/jsf/facelets">

<ui:composition>
<h:form>
    <p><b>Personas</b></p>
    <ul >
        <li type="disc"><h:commandLink value="Nueva Persona"
action="#{personaControlador.irNuevaPersona}" /></li>
        <li type="disc"><h:commandLink value="Buscar Personas"
action="verListaPersonas" /></li>
    </ul>
    <p><b>Reportes</b></p>
    <ul>
        <li type="disc"><h:commandLink value="Reporte de Personas"
action="irReportePersonas" /></li>
    </ul>
</h:form>
</ui:composition>
</html>

```

### **nuevaPersona.xhtml**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:ui="http://java.sun.com/jsf/facelets">

```

```

<ui:composition>
  <script src="/sis_ape/jquery/jquery.maskedinput-1.3.js"
type="text/javascript"></script>
  <script src="/sis_ape/jquery/livevalidation_standalone.js"
type="text/javascript"></script>
  <script>
    $j = jQuery.noConflict();
    $j(document).ready(function(){
      $j("#fono input").mask("(99) 9999-999");
    });
  </script>
  <h:form id="form_persona" acceptcharset="utf-8">
    <h:panelGrid columns="2">
      <h:outputLabel id="lbl_cedula" value="Cédula: " />
      <h:panelGroup>
        <h:inputText id="txt_cedula"
value="#{personaControlador.persona.cedula}" maxlength="10"
validator="numberValidator"/>
        <script type="text/javascript">
          var cedula = new
LiveValidation('form_persona:txt_cedula');
          cedula.add(Validate.Length, { is: 10,
wrongLengthMessage: "Deben ser 10 caracteres!" } );
        </script>
        <rich:message for="txt_cedula" showSummary="true" />
      </h:panelGroup>
      <h:outputLabel id="lbl_nombre" value="Nombre: " />
      <h:inputText id="txt_nombre"
value="#{personaControlador.persona.nombrePersona}" />
      <h:outputLabel id="lbl_apaterno" value="Apellido Paterno: " />
      <h:inputText id="txt_apaterno"
value="#{personaControlador.persona.apellidoPaterno}" />
      <h:outputLabel id="lbl_amaterno" value="Apellido Materno: " />
      <h:inputText id="txt_amaterno"
value="#{personaControlador.persona.apellidoMaterno}" />
      <h:outputLabel id="lbl_fecha_nac" value="Fecha de Nacimiento: "
/>
      <rich:calendar id="cal_fecha_nac"
value="#{personaControlador.persona.fechaNacimiento}"></rich:calendar>
      <h:outputLabel id="lbl_ciudad_nac" value="Ciudad de Nacimiento:
" />
      <h:inputText id="txt_ciudad_nac"
value="#{personaControlador.persona.ciudadNacimiento}"></h:inputText>
      <h:outputLabel id="lbl_sexo" value="Sexo: " />
      <h:selectOneRadio id="grupo_genero"
value="#{personaControlador.persona.sexo}">
        <f:selectItem id="rb_masculino" itemLabel="Masculino"
itemValue="m" />
        <f:selectItem id="rb_femenino" itemLabel="Femenino"
itemValue="f"/>
      </h:selectOneRadio>
      <h:outputLabel id="lbl_direccion" value="Dirección: " />

```

```

        <h:inputTextarea id="txta_direcc"
value="#{personaControlador.persona.direccionPersona}" style=" width: 195p px" />
        <h:outputLabel id="lbl_email" value="Email: " />
        <h:inputText id="txt_email"
value="#{personaControlador.persona.emailPersona}" />
        <h:outputLabel id="lbl_telef" value="Teléfono: " />
        <h:panelGroup>
        <span id="fono">
            <h:inputText id="txt_telef"
value="#{personaControlador.persona.telefonoPersona}" />
            <rich:message for="txt_telef" showSummary="true" />
        </span>
        </h:panelGroup>
    </h:panelGrid>
    <div id="botones" align="center" >
        <a4j:commandButton styleClass="btn_guardar"
action="#{personaControlador.guardarPersona}" value="Guardar" />
    </div>
</h:form>
</ui:composition>
</html>

```

En la Figura 3.8 se muestra la interfaz gráfica terminada que permite registrar Personas en el sistema.



Figura 3.8 Interfaz gráfica para el registro de Personas.

## Programación de la lógica del negocio correspondiente a Personas.

La implementación de la lógica del negocio en el sistema se codifica en las clases controladoras, las mismas que utilizan los métodos de los servicios del proyecto EJB.

Para el ejemplo se emplea la clase controladora *PersonaControlador.java*, la interfaz *ServicioPersonaLocal.java* y la clase *ServicioPersona.java* para modificar, ingresar o eliminar la información correspondiente a Personas en la base de datos.

### *PersonaControlador.java*

```
package com.edu.sisape.controladores;

import java.util.ArrayList;
import java.util.List;
import javax.ejb.EJB;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import com.edu.sisape.entidades.Persona;
import com.edu.sisape.servicios.ServicioPersonaLocal;

public class PersonaControlador {

    private Persona persona;
    private String textoBusqueda;
    private String criterioBusqueda;
    private List<Persona> listaPersonas;

    @EJB
    private ServicioPersonaLocal servicioPersona;

    public void setPersona(Persona persona) {
        this.persona = persona;
    }

    public String getTextoBusqueda() {
        return textoBusqueda;
    }

    public void setTextoBusqueda(String textoBusqueda) {
        this.textoBusqueda = textoBusqueda;
    }

    public String getCriterioBusqueda() {
```

```

        return criterioBusqueda;
    }

    public void setCriterioBusqueda(String criterioBusqueda) {
        this.criterioBusqueda = criterioBusqueda;
    }

    public List<Persona> getListaPersonas() {
        return listaPersonas;
    }

    public void setListaPersonas(List<Persona> listaPersonas) {
        this.listaPersonas = listaPersonas;
    }

    public String irNuevaPersona()
    {
        crearNuevaPersona();
        return "irNuevaPersona";
    }

    public void crearNuevaPersona()
    {
        persona = new Persona();
        persona.setEstado(true);
        persona.setCelebrante(false);
        persona.setTutor(false);
    }

    public Persona getPersona() {
        return persona;
    }

    public void guardarPersona()
    {
        int contador = 0;
        if(persona.getTelefonoPersona().trim().isEmpty())
            contador++;
        if(persona.getCelularPersona().trim().isEmpty())
            contador++;
        if(contador>1)
        {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_ERROR, "Debe ingresar al menos un teléfono
convencional o celular.", ""));
            return;
        }
        if(persona.getCedula()==null || persona.getCedula().trim().isEmpty())
        {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_ERROR, "Debe ingresar el número de cédula.",
""));
            return;
        }
    }

