

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:**

**INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:**

**Ingenieros de Sistemas**

**TEMA:**

**DESARROLLO DE UN MÓDULO GERENCIAL FINANCIERO QUE  
INTEGRE TODOS LOS SISTEMAS RUA DE LAS OBRAS Y CASAS  
SALESIANAS EN EL ECUADOR.**

**AUTORES:**

**CRISTIAN JAVIER PILLAJO PAILLACHO**

**LUIS PATRICIO TOAPANTA YANCHAGUANO**

**TUTOR:**

**DANIEL GIOVANNY DÍAZ ORTIZ**

**Quito, septiembre de 2020**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros Cristian Javier Pillajo Paillacho con documento de identificación N° 1721622866 y Luis Patricio Toapanta Yanchaguano con documento de identificación N° 0502895634, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: DESARROLLO DE UN MÓDULO GERENCIAL FINANCIERO QUE INTEGRE TODOS LOS SISTEMAS RUA DE LAS OBRAS Y CASAS SALESIANAS EN EL ECUADOR, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, que dando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, septiembre de 2020



.....  
CRISTIAN JAVIER  
PILLAJO PAILLACHO  
CI: 1721622866



.....  
LUIS PATRICIO  
TOAPANTA YANCHAGUANO  
CI: 0502895634

Quito, septiembre de 2020

**DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema **DESARROLLO DE UN MÓDULO GERENCIAL FINANCIERO QUE INTEGRE TODOS LOS SISTEMAS RUA DE LAS OBRAS Y CASAS SALESIANAS EN EL ECUADOR**, realizado por **Cristian Javier Pillajo Paillacho** y **Luis Patricio Toapanta Yanchaguano** obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, septiembre de 2020



.....

DANIEL GIOVANNY DÍAZ ORTIZ

CI: 1716975501

## **DEDICATORIA**

Este trabajo va dedicado primeramente a Dios que sin su ayuda y su bendición nada sería posible, a mis padres por haberme dado la vida y que hicieron que no me falte nada mientras me criaron con todo lo que estaba a su alcance, a mi esposa, mi hija Tiffany y mi hijo Lionel que más que ser una bendición en mi vida son mi inspiración mi motor, me apoyaron, me comprendieron y caminaron conmigo toda esta larga etapa de estudios hasta culminarla sembrando en mí siempre responsabilidad, dedicación para poder esforzarme y cumplir las metas que me proponga.

Sin olvidarme mis hermanos, mis suegros y de toda mi familia que en cualquier momento de este largo camino me apoyaron incondicionalmente.

Y no podían faltar mis compañeros de trabajo de Adelca que cada vez que necesitaba encontré apoyo, en especial a una persona que ya no trabaja aquí me inspiraron, me ayudaron para no rendirme y encontraba palabras de aliento siempre.

Cristian Javier Pillajo Paillacho

## **DEDICATORIA**

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

A mi amada novia Ana Dolores Tipán, por su apoyo y ánimo que me brinda día con día para alcanzar nuevas metas, tanto profesionales como personales. Siendo un pilar de motivación en las situaciones más difíciles, brindándome seguridad y confianza.

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

A mis hermanos y hermanas, por sus consejos y por su compañía que son una fuente de motivación, inspiración y felicidad.

Luis Patricio Toapanta Yanchaguano

## **AGRADECIMIENTO**

Agradecemos a la Universidad Politécnica Salesiana por acogernos muchos años de estudio y que han contribuido en nuestro desarrollo personal y académico a cada profesor que paso por nuestra vida estudiantil y sobre todo el mayor agradecimiento a Ing. Daniel Giovanny Díaz Ortiz quien fue nuestro tutor del proyecto y contamos siempre con su apoyo, por habernos guiado y orientado para poder culminar con éxito nuestro trabajo.

Agradecemos de la misma manera a la Lic. Lorena Sanchez quien apoyo y nos dio apertura para poder realizar nuestro trabajo de titulación en la Casa Inspectorial Salesiana.

Cristian Javier Pillajo Paillacho

Luis Patricio Topanta Yanchaguano

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
Antecedentes .....	1
Problemática .....	2
Justificación .....	3
Objetivos .....	4
Objetivo general .....	4
Objetivos específicos .....	4
Marco Metodológico.....	5
<b>CAPÍTULO I.....</b>	<b>8</b>
<b>1 ESTADO DEL ARTE .....</b>	<b>8</b>
1.1 Marco Institucional.....	8
1.1.1 Reseña Histórica.....	8
1.1.2 Organigrama Casa Inspectorial.....	8
1.1.3 Diagrama del proceso .....	9
1.2 Marco Teórico .....	13
1.2.1 Software empresarial y de pequeñas empresas .....	13
1.2.1.1 Gestión financiera.....	14
1.2.1.2 Gestión contable.....	14
1.2.2 Herramientas de desarrollo.....	16
1.2.2.1 Framework Jd Faces .....	17
1.2.2.2 Base de datos Postgress .....	18
1.2.2.3 GlassFish.....	19
1.2.2.4 Power Designer .....	19
1.2.2.5 Java .....	20
1.2.2.6 JDK.....	20
1.2.2.7 Netbeans.....	21
1.2.2.8 Scrum .....	21
1.2.2.9 Jmeter .....	27
<b>CAPÍTULO II .....</b>	<b>29</b>
<b>2 ANÁLISIS Y DISEÑO.....</b>	<b>29</b>
2.1 Análisis de requerimientos .....	29
2.2 Artefactos .....	31
2.2.1 Diagramas de casos de uso .....	31

2.2.1.1	Historias de usuario.....	31
2.2.2	Diagramas caso de uso .....	36
2.2.2.1	Caso de uso usuario administrador.....	36
2.2.2.2	Caso de uso usuario sistema RUA.....	36
2.2.2.3	Caso de uso usuarios externos.....	37
2.2.2.4	Caso de uso usuario contador general.....	37
2.2.2.5	Caso de uso usuario contador general.....	38
2.2.3	Diagramas de secuencia .....	38
2.3	Diagrama conceptual base de datos.....	41
<b>CAPÍTULO III.....</b>		<b>42</b>
<b>3</b>	<b>CONSTRUCCIÓN Y PRUEBAS.....</b>	<b>42</b>
3.1	Diagrama de clases módulo gerencial financiero.....	42
3.1.1	Código relevante .....	43
3.1.1.1	Versión del proyecto .....	43
3.1.1.2	Método activar período contable .....	44
3.1.1.3	Método activar mes contable.....	45
3.1.1.4	Método cerrar período contable.....	46
3.1.1.5	Método transferir Balance .....	47
3.1.1.6	Método validar archivos en Excel .....	48
3.1.1.7	Método generación de reportes gerenciales.....	49
3.2	Diagrama físico de base de datos.....	50
3.3	Diagrama de despliegue .....	55
3.3.1	Arquitectura.....	55
3.3.2	Requerimientos.....	57
3.3.2.1	Servidor físico .....	57
3.3.2.2	Software necesario .....	57
3.4	Plan de pruebas.....	58
3.4.1	Pruebas funcionales.....	58
3.4.1.1	Pruebas caja negra.....	58
3.4.1.2	Pruebas de validación.....	62
3.4.1.3	Pruebas de stress.....	64
3.5	Análisis de resultados de pruebas .....	68
3.5.1	Resultado pruebas caja negra, validación y conexión.....	68
3.5.2	Resultado pruebas de stress .....	69
3.5.2.1	Resultado pruebas stress 1.....	69
3.5.2.2	Resultado pruebas stress 2.....	71
3.5.2.3	Resultado pruebas stress 3.....	72

<b>CONCLUSIONES.....</b>	<b>74</b>
<b>RECOMENDACIONES.....</b>	<b>76</b>
<b>REFERENCIAS.....</b>	<b>78</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>81</b>

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Características Metodologías ágiles .....	6
<b>Tabla 2:</b> Requerimiento funcionales.....	29
<b>Tabla 3:</b> Requerimiento no funcionales.....	30
<b>Tabla 4:</b> Historia usuario administrador .....	31
<b>Tabla 5:</b> Historia de usuario sistema RUA. ....	32
<b>Tabla 6:</b> Historia de usuarios externos .....	33
<b>Tabla 7:</b> Historia de usuario contador general casa inspectorial.....	34
<b>Tabla 8:</b> Historia de usuario contador general casa inspectorial .....	35
<b>Tabla 9:</b> Glosario de términos y abreviaturas .....	52
<b>Tabla 10:</b> Formato de los datos .....	52
<b>Tabla 11:</b> Características módulo gerencial financiero.....	56
<b>Tabla 12:</b> Servidor físico.....	57
<b>Tabla 13:</b> Software necesario para que el sistema funcione. ....	57
<b>Tabla 14:</b> Prueba Caja Negra 1.....	59
<b>Tabla 15:</b> Prueba Caja Negra 2.....	59
<b>Tabla 16:</b> Prueba Caja Negra 3.....	60
<b>Tabla 17:</b> Prueba Caja Negra 4.....	60
<b>Tabla 18:</b> Prueba Caja Negra 5.....	61
<b>Tabla 19:</b> Prueba Caja Negra 6.....	61
<b>Tabla 20:</b> Formulario parametrización .....	62
<b>Tabla 21:</b> Generar balance Sistema RUA.....	62
<b>Tabla 22:</b> Carga de archivos Excel usuarios externos .....	63
<b>Tabla 23:</b> Consolidación de los balances generales y cédulas presupuestarias.....	63
<b>Tabla 24:</b> Generación de estados financieros. ....	64
<b>Tabla 25:</b> Parámetros para realizar las pruebas de stress.....	65

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Organigrama Casa Inspectorial Salesiana.....	9
<b>Figura 2:</b> Cierre contable mensual .....	10
<b>Figura 3:</b> Generación del balance general. ....	10
<b>Figura 4:</b> Envío de archivos Excel .....	11
<b>Figura 5:</b> Consolidación balances .....	11
<b>Figura 6:</b> Generación Estados Financieros .....	12
<b>Figura 7:</b> Presentación Estados Financieros .....	12
<b>Figura 8:</b> Estructura del framework dinámico .....	18
<b>Figura 9:</b> Ciclo desarrollo software.....	22
<b>Figura 10:</b> Reglas scrum .....	23
<b>Figura 11:</b> Diagramas caso de uso administrador.....	36
<b>Figura 12:</b> Diagramas caso de uso usuario sistema RUA.....	36
<b>Figura 13:</b> Diagramas caso de uso usuario usuarios externos.....	37
<b>Figura 14:</b> Diagramas caso de uso contador general .....	37
<b>Figura 15:</b> Diagramas caso de uso contador general .....	38
<b>Figura 16:</b> Diagrama de secuencia administrador .....	38
<b>Figura 18:</b> Diagrama de secuencia usuarios externos.....	39
<b>Figura 17:</b> Diagrama de secuencia usuario sistema RUA.....	39
<b>Figura 20:</b> Diagrama de secuencia usuario contador general - reportes .....	40
<b>Figura 19:</b> Diagrama de secuencia contador general .....	40
<b>Figura 21:</b> Diagrama conceptual de Base de datos módulo gerencial financiero.....	41
<b>Figura 22:</b> Diagrama clases de módulo gerencial financiero.....	42
<b>Figura 23:</b> Desarrollo de aplicación Netbeans.....	43
<b>Figura 24:</b> Método agregar y cerrar obra.....	44
<b>Figura 25:</b> Método agregar y cerrar períodos mensuales de las casas y obras .....	45
<b>Figura 26:</b> Método cierre de períodos mensuales de las casas y obras .....	46
<b>Figura 27:</b> Método transferir balances .....	47
<b>Figura 28:</b> Método validar archivos de Excel.....	48
<b>Figura 29:</b> Método generación de reportes.....	49
<b>Figura 30:</b> Modelo entidad relación.....	50
<b>Figura 31:</b> Diagrama físico parte 2.....	51

<b>Figura 32:</b> Estándar nombre tablas.....	53
<b>Figura 33:</b> Estándar nombre de campos .....	53
<b>Figura 34:</b> Estándar nombre campos de auditoría .....	54
<b>Figura 35:</b> Estándar nombre de dominios.....	54
<b>Figura 36:</b> Diagrama de despliegue.....	55
<b>Figura 37:</b> Configuración de hilos e intentos prueba stress 1 .....	65
<b>Figura 38:</b> Muestra de la prueba de stress 1 .....	66
<b>Figura 39:</b> Configuración de hilos e intentos prueba stress 2 .....	66
<b>Figura 40:</b> Muestra de la prueba de stress 2 .....	67
<b>Figura 41:</b> Configuración de hilos e intentos prueba stress 3 .....	67
<b>Figura 42:</b> Muestra de la prueba de stress 3 .....	68
<b>Figura 43:</b> Transacciones Exitosas en prueba de stress 1 .....	69
<b>Figura 44:</b> Distribución de datos .....	70
<b>Figura 45:</b> Tiempo de respuesta .....	70
<b>Figura 46:</b> Transacciones Exitosas en prueba de stress 2 .....	71
<b>Figura 47:</b> Distribución de datos .....	71
<b>Figura 48:</b> Tiempo de respuesta .....	72
<b>Figura 49:</b> Transacciones Exitosas en prueba de stress 3 .....	73
<b>Figura 50:</b> Distribución de datos .....	73
<b>Figura 51:</b> Tiempo de respuesta .....	73

## RESUMEN

La finalidad de este proyecto fue cumplir con la solución a una necesidad que se tenía dentro de la Casa Inspectorial Salesiana, debido a que no contaban con un sistema que pueda centralizar toda la información contable y financiera de las casas y obras salesianas, por lo cual cada uno de estas entidades llevaba sus registros por separado en su sistema contable o Excel y no contaban con reportes de balances generales y presupuestarios. Por tanto se realizó el análisis, diseño y desarrollo de un sistema que permita integrar y consolidar toda la información contable de todas las casas y obras salesianas.

El desarrollo de este proyecto se realizó utilizando un Framework denominado Jd Faces el cual está desarrollado, heredando y personalizando con los componentes JSF de Primefaces, que en la actualidad está entre las mejores tecnologías para el desarrollo de aplicaciones web.

El módulo integrador desarrollado, permitió obtener estados financieros en tiempos más cortos de esta manera se pudo optimizar el proceso que realizaba la contadora general en la consolidación de los balances; además permitiendo generar reportes de los estados financieros en cualquier momento y cuando los ecónomos lo requieran.

Las pruebas de carga y stress mostraron que el sistema es robusto y que puede tener gran capacidad de procesamiento así como también brinda confiabilidad en los resultados generados, tanto por los procesos validados como por el control de auditoria que posee el modulo, pudiendo ser controlado el usuario que carga información o genera cambios, con tiempo y hora.

El trabajo operativo manual sin trazabilidad y poco fiable ahora ya está solventado con la implementación de este módulo gerencial financiero que será administrado por su ente rector de la casa inspectorial salesiana.

## **ABSTRACT**

The purpose of this project was to meet the solution to a need that existed within the Salesian Provincial House, since they did not have a system that can centralize all the accounting and financial information of the Salesian houses and works, for which each of these entities kept their records separately in their accounting system or Excel and did not have reports on general and budget balances. Therefore, the analysis, design and development of a system that allows integrating and consolidating all the accounting information of all the Salesian houses and works was carried out.

The development of this project was carried out using a Framework called Jd Faces which is developed, inheriting and customizing with the JSF components of Primefaces, which is currently among the best technologies for developing web applications.

The integrating module developed, obtaining financial statements in shorter times, in this way it was possible to optimize the process carried out by the entity's accountant; in addition to generating reports of the financial statements at any time and when the treasurers require it.

The load and stress tests show that the system is robust and that it can have great processing capacity as well as provides reliability in the results generated, both by the validated processes and by the audit control that the module has, being able to control the user who uploads information or generates changes, with time and hour.

The untraceable and unreliable manual operative work is now solved with the implementation of this financial management module that will be administered by the governing body of the Salesian provincial house.

## INTRODUCCIÓN

### Antecedentes

El primer sistema que se implementó en las casas salesianas, fue el sistema contable de nombre Rendiconto que realiza la gestión contable del Economato Salesiano sin embargo las regularizaciones tributarias del Ecuador son muy cambiantes, por lo cual este sistema contable Rendiconto quedo obsoleto y nuevamente surge la necesidad de encontrar un nuevo sistema para el manejo del departamento contable financiero del Economato Salesiano y es así que en Octubre de 2011, se realiza una reunión de contadores donde realizan un profundo análisis y toman la decisión basado en los problemas y las necesidades de la casa inspectorial, donde solicitan la elaboración de un sistema contable que responda a las necesidades muy particulares de estas sociedades salesianas que son sin fines de lucro pero que cumplan con las exigencias y cambios dinámicos que tiene la tributación ecuatoriana. Esta propuesta fue aceptada por cuatro estudiantes de la Universidad Politécnica Salesiana sede Quito, quienes entregaron de proyecto final para obtener su título de ingenieros de sistemas.

En una carta dirigida al Padre Alfredo Espinoza en calidad de Ecónomo Inspectorial, por parte de su asistente financiero recibía lo concluido en la reunión, donde mencionaba que el sistema contable será elaborado para el departamento, además que podría ser implementado en todas las casas y obras salesianas que así lo requieran, es así que una vez concluido este sistema contable financiero denominado RUA en su primera versión, el consejo inspectorial en su resolución N.2013 – 059 con fecha diciembre 3 de 2013 dado en la ciudad de Quito resuelve aprobar este sistema con el nombre de “RUA versión 1”.

Tiempo después en la ley tributaria de Ecuador surge nuevos cambios, es así que en septiembre del 2015 se lleva a cabo la firma de un contrato entre la Universidad Politécnica Salesiana y profesionales que inicialmente desarrollaron el Sistema, para realizar la actualización del sistema Contable financiero “RUA versión 1”, estas actualizaciones fueron para la implementación de facturación electrónica, digitalización de comprobantes entre otros, ante los constantes cambios tributarios de nuestro país y el cambio necesario en el plan de cuentas de alienación con las normativas contables vigentes, se decide considerar la propuesta de pensar en consolidar en una nueva versión 2 del RUA.

En esta propuesta se considera para trabajar en más módulos como, comprobantes electrónicos, control de producción, inventarios, control de pensiones, inscripciones, matrículas, especies valoradas y distintos rubros de cobros, entre otros, esta versión 2 de RUA al igual que la versión 1 siguen respondiendo a su primera concepción para la cual fueron desarrollados siendo un sistema contable consolidado que está alineado a todas las necesidades para la cual fue realizado y todas las realidades que ofrecen todos los servicios salesianos.

### **Problemática**

La casa inspectorial salesiana del Ecuador como administradora financiera tiene problemas para cumplir con su rol de obtener, verificar, administrar, controlar y presupuestar todos los recursos financieros que asigna a todas las casas y obras salesianas del país. En la actualidad se tiene un sistema contable financiero instalado en estas sociedades salesianas que cada una de ellas maneja toda su actividad contable y financiera de forma independiente.

Por tanto al ser sometida a una auditoría financiera carece de información relevante para dar rendición de cuentas, movimientos y gestión sobre todas las casas

ya que al no contar con un software que facilite la centralización de operaciones, información y registros contable financiero, no se puede tener control sobre las mismas.

Es así que con todo lo antes mencionado, surge nuestra propuesta como plan de proyecto de titulación con carácter de vinculación con la sociedad, para desarrollar un nuevo módulo que centralice e integre todos los sistemas contables RUA de las casas salesiana del Ecuador, esto permitirá que se consolide toda la información contable y presupuestaria, para tener el control sobre los datos financieros, facilitando su disponibilidad y visualización mediante los estados financieros que exigen la normativa contable, con esto facilita la toma de decisiones por parte de la Casa Inspectorial.

### **Justificación**

La visión de la Sociedad Salesiana del Ecuador indica, “Para el año 2020 la Sociedad Salesiana en el Ecuador debe contar con un sistema contable financiero, implementando en todas las casas y obras salesianas” (UPS, 2019), para ayudar a cumplir con este propósito nuestro proyecto de titulación será; desarrollar un módulo integrador financiero a nivel gerencial mismo que permitirá la integración y centralización de todos los sistemas contables instalados en las casas y obras salesianas, que este coordinado y que además permita al órgano rector en este caso la casa inspectoral, administrar los recursos económicos y financieros de forma eficiente bajo todos los lineamientos, normas y políticas salesianas y gubernamentales.

Por tal motivo, la casa inspectoral como órgano rector de todas las instituciones, se encuentra en la necesidad de disponer con toda la información consolidada sobre sus movimientos económicos tanto contables como presupuestarios,

de tal manera que puedan tener una proyección financiera para administrar correctamente y lo más importante que no se tenga pérdida de información.

Este control de finanzas es la parte más relevante en cualquier tipo de negocio, por eso es importante el desarrollo de este módulo integrador que permita proporcionar información acerca de la situación de su capital, además que ayude a establecer los lineamientos para que no tengan gastos y pérdidas no planificadas e innecesarias y que indique los movimientos reales, verídicos y disponibles en cualquier momento que lo requieran.

Con el módulo desarrollado e integrado, permitirá al usuario final contar con una serie de reportes a nivel gerencial de los movimientos contables y presupuestarios en forma general o detallada de acuerdo a los formatos indicados en la normativa contable vigente, para su respectivo análisis.

## **Objetivos**

### **Objetivo general**

Analizar, diseñar y desarrollar un módulo gerencial financiero que integre todos los sistemas RUA de las obras y casas salesianas del Ecuador.

### **Objetivos específicos**

Analizar y evaluar las bases de datos de los sistemas RUA implementados, para lograr centralizar su información.

Diseñar el modelo de gestión financiera para la realización del módulo.

Desarrollar el módulo gerencial financiero que centralice todas las casas y obras salesianas.

Desarrollar reportes de cédulas presupuestarias ingresos y gastos detallados a nivel gerencial.

Realizar pruebas de funcionalidad del módulo financiero desarrollado, utilizando métricas para evaluar su rendimiento.

## **Marco Metodológico**

### **Metodologías Ágiles**

Para seleccionar la metodología a utilizar en este proyecto se realizó un estudio y valoración de algunas de las metodologías ágiles sobre las características que tienen para el desarrollo de software, por lo cual se realiza una tabla comparativa.

**Tabla 1:**

Características Metodologías ágiles

Características	Tipo	XP	Scrum	Kanban
Tamaño Proyectos	Pequeños	X	X	X
	Medianos	X	X	X
	Grandes		X	
Tamaño del Equipo	Menos de 10 miembros	X		X
	Múltiples equipos		X	
Desarrollo	Dinámico		X	
	Restringido	X		X
Tipo de Iteraciones	Fijo	X		
	Variable		X	X
Equipos Humano	Multifuncional		X	
	Especializado	X		X
Modelo Desarrollo	Incremental	X		X
	Evolutivo		X	
Variabilidad	Acepta cambio de requerimientos	X	X	
Innovación	Multiplataforma	X	X	X
Pruebas	Funcionales y no funcionales	X	X	
Trabajo	Tareas individuales para cada miembro del equipo		X	
Gestión	Administra el proyecto		X	X
Manejo	Desarrollo incremental e iterativo	X	X	X
Orientación	Cumplimiento de requisitos iniciales y cambios.		X	X
Calidad	Pruebas durante el desarrollo del producto.	X	X	X
Producto	Nuevas versiones al final del proyecto.		X	

**Nota:** Muestra la tabla comparativa de las metodologías ágiles para el desarrollo de software**Elaborado por:** Los autores

Para el desarrollo de este módulo se pudo determinar que la metodología a utilizar será scrum, de acuerdo con las características de la tabla comparativa realizada con los diferentes tipos que son más amigables para el desarrollo de software ya que es simple de entender, además se puede medir los resultados muy rápidamente teniendo iteraciones y corrigiendo errores sobre la marcha, de esta manera no necesariamente se tiene que terminar todo el módulo para poder corregir errores que puedan presentar en el desarrollo del mismo, se puede apreciar el progreso del trabajo y que todos sus procesos sean transparentes y de acuerdo a lo establecido.

También indicar que al ser una técnica de trabajo en equipo se puede asignar tareas de manera colaborativa asignando roles y procesos de manera eficiente teniendo una iteración directa con el cliente para que pueda ir validando y realizando las pruebas funcionales frecuentemente, con esto se logra poder entregar un módulo que cumpla con las expectativas y los requerimientos solicitados.