```

```

    }
    List<Persona> personasMismaCedula =
servicioPersona.consultarPersonasPorCedula(persona.getCedula(),
persona.getIdPersona());
    if(personasMismaCedula.size(>0)
    {
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_ERROR, "Ya existe una persona registrada con
esta cédula", ""));
        return;
    }

    try {
        persona = servicioPersona.guardarPersona(persona);
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "La información Personal ha sido
ingresada con éxito", ""));
    } catch (Exception e) {
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_ERROR, "La información no pudo ser guardada.
"+e.getMessage(), ""));
    }
}

public void buscarPersona()
{
    if (textoBusqueda == null || textoBusqueda.trim().isEmpty())
    {
        listaPersonas = servicioPersona.consultarPersonas();
    }
    if(textoBusqueda != null && criterioBusqueda.equals("nombre"))
    {
        listaPersonas =
servicioPersona.consultarPersonasPorNombre(textoBusqueda);
    }
    if(textoBusqueda != null && criterioBusqueda.equals("cedula"))
    {
        listaPersonas =
servicioPersona.consultarPersonasPorCedula(textoBusqueda);
    }
    if(textoBusqueda != null && criterioBusqueda.equals("apellido"))
    {
        listaPersonas =
servicioPersona.consultarPersonasPorApellido(textoBusqueda);
    }
    if (listaPersonas.size()==0)
    {
        listaPersonas = new ArrayList<Persona>();
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "No se encontró ninguna Persona
".concat(criterioBusqueda!="null?"con el criterio "+
criterioBusqueda.toUpperCase():""), ""));
    }
}
}

```



```

    public void eliminarPersona()
    {
        if(persona.getIdPersona()==1)
        {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "Esta persona no puede ser eliminada del
sistema", ""));
            return;
        }
        try {
            persona.setEstado(false);
            servicioPersona.guardarPersona(persona);
            listaPersonas.remove(persona);
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "Persona Eliminada. La lista de Personas
ha sido actualizada.", ""));
        } catch (Exception e) {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_ERROR, "La lista de Personas no pudo ser
actualizada"+e.getMessage(), ""));
        }
    }

    public String irEditarPersona()
    {
        return "irEditarPersona";
    }

    public String salir()
    {
        return "irInicioPersona";
    }
}

```

### ***ServicioPersonaLocal.java***

```

package com.edu.sisape.servicios;

import java.util.List;
import javax.ejb.Local;
import com.edu.sisape.entidades.Persona;

@Local
public interface ServicioPersonaLocal {

    public Persona guardarPersona(Persona persona) throws Exception;

    public List<Persona> consultarPersonas();

    public List<Persona> consultarPersonasPorNombre(String textobusqueda);

    public List<Persona> consultarTutores();
}

```

```

    public List<Persona> consultarPersonasPorCedula(String cedula);

    public List<Persona> consultarPersonasPorCedula(String cedula, int
idPersona);

    public List<Persona> consultarPersonasPorApellido(String apellido);

    public List<Persona> consultarCelebrantes();

    public List<Persona> consultarPersonaPorNombre(String nombre, Boolean
esTutor, Boolean esCelebrante);
}

```

### ***ServicioPersona.java***

```

package com.edu.sisape.servicios;

import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;
import com.edu.sisape.entidades.Persona;

/**
 * Session Bean implementation class ServicioPersona
 */
@Stateless
public class ServicioPersona implements ServicioPersonaLocal {

    @PersistenceContext
    private EntityManager em;

    public ServicioPersona() {
        // TODO Auto-generated constructor stub
    }

    public Persona guardarPersona(Persona persona) throws Exception {
        if(persona.getNombrePersona()==null ||
persona.getNombrePersona().trim().isEmpty())
            throw new Exception("Debe ingresar el nombre de la persona");
        if(persona.getApellidoPaterno()==null ||
persona.getApellidoPaterno().trim().isEmpty())
            throw new Exception("Debe ingresar el apellido paterno de la
persona");
        if(persona.getApellidoMaterno()==null ||
persona.getApellidoMaterno().trim().isEmpty())
            throw new Exception("Debe ingresar el apellido materno de la
persona");
        if(persona.getFechaNacimiento()==null )

```

```

        throw new Exception("Debe ingresar la fecha de nacimiento de la
persona");
        if(persona.getCiudadNacimiento()==null ||
persona.getCiudadNacimiento().trim().isEmpty())
            throw new Exception("Debe ingresar la ciudad de nacimiento de
la persona");
        if(persona.getSexo()==null || persona.getSexo().trim().isEmpty())
            throw new Exception("Debe seleccionar el sexo de la persona");
        return em.merge(persona);
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarPersonas()
    {
        Query consulta = em.createQuery("from Persona p where p.estado = 1
order by p.apellidoPaterno ");
        List<Persona> listaPersonas = consulta.getResultList();
        return listaPersonas;
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarPersonasPorNombre(String textobusqueda) {
        Query consulta = em.createQuery("from Persona p where p.nombrePersona
like ? and p.estado = 1 ");
        consulta.setParameter(1, textobusqueda+"%");
        List<Persona> resultado = consulta.getResultList();
        return resultado;
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarTutores()
    {
        Query consulta = em.createQuery("from Persona p where p.tutor = 1 and
p.estado = 1 ");
        List<Persona> resultadoTutores = consulta.getResultList();
        return resultadoTutores;
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarPersonasPorCedula(String cedula, int
idPersona) {
        Query consulta = em.createQuery("from Persona p where p.cedula like ?
and p.estado = 1 and p.idPersona <> ? ");
        consulta.setParameter(1, cedula+"%");
        consulta.setParameter(2, idPersona);
        List<Persona> resultadoPorCedula = consulta.getResultList();
        return resultadoPorCedula;
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarPersonasPorApellido(String apellido) {
        Query consulta = em.createQuery("from Persona p where
p.apellidoPaterno like ? and p.estado = 1 ");
        consulta.setParameter(1, apellido+"%");
    }

```

```

        List<Persona> resultadoPorApellido = consulta.getResultList();
        return resultadoPorApellido;
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarCelebrantes() {
        Query consulta = em.createQuery("from Persona p where p.celebrante =
1 and p.estado = 1 ");
        List<Persona> resultadoCelebrantes = consulta.getResultList();
        return resultadoCelebrantes;
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarPersonaPorNombre(String nombre, Boolean
esTutor, Boolean esCelebrante) {
        String sql="from Persona p where (p.apellidoPaterno like :apePadre "
+ "or p.apellidoMaterno like :apeMadre or p.nombrePersona like :nombres) " + "and
p.estado = 1 ";
        if(esTutor!=null){
            sql+="and p.tutor = :tutor ";
        }
        if(esCelebrante!=null){
            sql+="and p.celebrante = :celebrante ";
        }
        sql+="order by p.apellidoPaterno ";

        Query query=em.createQuery(sql);
        query.setParameter("apePadre", nombre+"%");
        query.setParameter("apeMadre", nombre+"%");
        query.setParameter("nombres", nombre+"%");

        if(esTutor!=null){
            query.setParameter("tutor", esTutor);
        }
        if(esCelebrante!=null){
            query.setParameter("celebrante", esCelebrante);
        }
        List<Persona> resultado=query.getResultList();
        if(resultado.size()==0){
            return null;
        }
        return resultado;
    }

    @SuppressWarnings("unchecked")
    public List<Persona> consultarPersonasPorCedula(String cedula) {
        Query consulta = em.createQuery("from Persona p where p.cedula like ?
and p.estado = 1 ");
        consulta.setParameter(1, cedula+"%");
        List<Persona> resultadoPorCedula = consulta.getResultList();
        return resultadoPorCedula;
    }
}

```

### 3.2.3 PRUEBAS

Las pruebas realizadas permiten verificar que la aplicación cumpla con los objetivos planteados y que satisfaga los requerimientos del usuario.

#### 3.2.3.1 Pruebas de Aceptación

Las pruebas de aceptación se realizan como actividad de cada ítem del Sprint Backlog para validar que cumple con el funcionamiento esperado y permite al usuario determinar su aceptación.

Las pruebas de aceptación son definidas y ejecutadas por el cliente, a continuación se muestran los resultados finales.

### *SPRINT 1*

Prueba de Aceptación	
<b>Código</b>	<b>1</b>
<b>Nombre de la Prueba:</b>	Prueba de Registro de Persona
<b>Ítem Correspondiente:</b>	Registrar Persona
<b>Descripción:</b>	Se ingresan la información de una persona: nombres, apellidos, cédula, fecha de nacimiento, sexo, teléfono, celular, dirección y correo electrónico
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Personas</li> <li>3. Ir a la opción Nueva Persona</li> <li>4. Ingresar datos de la persona</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	La información de la persona se registra correctamente en el sistema y se muestra un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se muestra un mensaje que confirma el registro de la información de la persona en el sistema.
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.6 Prueba de Aceptación Registro de Persona

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>2</b>
<b>Nombre de la Prueba:</b>	Prueba de Búsqueda de Personas
<b>Ítem Correspondiente:</b>	Buscar Personas
<b>Descripción:</b>	Se realiza la búsqueda de una persona por criterio y por texto de búsqueda y se muestra una lista de resultados.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Personas</li> <li>3. Ir a la opción Buscar Personas</li> <li>4. Ingresar el texto de búsqueda.</li> <li>5. Seleccionar el criterio de búsqueda.</li> <li>6. Clic en el botón Buscar</li> </ol>
<b>Resultados Esperados:</b>	<p>Mostrar una lista de personas de acuerdo al texto de búsqueda y al criterio seleccionado.</p> <p>Cada elemento de la lista tiene un botón que permite eliminar la información de la persona del sistema.</p> <p>Cada elemento de la lista tiene un botón que permite editar la información de la persona.</p>
<b>Resultados Obtenidos:</b>	<p>Se muestra la lista de personas.</p> <p>Se elimina la información de la persona del sistema.</p> <p>Se muestra la página de edición con la información de la persona.</p>
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.7 Prueba de Aceptación Búsqueda de Personas**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>3</b>
<b>Nombre de la Prueba:</b>	Prueba de registro de Eventos
<b>Ítem Correspondiente:</b>	Registrar Evento
<b>Descripción:</b>	Se ingresa la información del evento: nombre, lugar, día, horario y celebrante.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Eventos</li> <li>3. Ir a la opción Nuevo Evento</li> <li>4. Ingresar los datos del evento</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	La información del evento se registra correctamente en el sistema y se muestra un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se muestra un mensaje que confirma el registro de la información del evento en el sistema.
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.8 Prueba de Aceptación Registro de Eventos**

Prueba de Aceptación	
<b>Código</b>	4
<b>Nombre de la Prueba:</b>	Prueba de Listado de Eventos
<b>Ítem Correspondiente:</b>	Listar Eventos
<b>Descripción:</b>	Se muestra una lista con todos los eventos existentes.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Eventos</li> <li>3. Ir a la opción Lista de Eventos</li> </ol>
<b>Resultados Esperados:</b>	<p>Mostrar una lista con todos los eventos registrados en el sistema. Cada elemento de la lista tiene un botón que permite eliminar la información del evento del sistema. Cada elemento de la lista tiene un botón que permite editar la información del evento.</p>
<b>Resultados Obtenidos:</b>	<p>Se muestra la lista de eventos. Se elimina la información del evento. Se muestra la página de edición con la información del evento.</p>
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.9 Prueba de Aceptación Listado de Eventos

Prueba de Aceptación	
<b>Código</b>	5
<b>Nombre de la Prueba:</b>	Prueba de registro de Celebración
<b>Ítem Correspondiente:</b>	Registrar Celebración
<b>Descripción:</b>	Se ingresa la información de la celebración: fecha, nombre del evento, lugar, horario, celebrante, detalle, observación, participantes.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Eventos</li> <li>3. Ir a la opción Nueva Celebración</li> <li>4. Ingresar los datos de la celebración</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	La información de la celebración se registra correctamente en el sistema y se muestra un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se muestra un mensaje que confirma el registro de la información de la celebración en el sistema.
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.10 Prueba de Aceptación Registro de Celebración