## **CAPÍTULO I**

### **1 ESTADO DEL ARTE**

#### **1.1 Marco Institucional**

##### **1.1.1 Reseña Histórica**

La Casa Inspectorial de la Inspectoría Salesiana, durante mucho tiempo funcionó en el Colegio Don Bosco de la Tola, con el tiempo al irse expandiendo las obras y casas de la Inspectoría y del Vicariato, la casa inspectorial necesitó de unas dependencias más amplias y funcionales.

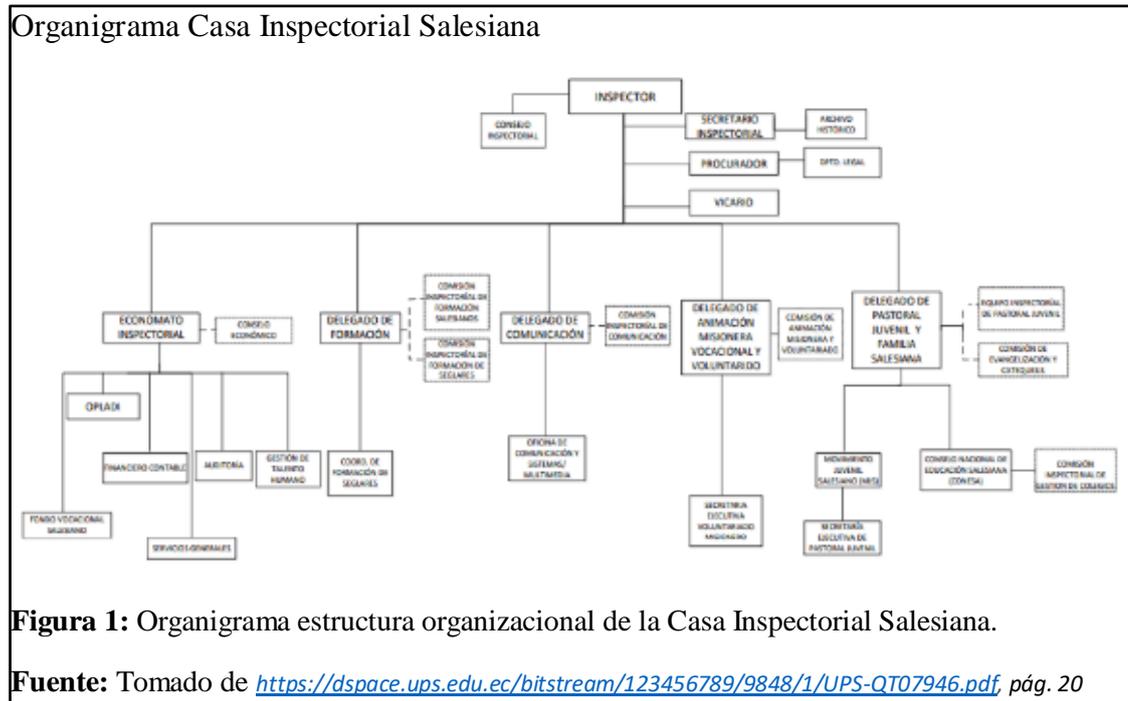
En el año de 1973, siendo inspector el padre Idelfonso Gil, se decidió empezar la construcción en la zona norte de Quito, en la calle Alpallana y 6 de Diciembre 144. La generosa colaboración del padre John Porter facilitó las cosas. La nueva casa inspectorial cuenta con más de una docena de habitaciones para los salesianos visitantes, las oficinas administrativas, capilla, etc.

La construcción de la nueva casa inspectorial salesiana de Quito inicia los trabajos en agosto de 2002, ubicada en los terrenos del Colegio Spellman que fue trasladada a Cumbayá, y fue inaugurada en abril de 2004. (SALESIANOS, 2019)

##### **1.1.2 Organigrama Casa Inspectorial**

Este organigrama que presenta la casa inspectorial es del tipo vertical y jerárquico, con todos sus servicios inspectoriales de acuerdo a la administración de gestión humana.

## Organigrama Casa Inspectorial Salesiana



**Figura 1:** Organigrama estructura organizacional de la Casa Inspectorial Salesiana.

**Fuente:** Tomado de <https://dspace.ups.edu.ec/bitstream/123456789/9848/1/UPS-QT07946.pdf>, pág. 20

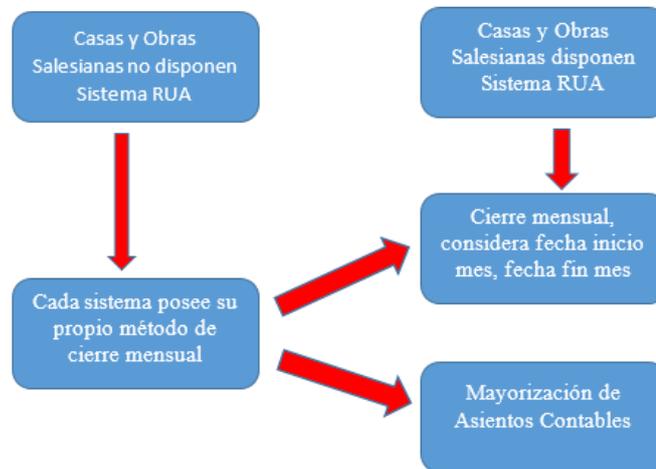
### 1.1.3 Diagrama del proceso

Una vez realizado el análisis de los requerimientos se elaboró los diagramas de procesos que actualmente se realiza de forma manual, dentro de esto se determina la implementación del módulo gerencial que automatice estos procesos, que muestre datos consolidados y permita la generación de estados financieros mensuales de las casas y obras salesianas en el Ecuador.

Los procesos para la consolidación de los estados financieros de las casas y obras salesianas son los siguientes.

Proceso 1.- Cierre contable mensual, en cada una de las casas y obras salesianas en los sistemas que cada uno de ellos posee.

### Cierre contable mensual

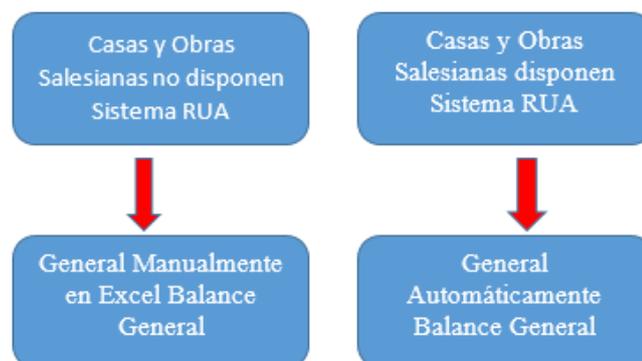


**Figura 2:** Proceso manual de cierre de periodo contable que mantienen las casas y obras salesianas.

**Elaborado por:** Los autores

Proceso 2.- Generación del balance general, cada casa u obra salesiana lo realiza manualmente por separado.

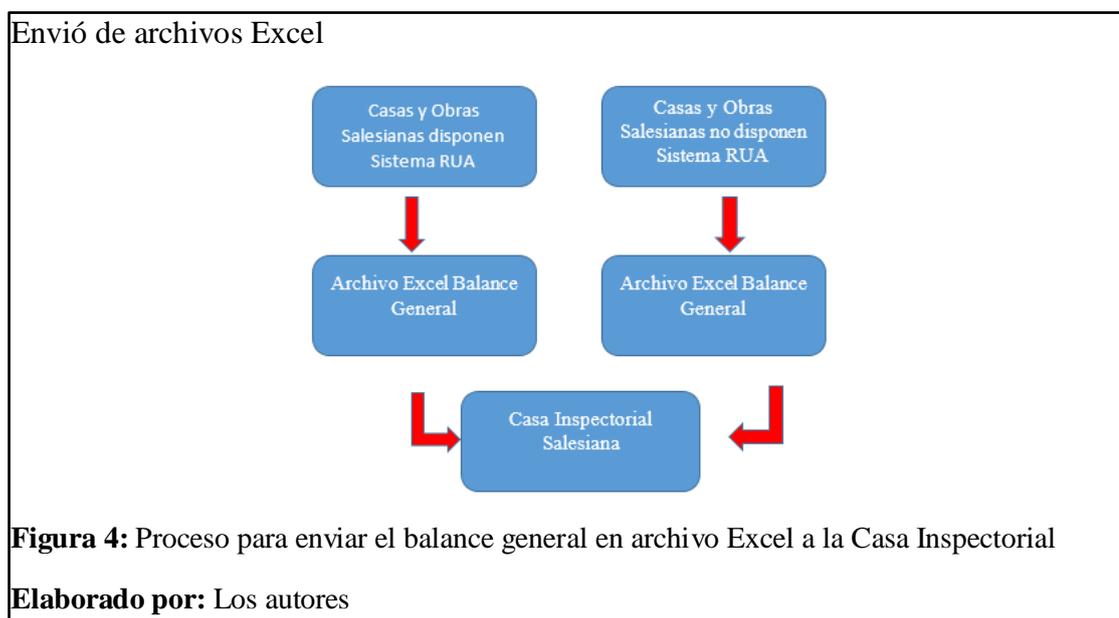
### Generación del balance general.



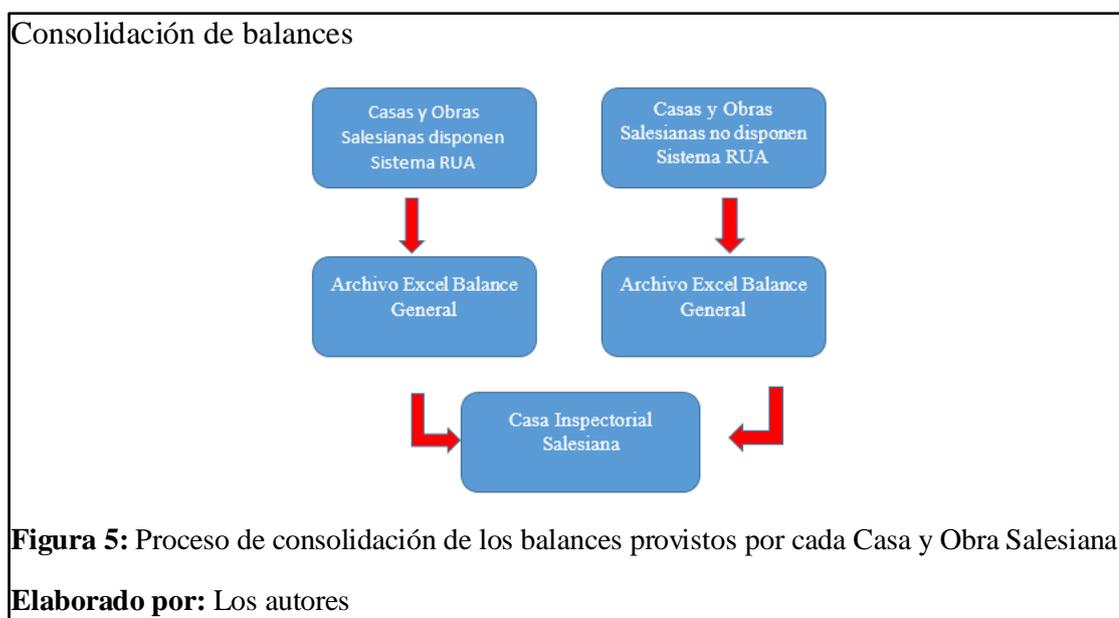
**Figura 3:** Proceso de generación del balance general que realiza el contador general de cada Casa y Obra Salesiana.

**Elaborado por:** Los autores

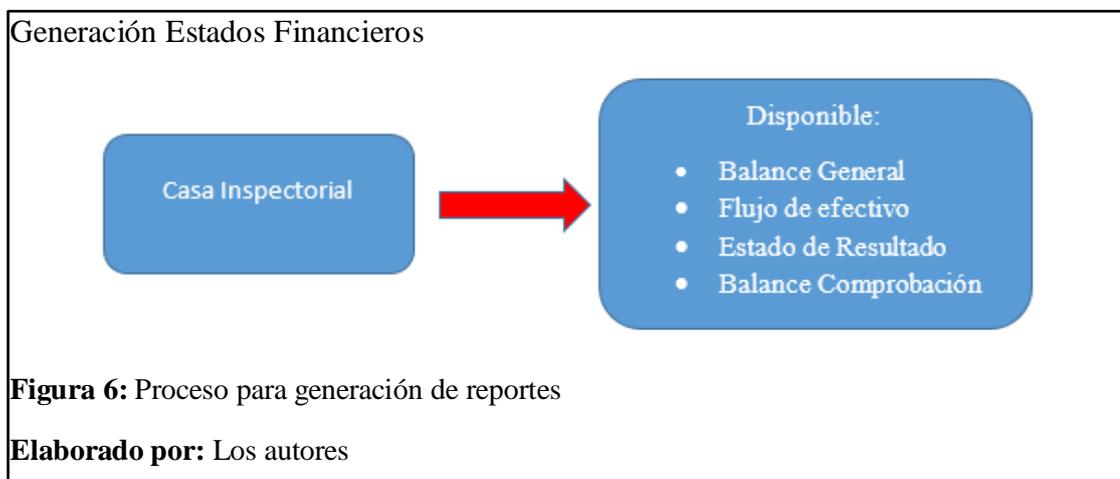
Proceso 3.- Envío de archivos Excel del balance general a la casa Inspectorial salesiana por parte de cada una de las casas y obras.



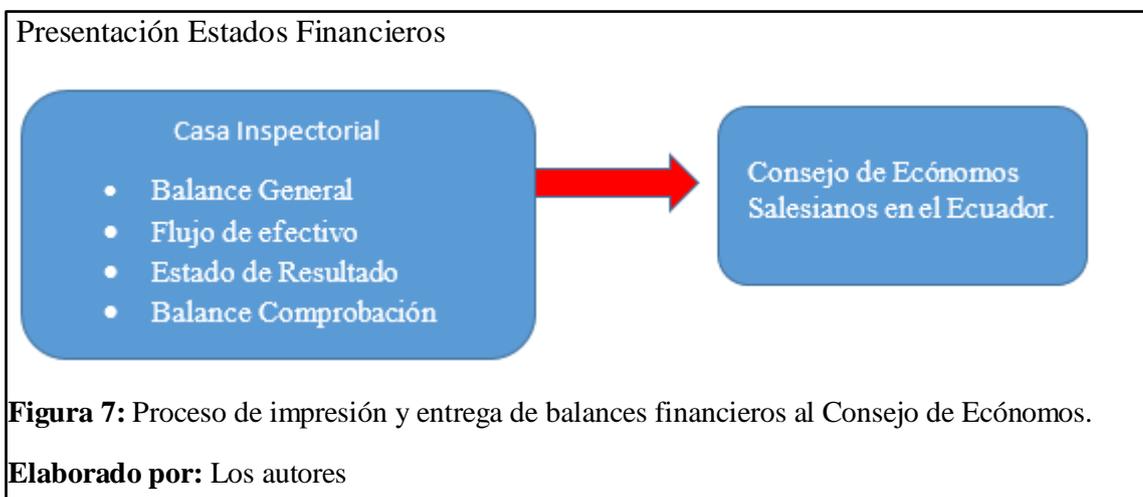
Proceso 4.- En la casa Inspectorial la contadora general, consolida los archivos Excel enviados por cada una de las casas y obras salesianas.



Proceso 5. Consolidación de los archivos Excel, se procede con la generación de cada uno de los reportes contables detallados.



Proceso 6. Generado los estados financieros la contadora general pone en conocimiento al consejo de ecónomos Salesianos en el Ecuador para su respectiva toma de decisiones.



## **1.2 Marco Teórico**

### **1.2.1 Software empresarial y de pequeñas empresas**

En las últimas décadas las pequeñas y medianas empresas del Ecuador se ven en la necesidad de tener alguna herramienta informática, los programas "Small business server" y "Office xp", son sistemas de facturación que cuenta con características para gestionar, controlar, presupuestar y facturar los bienes o servicios que brindan a sus clientes, esto dentro del negocio ayudará a mantener la gestión empresarial al día. Esta es una herramienta básica que asegura el mejor funcionamiento del negocio, además ayuda a tener todos sus movimientos y registros de información ordenados y disponibles en cualquier momento, de esta manera sirve además como guía importante para la toma de decisiones.

Estos sistemas se encuentran orientados a todo tipo de empresas entre los principales servicios que brindan son: contabilidad, finanzas, compras, ventas, inventarios, logística entre otras, estas soluciones pueden contar con profundas capacidades especializadas desde el soporte a la planificación, la gestión y el análisis empresarial.

Se debe implementar la gestión financiera para tener el control general sobre la contabilidad y finanzas, esta gestión financiera se divide en dos módulos principales. El módulo de contabilidad permite la gestión global del propio entorno contable, especialmente: contabilidad general, fiscalidad, contabilidad analítica, contabilidad presupuestaria. En segundo lugar, el módulo de tesorería agrupa todas las funciones relacionadas con las cuentas por cobrar y las cuentas por pagar. (DBC, 2019)

Este tipo de software tiene características, ventajas y aplicaciones que permite la comunicación entre módulos, centralizar información, contacto instantáneo, evitar

duplicidad de información, entre otras. Además provee ciertas herramientas de productividad, control de tiempo, agenda y comunicación. (Salazar, 2019)

#### **1.2.1.1 Gestión financiera**

Este tipo de sistemas permite transformar el desempeño de la empresa, se tiene una vista en vivo de todos sus números y análisis de datos, también se puede obtener la información relevante que se necesite para ofrecer conocimiento estratégico basado en datos, además permite crear procesos más eficientes y que se entregue informes y análisis robustos.

Entre sus principales características tenemos.

- Adaptarse a los requisitos empresariales cambiantes al adaptar el plan de cuentas, las dimensiones y las reglas.
- Proporcionar controles mediante la automatización de procesos para el control y planificación presupuestaria.
- Realizar un seguimiento eficiente de los valores que reduzca los errores de informes mediante la administración automática.
- Cumplir con los requisitos del sector público para gravámenes y presupuestos.
- Impulsar la estrategia corporativa y el crecimiento.
- Disminuir las deudas mediante una gestión eficiente de la recaudación.
- Obtener información financiera inmediata. (NOVATECH, 2015)

#### **1.2.1.2 Gestión contable**

Se puede confiar en este tipo de software para utilizar en informes específicos de permitan realizar un seguimiento de los problemas contables, a fin de controlar mejor su flujo de caja. El software de gestión financiera y contable puede aumentar la velocidad y la eficiencia del departamento de contabilidad a través de la

automatización. Otra función importante del sistema de contabilidad es asegurar una correcta tributación y el cumplimiento de las leyes tributarias, también se prioriza la optimización de procesos, estos sistemas contables están diseñados para facilitar los procesos contables y la gestión financiera. (myabcm, 2018)

Sus características incluyen una mayor precisión y eficiencia debido a que las actividades y transacciones contables se registran directamente en el sistema sin perder ni duplicar información, reduce en gran medida la posibilidad de errores, también proporciona información específica sobre el estado financiero de la empresa, además muchas de estas herramientas brindan otras opciones útiles para analizar y presentar estos datos, implementar normativas o generar informes, y también pueden ahorrar más costos, y pueden reducir el uso de software y el gasto económico de acuerdo con las características de la empresa y dedica tiempo a la gestión financiera. Por otro lado, dado que la herramienta está diseñada para facilitar el cumplimiento de las obligaciones administrativas y mantener un control preciso sobre la cuenta, facilitando así la toma de decisiones, uno de los principales usos de la herramienta es reducir posibles multas legales, es una ventaja porque brinda información clave sobre la situación económica de la empresa y puede ayudar a los gerentes a tomar decisiones sobre acciones futuras. (SOFINA, 2020)

Es así que bajo estos criterios se debería contar con dos tipos de herramientas para que puedan cumplir con los requerimientos solicitados por la casa inspectoral salesiana, en nuestro caso se desarrollara un sistema que integre las dos tipos de sistemas que además utilice herramientas de desarrollo de software que sea de libre

utilización y que no pueda generar costos de licencias, patentes, mantenimientos y ataduras con los proveedores de estos sistemas.

En nuestro caso se trabajara con una aplicación con el cual el personal de la institución ya está familiarizado y con el cual han venido trabajando varios años por que la adaptación y la utilización serán simple y amigable para los usuarios que lo administraran.

### **1.2.2 Herramientas de desarrollo**

Considerando la descripción de los conceptos relacionados con cada concepto desde la perspectiva de la aplicación, las siguientes herramientas son fundamentales para trabajar en el proyecto, esto permite conocer con más detalle las consideraciones y herramientas que se tomaron en cuenta para utilizar en el desarrollo del mismo.

Así mismo se verificó la disponibilidad y la compatibilidad entre las herramientas seleccionadas para el proyecto con el módulo a desarrollar, que sean de amplia capacidad operativa y de uso libre sin generar altos costos en la implementación del sistema.

A continuación se detalla las versiones de cada software.

- Framework PrimeFaces
- GlassFish ver.4.1.13
- postgresSQL ver.9.4
- PgAdminIII ver.3
- Netbeans ver. 8.1
- JDK (Java SE Development Kit) ver.7
- Power Designer ver. 16.1

### **1.2.2.1 Framework Jd Faces**

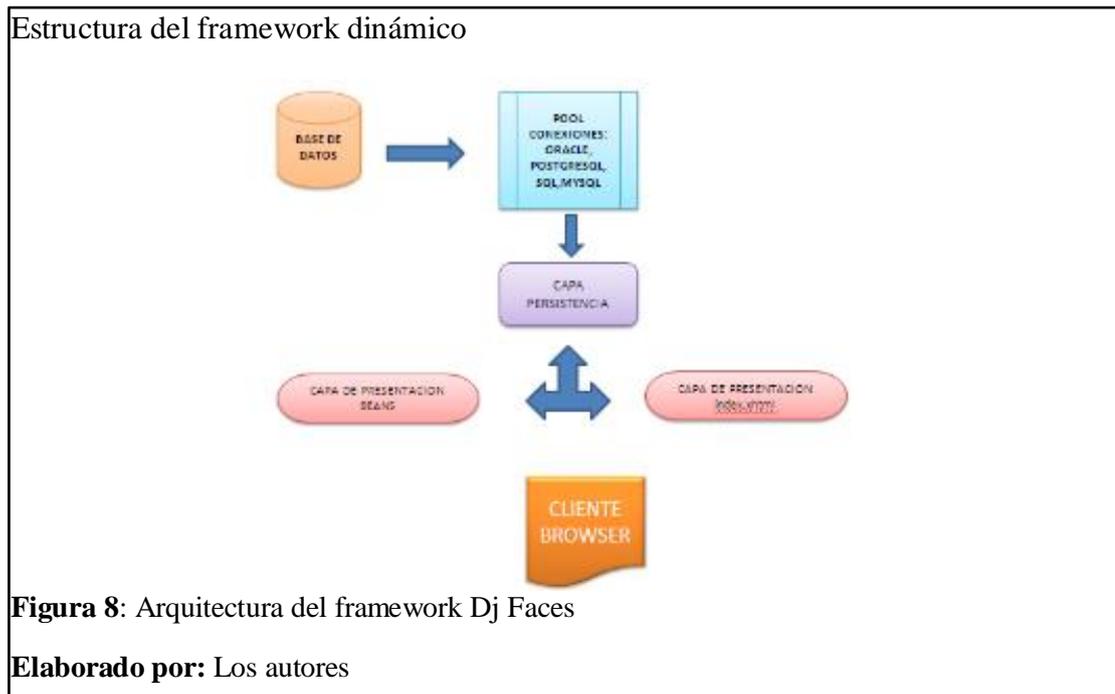
El sistema se desarrolló heredando y personalizando los componentes JSF de Primefaces que actualmente para el desarrollo de aplicaciones web es una de las mejores tecnologías, Primefaces tiene 117 conjuntos de componentes, además de conjuntos de componentes estándar (como editor Html, gráficos, calendario de fechas y Excel, PDF, exportador de datos de Word), también incluye otros componentes. También proporciona funciones desde aplicaciones de escritorio hasta aplicaciones web, como arrastrar y soltar, menús contextuales y enviar mensajes a los usuarios en forma de cuadros de diálogo. Una característica importante del marco es que evita que los desarrolladores tengan que ejecutar cualquier tipo de programa relacionado con AJAX, y el marco puede manejarlos de forma transparente. (javaHispano, 2012)

### **Framework Dinámico**

Está desarrollado bajo java y conteniendo las librerías de primefaces, para seguridades de autenticación spring framework security.