Prueba de Aceptación	
<b>Código</b>	6
<b>Nombre de la Prueba:</b>	Prueba de Búsqueda de Celebraciones
<b>Ítem Correspondiente:</b>	Buscar Celebraciones
<b>Descripción:</b>	Se realiza la búsqueda de una celebración por criterio y por texto de búsqueda y se muestra una lista de resultados.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Eventos</li> <li>3. Ir a la opción Lista de Eventos</li> </ol>
<b>Resultados Esperados:</b>	<p>Mostrar una lista de celebraciones de acuerdo al texto de búsqueda y al criterio seleccionado.</p> <p>Cada elemento de la lista tiene un botón que permite eliminar la información de la celebración.</p> <p>Cada elemento de la lista tiene un botón que permite editar la información de la celebración.</p>
<b>Resultados Obtenidos:</b>	<p>Se muestra la lista de celebraciones.</p> <p>Se elimina la información de la celebración del sistema.</p> <p>Se muestra la página de edición con la información de la celebración.</p>
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.11 Prueba de Aceptación Búsqueda de Celebraciones

Prueba de Aceptación	
<b>Código</b>	7
<b>Nombre de la Prueba:</b>	Prueba de Emisión de Certificados
<b>Ítem Correspondiente:</b>	Emitir Certificados
<b>Descripción:</b>	Se emiten certificados de bautizo y de matrimonio.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Certificados</li> <li>3. Seleccionar el tipo de certificado</li> <li>4. Ingresar el nombre de la persona</li> <li>5. Clic en el botón Buscar</li> <li>6. Clic en el botón Emitir Certificado</li> </ol>
<b>Resultados Esperados:</b>	<p>Mostrar la lista de las personas que coinciden con el texto de búsqueda.</p> <p>Emitir el certificado correspondiente.</p>
<b>Resultados Obtenidos:</b>	<p>Se muestra la lista de personas</p> <p>Se emite el certificado.</p>
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.12 Prueba de Aceptación Emisión de Certificados



Prueba de Aceptación	
<b>Código</b>	8
<b>Nombre de la Prueba:</b>	Prueba de Creación de Usuarios
<b>Ítem Correspondiente:</b>	Crear Usuario
<b>Descripción:</b>	Se crea un usuario para el ingreso al sistema. Se registran los datos del usuario: nombre, username, contraseña y perfil.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Administración</li> <li>3. Ir a la opción Nuevo Usuario</li> <li>4. Ingresar los datos del usuario.</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	La información del usuario se registra correctamente en el sistema y se muestra un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se muestra un mensaje que confirma el registro de la información del usuario en el sistema.
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.13 Prueba de Aceptación Creación de Usuarios.

Prueba de Aceptación	
<b>Código</b>	9
<b>Nombre de la Prueba:</b>	Prueba de Búsqueda de Usuarios
<b>Ítem Correspondiente:</b>	Buscar Usuarios
<b>Descripción:</b>	Se realiza la búsqueda de un usuario por criterio y por texto de búsqueda y se muestra una lista de resultados.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Administración</li> <li>3. Ir a la opción Buscar Usuarios</li> <li>4. Ingresar el texto de búsqueda</li> <li>5. Seleccionar el criterio de búsqueda</li> <li>6. Clic en el botón Buscar</li> </ol>
<b>Resultados Esperados:</b>	<p>Mostrar una lista de usuarios de acuerdo al texto de búsqueda y al criterio seleccionado.</p> <p>Cada elemento de la lista tiene un botón que permite eliminar la información del usuario del sistema.</p> <p>Cada elemento de la lista tiene un botón que permite editar la información del usuario.</p>
<b>Resultados Obtenidos:</b>	<p>Se muestra la lista de usuarios.</p> <p>Se elimina la información del usuario del sistema.</p> <p>Se muestra la página de edición con la información del usuario.</p>
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.14 Prueba de Aceptación Búsqueda de Usuarios.

**SPRINT 2**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>10</b>
<b>Nombre de la Prueba:</b>	Prueba de Asignación de Personas como Tutores
<b>Ítem Correspondiente:</b>	Asignar a una persona como tutor
<b>Descripción:</b>	Se agrega una persona a la lista de Tutores
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Administración</li> <li>3. Ir a la opción Configuración General</li> <li>4. Ir a la pestaña Tutores</li> <li>5. Clic en el botón Agregar</li> <li>6. Seleccionar una persona</li> <li>7. Clic en Guardar</li> </ol>
<b>Resultados Esperados:</b>	Agregar una persona a la lista de Tutores y mostrar un mensaje informativo
<b>Resultados Obtenidos:</b>	Se agrega la persona a la lista y se muestra el mensaje
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.15 Prueba de Aceptación Asignación de Personas como Tutores.**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>11</b>
<b>Nombre de la Prueba:</b>	Prueba de Asignación de Personas como Celebrantes
<b>Ítem Correspondiente:</b>	Asignar a una persona como celebrante
<b>Descripción:</b>	Se agrega una persona a la lista de Celebrantes
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Administración</li> <li>3. Ir a la opción Configuración General</li> <li>4. Ir a la pestaña Celebrantes</li> <li>5. Clic en el botón Agregar</li> <li>6. Seleccionar una persona</li> <li>7. Clic en Guardar</li> </ol>
<b>Resultados Esperados:</b>	Agregar una persona a la lista de Celebrantes y mostrar un mensaje informativo
<b>Resultados Obtenidos:</b>	Se agrega la persona a la lista y se muestra el mensaje
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.16 Prueba de Aceptación Asignación de Personas como Celebrantes.**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>12</b>
<b>Nombre de la Prueba:</b>	Prueba de Registro de Lugares
<b>Ítem Correspondiente:</b>	Registrar Lugares
<b>Descripción:</b>	Se agrega un lugar a la lista de Lugares
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Administración</li> <li>3. Ir a la opción Configuración General</li> <li>4. Ir a la pestaña Lugares</li> <li>5. Clic en el botón Agregar</li> <li>6. Ingresar los datos del lugar</li> <li>7. Clic en Guardar</li> </ol>
<b>Resultados Esperados:</b>	Agregar un lugar a la lista de Lugares y mostrar un mensaje informativo
<b>Resultados Obtenidos:</b>	Se agrega el lugar a la lista y se muestra el mensaje
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.17 Prueba de Aceptación Registro de Lugares.**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>13</b>
<b>Nombre de la Prueba:</b>	Prueba de Modificación de Parámetros
<b>Ítem Correspondiente:</b>	Modificar Parámetros
<b>Descripción:</b>	Se modifican los parámetros del sistema correspondientes a número de faltas, número de atrasos y nota mínima de aprobación
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Administración</li> <li>3. Ir a la opción Parámetros</li> <li>4. Ingresar los nuevos valores de cada parámetro</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	Valores de cada parámetro actualizados y mostrar un mensaje informativo
<b>Resultados Obtenidos:</b>	Se actualizan los valores y se muestra el mensaje
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.18 Prueba de Aceptación Modificación de Parámetros.**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>14</b>
<b>Nombre de la Prueba:</b>	Prueba de Registro de Grupos
<b>Ítem Correspondiente:</b>	Registrar grupo
<b>Descripción:</b>	Se registran los datos del grupo: nombre, tutor, horario, lugar, fechas de inicio y fin
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Grupos</li> <li>3. Ir a la opción Nuevo Grupo</li> <li>4. Ingresar los datos del grupo</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	La información del grupo se registra correctamente en el sistema y se muestra un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se muestra un mensaje que confirma el registro de la información del grupo en el sistema.
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.19 Prueba de Aceptación Registro de Grupos.**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>15</b>
<b>Nombre de la Prueba:</b>	Prueba de Búsqueda de Grupos
<b>Ítem Correspondiente:</b>	Buscar Grupos
<b>Descripción:</b>	Se realiza la búsqueda de un grupo por criterio y por texto de búsqueda y se muestra una lista de resultados.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Grupos</li> <li>3. Ir a la opción Buscar Grupos</li> <li>4. Ingresar el texto de búsqueda</li> <li>5. Seleccionar el criterio de búsqueda</li> <li>6. Clic en el botón Buscar</li> </ol>
<b>Resultados Esperados:</b>	<p>Mostrar una lista de grupos de acuerdo al texto de búsqueda y al criterio seleccionado.</p> <p>Cada elemento de la lista tiene un botón que permite eliminar la información del grupo del sistema.</p> <p>Cada elemento de la lista tiene un botón que permite editar la información del grupo.</p>
<b>Resultados Obtenidos:</b>	<p>Se muestra la lista de grupos.</p> <p>Se elimina la información del grupo.</p> <p>Se muestra la página de edición con la información del grupo.</p>
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.20 Prueba de Aceptación Búsqueda de Grupos.**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>16</b>
<b>Nombre de la Prueba:</b>	Prueba de Registro de Inscripciones
<b>Ítem Correspondiente:</b>	Registrar Inscripción
<b>Descripción:</b>	Se registra la inscripción de una persona a uno de los grupos
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Grupos</li> <li>3. Ir a la opción Nueva Inscripción</li> <li>4. Ingresar los datos para la inscripción</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	La inscripción se registra correctamente en el sistema y se muestra un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se muestra un mensaje que confirma el registro de la inscripción en el sistema.
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.21 Prueba de Aceptación Registro de Inscripciones.**

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>17</b>
<b>Nombre de la Prueba:</b>	Prueba de Búsqueda de Inscripciones
<b>Ítem Correspondiente:</b>	Buscar Inscripciones
<b>Descripción:</b>	Se realiza la búsqueda de una inscripción por criterio y por texto de búsqueda y se muestra una lista de resultados.
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Grupos</li> <li>3. Ir a la opción Buscar Inscripciones</li> <li>4. Ingresar el texto de búsqueda</li> <li>5. Seleccionar el criterio de búsqueda</li> <li>6. Clic en el botón Buscar</li> </ol>
<b>Resultados Esperados:</b>	<p>Mostrar una lista de inscripciones de acuerdo al texto de búsqueda y al criterio seleccionado.</p> <p>Cada elemento de la lista tiene un botón que permite eliminar la inscripción del sistema.</p> <p>Cada elemento de la lista tiene un botón que permite editar la información de la inscripción.</p>
<b>Resultados Obtenidos:</b>	<p>Se muestra la lista de inscripciones.</p> <p>Se elimina la inscripción.</p> <p>Se muestra la página de edición con la información de la inscripción.</p>
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.22 Prueba de Aceptación Búsqueda de Inscripciones.**

Prueba de Aceptación	
<b>Código</b>	<b>18</b>
<b>Nombre de la Prueba:</b>	Prueba de Listado de Mis Grupos
<b>Ítem Correspondiente:</b>	Listar Mis Grupos
<b>Descripción:</b>	Se muestra una lista de los grupos correspondientes al usuario que ingresa con perfil de Instructor
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Grupos</li> <li>3. Ir a la opción Mis Grupos</li> </ol>
<b>Resultados Esperados:</b>	Mostrar una lista de grupos de acuerdo al usuario que ingresa con perfil de Instructor.
<b>Resultados Obtenidos:</b>	Se muestra la lista de grupos.
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.23 Prueba de Aceptación Listado de Mis Grupos.**

Prueba de Aceptación	
<b>Código</b>	<b>19</b>
<b>Nombre de la Prueba:</b>	Prueba de Registro de Asistencia
<b>Ítem Correspondiente:</b>	Registrar Asistencia
<b>Descripción:</b>	Se registra la asistencia de los integrantes de un grupo
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Grupos</li> <li>3. Ir a la opción Registro de Asistencia</li> <li>4. Ingresar la información de Asistencia</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	Registrar la asistencia de los alumnos de un grupo y mostrar un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se registra la asistencia al grupo y se muestra el mensaje.
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.24 Prueba de Aceptación Registro de Asistencia.**

Prueba de Aceptación	
<b>Código</b>	<b>20</b>
<b>Nombre de la Prueba:</b>	Prueba de Registro de Notas
<b>Ítem Correspondiente:</b>	Registrar Notas
<b>Descripción:</b>	Se registra las notas de los integrantes de un grupo
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema</li> <li>2. Ir al módulo Grupos</li> <li>3. Ir a la opción Registro de Notas</li> <li>4. Ingresar la información de Notas</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Resultados Esperados:</b>	Registrar las notas de los alumnos de un grupo y mostrar un mensaje informativo.
<b>Resultados Obtenidos:</b>	Se registra las notas de los alumnos y se muestra el mensaje.
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.25 Prueba de Aceptación Registro de Notas.