Es un framework de desarrollo nacional que brinda todas las características de contenidos Primefaces, para aplicaciones webs y dispositivos móviles.

El framework proporciona una fácil administración de aplicaciones webs, tal como una reducción completa de código fuente para la implementación de una página, ya que todo se encuentra en un contenedor genérico, el mismo que solo se configura desde la aplicación sin necesidad de tener conocimientos avanzados en programación o desarrollo de aplicaciones webs. (Sierra, F., Acosta, J., Ariza, J., & Salas, M., 2017)



### 1.2.2.2 Base de datos Postgress

PostgreSQL es un sistema de administración de bases de datos relacionales que está orientado a objetos, multiplataforma y de código abierto, este sistema open source se utiliza para administrar bases de datos convencionales y bases de datos relacionales (RDBMS). (Gil, 2018)

Este tipo de base de datos en su mercado actual mantiene estándares de alta tecnología y el potencial efectivo para procesar grandes cantidades de información ha mejorado por completo la concurrencia que se distingue por su rica lista de funciones, también permite competir con cualquier DBMS comercial, tiene un amplio conjunto

de tipos de datos y permite que sus extensiones sean definidas por el usuario y programables. Se puede confirmar en los siguientes puntos qué organizaciones eligen usar PostgreSQL, resulta que es el mejor DBMS para cualquier situación y va mejorando cada vez en sus nuevas versiones. (Henríquez,N.Iglesias,A.Amaris Ramos,L.Ropain,Y., 2017).

### **1.2.2.3 GlassFish**

GlassFish es un servidor de aplicaciones open source creado por Sun Microsystems para la plataforma JavaEE y ahora está patrocinado por Oracle Corporation. La versión admitida es Sun GlassFish Enterprise Server tiene una licencia doble de software libre: Licencia de desarrollo y distribución común (CDDL) y Licencia pública general GNU (GPL), excepto para la ruta de clase. GlassFish es una implementación de referencia de JavaEE, por lo que es compatible con EJB, JPA, JavaServerFaces, JMS, RMI, JavaServer Pages, Servlet, etc. Esto permite a los desarrolladores crear aplicaciones empresariales portátiles y escalables, y que se integran con las tecnologías; los componentes opcionales también se pueden instalar para servicios adicionales. (Andrango,B.Jácome,D., 2013)

### **1.2.2.4 Power Designer**

Power Designer es un conjunto de herramientas de modelado que combina diferentes tecnologías estándar: modelado de aplicaciones a través de UML, tecnología de modelado de procesos de negocios y tecnología de modelado de bases de datos tradicional. (Quezada, 2008)

Power Designer es una herramienta que le permite crear y administrar modelos de datos conceptuales, administrar y generar modelos de datos físicos, generar scripts de creación de bases de datos, sincronizar modelos de datos físicos con bases de datos,

usar MetaWorks para administrar modelos grandes y generar información detallada como disparadores. Implementación de procesos y vistas, ingeniería inversa de bases de datos existentes para crear modelos físicos, conceptuales y orientados a objetos, y finalmente crear informes sobre modelos de Power Designer. Además, la interfaz es muy amigable y fácil de usar con las herramientas incluidas en Power Designer. (Novalys, 2019)

#### **1.2.2.5 Java**

Java es un lenguaje de programación orientado a objetos, java también es de código abierto, java es gratuito y otros desarrolladores pueden usar y modificar el código fuente. Además también se puede desarrollar aplicaciones web o fuera de línea para cualquier tipo de sistema operativo o Sistemas como Windows, Linux y Mac OS.

Java es un lenguaje de programación y como plataforma informática lanzada por primera vez por Sun Microsystems en 1995. A menos que instale java y cree más aplicaciones y sitios web todos los días no funcionarán, java es rápido, seguro y confiable para utilizar desde computadoras portátiles hasta centros de datos, desde consolas de juegos hasta supercomputadoras, desde teléfonos móviles hasta internet, java está en todas partes. (ORACLE, 2020)

#### **1.2.2.6 JDK**

JDK es el Java Development Kit, que se traduce al español, es decir Java Development Tools, en definitiva es un conjunto de herramientas (librerías y programas) que te permiten desarrollar en el lenguaje java. Cuando hablamos de desarrollo incluye compilar, ejecutar, generar documentación, etc. Puede definirse como un conjunto de herramientas, utilidades, documentos y ejemplos utilizados para

desarrollar aplicaciones java, que constan de una serie de aplicaciones y componentes para realizar todas las tareas que puede realizar. (ROSERO, 2020)

### 1.2.2.7 Netbeans

Netbeans es un entorno de desarrollo integrado gratuito que se utiliza principalmente en el lenguaje de programación java. Hay muchos módulos para ampliarlo. Netbeans IDE2 es un producto gratuito sin restricciones. En resumen, Netbeans no es un lenguaje de programación, se considera un "entorno de desarrollo" lo que significa que es un programa, una especie de software, en el que los programadores pueden ejecutar proyectos de una determinada manera para desarrollar programas utilizando un lenguaje de programación específico. Es más fácil y rápido que la forma tradicional es decir, usando un editor de texto. (netbeans, 2020)

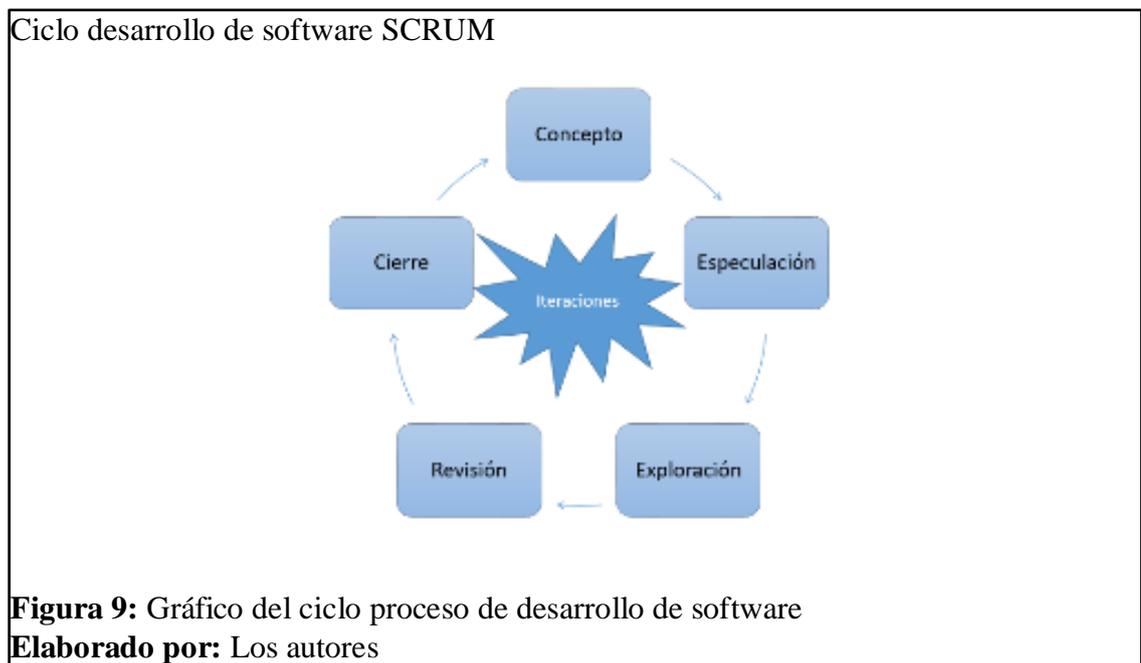
### 1.2.2.8 Scrum

El concepto de esta metodología en (Trigás, 2012) indica que Takeuchi y Nokana publican un artículo donde dan a conocer que para gestionar proyectos de desarrollo de software existe una nueva forma con características de metodologías ágiles y flexibles con entornos cambiantes que trabajen de manera colaborativa con características de auto organización, control moderado en la transmisión del conocimiento y que manejen incertidumbre.

En SCRUM es necesario tener el conocimiento de las 5 fases del ciclo de desarrollo de software.

- **Concepto:** Definiciones de las características del producto a desarrollar y se forma un equipo de trabajo.
- **Especulación:** Revisar los requisitos generales, costos, agendas de trabajo y como se construirá el producto.

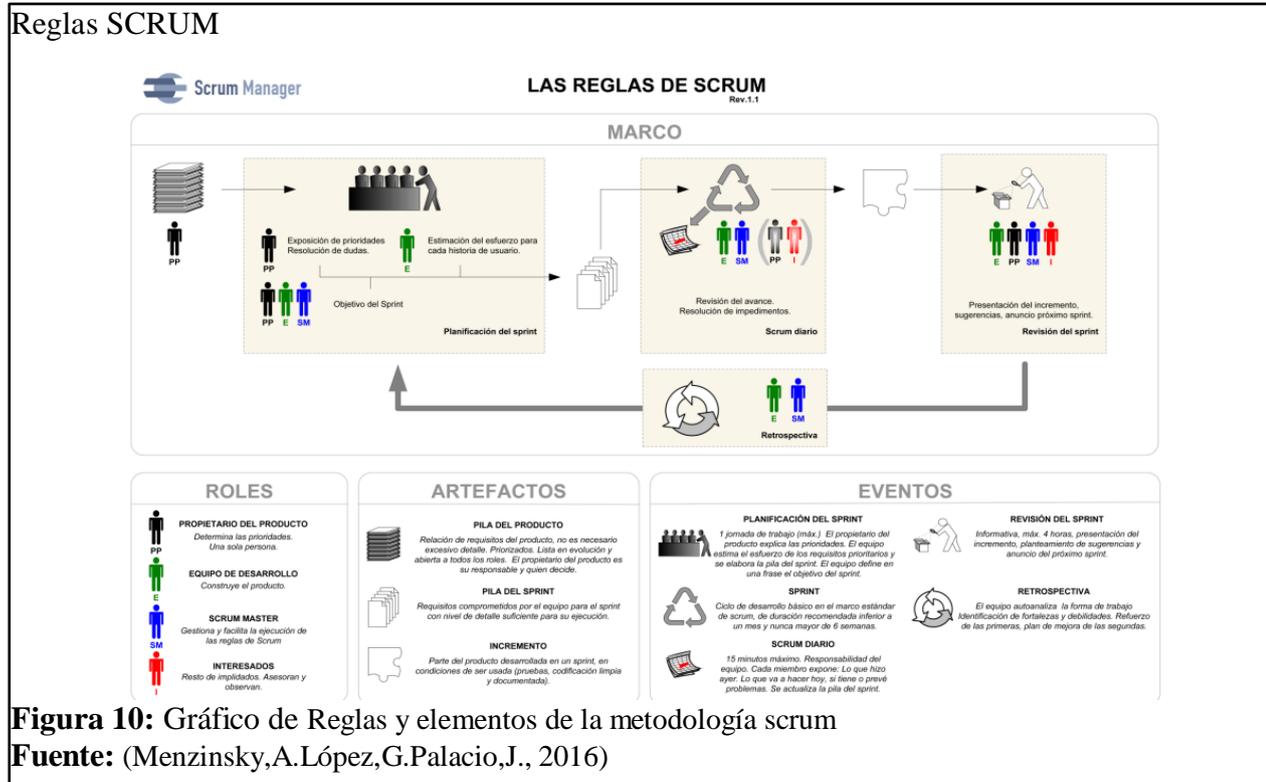
- **Exploración:** Se incluyen las funcionalidades de la fase de especulación para incrementar al producto.
- **Revisión:** Informe de avances del proyecto según lo planificado contra los trabajos realizados.
- **Cierre:** Este cierre no indica que se ha finalizado el proyecto, se seguirán realizando cambios a los cuales se les denomina mantenimiento. (Trigás, 2012)



En conclusión nos ayuda para obtener los mejores resultados posibles al terminar el proyecto, además que tenga principios de imposición, adaptación y facilite el trabajo en equipo. Los procesos que se van a llevar a cabo se dividirán en sprint esto facilitará la entrega en tiempos cortos y fijos y que cada iteración proporcionara un resultado completo.

## Reglas de SCRUM

En la siguiente imagen se indica las reglas básicas de funcionamiento del marco de trabajo de scrum que fue necesario para el desarrollo del proyecto.



## Scrum técnico

A continuación se detalla el marco técnico de scrum.

### Roles

En los roles básicamente consta la relación de los miembros que intervienen en el desarrollo del proyecto.

### Dueño del producto

El propietario del producto o también llamado product owner es la persona encargada del desarrollo de todo el proyecto, es quien será responsable de tomar las mejores decisiones para todo el equipo de trabajo y que estará en contacto con el cliente. Para que el miembro del equipo sea idóneo para este puesto debe tener varios conocimientos propios del mismo entre ellos es conocer el modelo de negocio del

cliente las necesidades y sus objetivos, debe saber priorizar todas las actividades de forma efectiva para optimizar recursos y tener la capacidad de tomar sugerencias por parte de los demás miembros del equipo. (Menzinsky, A. López, G. Palacio, J., 2016)

### **Equipo de desarrollo**

A este grupo de miembros del equipo se les denomina team, la recomendación para formar los equipos scrum es que no tengan más de 9 personas por equipo pero tampoco pueden tener menos de 3 personas, esto es básicamente por la forma en que se desarrollara el proyecto ya que al ser un grupo grande dificulta de gran manera en la dinámica del grupo y obviamente hace que se vuelva muy complicada la comunicación directa entre todos al mismo tiempo. Una gran ventaja es que es un equipo muy funcional ya que todos trabajarán de forma solidaria y tendrán responsabilidades compartidas eso dará resultado en que todos se apoyaran mutuamente y tratarán de ayudar para cumplir las múltiples actividades. (Menzinsky, A. López, G. Palacio, J., 2016)

### **Scrum master**

Esta persona es la encargada de dar el cumplimiento de todas las reglas del marco técnico de scrum dentro del desarrollo del proyecto para que todo el equipo entienda la organización y trabajen conforme a ello. Las habilidades principales del scrum master es que proporcione una asesoría y formación a todo el equipo para que puedan trabajar de forma organizada y de forma responsable, será quien revise y valide la pila del producto, sea el moderador de las reuniones, y que sepa gestionar las dificultades que puedan aparecer dentro del equipo. (Menzinsky, A. López, G. Palacio, J., 2016)

## **Artefactos**

Son todos los elementos físicos que se producen a través de la aplicación de scrum en el desarrollo de software.

## **Pila del producto**

Constan de todos los requisitos del cliente o propietario del producto donde indica todos sus requerimientos que crea necesarios para el mismo, esto se da en una lista de forma ordenada.

La característica de la pila del producto es que nunca se da por terminada ya que su trabajo se lo mide con la continua evolución del producto y desarrollo del proyecto, para realizar este elemento se realiza en primera instancia la preparación que será la priorización de las actividades y estimar los esfuerzos necesarios para cumplir las tareas que serán continuas y de manera colaborativa con el propietario del producto, luego se dará el formato de la pila del producto para scrum se prefiere tener una comunicación verbal y de visualización directa. (Menzinsky,A.López,G.Palacio,J., 2016)

## **Pila de sprint**

En la reunión de planificación de sprint el equipo indicará una lista de todas las tareas que sean necesarias para la construcción de las historias de usuarios, que sean de tamaño adecuado para poder medir de forma diaria el avance de los mismos. Los soportes y formatos más habituales son 3; un tablero físico o pizarra, las hojas de cálculo Excel y las herramientas informáticas para la gestión de proyectos. (Menzinsky,A.López,G.Palacio,J., 2016)

## **Incremento**

La parte del producto debe estar terminada y probada para ser considerada como incremento, esto se refiere a funcionalidades entregables y no que sean prototipos o partes independientes de integración.

## **Eventos**

Son eventos con la finalidad de describir todas las reuniones del equipo

## **Sprint**

La duración no puede ser mayor a 4 semanas dependiendo el tipo de incremento del producto que se esté desarrollando, esto es considerado como el evento contenedor de todos los eventos.

## **Reunión de planificación de sprint**

Esta reunión puede durar hasta una jornada de trabajo donde el scrum master será quien sea el moderador y deben asistir todos los miembros del equipo, la reunión se realiza antes de iniciar cada sprint y ayuda en que el propietario del producto valide que se cumpla con todos los requerimientos solicitados.

(Menzinsky,A.López,G.Palacio,J., 2016)

## **Scrum diario**

Es una reunión en la que el equipo organiza las actividades para el siguiente día de trabajo, el tiempo de duración de esta reunión breve no será más de 15 minutos, esto ayuda para saber cómo se está avanzado en el desarrollo del producto cuales pueden ser las posibles complicaciones que tienen y que con la comunicación todos estén enterados las tareas que ya se cumplieron y las que faltan por terminar.

## **Revisión del sprint**

Esta es una reunión que se debe realizar al final de cada sprint que su duración será menos 4 horas, a esta reunión pueden asistir también los ingenieros y

programadores para que puedan enterarse de las tecnologías con las que se está trabajando, además pueden ayudar con un feedback para la revisión de la pila del producto. (Menzinsky,A.López,G.Palacio,J., 2016)

### **Retrospectiva del sprint**

Esta reunión se realiza después de revisar cada sprint y antes de realizar la planificación del siguiente, el equipo aquí realizara un autoanálisis de la forma en que están trabajando para identificar sus debilidades y fortalezas. El objetivo de esta reunión es verificar si hay problemas y dar soluciones a los posibles errores, al analizar cada sprint se realiza las siguientes preguntas “QUÉ” se está construyendo, “CÓMO” se lo está construyendo: “CÓMO” se está trabajando.

(Menzinsky,A.López,G.Palacio,J., 2016)

#### **1.2.2.9 Jmeter**

JMeter es un software de código abierto esta aplicación de escritorio está desarrollado en lenguaje java puro al 100%, fue creado por Stefano Mazzocchi de la Apache Software Foundation, su fin es para utilizar en los aplicativos web y servidores por medio del protocolo HTTP al realizan pruebas de carga y stress, al ser muy usado actualmente incluye para otros tipos de protocolos es por eso que se ha convertido en la herramienta más buscada y utilizada para trabajar y medir los desempeño los sistemas informáticos. Jmeter trabaja de manera que simula todas las peticiones que pueda recibir al protocolo HTTP y no es un navegador ni tampoco compila el código de programación. (JMeter, 2020)

Actualmente esta herramienta se utiliza para realizar pruebas funcionales, conexiones de bases de datos, prueba con el servidor web, entre otras más. Las pruebas de stress y carga nos ayudan a conocer la capacidad máxima de procesamiento de datos con esto se puede determinar que causas puedan condicionar su bajo rendimiento, su

procedimiento de trabajo se base en 3 fases: para la instalación desde su página de Apache Jmeter procedemos con la descarga del software, en la ejecución no es más que ingresar a su carpeta bin y ejecutar el archivo .jar, para realizar las pruebas debemos simular un grupo de hilos creando un plan de pruebas donde al ejecutar simularemos la cantidad de peticiones y el comportamiento de los usuarios dentro del sistema. (Avalos, 2019)

## CAPÍTULO II

### 2 ANÁLISIS Y DISEÑO

#### 2.1 Análisis de requerimientos

**Tabla 2:**

Requerimientos funcionales

LISTADO DE REQUERIMIENTOS FUNCIONALES	
RF ID	Descripción
RF-S001	Carga de balances en formato xls, de las casas y obras que no disponen de acceso al sistema RUA.
RF-S002	Almacenar los balances de las casas u obras que no disponen de acceso al sistema RUA, año vigente y años anteriores.
RF-S003	Permitir la sustitución de un balance de las casas u obras que no disponen de acceso al sistema RUA.
RF-S004	Almacenar usuario, fecha, hora del usuario que realice la carga del balance de comprobación, en el caso que una casa u obra salesiana no tenga instalado el sistema contable RUA.
RF-S005	Validar los balances de las casas y obras que no disponen de acceso al sistema RUA, que los mismos se encuentren en el catálogo de cuentas a nivel 5.
RF-S006	Permitir que la generación de reportes y estados financieros sean visibles, de acuerdo al perfil que el administrador del sistema.
RF-S007	Al imprimir los reportes y estados financieros, que estos impriman con sus respectivos logotipos y datos de acuerdo su casa u obras.
RF-S008	Permitir generar el balance de comprobación entre dos niveles
RF-S009	Disponer un catálogo de cuentas presupuestario y contable unificado a nivel 5
RF-S010	Generar balance general consolidado.
RF-S011	Generar balance general por casas y obras.
RF-S012	Generar balance de comprobación consolidado.
RF-S013	Generar balance de comprobación por casas y obras.
RF-S014	Generar estado de resultados consolidado.
RF-S015	Generar estado de resultados por casas y obras
RF-S016	Generar cédula presupuestaria de ingreso y gastos consolidados
RF-S017	Generar cédula presupuestaria de ingreso y gastos por casas y obras
RF-S018	Generar consolidación presupuestaria vs contable general
RF-S019	Generar consolidación presupuestaria vs contable por casas y obras
RF-S020	Generar reporte flujo del efectivo.

**Nota:** Listado de Requerimientos funcionales iniciales.

**Elaborado por:** Los autores

**Tabla 3:**

## Requerimientos no funcionales

LISTADO DE REQUERIMIENTOS NO FUNCIONALES	
RNF ID	Descripción
RNF-S001	El acceso al módulo y su funcionalidad debe utilizar tipos de datos estándar de ancho de banda e internet.
RNF-S002	La navegación e interacción del módulo con el usuario debe ser amigable para el uso del mismo menú, barra de herramientas, accesos directos.
RNF-S003	Los mensajes de errores deberán estar en el idioma apropiado y entendible que deberá ser reportado por la misma aplicación.
RNF-S004	El módulo deberá extraer información de los sistemas RUA conectados en tiempo real y dar respuesta en el menor tiempo posible desde su base de datos.
RNF-S005	Realizar un manual de usuario detallando el uso del módulo con todas sus herramientas y funciones, además será versionable para corregir posibles errores y añadir nuevas funcionalidades.
RNF-S006	La información estará protegida para el usuario de la casa inspectorial en contra de accesos no autorizados utilizando validaciones a nivel de acceso, roles y perfiles.
RNF-S007	El módulo cumplirá con los lineamientos políticos del ente regulador que utilice este sistema.
RNF-S008	El módulo tendrá tolerancia a fallos y las operaciones que realice este deben ser transaccionales.
RNF-S009	El módulo será multiplataforma que podrá ejecutarse en diferentes sistemas operativos Windows, Linux, etc, también tendrá la capacidad para guardar en diferentes motores de base de datos todos sus registros.

**Nota:** Lista de requerimientos no funcionales iniciales

**Elaborado por:** Los autores

## 2.2 Artefactos

### 2.2.1 Diagramas de casos de uso

#### 2.2.1.1 Historias de usuario

**Tabla 4:**

Historia usuario administrador

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 1	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Parametrizar catálogos contables, presupuestarios, períodos fiscales anuales y mensuales.	
<b>Prioridad en negocio:</b> Alta	
<b>Programador responsable:</b> Cristian Pillajo	
<b>Descripción:</b> El sistema debe permitir registrar, modificar y eliminar los catálogos contables, presupuestarios, períodos fiscales anuales y mensuales.	
<b>Validación:</b> El usuario que accede y tenga el perfil de administrador tendrá la potestad de habilitar y deshabilitar los períodos fiscales anuales y mensuales.	
<b>Flujo Normal</b> <ol style="list-style-type: none"><li>1. El usuario administrador accede a la opción de parametrizar los catálogos contables, presupuestarios, habilitar y deshabilitar períodos fiscales tanto anuales como mensuales en el sistema.</li><li>2. El sistema carga el formulario en pantalla con los datos a parametrizar.</li><li>3. El usuario realiza los registros, modificaciones y eliminación de las opciones de parametrización contable y presupuestario en el sistema.</li><li>4. El sistema guarda los cambios que van a ser aplicados y presenta un mensaje de proceso satisfactorio.</li></ol>	
<b>Flujo Alternativo</b> <ol style="list-style-type: none"><li>2.a El sistema no carga el formulario de parametrización tanto contable como presupuestario e informa al usuario a través de un mensaje de alerta que existe un error en el mismo.</li><li>3.a En caso de cancelar acción el sistema va a finalizar toda acción sin guardar cambio.</li><li>4.a En caso que la base de datos de error de conexión el sistema muestra un mensaje al usuario.</li></ol>	

**Nota:** Detalla la historia de usuarios del usuario administrador del sistema

**Elaborado por:** Los autores

**Tabla 5:**

Historia de usuario sistema RUA.