Prueba de Aceptación	
<b>Código</b>	<b>21</b>
<b>Nombre de la Prueba:</b>	Prueba de Autenticación de Usuarios
<b>Ítem Correspondiente:</b>	Autenticación de usuario
<b>Descripción:</b>	Permite el ingreso de un usuario al sistema según el perfil asignado
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Abrir una ventana de Mozilla Firefox</li> <li>2. Digitar la dirección URL de la aplicación Web</li> <li>3. Digitar el username y la contraseña correspondiente</li> <li>4. Clic en Entrar</li> </ol>
<b>Resultados Esperados:</b>	Mostrar la página de Bienvenida al sistema.
<b>Resultados Obtenidos:</b>	Se muestra la página de Bienvenida.
<b>Evaluación de la Prueba:</b>	Exitosa

Tabla 3.26 Prueba de Aceptación Autenticación de Usuarios.

<b>Prueba de Aceptación</b>	
<b>Código</b>	<b>22</b>
<b>Nombre de la Prueba:</b>	Prueba de Generación de Reportes
<b>Ítem Correspondiente:</b>	Generar Reportes
<b>Descripción:</b>	Permite generar reportes informativos
<b>Procedimiento:</b>	<ol style="list-style-type: none"> <li>1. Escoger uno de los Módulos del sistema</li> <li>2. Ir a la opción de Reportes</li> <li>3. Seleccionar los filtros</li> <li>4. Seleccionar el tipo de reporte</li> <li>5. Clic en Generar Reporte</li> </ol>
<b>Resultados Esperados:</b>	Generar el reporte en formato PDF Generar el reporte vía Web
<b>Resultados Obtenidos:</b>	Se descarga el reporte en formato PDF Se muestra una página Web con la información del reporte
<b>Evaluación de la Prueba:</b>	Exitosa

**Tabla 3.27 Prueba de Aceptación Generación de Reportes.**



### **3.3 INSPECCIÓN Y ADAPTACIÓN**

En el presente proyecto se realiza la inspección por medio del seguimiento diario del progreso de cada sprint, lo que permite gestionar el trabajo restante adaptándolo para cumplir con el objetivo del sprint.

#### **3.3.1 SEGUIMIENTO DEL PROGRESO DEL SPRINT**

El seguimiento consiste en identificar en cualquier momento de la realización del sprint el trabajo que se necesita completar para alcanzar el objetivo.

Scrum especifica que el cliente debe realizar este seguimiento al menos una vez por cada revisión de sprint. El cliente compara el trabajo restante actual con el de revisiones de sprint previas para evaluar el progreso del trabajo proyectado en el tiempo deseado; por otro lado, el equipo de desarrollo realiza el seguimiento diariamente para gestionar su progreso a lo largo del sprint.

Existen varias prácticas de proyección que se pueden utilizar para predecir el progreso, entre estas tenemos la gráfica de tendencia de trabajo consumido (Burndown).

En el presente proyecto se utiliza el diagrama Burndown para el seguimiento de cada sprint.

##### **3.3.1.1 Sprint 1**

En la Figura 3.9 se muestra el diagrama Burndown de seguimiento a la fecha de finalización del sprint 1.

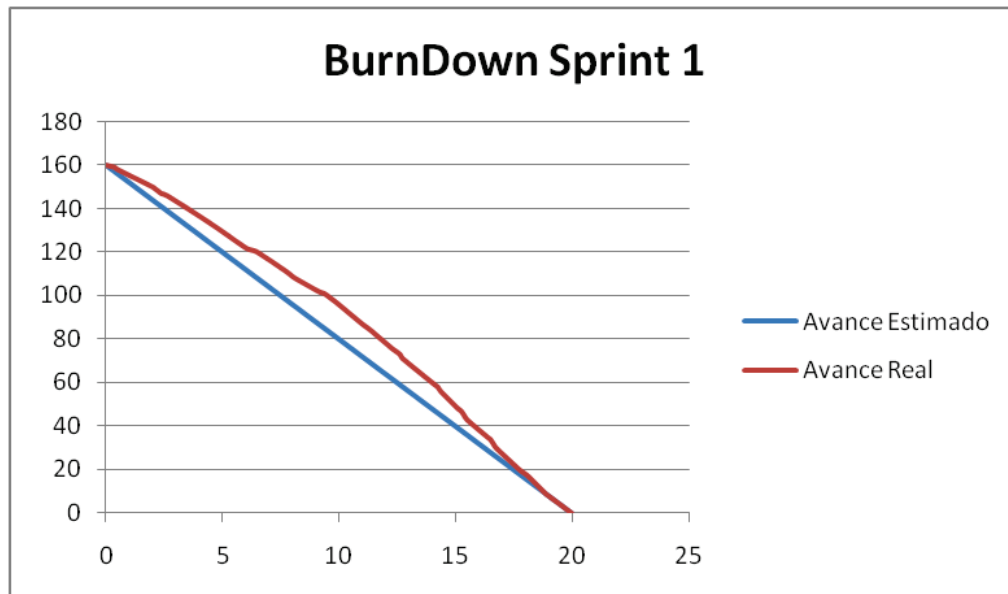


Figura 3.9 Gráfica Burndown para el Sprint 1

Como se observa en el diagrama existe un retraso en las tareas realizadas. Del análisis del equipo de desarrollo se identifican las siguientes causas: entrenamiento inicial en el framework JSF, malas estimaciones que luego fueron ajustadas.

Como resultado del análisis del gráfico se corrigen los problemas que surgen, se estiman correctamente las tareas y se cumple con el plazo definido al inicio del Sprint.

### 3.3.1.1 Sprint 2

En la Figura 3.10 se muestra el diagrama Burndown de seguimiento a la fecha de finalización del sprint 2.