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 2	<b>Usuario:</b> Usuario sistema RUA
<b>Nombre historia:</b> Cierre contable en el sistema RUA, generación de balance general y cédulas presupuestarias.	
<b>Prioridad en negocio:</b> Alta	
<b>Programador responsable:</b> Cristian Pillajo.	
<b>Descripción:</b> El sistema debe permitir generar el balance general y cédula presupuestaria permitiendo ser registrados automáticamente en el módulo gerencial de consolidación.	
<b>Validación:</b> El usuario que accede y tenga el perfil de contador general, debe generar el balance general y cédula presupuestaria, para realizar el traspaso automático del módulo gerencial.	
<b>Flujo Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario del sistema RUA debe realizar la acción de cerrar el período contable.</li> <li>2. La casa y obras salesianas que tengan instalado el sistema RUA realizan un traspaso automático de los balances y cédulas presupuestarias al módulo gerencial de la Casa Inspectorial Salesiana y generar reportes.</li> <li>3. El usuario contador general debe generar en el sistema la ejecución de los balances y cédulas presupuestarias.</li> <li>4. El sistema consolida todos los movimientos contables de todas las Casas y obras salesianas que tengan habilitado el sistema RUA.</li> </ol>	
<b>Flujo alternativo.</b>	
<ol style="list-style-type: none"> <li><b>1.a</b> El sistema muestra un mensaje cuando los usuarios de sistema RUA no cierren sus períodos contables.</li> <li><b>2.a</b> El sistema no encuentra información contable y presupuestaria, mediante un mensaje indicará al usuario que no se puede continuar con el cierre del período fiscal.</li> <li><b>3.a</b> En caso de cancelar acción el sistema va a finalizar toda acción sin guardar cambio alguno.</li> <li><b>4.a</b> El sistema cuando tenga un error al generar las tareas de consolidación muestra un mensaje indicando la razón porque no se pudo generar.</li> </ol>	

**Nota:** Detalla la historia de usuarios del usuario del sistema RUA**Elaborado por:** Los autores

**Tabla 6:**

Historia de usuarios externos

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 3	<b>Usuario:</b> Usuarios Externos
<b>Nombre historia:</b> Carga de archivos Excel del balance general y cédulas presupuestarias de ingreso y gastos.	
<b>Prioridad en negocio:</b> Alta	
<b>Programador responsable:</b> Luis Toapanta	
<b>Descripción:</b> El sistema debe permitir la carga de archivos en formato xls tanto balance general como la cédula presupuestaria para ser registrados en el módulo gerencial de consolidación.	
<b>Validación:</b> El usuario que accede y tenga el perfil de contador general de la casa Inspectorial salesiana, debe realizar la carga del archivo en formato xls el balance general y cédula presupuestaria, para realizar el traspaso automático del módulo gerencial.	
<p><b>Flujo Normal.</b></p> <ol style="list-style-type: none"> <li>1. El usuario de casas y obras salesianas que no tengan instalado el sistema RUA deberán generar y enviar un archivo Excel de los movimientos contables y presupuestarios.</li> <li>2. El usuario contador general de la Casa Inspectorial Salesiana, recibe el archivo de Excel para realizar la carga manual al módulo gerencial.</li> <li>3. El sistema carga en sus registros todos los datos del archivo Excel.</li> <li>4. El sistema consolida los registros del archivo Excel en el módulo gerencial, para la generación de los estados financieros.</li> </ol>	
<p><b>Flujo Alternativo.</b></p> <ol style="list-style-type: none"> <li><b>1.a</b> El sistema no permite archivos de otras extensiones diferentes a Excel.</li> <li><b>3.a</b> El sistema elimina las tareas asociadas en caso de error de generar petición.</li> <li><b>4.a</b> El sistema cuando tenga un error al generar las tareas de consolidación muestra un mensaje indicando la razón porque no se pudo generar.</li> </ol>	

**Nota:** Detalla la historia de usuarios de los usuarios que no tienen el sistema RUA**Elaborado por:** Los autores

**Tabla 7:**

Historia de usuario contador general casa inspectorial

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 4	<b>Usuario:</b> Contador general Casa Inspectorial
<b>Nombre historia:</b> Consolidación de los balances generales y cédulas presupuestarias.	
<b>Prioridad en negocio:</b> Alta	
<b>Programador responsable:</b> Luis Toapanta	
<b>Descripción:</b> El sistema debe permitir consolidar el balance general como la cédula presupuestaria tanto consolidación general como por casa u obras salesianas.	
<b>Validación:</b> El usuario que accede y tenga el perfil de contador general de la casa Inspectorial salesiana, debe realizar la consolidación del balance general y cédula presupuestaria, sea de manera general como por casa u obra salesiana.	
<b>Flujo Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario contador general revisa que todos los sistemas RUA estén cerrados los períodos y los archivo de Excel cargados.</li> <li>2. El sistema realiza la validación del acceso al usuario que va a generar la consolidación de los movimientos contables para que muestre las opciones necesarias.</li> <li>3. El usuario contador general ejecuta en el sistema la acción para ejecutar el proceso automático de consolidación tanto de los balances contables como cédulas presupuestarias.</li> <li>4. El sistema muestra los mensajes de confirmación de proceso concluido con éxito.</li> </ol>	
<b>Flujo Alternativo</b>	
<ol style="list-style-type: none"> <li>1.a El sistema no acepta archivos de otras extensiones diferentes a Excel.</li> <li>2.a El usuario que no tenga el perfil contador general no tendrá acceso al menú de consolidaciones contables.</li> <li>3.a El sistema indica un mensaje en caso de que todos los datos no estén completos para la ejecución del proceso de consolidación.</li> </ol>	

**Nota:** Detalla la historia de usuarios del contador general**Elaborado por:** Los autores

**Tabla 8:**

Historia de usuario contador general casa inspectorial

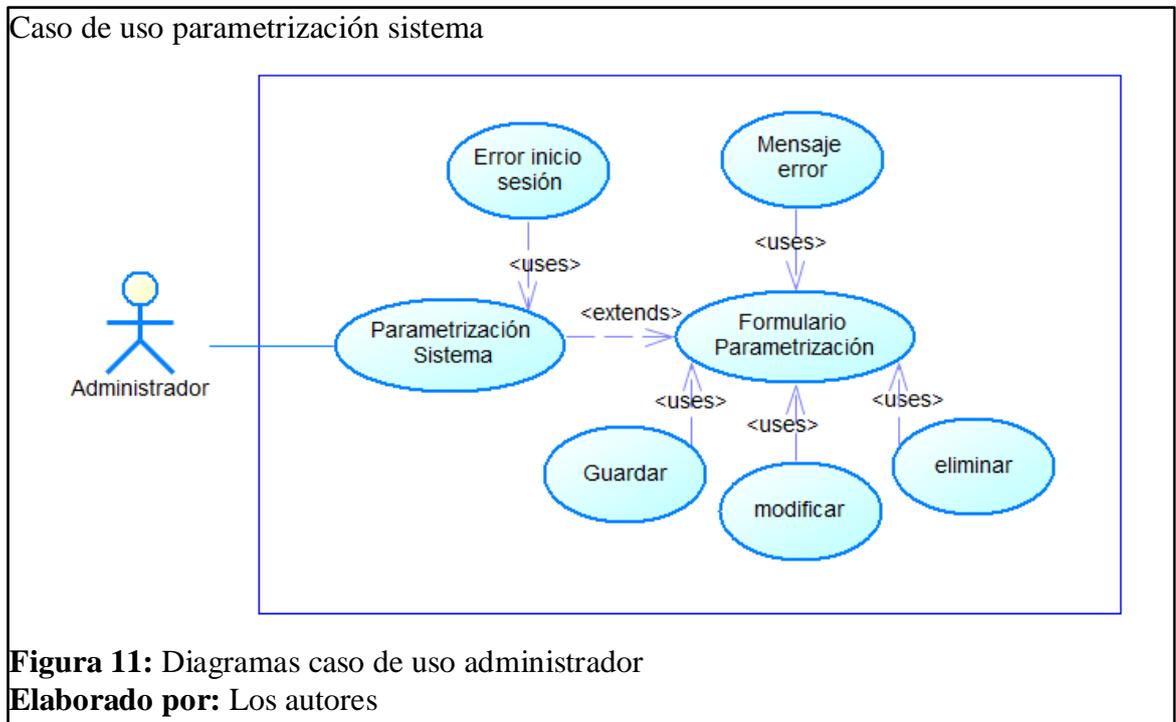
<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 5	<b>Usuario:</b> Contador general Casa Inspectorial
<b>Nombre historia:</b> Generación de estados financieros.	
<b>Prioridad en negocio:</b> Alta	
<b>Programador responsable:</b> Luis Toapanta, Cristian Pillajo	
<p><b>Descripción:</b> El sistema debe permitir la generación de estados financieros como:</p> <p>Estados financieros contables</p> <ul style="list-style-type: none"> <li>- Balance de comprobación.</li> <li>- Estado de resultados.</li> <li>- Flujo de Efectivo.</li> <li>- Balance General.</li> </ul> <p>Estados financieros presupuestarios.</p> <ul style="list-style-type: none"> <li>- Cédula de ingresos.</li> <li>- Cédula de gastos.</li> </ul>	
<p><b>Validación:</b> El usuario que accede y tenga el perfil de contador general de la casa inspectorial salesiana, debe realizar la generación de estados financieros tanto contables como presupuestarios.</p>	
<p><b>Flujo Normal</b></p> <ol style="list-style-type: none"> <li>1. El usuario contador general tendrá el perfil para generación de reportes gerenciales.</li> <li>2. El sistema habilita todas las opciones para que el usuario ejecute la generación de reportes financieros contables y presupuestarios.</li> <li>3. El usuario selecciona la opción que se ejecutará.</li> </ol>	
<p><b>Flujo Alternativo</b></p> <p><b>1.a</b> En caso que el usuario no tenga el perfil de contador gerencial en el sistema no se activan las opciones de reportes.</p> <p><b>2.a</b> En caso que la base de datos de error de conexión el sistema muestra un mensaje al usuario.</p>	

**Nota:** Detalla la historia de usuarios de contador general**Elaborado por:** Los autores

## 2.2.2 Diagramas caso de uso

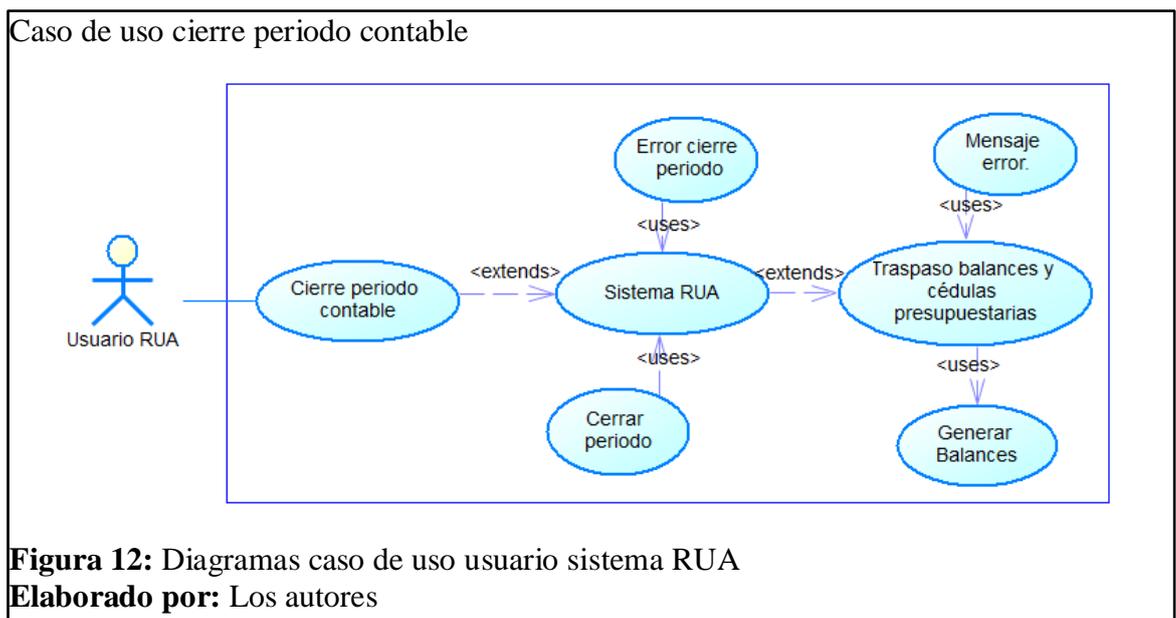
### 2.2.2.1 Caso de uso usuario administrador

El usuario administrador accede y realiza la parametrización en el sistema.



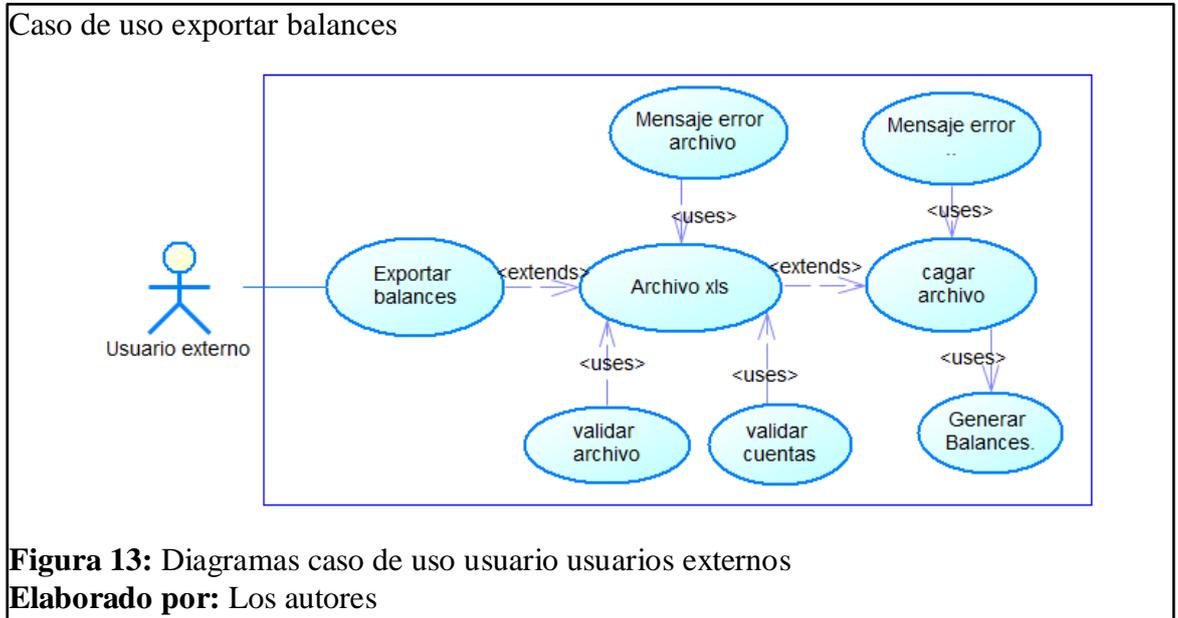
### 2.2.2.2 Caso de uso usuario sistema RUA

El usuario del sistema RUA debe realizar el cierre contable para el envío de los datos al módulo financiero.



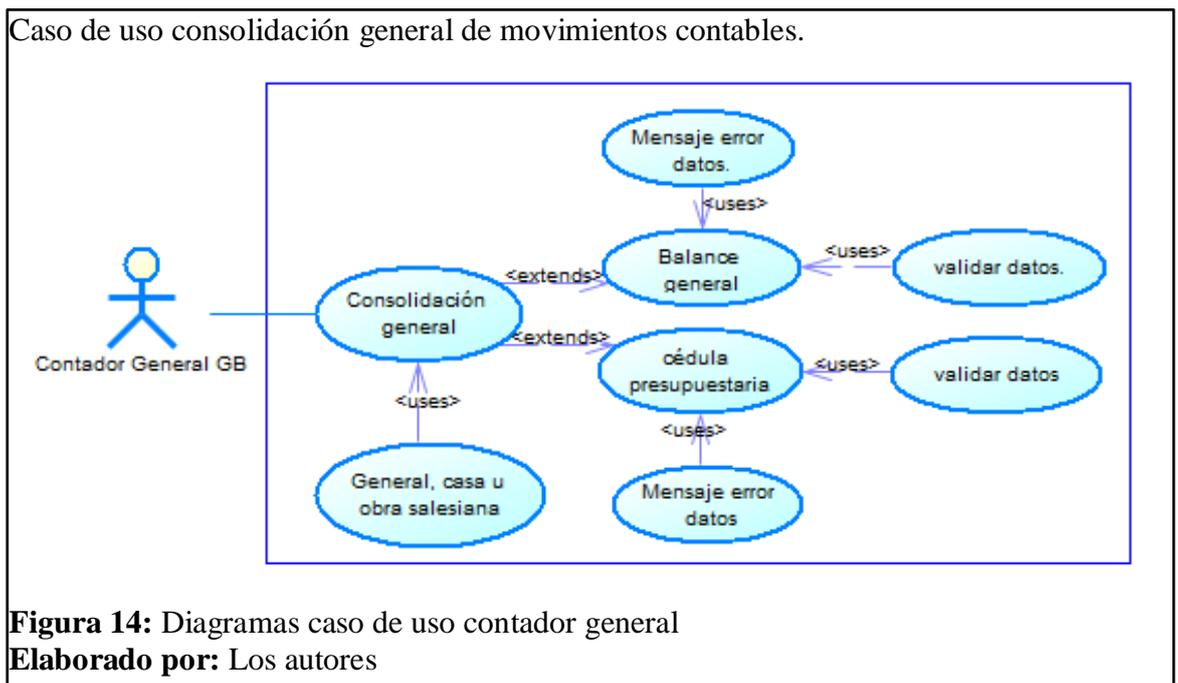
### 2.2.2.3 Caso de uso usuarios externos

Los usuarios externos que no cuenten con el sistema RUA deben enviar el archivo Excel con todos los movimientos contables para que reciba el contador general y cargue manualmente al módulo gerencial.



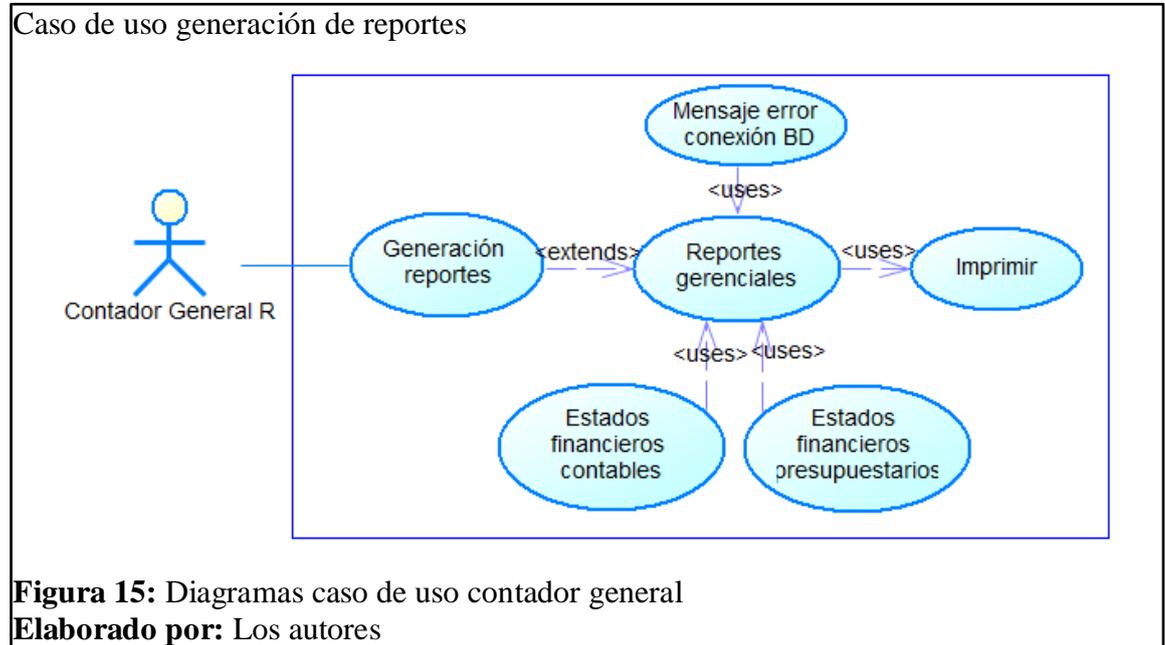
### 2.2.2.4 Caso de uso usuario contador general

El usuario contador general es el encargado de ejecutar las consolidaciones en el módulo.

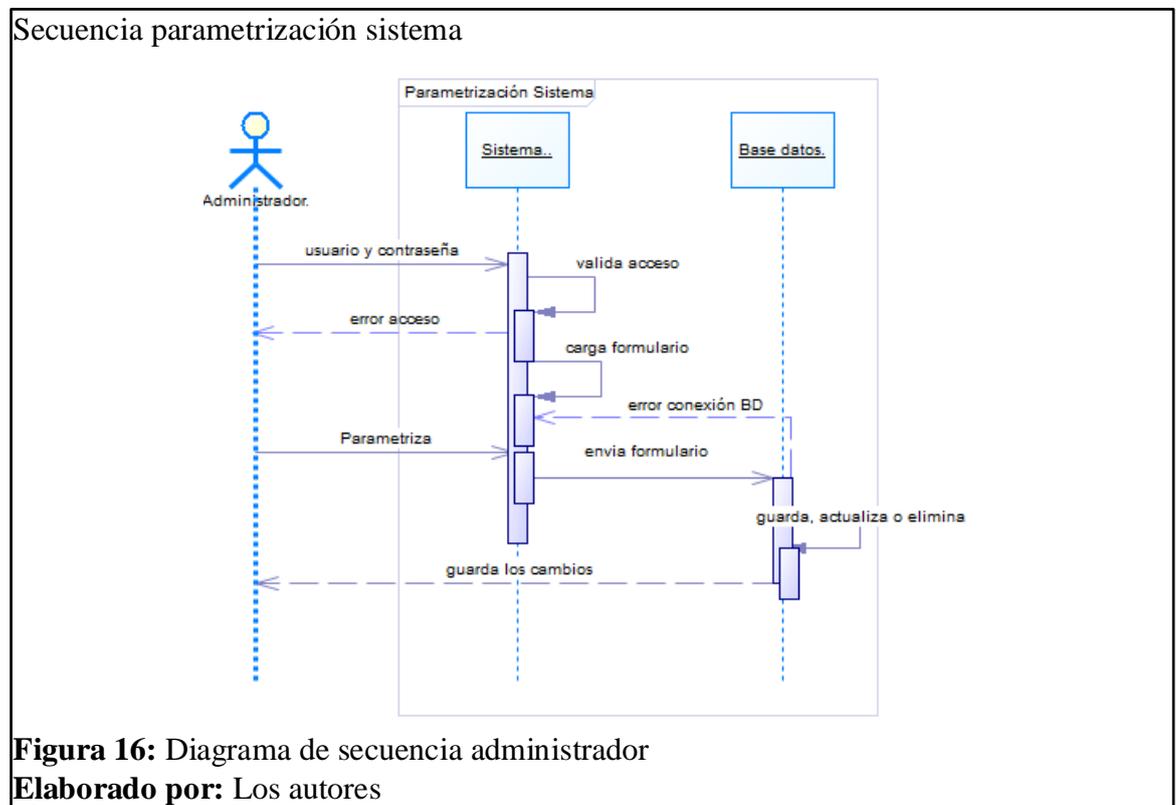


### 2.2.2.5 Caso de uso usuario contador general

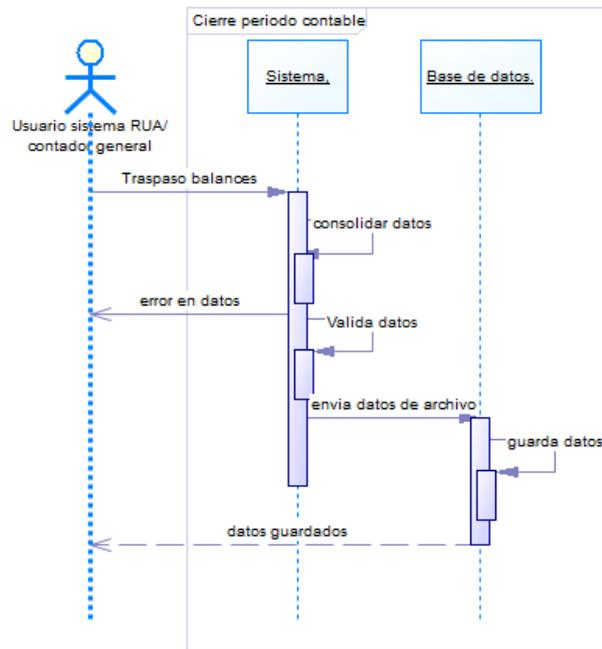
El usuario contador general es el encargado de generar los reportes gerenciales de estados financieros contables y presupuestarios.



### 2.2.3 Diagramas de secuencia



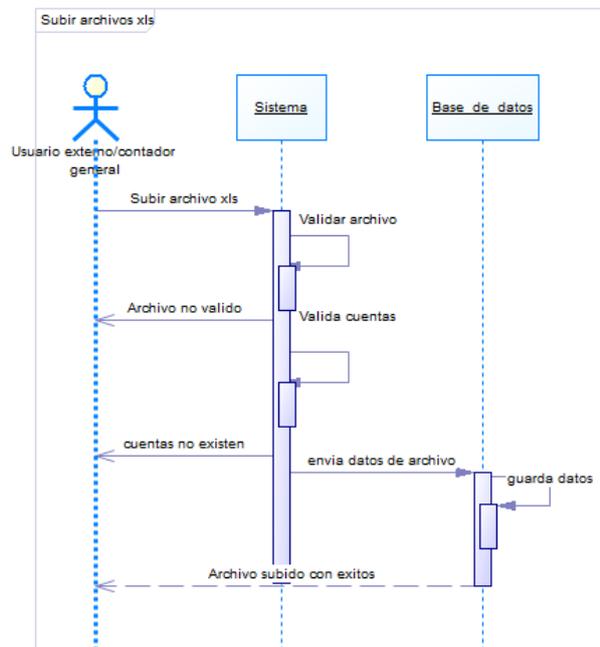
### Secuencia cierre período contable



**Figura 18:** Diagrama de secuencia usuario sistema RUA

**Elaborado por:** Los autores

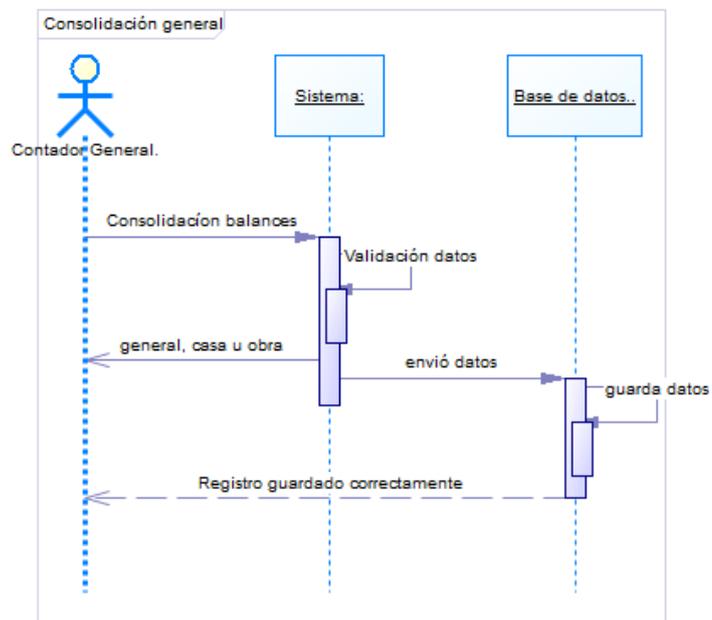
### Secuencia subir archivos Excel



**Figura 17:** Diagrama de secuencia usuarios externos

**Elaborado por:** Los autores

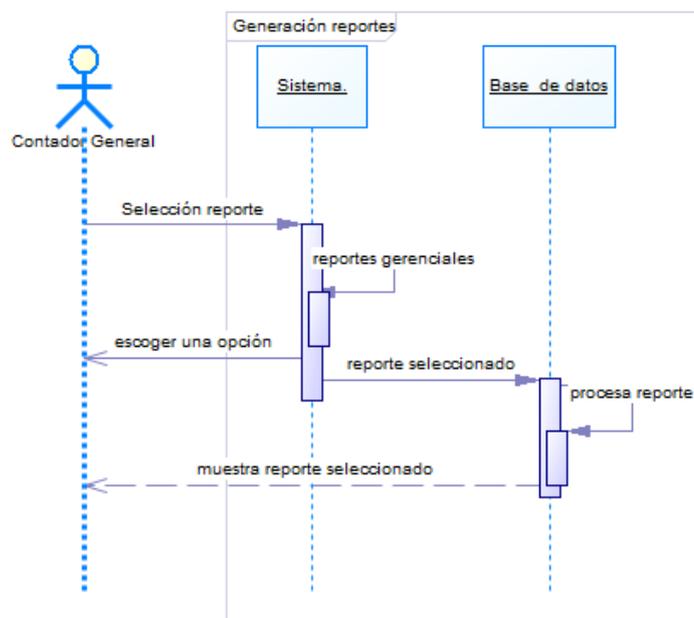
## Secuencia consolidación general balances



**Figura 20:** Diagrama de secuencia contador general

**Elaborado por:** Los autores

## Secuencia generación de reportes



**Figura 19:** Diagrama de secuencia usuario contador general - reportes

**Elaborado por:** Los autores



# CAPÍTULO III

## 3 CONSTRUCCIÓN Y PRUEBAS

### 3.1 Diagrama de clases módulo gerencial financiero

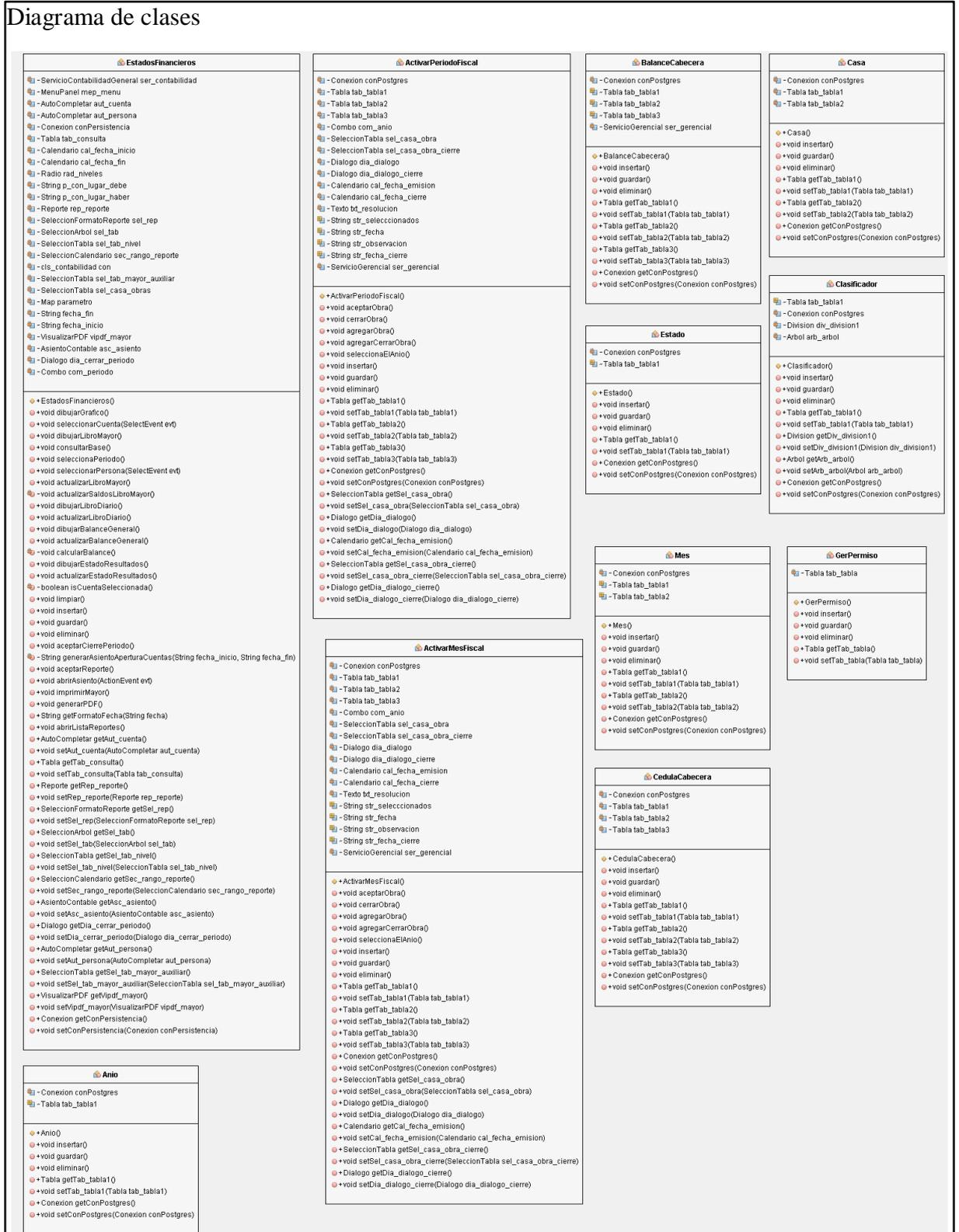
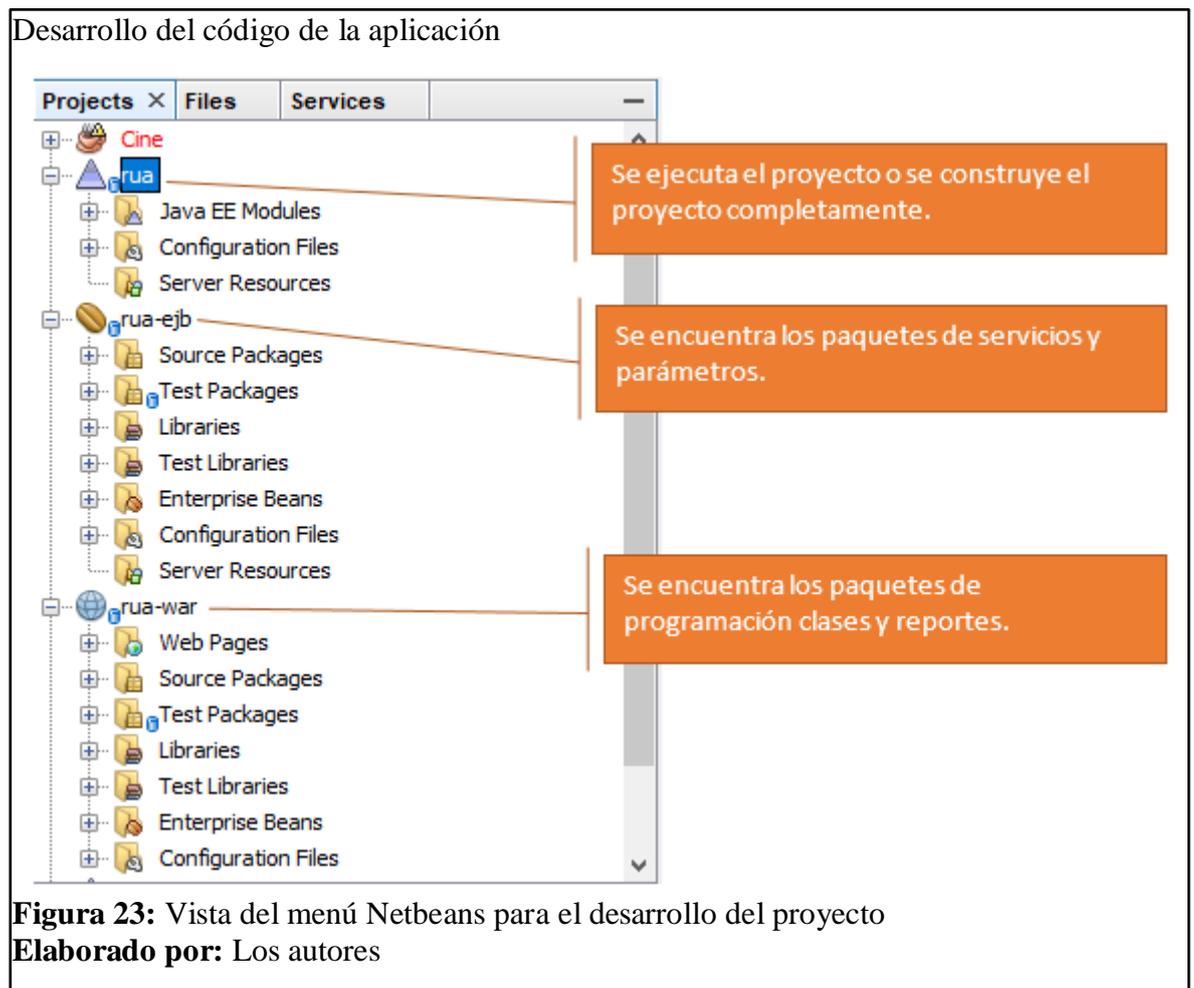


Figura 22: Diagrama clases de módulo gerencial financiero.  
Elaborado por: Los autores

### 3.1.1 Código relevante

#### 3.1.1.1 Versión del proyecto

A continuación, se muestra de forma detallada como se construye el sistema en cuanto al trabajo de programación y ejecución de cada una de las pantallas, mismo que consta de requerimientos, paquetes, clases y complementos para la ejecución del mismo.



### 3.1.1.2 Método activar período contable

Este método tiene la programación para que el sistema permita realizar la apertura y el cierre de los períodos contables en años para cada una de las casas y obras salesianas y posteriormente se pueda consolidar información y generar reportes.

#### Método agregar y cerrar períodos de las casas y obras

```
public void agregarObra() {
    if (cal_fecha_emision.getValue() == null || cal_fecha_emision.getValue().toString().isEmpty()) {
        utilitario.agregarMensajeInfo("ADVERTENCIA", "Ingrese la Fecha de Apertura");
        return;
    } else if (txt_resolucion.getValue() == null || txt_resolucion.getValue().toString().isEmpty()) {
        utilitario.agregarMensajeInfo("ADVERTENCIA", "Ingrese alguna Observación");
        return;
    } else {
        if (dia_dialogo.isVisible()) {
            str_fecha = cal_fecha_emision.getFecha();
            str_observacion = txt_resolucion.getValue().toString();
            dia_dialogo.cerrar();
            sel_casa_obra.getTab_seleccion().setSql(ser_gerencia.getCasaObra("2", "", com_anio.getValue().toString()));
            sel_casa_obra.getTab_seleccion().ejecutarSql();
            sel_casa_obra.dibujar();
        } else if (sel_casa_obra.isVisible()) {
            str_seleccionados = sel_casa_obra.getSeleccionados();
            if (str_seleccionados == null) {
                utilitario.agregarMensajeError("Seleccione un valor", "Seleccione una casa u obra");
            } else {
                sel_casa_obra.cerrar();
                TablaGenerica tab_obra = new TablaGenerica();
                List list_obras = tab_obra.getConexion().consultar(ser_gerencia.getCasaObra("1", str_seleccionados,
                    ""));
                if (list_obras.size() > 0) {
                    for (Object li : list_obras) {
                        Object[] fila = (Object[]) li;
                        tab_tabla.insertar();
                        tab_tabla.setValor("ide_gerobr", String.valueOf(fila[0]));
                        tab_tabla.setValor("responsable_gerobc", utilitario.getVariable("NICK"));
                        tab_tabla.setValor("ide_gerani", com_anio.getValue() + "");
                        tab_tabla.setValor("ide_gerest", utilitario.getVariable("p_ger_estado_activo"));
                        tab_tabla.setValor("fecha_apert_gerobc", str_fecha);
                        tab_tabla.setValor("observacion_gerobc", str_observacion);
                    }
                }
            }
        }
    }
}

public void agregarCerrarObra() {
    if (cal_fecha_cierre.getValue() == null || cal_fecha_cierre.getValue().toString().isEmpty()) {
        utilitario.agregarMensajeInfo("ADVERTENCIA", "Ingrese la Fecha de Cierre");
        return;
    } else {
        if (dia_dialogo_cierre.isVisible()) {
            str_fecha_cierre = cal_fecha_cierre.getFecha();
            dia_dialogo_cierre.cerrar();
            sel_casa_obra_cierre.getTab_seleccion().setSql(ser_gerencia.getCasaObraPeriodoFiscal(
                utilitario.getVariable("p_ger_estado_activo"), com_anio.getValue().toString(), "1", "1"));
            sel_casa_obra_cierre.getTab_seleccion().ejecutarSql();
            sel_casa_obra_cierre.dibujar();
        } else if (sel_casa_obra_cierre.isVisible()) {
            sel_casa_obra_cierre.cerrar();
            str_seleccionados = sel_casa_obra_cierre.getSeleccionados();
            TablaGenerica tab_obra = new TablaGenerica();
            List list_obras = tab_obra.getConexion().consultar(ser_gerencia.getCasaObraPeriodoFiscal("-1", "-1", "2",
                str_seleccionados));
            if (list_obras.size() > 0) {
                for (Object li : list_obras) {
                    Object[] fila = (Object[]) li;
                    utilitario.getConexion().setUnidad_persistencia("rua_gerencia");
                    utilitario.getConexion().ejecutarSql("update ger_cont_balance_cabecera set fecha_cierre_gerobc="
                        + str_fecha_cierre + " where ide_gerobc=" + String.valueOf(fila[0]));
                }
            }
            tab_tabla.ejecutarSql();
            utilitario.addUpdate("tab_tabla");
        }
    }
}
```

**Figura 24:** Código fuente del método agregar y cerrar obra  
**Elaborado por:** Los autores

### 3.1.1.3 Método activar mes contable

Este método tiene la programación para que el sistema permita la opción de abrir períodos contables mensuales, sea de tipo inicial y mensual según corresponda para cada una de las casas y obras salesianas.

#### Método agregar y cerrar períodos mensuales de las casas y obras

```
public void agregarObra() {  
  
    if (cal_fecha_emision.getValue() == null || cal_fecha_emision.getValue().toString().isEmpty()) {  
        utilitario.agregarMensajeInfo("ADVERTENCIA", "Ingrese la Fecha de Apertura");  
        return;  
    } else if (txt_resolucion.getValue() == null || txt_resolucion.getValue().toString().isEmpty()) {  
        utilitario.agregarMensajeInfo("ADVERTENCIA", "Ingrese alguna Observación");  
        return;  
    } else if (dia_dialogo.isVisible()) {  
        str_fecha = cal_fecha_emision.getFecha();  
        str_observacion = txt_resolucion.getValue().toString();  
        dia_dialogo.cerrar();  
  
        sel_mes_apertura.dibujar();  
    } else if (sel_mes_apertura.isVisible()) {  
  
        str_mes_seleccionado = sel_mes_apertura.getValorSeleccionado();  
        if (str_mes_seleccionado == null) {  
            utilitario.agregarMensajeError("Seleccione un valor", "Seleccione un mes");  
        } else {  
            sel_mes_apertura.cerrar();  
            sel_tipo_balance.getTab_seleccion().setSql(ser_gerencia.getTipoBalance("1", str_mes_seleccionado));  
            sel_tipo_balance.getTab_seleccion().ejecutarSql();  
            sel_tipo_balance.dibujar();  
        }  
    } else if (sel_tipo_balance.isVisible()) {  
  
        str_tipo_balance_seleccionado = sel_tipo_balance.getValorSeleccionado();  
        if (str_tipo_balance_seleccionado == null) {  
            utilitario.agregarMensajeError("Seleccione un valor", "Seleccione el tipo de balance");  
        }  
    }  
  
    else {  
        sel_tipo_balance.cerrar();  
        sel_casa_obra.getTab_seleccion().setSql(ser_gerencia.getCasaObraMensual("1", com_anio.getValue().toString(), str_mes_seleccionado, str_tipo_balance_seleccionado, ""));  
        sel_casa_obra.getTab_seleccion().ejecutarSql();  
        sel_casa_obra.dibujar();  
    }  
} else if (sel_casa_obra.isVisible()) {  
  
    str_seleccionados = sel_casa_obra.getSeleccionados();  
    if (str_seleccionados == null) {  
        utilitario.agregarMensajeError("Seleccione un valor", "Seleccione una casa u obra");  
    } else {  
        sel_casa_obra.cerrar();  
        TablaGenerica tab_obras_mensual = utilitario.consultar(ser_gerencia.getCasaObraMensual("2", com_anio.getValue().toString(), str_mes_seleccionado, str_tipo_balance_seleccionado, str_seleccionados));  
        if (tab_obras_mensual.getTotalFilas() > 0) {  
            for (int i = 0; i < tab_obras_mensual.getTotalFilas(); i++) {  
                tab_tabla2.insertar();  
                tab_tabla2.setValor("ide_gecobc", tab_obras_mensual.getValor(i, "ide_gecobc"));  
                tab_tabla2.setValor("ide_gemes", str_mes_seleccionado);  
                tab_tabla2.setValor("ide_getiba", str_tipo_balance_seleccionado);  
                tab_tabla2.setValor("fecha_apert_gesame", str_fecha);  
                tab_tabla2.setValor("responsable_gesame", utilitario.getVariable("NICK"));  
                tab_tabla2.setValor("observacion_gesame", str_observacion);  
                tab_tabla2.setValor("ide_gerest", utilitario.getVariable("p_ger_estado_activo"));  
            }  
        }  
        tab_tabla2.guardar();  
        guardarPantalla();  
        tab_tabla2.ejecutarValorForanea(tab_tabla1.getValorSeleccionado());  
        utilitario.addUpdate("tab_tabla2");  
    }  
}
```

**Figura 25:** Código fuente del método agregar y cerrar períodos mensuales  
**Elaborado por:** Los autores

### 3.1.1.4 Método cerrar período contable

Este método tiene la programación para que el sistema permita la opción de cerrar los períodos contables mensuales y anuales, sea de tipo inicial o mensual según corresponda para cada una de las casas y obras salesianas con este proceso ya se puede generar los balances.

#### Método cerrar períodos mensuales de las casas y obras

```
public void agregarCerrarObra() {  
    if (cal_fecha_cierre.getValue() == null || cal_fecha_cierre.getValue().toString().isEmpty()) {  
        utilitario.agregarMensajeInfo("ADVERTENCIA", "Ingrese la Fecha de Cierre");  
        return;  
    } else if (dia_dialogo_cierre.isVisible()) {  
        str_fecha_cierre = cal_fecha_cierre.getFecha();  
        dia_dialogo_cierre.cerrar();  
        sel_casa_obra_cierre.getTab_seleccion().setSql(sel_gerencial.getCasaObraPeriodoFiscal(utilitario.getVariable  
        ("p_ger_estado_activo"), com_anio.getValue().toString(), "1", "1"));  
        sel_casa_obra_cierre.getTab_seleccion().ejecutarSql();  
        sel_casa_obra_cierre.dibujar();  
    } else if (sel_casa_obra_cierre.isVisible()) {  
        sel_casa_obra_cierre.cerrar();  
        str_seleccionados = sel_casa_obra_cierre.getSeleccionados();  
        TablaGenerica tab_obra = new TablaGenerica();  
        List list_obras = tab_obra.getConexion().consultar(sel_gerencial.getCasaObraPeriodoFiscal("-1", "-1", "2",  
        str_seleccionados));  
        if (list_obras.size() > 0) {  
            for (Object li : list_obras) {  
                Object[] fila = (Object[]) li;  
                utilitario.getConexion().setUnidad_persistencia("rua_gerencial");  
                utilitario.getConexion().ejecutarSql("update ger_cont_balance_cabecera set fecha_cierre_gecobc=" +  
                + str_fecha_cierre + "' where ide_gecobc=" + String.valueOf(fila[0]));  
            }  
            tab_tabla1.ejecutarSql();  
            utilitario.addUpdate("tab_tabla1");  
        }  
    }  
}
```

**Figura 26:** Código fuente método cierre de períodos mensuales  
**Elaborado por:** Los autores

### 3.1.1.5 Método transferir Balance

Este método tiene la función de transferir todos los balances contables de las casas y obras salesianas que tengan instalado y lleven su contabilidad en el sistema RUA, de esta manera se tiene toda la información para que el sistema pueda realizar la consolidación de información y posterior generación de reportes.