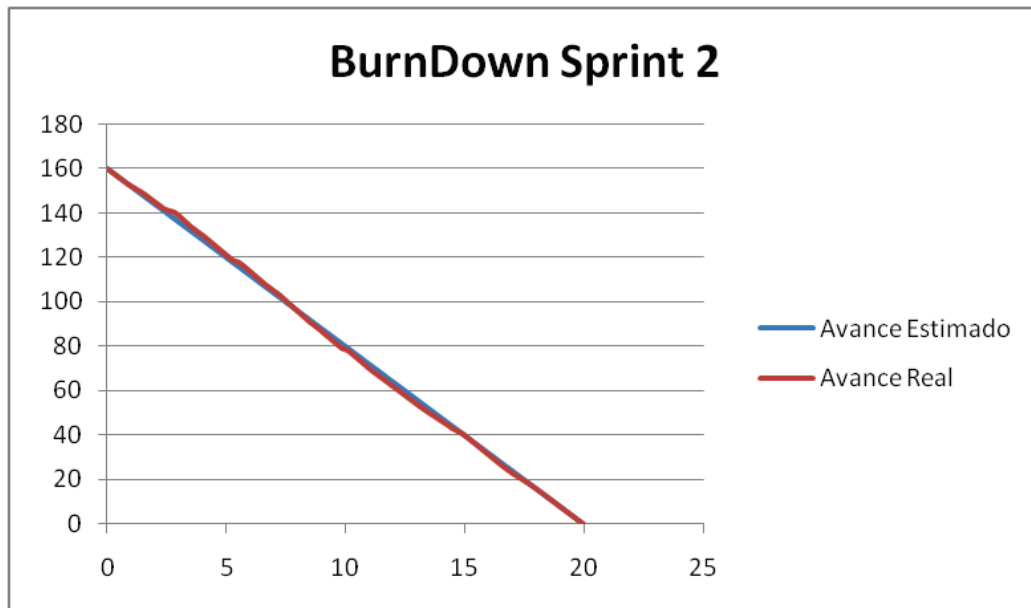


Figura 3.10 Gráfica Burndown para el Sprint 2

Para el Sprint 2 se observa que la mayor parte de las tareas se cumplen dentro de los plazos establecidos sin mayores inconvenientes y no existen problemas de gran relevancia.

Este sprint se desarrolla con normalidad ya que el equipo cuenta con más experiencia en el manejo de las herramientas y la metodología utilizadas y el cliente se muestra satisfecho con los resultados.

## CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 CONCLUSIONES

- El sistema de Administración de las actividades eclesíásticas de la Iglesia Católica Parroquial Santiago Apóstol permite mejorar la calidad del servicio y facilita el acceso a la información de manera ordenada y clara.
- Al ser una aplicación Web está diseñada para permitir a futuro la entrega de nuevos servicios a la comunidad, a los que se podrá acceder usando un navegador.
- La metodología Scrum se caracteriza por ser incremental e iterativa, permitiendo una mayor participación del cliente en el proyecto y una interacción continua con el equipo de desarrollo, para obtener un producto de software de calidad y que cumpla con sus expectativas.
- Los cambios en los requerimientos solicitados por el cliente se realizaron con éxito y sin grandes impactos en el sistema, lo que se logró gracias a que Scrum sugiere la entrega de pequeñas versiones del sistema al final de cada sprint para que sean analizadas y aprobadas por el cliente.
- El uso de software libre en el presente proyecto permite que el equipo de desarrollo se enfoque entregar un producto de software con la mayor calidad posible sin preocuparse por el costo de licencias y permisos.
- La selección de las herramientas de desarrollo apropiadas para el presente proyecto permite que se aprovechen las capacidades, conocimiento y experiencia de la desarrolladora, proporcionándole un mejor entendimiento, habilidad y práctica al usarlas.

## 4.2 RECOMENDACIONES

- Se recomienda el uso de software libre para reducir los costos en un proyecto de desarrollo de software.
- El uso de la metodología Scrum en proyectos de desarrollo de software es altamente recomendable, ya que garantiza que el cliente obtenga un producto de calidad, en el tiempo establecido y que cumpla los objetivos propuestos.
- Se recomienda ampliar la zona de cobertura de la red inalámbrica existente en la Iglesia Católica Parroquial Santiago Apóstol, para que el sistema brinde un mejor servicio y esté disponible a la comunidad.
- Se recomienda actualizar diariamente el gráfico Burndown, para poder realizar el seguimiento del proyecto, tomar las decisiones que permitan detectar y corregir problemas a tiempo y cumplir con el plazo establecido.
- Establecer un horario fijo para las reuniones diarias que dure aproximadamente 15 minutos, así el equipo tendrá una mejor comunicación y se organizarán las tareas oportunamente.
- Se recomienda especificar normas y estándares para el ingreso de la información en el sistema y para su gestión en general dentro de la Iglesia, con el objetivo de evitar errores y futuros inconvenientes por un manejo inadecuado.

## REFERENCIAS BIBLIOGRÁFICAS

### Tesis de Referencia

- MALDONADO María Fernanda, TIPÁN Ana. Implementación de un Sistema Web para manejo de datos meteorológicos del laboratorio de Energías Alternativas y Eficiencia Energética de la Escuela Politécnica Nacional. Proyecto de Titulación. Quito, Octubre 2011. E.P.N. 260 p.
- VILLENAS Andrés. Desarrollo de un sistema de publicidad de proximidad. Proyecto de Titulación. Quito, Mayo 2012. E.P.N. 107 p.
- JERÉZ Eleana. Desarrollo del Portal Web de la intranet para la empresa Servicios y Lujos Only Cars sc. Proyecto de Titulación. Quito, Mayo 2012. E.P.N. 158 p.
- REDÍN Ricardo. Aplicación de Scrum al desarrollo del sistema para la calificación y asignación de montos y plazos de crédito. Proyecto de Titulación. Quito, Junio 2012. E.P.N. 176 p.
- ARELLANO Fabricio. Desarrollo e implementación del sistema de reservación de laboratorios para el laboratorio de la facultad de Ingeniería de Sistemas. Proyecto de Titulación. Quito, Febrero 2011. E.P.N. 144 p.
- CASAMAYOR Alejandro. Mejora de la Metodología Scrum. Proyecto de Titulación. Buenos Aires, Febrero 2010. Universidad Belgrano. 68 p.

### Libros de Referencia

- PRESSMAN Roger. Ingeniería del Software. Un enfoque práctico. Quinta Edición. McGraw-Hill. 2002.
- Sun Microsystems. The Java EE 5 Tutorial. 2007.
- GONZALEZ José, CORDERO Juan. Diseño de Páginas Web: Iniciación y Referencia. Segunda Edición. MCGraw-Hill / Interamericana de España S.A. 2004.