#### Método transferir Balances

```
public void transferirBalance() {
    if(cal_fecha_inicio.getFecha()==null||cal_fecha_fin.getFecha()==null){
        utilitario.agregarMensajeError("Fechas no validas", "Necesita seleccionar la fecha inicial o fecha final");
        return;
    }
    if(utilitario.isFechasValidas(cal_fecha_inicio.getFecha(), cal_fecha_fin.getFecha())){
        utilitario.agregarMensajeError("Fechas no validas", "Las fecha final del reporte es mayor a la fecha inicial");
        return;
    }
    if(com_estado_mes_fiscal.getValue().toString().equals(utilitario.getVariable("p_ger_estado_activo"))){
        String estado_normal = utilitario.getVariable("p_con_estado_comprobante_normal");
        String estado_inicial = utilitario.getVariable("p_con_estado_comp_inicial");
        String estado="-1";
        if(com_tipo_balance.getValue().toString().equals("1")){ //Inicial
            estado=estado_inicial;
        }
        else if(com_tipo_balance.getValue().toString().equals("2")){ //mensual
            estado=estado_normal;
        }
    }
    TablaGenerica tab_cuenta=utilitario.consultar(ser_gerencial.getTranferirAsientos(estado,
    utilitario.getVariable("ide_sucu"), cal_fecha_inicio.getFecha(), cal_fecha_fin.getFecha()));
    utilitario.getConexion().ejecutarSql(ser_gerencial.deleteTempBalance(utilitario.getVariable("ide_usua")));
    if(tab_cuenta.getTotalFilas()>0){
        for(int i=0;i<tab_cuenta.getTotalFilas();i++){
            //System.out.println("primer insert i "+i);
            utilitario.getConexion().ejecutarSql(ser_gerencial.insertTempBalance(tab_cuenta.getValor(i, "ide_cndpc"),
            tab_cuenta.getValor(i, "debe"),tab_cuenta.getValor(i, "haber"),utilitario.getVariable("ide_usua"),
            tab_cuenta.getValor(i, "ide_cnncu") ));
        }

        int nivel_nuevo=0;
        for(int j=7;j>2; j--){
            nivel_nuevo=j-1;
            System.out.println("j "+j);
            TablaGenerica tab_temporal=utilitario.consultar(ser_gerencial.getCalTemBalance(utilitario.getVariable
            ("ide_usua"), j+""));
            for(int k=0;k<tab_temporal.getTotalFilas();k++){
                // System.out.println("segundo insert K "+k);
                utilitario.getConexion().ejecutarSql(ser_gerencial.insertTempBalance(tab_temporal.getValor(k,
                "con_ide_cndpc"),tab_temporal.getValor(k, "debe"),tab_temporal.getValor(k, "haber"),
                utilitario.getVariable("ide_usua"),nivel_nuevo+" "));
            }
        }
        TablaGenerica tab_temporal_insert=utilitario.consultar(ser_gerencial.getCalTemBalance(utilitario.getVariable
        ("ide_usua"), "2,3,4,5,6"));
        for(int k=0;k<tab_temporal_insert.getTotalFilas();k++){
            tab_tabla3.insertar();
            tab_tabla3.setValor("ide_cndpc",tab_temporal_insert.getValor(k,"con_ide_cndpc"));
            tab_tabla3.setValor("ide_gebame",ide_gebame);
            tab_tabla3.setValor("valor_debe_gebade",tab_temporal_insert.getValor(k,"debe"));
            tab_tabla3.setValor("valor_haber_gebade",tab_temporal_insert.getValor(k,"haber"));
        }
    }
    tab_tabla3.guardar();
    guardarPantalla();
    tab_tabla3.setCondicion("ide_gebame="+ide_gebame);
    tab_tabla3.ejecutarSql();
} else{
    utilitario.agregarMensajeError("Período no habilitado", "NO se puede realizar la transferencia, "
    + "el periodo se encuentra inhabilitado");
}
}
```

**Figura 27:** Código fuente Método transferir balances  
**Elaborado por:** Los autores

### 3.1.1.6 Método validar archivos en Excel

Este método tiene la programación para que el sistema permita la opción la carga del archivo Excel al sistema con los balances de usuarios que no tienen instalado el sistema RUA, la opción más importante es la validación del formato a cargar en el sistema.

#### Método validar archivos

```
public void validarArchivo(FileUploadEvent evt) {
    //Leer el archivo
    String str_msg_info = "";
    String str_msg_adve = "";
    String str_msg_erro = "";
    String codigo_valdap = "";
    String codigo_discapacidad = "";
    String str_result2 = "";
    //System.out.println(" valor de resultado " + str_resultado);
    if (validarArchivoMensa(evt).equals("1")) {
        // System.out.println(" entre if_valida " + str_resultado);
        utilitario.getConnection().ejecutarSql(ser_gerencia.deleteTempBalance(utilitario.getVariable("ide_usua")));
        try {
            Workbook archivoExcel = Workbook.getWorkbook(evt.getFile().getInputStream());
            Sheet hoja = archivoExcel.getSheet(0); //LEE LA PRIMERA HOJA
            if (hoja == null) {
                utilitario.agregarMensajeError("No existe ninguna hoja en el archivo seleccionado", "");
                return;
            }
            int int_fin = hoja.getRows();
            upl_archivo.setNombreReal(evt.getFile().getFileName());
            str_msg_info += getFormatoInformacion("El archivo " + upl_archivo.getNombreReal() + " contiene " +
                int_fin + " filas que se subieron con éxito, dar clic en aceptar y actualizar la pantalla de "
                + "inscripciones");
            System.out.println(" valor int_fin " + int_fin);
            for (int i = 0; i < int_fin; i++) {
                String casa_obra = hoja.getCell(0, i).getContents();
                casa_obra = casa_obra.trim();
                String tipo_balance = hoja.getCell(1, i).getContents();
                tipo_balance = tipo_balance.trim();
                String anio = hoja.getCell(2, i).getContents();
                anio = anio.trim();
                String mes = hoja.getCell(3, i).getContents();
                mes = mes.trim();

                int nivel_nuevo=0;
                for(int j=5;j>2; j--){
                    nivel_nuevo=j-1;
                    // System.out.println("j "+j);
                    TablaGenerica tab_temporal=utilitario.consultar(ser_gerencia.getCalTemBalance(utilitario.getVariable("ide_usua"), j+""));
                    for(int k=0;k<tab_temporal.getTotalFilas();k++){
                        // System.out.println("segundo insert k "+k);
                        utilitario.getConnection().ejecutarSql(ser_gerencia.insertTempBalance(tab_temporal.getValor(k, "con_ide_cndpc"), tab_temporal.getValor(k, "debe"), tab_temporal.getValor(k, "haber"), utilitario.getVariable("ide_usua"), nivel_nuevo+" "));
                    }
                }
                TablaGenerica tab_temporal_insert=utilitario.consultar(ser_gerencia.getCalTemBalance(utilitario.getVariable("ide_usua"), "2,3,4,5,6"));
                for(int k=0;k<tab_temporal_insert.getTotalFilas();k++){
                    tab_tabla3.insertar();
                    tab_tabla3.setValor("ide_cndpc", tab_temporal_insert.getValor(k, "con_ide_cndpc"));
                    tab_tabla3.setValor("ide_gebame", tab_tabla2.getValor(tab_tabla2.getFilaActual(), "ide_gebame"));
                    tab_tabla3.setValor("valor_debe_gebade", tab_temporal_insert.getValor(k, "debe"));
                    tab_tabla3.setValor("valor_haber_gebade", tab_temporal_insert.getValor(k, "haber"));
                }
                // hasta aqui termino la insercion
                edi_mensajes.setValue(str_resultado);
                utilitario.addUpdate("edi_mensajes");
                tab_tabla3.guardar();
                guardarPantalla();
                tab_tabla3.ejecutarValorForanea(tab_tabla2.getValorSeleccionado());
            } else {
                edi_mensajes.setValue(str_resultado);
                utilitario.addUpdate("edi_mensajes");
            }
        } catch (Exception e) {
            // Manejo de excepciones
        }
    }
}
```

Figura 28: Código fuente método validar archivos de Excel.

Elaborado por: Los autores

### 3.1.1.7 Método generación de reportes gerenciales.

Este método tiene la programación para que el sistema permita la opción de la generación de los reportes gerenciales detallados a nivel 5 de cuentas contables de casa y obras salesianas.

#### Generación balance general consolidado y de comprobación

```
parametro.put("titulo", "BALANCE GENERAL CONSOLIDADO");
parametro.put("pmeses", lis_meses.getSeleccionados());
parametro.put("pmes_inicial", tab_mesi_desc.getValor("nombre_gemes"));
parametro.put("pmes_final", tab_mesf_desc.getValor("nombre_gemes"));
parametro.put("pnivel_inicial", Integer.parseInt(rad_nivel_inicial.getValue().toString()));
parametro.put("pnivel_final", Integer.parseInt(rad_nivel_final.getValue().toString()));
parametro.put("pperiodo", tab_periodo.getValor("nom_geani"));
parametro.put("ptipo_balance", lis_tipo_balance.getSeleccionados());
parametro.put("pusuario", utilitario.getVariable("nick"));
parametro.put("pobra", tab_obras.getValor("obra"));
parametro.put("pide_obra", seleccionado);
parametro.put("pide_geani", Integer.parseInt(com_periodo_financiero.getValue().toString()));
parametro.put("p_tot_activo", tot_activo);
parametro.put("p_total", total);
parametro.put("p_utilidad_perdida", utilidad_perdida);
parametro.put("p_tot_pasivo", tot_pasivo);
parametro.put("p_tot_patrimonio", (tot_patrimonio));
parametro.put("pfirma1", utilitario.getVariable("p_ger_nom_firma1"));
parametro.put("pcargol", utilitario.getVariable("p_ger_cargo_firma1"));
parametro.put("pfirma2", utilitario.getVariable("p_ger_nom_firma2"));
parametro.put("pcargo2", utilitario.getVariable("p_ger_cargo_firma2"));
vipdf_mayor.setVisualizarPDF("rep_gerencial/rep_balance_general_ger.jasper", parametro);
vipdf_mayor.dibujar();

parametro.put("titulo", "BALANCE DE COMPROBACIÓN ");
parametro.put("pmeses", lis_meses.getSeleccionados());
parametro.put("pmes_inicial", tab_mesi_desc.getValor("nombre_gemes"));
parametro.put("pmes_final", tab_mesf_desc.getValor("nombre_gemes"));
parametro.put("pnivel_inicial", Integer.parseInt(rad_nivel_inicial.getValue().toString()));
parametro.put("pnivel_final", Integer.parseInt(rad_nivel_final.getValue().toString()));
parametro.put("pperiodo", tab_periodo.getValor("nom_geani"));
parametro.put("ptipo_balance", lis_tipo_balance.getSeleccionados());
parametro.put("pbal_inicial", bal_inicial);
parametro.put("pbal_mes", bal_mensual);
parametro.put("pmes_periodo", tab_mes_max.getValor("valor"));
parametro.put("pusuario", utilitario.getVariable("nick"));
parametro.put("pobra", tab_obras.getValor("obra"));
parametro.put("pide_obra", seleccionado);
parametro.put("pide_geani", Integer.parseInt(com_periodo_financiero.getValue().toString()));
parametro.put("pfirma1", utilitario.getVariable("p_ger_nom_firma1"));
parametro.put("pcargol", utilitario.getVariable("p_ger_cargo_firma1"));
parametro.put("pfirma2", utilitario.getVariable("p_ger_nom_firma2"));
parametro.put("pcargo2", utilitario.getVariable("p_ger_cargo_firma2"));
parametro.put("pdebe_ini", Double.parseDouble(tab_totales.getValor("tot_debe_inicial")));
parametro.put("phaber_ini", Double.parseDouble(tab_totales.getValor("tot_haber_inicial")));
parametro.put("pdebe_per", Double.parseDouble(tab_totales.getValor("tot_debe_per")));
parametro.put("phaber_per", Double.parseDouble(tab_totales.getValor("tot_haber_per")));
parametro.put("pdebe_acum", Double.parseDouble(tab_totales.getValor("tot_debe_acum")));
parametro.put("phaber_acum", Double.parseDouble(tab_totales.getValor("tot_haber_acum")));
parametro.put("pdebe_sal", Double.parseDouble(tab_totales.getValor("tot_debe_sal")));
parametro.put("phaber_sal", Double.parseDouble(tab_totales.getValor("tot_haber_sal")));
vipdf_mayor.setVisualizarPDF("rep_gerencial/rep_balance_compro_ger.jasper", parametro);
vipdf_mayor.dibujar();
```

**Figura 29:** Código fuente método generación de reportes

**Elaborado por:** Los autores

### 3.2 Diagrama físico de base de datos

Diagrama físico de base de datos

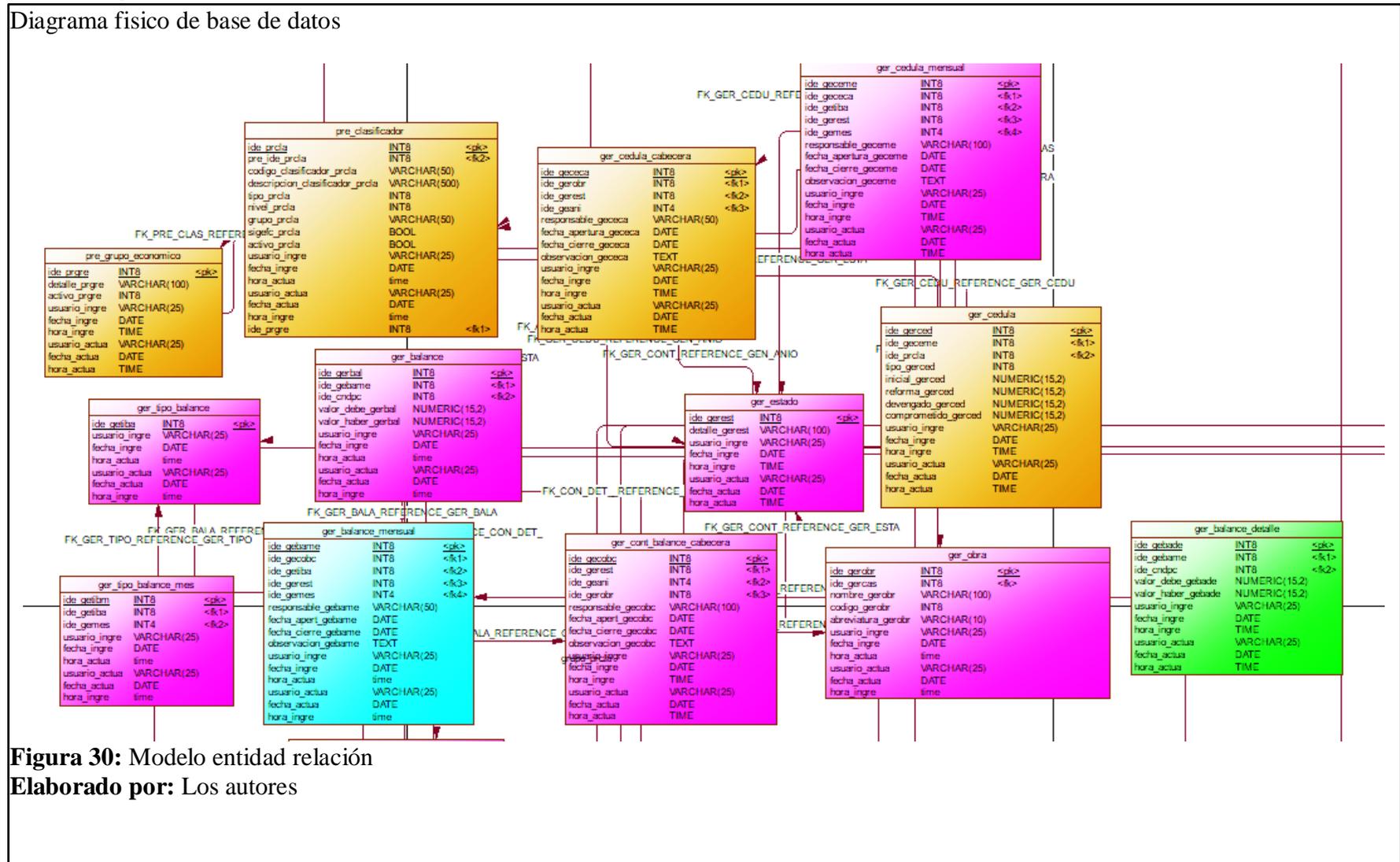


Figura 30: Modelo entidad relación

Elaborado por: Los autores

Diagrama fisicon de base de datos parte 2

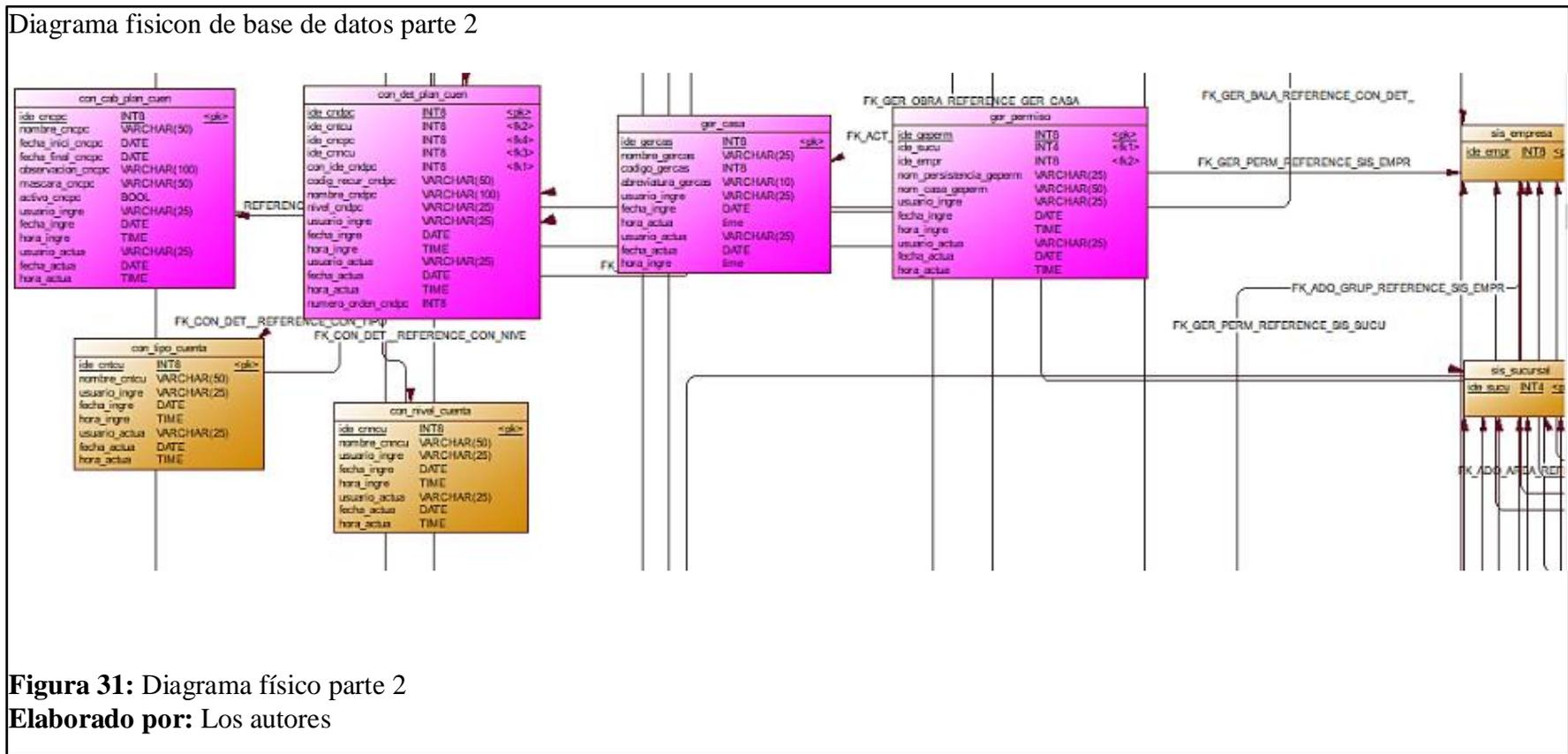


Figura 31: Diagrama físico parte 2

Elaborado por: Los autores

**Tabla 9:**

Glosario de términos y abreviaturas

<b>Término/Abreviatura</b>	<b>Significado</b>
<b>PK</b>	Primary Key
<b>FK</b>	Foreing Key

**Nota:** Tabla de identificativo de términos para las bases de datos

**Elaborado por:** Los autores

**Tabla 10:**

Formato de los datos

<b>Abreviatura</b>	<b>Significado</b>
<b>Integer</b>	Entero con signo, 4 bytes
<b>Int 8</b>	Entero con signo de 8 bytes
<b>Numeric</b>	Numérico exacto con precisión modificable
<b>Date</b>	Fecha (año, mes, día)
<b>Bool</b>	Lógico (true/false)
<b>Text</b>	Campo de caracteres de longitud variable
<b>Varchar</b>	Campo de caracteres de longitud variable
<b>Timestamp</b>	Fecha y hora

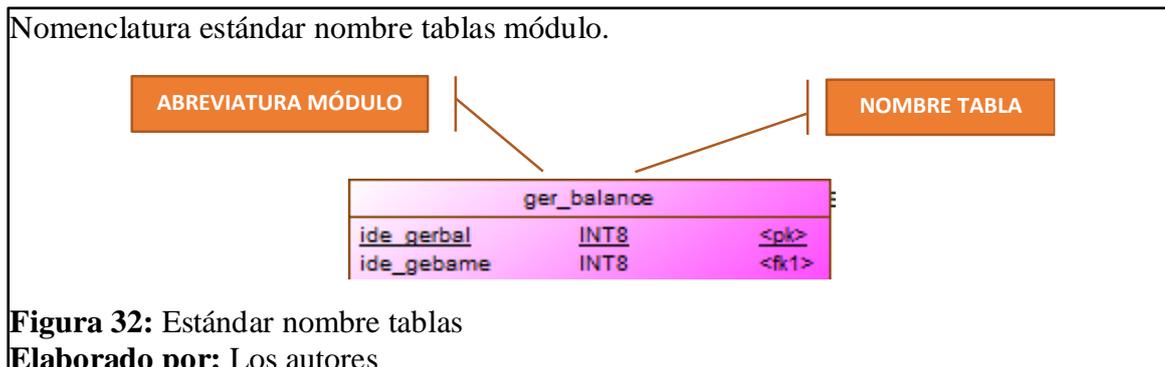
**Nota:** Listado de formato de datos a utilizar en la base de datos

**Elaborado por:** Los autores

**Estándar para Nombre de las tablas**

3 primeras letras del nombre del módulo + guion bajo + nombre completo de la tabla (la sumatoria debe de ser hasta 30 caracteres no puede exceder).

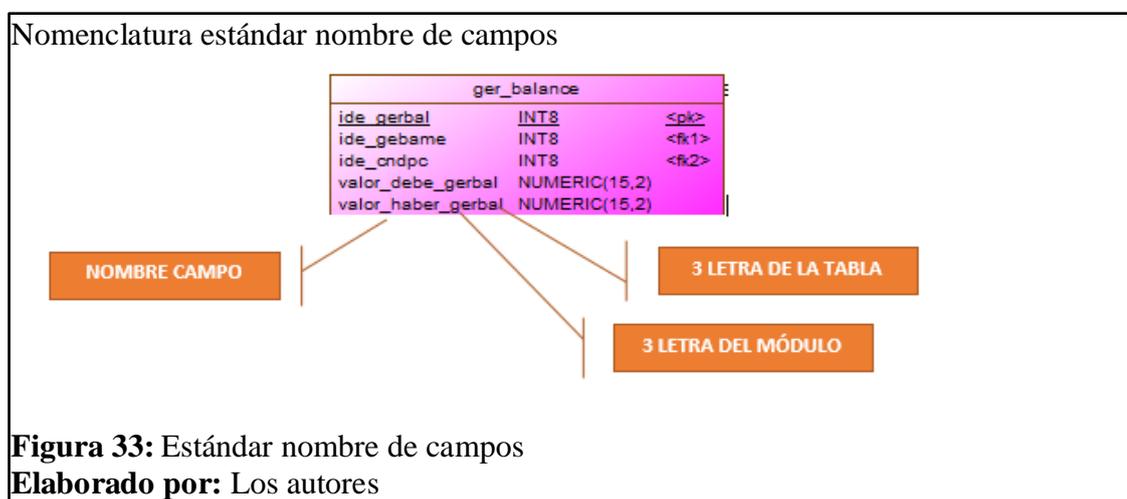
Ejemplo: En este caso nuestro módulo es gerencial por lo cual las 3 primeras letras serian **ger** y el nombre de la tabla que en este caso es **balance**.



### Estándar Nombre del campo

Nombre del campo + guion bajo + 3 letras del módulo seguido de las 3 primeras letras de la tabla (la sumatoria debe de ser hasta 30 caracteres no puede exceder).

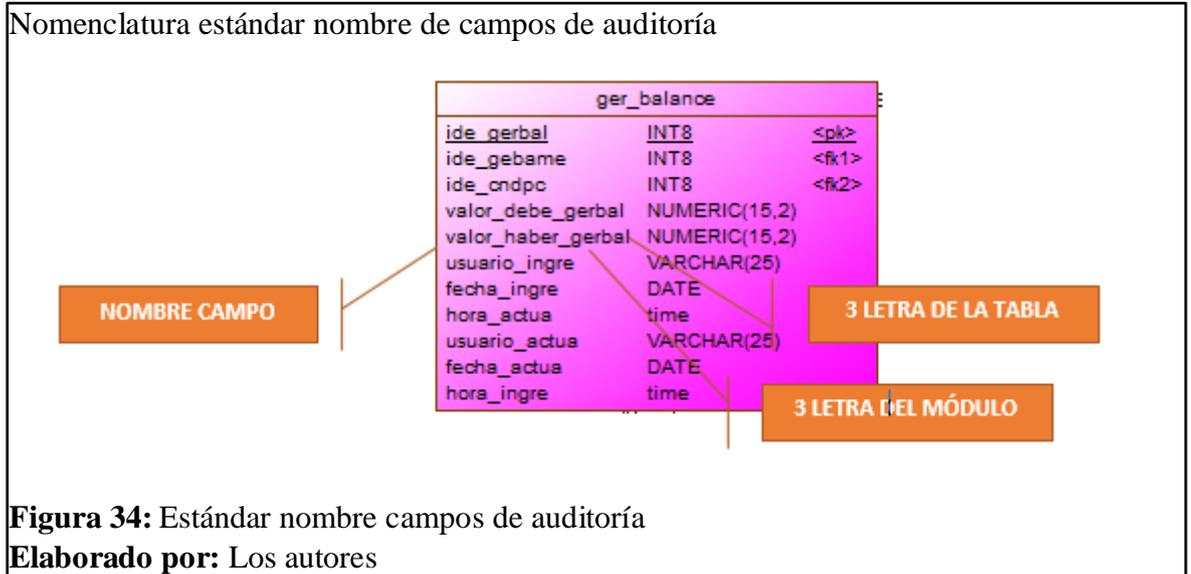
Ejemplo: En este caso nuestro campo es **valor haber**, luego nuestro módulo es gerencial por lo cual las 3 primeras letras serian **ger** y el nombre de la tabla que en este caso es **balance**.



### Estándar Nombre de campos de auditoria

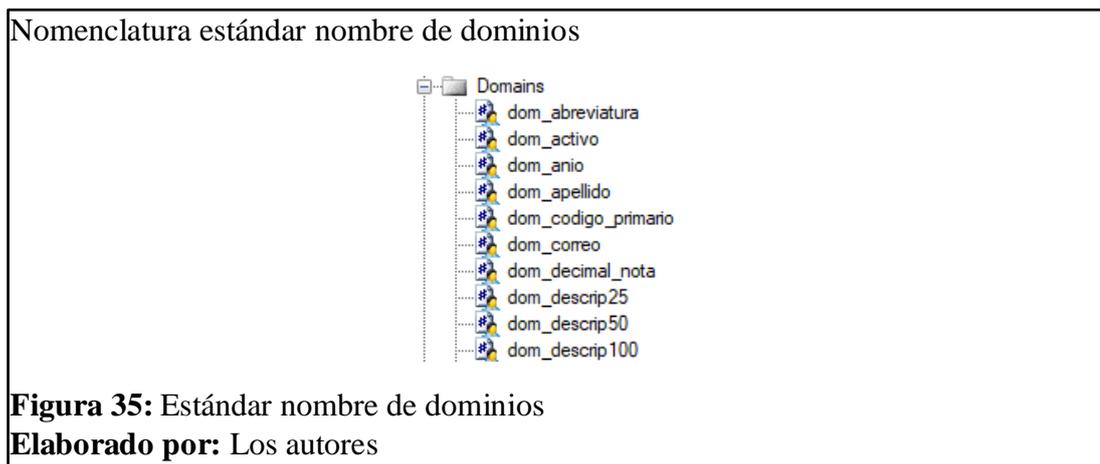
Todas las tablas tendrán 6 campos de auditoría contendrá el nombre del campo + guion bajo + el registro de ingreso **ingre** o actualización **actua**.

Ejemplo: En este caso nuestro campo es **valor haber**, luego nuestro módulo es gerencial por lo cual las 3 primeras letras serian **ger** y el nombre de la tabla que en este caso es **balance**.



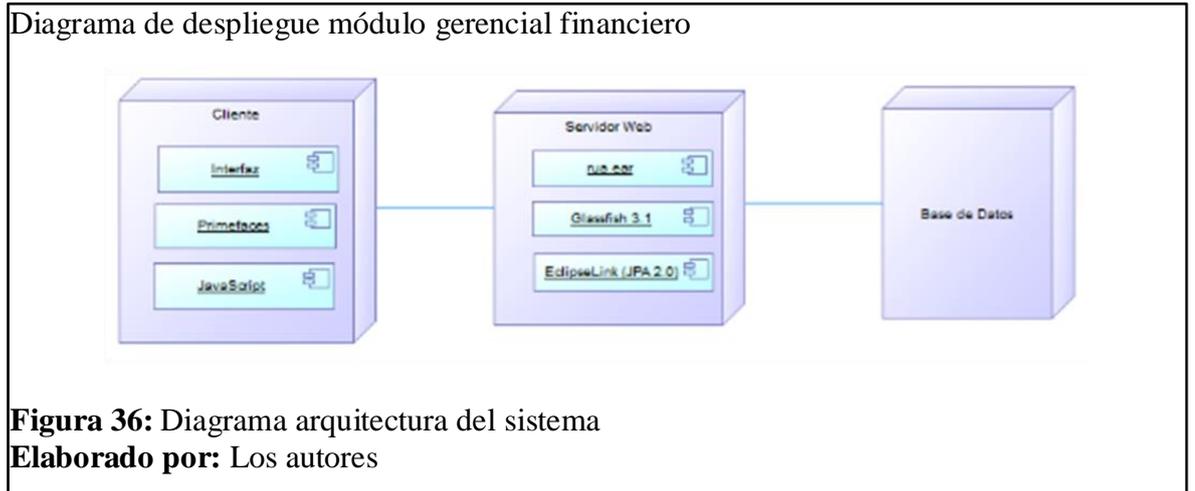
### Estándar de dominios

3 primeras letras de la palabra dominio + guion bajo + nombre de la respectiva base de datos que lleve el dominio + su número o tipo de carácter, (la sumatoria debe de ser hasta 30 caracteres no puede exceder).



### 3.3 Diagrama de despliegue

El diagrama permite exponer la arquitectura para el óptimo funcionamiento así como todos los componentes se ejecutan en el sistema y como intervienen cada uno de ellos, además se puede evidenciar como interactúa el usuario con el sistema.



#### 3.3.1 Arquitectura

El sistema que se desarrolla mantiene un modelo vista controlador

**Tabla 11:**

Características módulo gerencial financiero

NODO	CONEXIÓN	CARACTERÍSTICAS
CLIENTE	Interfaz	Es donde interactúa el cliente con el servidor web y automáticamente sincroniza sus estados, en el servidor se ejecuta las aplicaciones y también accede a sus servicios.
	Primefaces	Esta optimiza el uso del framework y además da una serie de componentes que enriquecen a la interfaz.
	JavaScript	El motor de Primefaces genera automáticamente el código javascript.
SERVIDOR WEB	rua.ear	La lógica del negocio es manejada por ejb esta entidad es la que automáticamente sincroniza la base de datos con el servidor web.
	Glassfish 3.1	Permite el manejo de persistencia y concurrencia para utilizar la base de datos que conecten con rua.ear y se logre tener escalabilidad y velocidad. Además se maneja un pool de conexión para lograr concurrencia con los usuarios.
	EclipseLink (JPA 2.0)	El módulo es desarrollado en lenguaje java permitiendo tener control sobre la concurrencia de múltiples usuarios y conexión de todas sus bases de datos.
BASE DE DATOS	PostgresSQL	Este es un gestor que tiene un modelo de sistema que permite mantener la integridad de las bases datos y accesibilidad en todo momento.

**Nota:** Listado a detalle de todos los componentes de la arquitectura**Elaborado por:** Los autores

### 3.3.2 Requerimientos

Al finalizar la construcción del sistema se procede con la implementación del mismo en la Casa Inspectorial Salesiana creando un nuevo menú con todas sus opciones, a continuación todos los detalles referentes a esta tarea.

#### 3.3.2.1 Servidor físico

A continuación se detalla el servidor donde se aloja el módulo gerencial financiero con cada una de sus características.

**Tabla 12:**

Servidor físico

Hardware	Detalle
<b>Sistema Operativo</b>	CentOS 7 servidor web
<b>Memoria</b>	8 GB de RAM
<b>Procesador</b>	8.024e-6 GB
<b>Arquitectura</b>	Trabaja a nivel de cliente servidor
<b>Dirección IP servidor</b>	192.168.1.121

**Nota:** Lista de componentes necesarios para el servidor web

**Elaborado por:** Los autores

#### 3.3.2.2 Software necesario

A continuación se detalla toda la lista de software es necesarios para que el sistema funcione correctamente.

**Tabla 13:**

Software necesario para que el sistema funcione.

Software	Detalle
<b>Base de datos</b>	PostgreSQL 9.4
<b>Complemento java</b>	JDK 1.8
<b>Servidor de aplicaciones</b>	GlassFish 4.1

**Nota:** Lista de componentes para implementar el sistema

**Elaborado por:** Los autores

### **3.4 Plan de pruebas**

Con el sistema ya desarrollado es necesario realizar un plan de pruebas para poder descubrir todos los posibles errores que se puedan presentar antes de entregar al cliente, en esta sección se detalla los tipos de pruebas que se van a ejecutar en el sistema desarrollado.

#### **3.4.1 Pruebas funcionales**

Estas pruebas nos ayudan a verificar la calidad del software para así evitar realizar menos reprocesos, aumentar la velocidad y calidad en el desarrollo.

Además nos sirve para verificar el correcto funcionamiento del software, así como para realizar la revisión, retroalimentación y medir el cumplimiento de todos los requerimientos solicitados para este proyecto.

##### **3.4.1.1 Pruebas caja negra**

Las pruebas de tipo caja negra nos ayuda a verificar la correcta ejecución del sistema sin tomar en cuenta ninguna estructura interna, ni el código de desarrollo.

A continuación se muestra varias pruebas realizadas las mismas que nos indica cómo se relaciona el sistema al tomar la entrada y mostrar la salida correcta en base al cumplimiento de los requisitos.

**Tabla 14:**

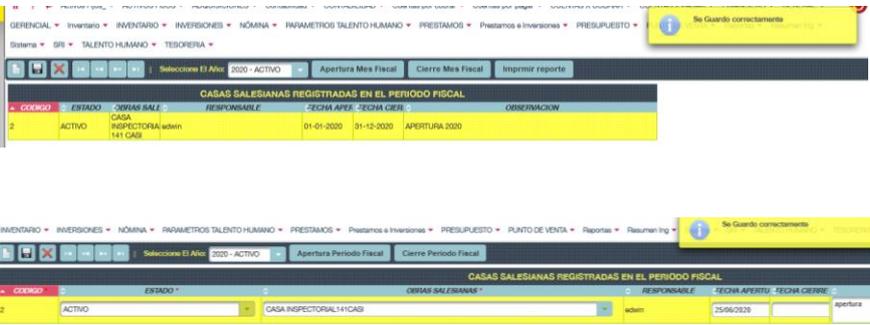
Prueba Caja Negra 1

<b>NOMBRE CASO PRUEBA:</b>	Generar formularios de parametrización de catálogos contables y presupuestarios.
<b>DATOS ENTRADA:</b>	Usuario: Administrador
<b>EJECUCIÓN:</b>	El formulario de parametrización debe conectar con la base de datos.
<b>SALIDA:</b>	Formularios guarda exitosamente todos los cambios. 

**Nota:** Registro de las pruebas de generación de formularios de parametrización  
**Elaborado por:** Los autores

**Tabla 15:**

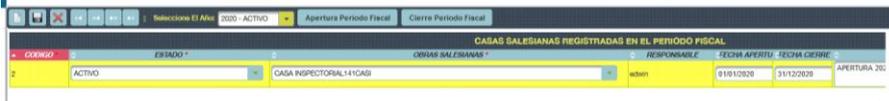
Prueba Caja Negra 2

<b>NOMBRE CASO PRUEBA:</b>	Generar formularios de parametrización de períodos fiscales mensuales y anuales
<b>DATOS ENTRADA:</b>	Usuario: Administrador
<b>EJECUCIÓN:</b>	El formulario de parametrización debe conectar con la base de datos.
<b>SALIDA:</b>	Formularios guarda exitosamente todos los cambios. 

**Nota:** Registro de las pruebas de parametrización periodos fiscales  
**Elaborado por:** Los autores

**Tabla 16:**

Prueba Caja Negra 3

<b>NOMBRE CASO PRUEBA:</b>	Cierre contable del sistema RUA
<b>DATOS ENTRADA:</b>	Usuario: Usuario sistema RUA
<b>EJECUCIÓN:</b>	El formulario de cierre debe conectar con la base de datos.
<b>SALIDA:</b>	Formulario guarda exitosamente todos los cambios para el cierre contable. 

**Nota:** Registro de las pruebas cierre contable

**Elaborado por:** Los autores

**Tabla 17:**

Prueba Caja Negra 4

<b>NOMBRE CASO PRUEBA:</b>	Generar balance general y cédulas presupuestarias.
<b>DATOS ENTRADA:</b>	Usuario: Administrador
<b>EJECUCIÓN:</b>	El formulario de reportes debe conectar con la base de datos.
<b>SALIDA:</b>	Formulario genera los reportes de balance general y cédulas presupuestarias. 

**Nota:** Registro de las pruebas generación de balances

**Elaborado por:** Los autores

**Tabla 18:**

Prueba Caja Negra 5

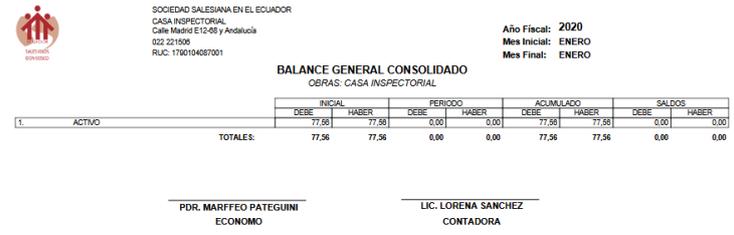
<b>NOMBRE CASO PRUEBA:</b>	Carga de archivos Excel, tanto balance general como cédulas presupuestarias de ingresos y gastos
<b>DATOS ENTRADA:</b>	Usuario: Usuarios Externos
<b>EJECUCIÓN:</b>	El formulario permite cargar el balance general y cédula presupuestaria de ingresos y gastos.
<b>SALIDA:</b>	Formulario carga exitosamente los datos de Excel del balance general y cédulas presupuestarias. 

**Nota:** Registro de las pruebas de carga de archivos de excel

**Elaborado por:** Los autores

**Tabla 19:**

Prueba Caja Negra 6

<b>NOMBRE CASO PRUEBA:</b>	Generar estados financieros
<b>DATOS ENTRADA:</b>	Usuario: Contador general Casa Inspectorial
<b>EJECUCIÓN:</b>	El formulario de estados financieros debe conectar con la base de datos.
<b>SALIDA:</b>	Formulario genera exitosamente todos los estados financieros. 

**Nota:** Registro de las pruebas de generación estados financieros

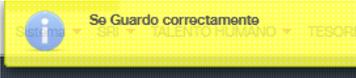
**Elaborado por:** Los autores

### 3.4.1.2 Pruebas de validación

Estas pruebas sirven para validar de que el sistema desarrollado cumpla con todas las especificaciones y que termine las tareas completamente, también valida los tipos de datos en los campos de ingreso al sistema por lo que debe mostrar mensajes claros y fáciles de entender para el usuario cuando un error se presente.

**Tabla 20:**

Formulario parametrización

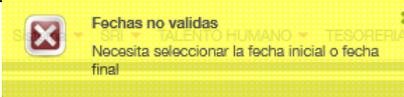
Formulario parametrización				
Tipo dato	Campo	Regla	Ejemplo	Mensaje del Sistema
Texto	Nombre Casa Salesiana	Opcional	Campo vacío	
Texto	Código Casa Salesiana	Opcional	Campo vacío	

**Nota:** Registro de las pruebas parametrización

**Elaborado por:** Los autores

**Tabla 21:**

Generar balance Sistema RUA

Generar balance Sistema RUA				
Tipo dato	Campo	Regla	Ejemplo	Mensaje del Sistema
texto	Nombre casa salesiana	Requerido	Lista casas y obras salesianas	Casa-Obra: QUITO - CASA INSPECTORIAL CASA INSPECTORIAL
date	Fecha generación	Requerido	Selección de fechas	Año Fiscal: 2020 - ACTIVO
entero	Período inicial	Requerido	Selección de fecha	
entero	Período final	Requerido	Selección de fecha	

**Nota:** Registro de las pruebas de generación de balances

**Elaborado por:** Los autores

**Tabla 22:**

Carga de archivos Excel usuarios externos

Carga de archivos Excel usuarios externos				
Tipo dato	Campo	Regla	Ejemplo	Mensaje del Sistema
Texto	Nombre casa salesiana	Requerido	Lista casas y obras salesianas	<b>OBRAS SALESIANAS:</b> CASA INSPECTORIAL 141 CASI
Date	Fecha carga archivo	Requerido	Selección de fechas	<b>Seleccione el Año:</b> 2020 - ACTIVO
Date	Período inicial	Requerido	Selección de fecha	<b>FECHA APERTURA:</b> 25/06/2020
Date	Período final	Requerido	Selección de fecha	<b>FECHA CIERRE:</b> <input type="text"/>
Texto	Selección formato excel	Requerido	Carga formato excel	<b>ERRORES</b> * El código de obra << 01 >> no se encuentra registrado en la base de datos revisar en la Fila: 1 no es válido * El código tipo de balance << 0 >> no se encuentra registrado en la base de datos revisar en la Fila: 1 no es válido * El código de obra << 01 >> no se encuentra registrado en la base de datos revisar en la Fila: 2 no es válido * El código tipo de balance << 0 >> no se encuentra registrado en la base de datos revisar en la Fila: 2 no es válido * El código de obra << 01 >> no se encuentra registrado en la base de datos revisar en la Fila: 3 no es válido * El código tipo de balance << 0 >> no se encuentra registrado en la base de datos revisar en la Fila: 3 no es válido * El código de obra << 01 >> no se encuentra registrado en la base de datos revisar en la Fila: 4

**Nota:** Registro de las pruebas de carga de archivos de excel

**Elaborado por:** Los autores

**Tabla 23:**

Consolidación de los balances generales y cédulas presupuestarias.

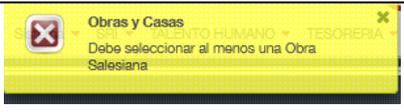
Consolidación de los balances generales y cédulas presupuestarias.				
Tipo dato	Campo	Regla	Ejemplo	Mensaje del Sistema
Texto	Nombre casa salesiana	Requerido	Lista casas y obras salesianas	<b>OBRAS SALESIANAS:</b> CASA INSPECTORIAL 141 CASI
Date	Período inicial	Requerido	Selección de fechas	<b>FECHA APERTURA:</b> 25/06/2020
Date	Período final	Requerido	Selección de fecha	<b>FECHA CIERRE:</b> <input type="text"/>

**Nota:** Registro de las pruebas de consolidación de balances

**Elaborado por:** Los autores

**Tabla 24:**

Generación de estados financieros.

<b>Generación de estados financieros.</b>				
<b>Tipo dato</b>	<b>Campo</b>	<b>Regla</b>	<b>Ejemplo</b>	<b>Mensaje del Sistema</b>
Entero	Tipo balance	Requerido	Inicial o Mensual	
Entero	Casas salesianas	Requerido	Selección de obra o casa	
Entero	Mes	Requerido	Selección del mes	

**Nota:** Registro de las pruebas de generación de estados financieros

**Elaborado por:** Los autores

### 3.4.1.3 Pruebas de stress

Estas pruebas son centradas en validar las características de rendimiento del sistema, al ser sometido a condiciones de uso extremo bajo condiciones muy superiores a lo que normalmente debería trabajar con esto se puede verificar la robustez y confiabilidad del mismo.

Para realizar estas pruebas dentro del sistema se utilizó una herramienta de tipo open source denominada “JMeter” y como resultado nos entregó resultados de tipo gráfico donde se puede visualizar el comportamiento, muestra los errores y cantidades de paquetes enviados y recibidos.

Para realizar las pruebas de stress del sistema se realizó 3 sesiones para así poder evidenciar como está funcionando.

**Tabla 25:**

Parámetros para realizar las pruebas de stress.

Prueba	Usuarios	Peticiones	Transacciones
1	40	80	3200
2	80	10	800
3	100	30	3000

**Nota:** Registro de las pruebas que se realizan

**Elaborado por:** Los autores

### 3.4.1.3.1 Prueba de stress 1

Se realizó las pruebas de stress utilizando los siguientes parámetros 40 usuarios y 80 peticiones al sistema.

Grupo de Hilos 1

Grupo de Hilos

Nombre: Thread Group

Comentarios

Acción a tomar después de un error de Muestreador

Continuar  Comenzar siguiente iteración  Parar Hilo  Parar Test  Parar test ahora

Propiedades de Hilo

Número de Hilos: 40

Periodo de Subida (en segundos): 1

Contador del bucle:  Sin fin 80

**Figura 37:** Configuración de hilos e intentos prueba stress 1

**Elaborado por:** Los autores

### Muestra de resultados prueba 1

```
Resultado del Muestreador  Petición  Datos de Respuesta
Nombre del hilo:Grupo de Hilos 1-40
Comienzo de muestra:2020-06-25 18:18:29 COT
Tiempo de carga:35
Connect Time:0
Latencia:35
Tamaño en bytes:4859
Sent bytes:0
Headers size in bytes:354
Body size in bytes:4505
Conteo de muestra:1
Conteo de error:0
Data type ("text"|"bin"|"):text
Código de respuesta:200
Mensaje de respuesta:OK
```

**Figura 39:** Pantalla de datos de la prueba de stress 1

**Elaborado por:** Los autores

### 3.4.1.3.2 Prueba de stress 2

Se realizó las pruebas de stress utilizando los siguientes parámetros 80 usuarios y 10 peticiones al sistema.