- NIELSEN Jakob. Usabilidad: Diseño de sitios Web. Prentice Hall. 2002.
- PALACIO Juan. Flexibilidad con Scrum: Principios de diseño e implantación de campos de Scrum. Autoedición. 2008.
- COHN Mike. Agile Estimating and Planning. Prentice Hall. 2005.
- SCHILDT Herbert. Java: A Beginner's Guide. Tercera Edición. McGraw-Hill / Osborne. 2005.
- ECKEL Bruce. Piensa en Java. Segunda Edición. Prentice Hall. 2002.

### Sitios y Documentos Web de Referencia

- SCHWABER Ken, SUTHERLAND Jeff. La Guía Definitiva de Scrum: Las Reglas del Juego. Octubre 2011. Disponible en:  
[https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum\\_Guide%202011%20-%20ES.pdf](https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide%202011%20-%20ES.pdf)
- Red Hat Inc. Plataforma de Aplicaciones JBoss Enterprise para portales. 2007. Disponible en:  
[http://es.redhat.com/pdf/jboss/JB\\_PORTAL\\_PLATFORM\\_ES\\_web.pdf](http://es.redhat.com/pdf/jboss/JB_PORTAL_PLATFORM_ES_web.pdf)
- Wikipedia, la enciclopedia libre. Scrum. <http://es.wikipedia.org/wiki/Scrum>
- Softeng Software Engineers. Metodología Scrum. <http://www.softeng.es/es-empresa/metodologias-de-trabajo/metodologia-scrum.html>
- Scrum, gestión ágil de proyectos / agile Project management. <http://www.marblestation.com/?p=663>
- Tutorial de UML – Modelo de Clases.  
<http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>
- GONZÁLEZ José. El lenguaje de Modelado UML. <http://www.docirs.cl/uml.htm>
- Tutorial de Java Server Faces. Disponible en:  
<http://www.sicama.uma.es/sicama/Formacion/documentacion/JSF.pdf>
- Xtreme Programming. <http://xprogramming.com/what-is-extreme-programming>
- LETELIER Patricio, PENADÉS Carmen. Metodologías ágiles para el desarrollo de software. [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)
- IS-T. METODOLOGÍA SCRUM. <http://is-t.wikispaces.com/metodologia+scrum>

- Gráfico de Burndown.  
[http://www.dosideas.com/wiki/Gr%C3%A1fico\\_de\\_Burn-Down](http://www.dosideas.com/wiki/Gr%C3%A1fico_de_Burn-Down)
- VERHEYEN, Gunther. Introduction to the Professional Scrum Developer program. <http://mscop.be.capgemini.com/tag/scrum/>
- JSF 2.0 Components Showcase. With some jQuery UI goodness.  
<http://componentsapp.appspot.com/inputText.jsf>
- RichFaces. Live Demo. <http://livedemo.exadel.com/richfaces-demo/>
- MySQL. 5.0 Reference Manual.  
<http://dev.mysql.com/doc/refman/5.0/es/features.html>
- W3Schools. <http://www.w3schools.com/>
- W3C. Web Design and Applications.  
<http://www.w3.org/standards/webdesign/htmlcss>



## GLOSARIO

- **API**

Interfaz de programación de aplicaciones (Application Programming Interface). Representa una librería de Java extensiva provista por los creadores de Java y que da a los programadores los medios para desarrollar aplicaciones Java.

- **Burndown**

Es un gráfico que facilita el seguimiento del Sprint. Muestra cuánto le falta al equipo para completar con el compromiso del Sprint.

- **CSS**

Hojas de Estilo en Cascada (Cascading Style Sheets). Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar estos documentos.

- **EJB**

Siglas en Inglés de Enterprise Java Beans. Definen un modelo para el desarrollo y distribución de componentes reutilizables del lado del servidor Java. Se usan para encapsular varios objetos en un único objeto.

- **Entity Beans**

EJB de Entidad. Encapsulan los objetos del lado del servidor que almacena los datos. Representan un objeto existente en la base de datos de la institución.

- **HTML**

Siglas en inglés de Hypertext Markup Lenguaje. Es un lenguaje que permite crear documentos para Internet mediante el uso de etiquetas. El organismo que gestiona la especificación del lenguaje HTML es W3C (World Wide Web Consortium).

- **IDE**

Entorno de desarrollo integrado (Integrated Development Environment). Es un programa informático compuesto por un conjunto de herramientas de programación.

- **Java**

Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995 de propósito general, concurrente, orientado a objetos y basado en clases.

- **Java EE**

Java Enterprise Edition o Java EE es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas.

- **JavaScript**

JavaScript es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página. Es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML.

- **JDK**

Siglas en inglés de Java Development Kit. Es un software que provee herramientas de desarrollo para la creación de programas en Java.

- **JBoss**

JBoss es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Puede ser utilizado en cualquier sistema operativo que soporte Java.

- **JSF**

Siglas en inglés de Java Server Faces. Es una tecnología y marco de trabajo para aplicaciones Java basadas en Web que simplifica el desarrollo de interfaces de

usuario en aplicaciones Java JEE. Está orientado a la interfaz gráfica de usuario (GUI).

- **MVC**

Abreviatura de Modelo Vista Controlador. Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

- **MySQL**

Es un sistema de gestión de base de datos desarrollado por Sun Microsystems. Permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos.

- **Product Backlog**

Pila del Producto. Es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requerimientos para cualquier cambio a realizarse en el producto.

- **Scrum**

Es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

- **Session Beans**

EJB de Sesión. Gestionan el flujo de información en el servidor. Representan un proceso o acción del negocio.

- **Sprint**

Es un periodo de tiempo de un mes o menos durante el cual se crea un incremento de producto “Hecho”, utilizable y potencialmente entregable.

- **Sprint Backlog**

Pila del Sprint. Es el conjunto de elementos de la Pila de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint.

- **UML**

Lenguaje de Modelamiento Unificado (Unified Modeling Language). Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

- **XML**

Lenguaje de marcas extensible (eXtensible Markup Language). Desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

## ANEXOS

**ANEXO A:** Manual de Usuario del sistema SISAPE.

Anexo Digital

CD-ROOM: \Anexos\Manuales\Manual de Usuario Sistema SISAPE

**ANEXO B:** Manual de Instalación y Configuración del sistema SISAPE.

Anexo Digital

CD-ROOM: \Anexos\Manuales\Manual de Instalación y Configuración Sistema SISAPE

**ANEXO C:** Paquete de Instalación Sistema SISAPE

Anexo Digital

CD-ROOM: \Anexos\SISAPE\Sistema SISAPE