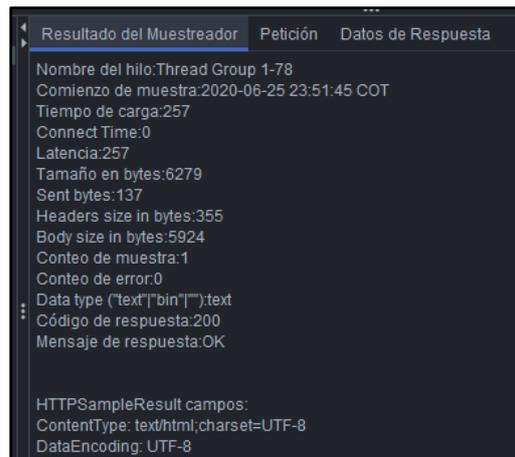
### Grupo de Hilos 2

<b>Grupo de Hilos</b>	
Nombre:	Grupo de Hilos
Comentarios	prueba stress sistema gerencial
Acción a tomar después de un error de Muestreador	
<input checked="" type="radio"/> Continuar <input type="radio"/> Comenzar siguiente iteración <input type="radio"/> Parar Hilo <input type="radio"/> Parar Test <input type="radio"/> Parar test ahora	
Propiedades de Hilo	
Número de Hilos	80
Periodo de Subida (en segundos):	1
Contador del bucle: <input type="checkbox"/> Sin fin	10

**Figura 41:** Configuración de hilos e intentos prueba stress 2

**Elaborado por:** Los autores

## Muestra de resultados prueba 2



**Figura 43:** Pantalla de datos de la prueba de stress 2

**Elaborado por:** Los autores

### 3.4.1.3.3 Prueba de stress 3

Se realizó las pruebas de stress utilizando los siguientes parámetros 100 usuarios y 30 peticiones al sistema.

## Grupo de Hilos 3

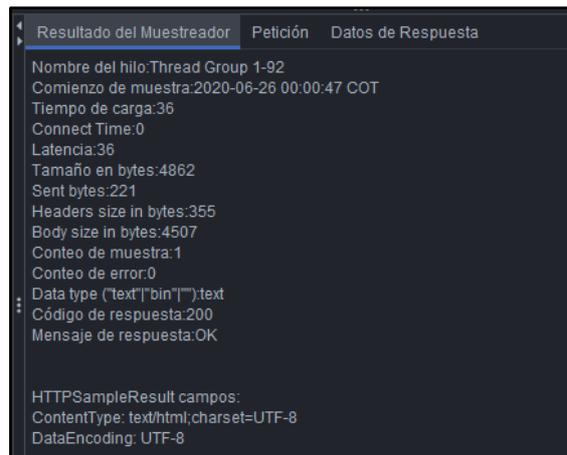
The screenshot shows the 'Grupo de Hilos' configuration window with the following settings:

- Nombre: Grupo de Hilos
- Comentarios: prueba stress sistema gerencial
- Acción a tomar después de un error de Muestreador:  Continuar,  Comenzar siguiente iteración,  Parar Hilo,  Parar Test,  Parar test ahora
- Propiedades de Hilo:
  - Número de Hilos: 100
  - Periodo de Subida (en segundos): 1
  - Contador del bucle:  Sin fin, 30

**Figura 45:** Configuración de hilos e intentos prueba stress 3

**Elaborado por:** Los autores

## Muestra de resultados prueba 3



```
Resultado del Muestreador  Petición  Datos de Respuesta
Nombre del hilo:Thread Group 1-92
Comienzo de muestra:2020-06-26 00:00:47 COT
Tiempo de carga:36
Connect Time:0
Latencia:36
Tamaño en bytes:4862
Sent bytes:221
Headers size in bytes:355
Body size in bytes:4507
Conteo de muestra:1
Conteo de error:0
Data type ("text"|"bin"|"):text
Código de respuesta:200
Mensaje de respuesta:OK

HTTPSampleResult campos:
ContentType: text/html;charset=UTF-8
DataEncoding: UTF-8
```

**Figura 47:** Pantalla de datos de prueba de stress 3

**Elaborado por:** Los autores

## 3.5 Análisis de resultados de pruebas

### 3.5.1 Resultado pruebas caja negra, validación y conexión

Se realizaron las debidas pruebas de caja negra las cuales están verificados y ajustadas de acuerdo al flujo alternativo de las historias de usuarios, se puede verificar que se cumpla con todos los campos validados para que el usuario lo pueda utilizar sin problema, también las conexiones para que el sistema pueda ser portable y multiplataforma.

Se evidencia de la misma manera que el sistema cumple con todas las especificaciones y logra terminar todos los procesos cumpliendo con lo que el usuario requiere, se logra verificar que las bases de datos realizan todas las conexiones entre sí, su comunicación sea correcta y todas las opciones del menú de usuario se habiliten.

### 3.5.2 Resultado pruebas de stress

#### 3.5.2.1 Resultado pruebas stress 1

Para esta prueba se utilizó los parámetros:

40 usuarios.

80 peticiones.

3200 solicitudes

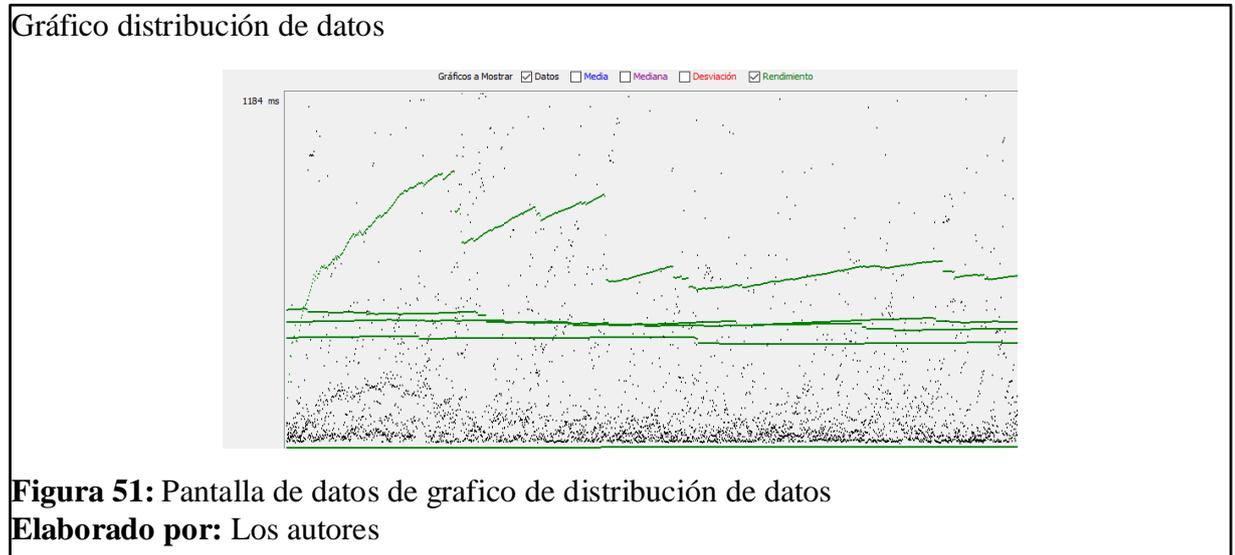
El tiempo de procesamiento de la solicitud es de 20 minutos y 5 segundos, como resultado se dio la caída del servidor evidenciando así la capacidad de procesar las peticiones y su rendimiento en horas pico.

Transacción exitosa prueba 1

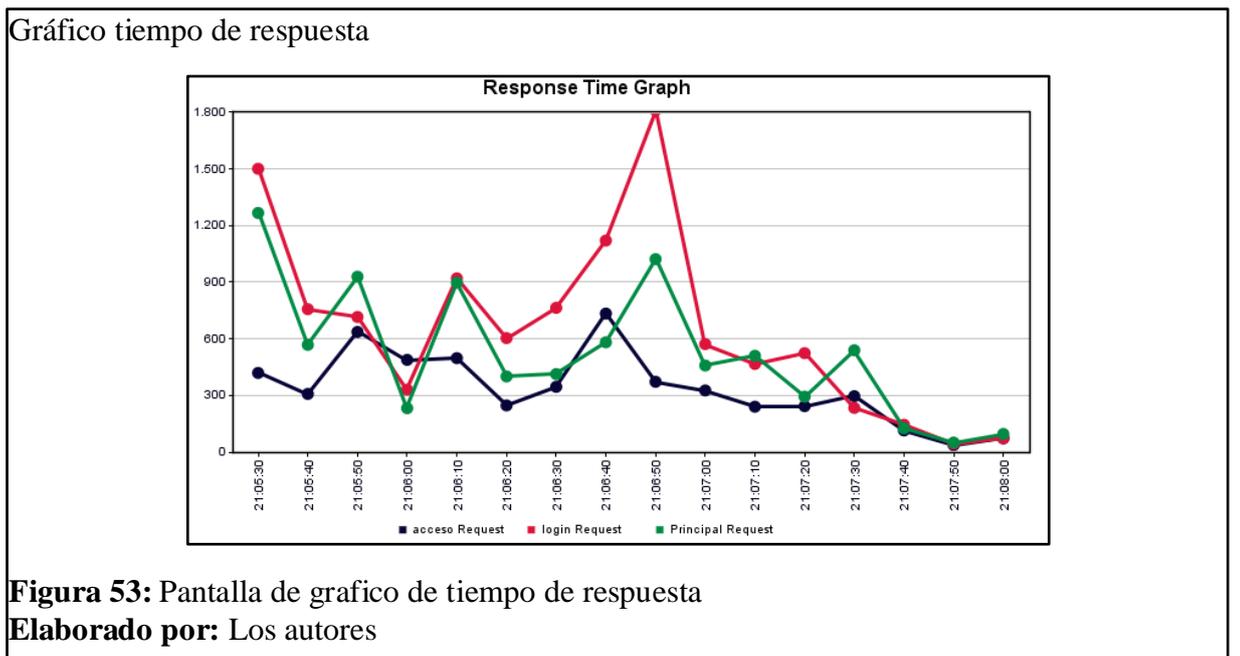
Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
acceso Request	3200	10887	0	5427678	234152,50	2,41%	32,7/min	1,30	0,07	2439,4
login Request	3200	6139	0	5417170	165709,40	2,22%	32,7/min	3,30	0,07	6201,7
Principal Request	3200	5949	0	5416622	165480,08	2,34%	32,7/min	2,56	0,11	4812,3
Total	9600	7658	0	5427678	191212,20	2,32%	1,6/sec	7,15	0,25	4484,5

**Figura 49:** Pantalla de transacciones Exitosas en prueba de stress 1  
**Elaborado por:** Los autores

La distribución de datos se evidencia en el siguiente gráfico con un rendimiento de 1,6 sec utilizando 7,15 kb/sec al simular que se tiene trabajando a 40 usuarios dentro del sistema de generación de balances.



El tiempo de respuesta con 3200 solicitudes es de 1800ms con una velocidad de envío de 0,25 kb/sec promedio. Lo que difiere en este tiempo es que se realizó con una cantidad de peticiones en total 80.



### 3.5.2.2 Resultado pruebas stress 2

Para esta prueba se utilizó los parámetros:

80 usuarios.

10 peticiones.

800 solicitudes

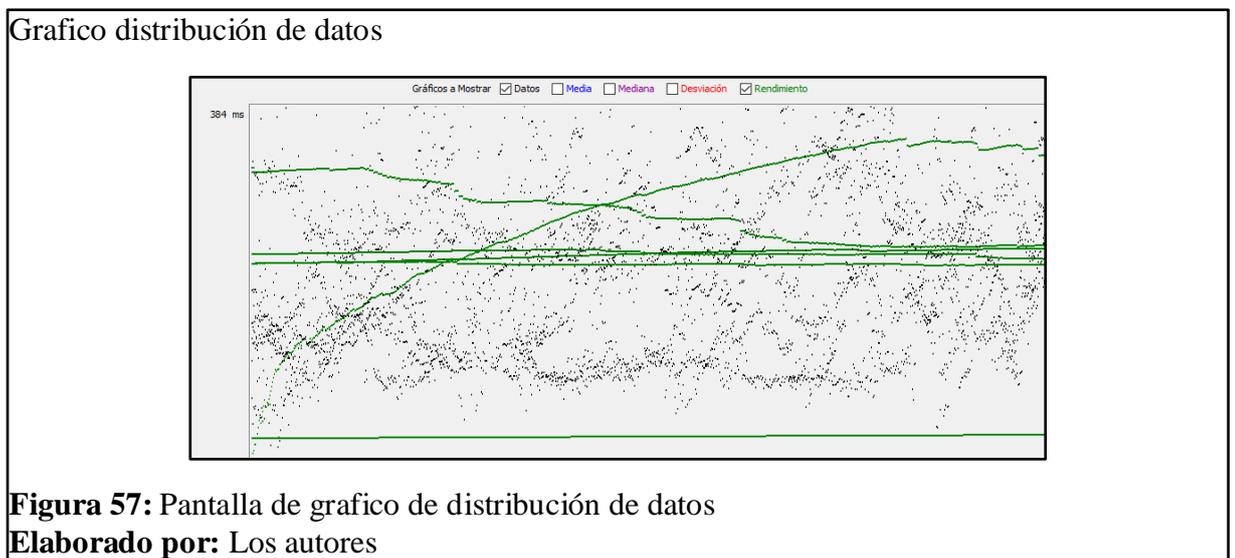
El tiempo de procesamiento de la solicitud es de 12 minutos y 4 segundos, como resultado se evidencia que el sistema puede trabajar de manera normal y así se verifica la capacidad de procesar las peticiones y su rendimiento en horas pico.

Transacción exitosa prueba 2

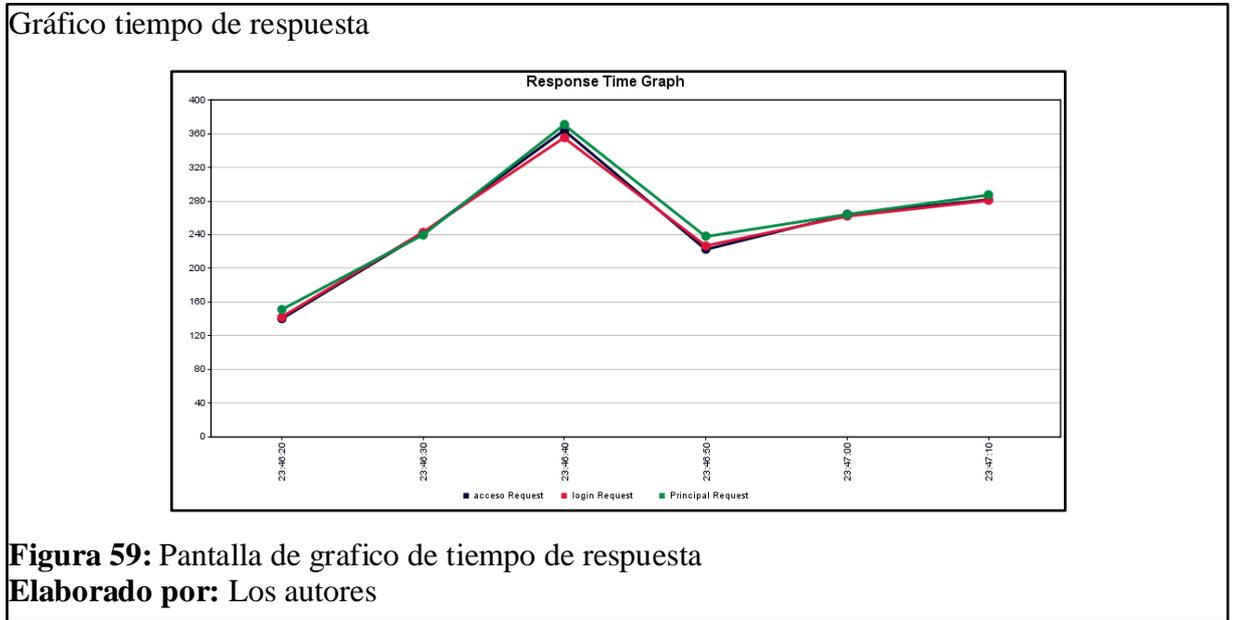
Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
acceso Request	4000	261	39	1289	174,50	0,00%	12,4/sec	29,52	1,55	2431,0
login Request	4000	265	38	1304	175,64	0,00%	12,4/sec	76,25	1,66	6278,7
Principal Request	4000	268	33	1283	175,23	0,00%	12,4/sec	59,05	2,68	4861,5
Total	12000	264	33	1304	175,15	0,00%	37,3/sec	164,74	5,90	4523,7

**Figura 55:** Pantalla de transacciones Exitosas en prueba de stress 2  
**Elaborado por:** Los autores

La distribución se evidencia en el siguiente gráfico con un rendimiento de 37,3 sec utilizando 164,74 kb/sec al simular que se tiene trabajando a 80 usuarios dentro del módulo.



El tiempo de respuesta con 800 solicitudes es de 360ms con una velocidad de envío de 5,90 kb/sec promedio. Lo que difiere en este tiempo es que se realizó únicamente 10 peticiones.



### 3.5.2.3 Resultado pruebas stress 3

Para esta prueba se utilizó los parámetros:

100 usuarios.

30 peticiones.

3000 solicitudes

El tiempo de procesamiento de la solicitud es de 01 minuto y 26 segundos, como resultado se evidencia que el sistema puede trabajar de manera normal y así se verifica la capacidad de procesar las peticiones y su rendimiento en horas pico.

### Transacción exitosa prueba 3

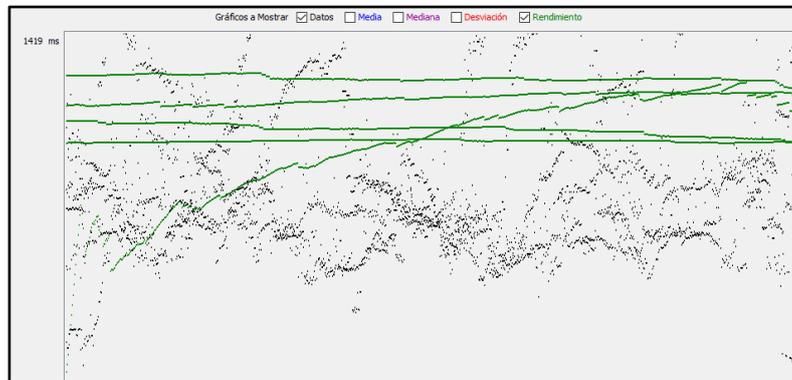
Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
acceso Request	3000	898	80	2466	458,63	0,00%	36,1/sec	85,67	4,51	2431,0
login Request	3000	880	57	2499	445,58	0,00%	36,2/sec	221,75	4,84	6278,8
Principal Request	3000	916	36	3086	470,23	0,00%	36,2/sec	171,96	7,82	4861,5
Total	9000	898	36	3086	458,50	0,00%	108,1/sec	477,71	17,11	4523,7

**Figura 61:** Pantalla de transacciones Exitosas en prueba de stress 3

**Elaborado por:** Los autores

La distribución se evidencia en el siguiente gráfico con un rendimiento de 108,1 sec utilizando 477,71 kb/sec al simular que se tiene trabajando a 100 usuarios dentro del módulo.

### Gráfico distribución de datos

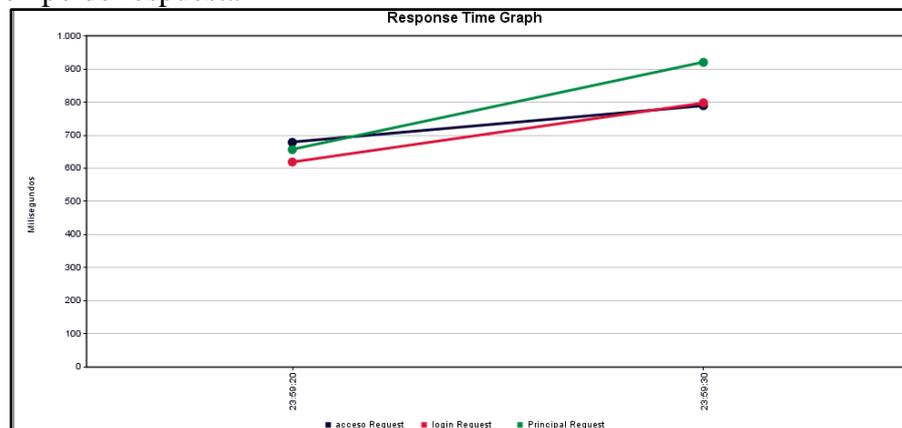


**Figura 63:** Pantalla de grafico de distribución de datos

**Elaborado por:** Los autores

El tiempo de respuesta con 3000 solicitudes es de 900ms con una velocidad de envío de 17,11 kb/sec promedio. Lo que difiere en este tiempo es que se realizó únicamente 30 peticiones.

### Gráfico tiempo de respuesta



**Figura 65:** Pantalla de grafico de tiempo de respuesta

**Elaborado por:** Los autores

## CONCLUSIONES

La casa inspectorial salesiana dirigida por su ente rector cuenta ahora con un sistema de gestión financiera de muy alta fiabilidad contable y tributaria cumpliendo con las necesidades establecidas para este proyecto.

La información contable de las casas y obras salesianas ahora ya está centralizada y consolidada, a través de la conexión con las bases de datos de los sistemas RUA que nos entregará los balances generales y presupuestarios.

El sistema implementado ayuda significativamente en el trabajo operativo de las personas que llevan la contabilidad. Debido a que las entidades que no tienen instalado el sistema RUA debían llevar sus registros contables en una plantilla de Excel y no contaban con una consolidación general de sus balances.

La arquitectura que utiliza el framework JSF tiene un contenedor para todos sus controladores, esto ayuda a que continúe con su secuencia de flujo sin que tenga la necesidad de enviar solicitudes HTTP Request al servidor. De esta manera tiene una extra seguridad para ataque de fuerza bruta, siendo transparentes todas las solicitudes recibidas en un HTTP Proxy, además permite ahorro de tiempo, esfuerzo y de redundancia en el código.

Se pudo constatar que el servidor de aplicaciones Glasfish ayuda para que el sistema de respuesta a las peticiones, su rapidez aumenta en un 50%. Esto debido a que se implementa con la plataforma java y resulta compatible.

Al realizar las pruebas de caja negra y validación se puede evidenciar que el sistema está funcionando de manera correcta, ya que se han obtenido resultados acordes de acuerdo al funcionamiento dentro del escenario con datos reales.

Se midió el rendimiento con ayuda de jmeter donde se realizó 3 pruebas de stress con 10, 30 y 80 peticiones, la prueba uno y dos respondieron a las peticiones sin problema en el tiempo de 12 con 4 segundos y 01 minuto con 26 segundos respectivamente. La tercera prueba mostro la caída del servidor en el minuto 20 con 5 segundos, esto se dio debido a que fue la mayor carga que se le puso con 80 peticiones y 40 usuarios ejecutando la transferencia de balances.

## RECOMENDACIONES

El usuario administrador de este sistema será el encargado de realizar las parametrizaciones iniciales, consolidar balances y generar reportes a nivel gerencial que mostrará información única y veraz para la toma de decisiones oportunas.

Se recomienda crear todos los usuarios asignando a cada uno sus roles y perfiles para que pueda mostrar todas las opciones y los menús que tengan asignado para cada uno de los actores que estarán a cargo de la administración del mismo.

Se recomienda revisar el manual del usuario para saber las múltiples opciones de reportes que este sistema puede mostrar, cada uno de ellos tiene su respectivo detalle ya sea a nivel general o específicamente por casa u obra salesiana.

El contador general debe guiarse del manual de usuario para que todos los procesos del sistema sean exitosos, con esto se garantiza que ningún registro quede por fuera y esto pueda distorsionar en los resultados finales que el sistema va a mostrar.

Los posibles sistemas a implementar para la administración de la casa inspectorial salesiana se daría en la optimización del sistema RUA, que contemple a todos los módulos y puedan arrojar datos consolidados como puede ser inventarios, bodegas, presupuestos. Esto debido a la gran cantidad de datos que manejan y se necesitaría tenerlos centralizados y pueda mostrar información disponible en todo momento.

Para poder realizar una mejora a este sistema se recomienda trabajar con el mismo framework que ayuda en aplicaciones de tipo empresariales con la gran ventaja de optimizar los recursos del servidor donde está alojado toda la información, de esta

manera se mantiene todos los procesos alineados y con disponibilidad de soporte a cambios y versionamientos con el mismo estándar.

## REFERENCIAS

### Artículos Académicos

Henríquez, N., Iglesias, A., Amaris Ramos, L., Ropain, Y. (2017). *Postgresql una alternativa efectiva en las empresas*. Obtenido de <https://revistas.unisimon.edu.co/index.php/identific/article/download/2482/2375>

Menzinsky, A., López, G., Palacio, J. (julio de 2016). *Scrum Manager*. Obtenido de [https://www.scrummanager.net/files/sm\\_proyecto.pdf](https://www.scrummanager.net/files/sm_proyecto.pdf)

Sierra, F., Acosta, J., Ariza, J., & Salas, M. (2017). *Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web*. Obtenido de <https://revistas.unisimon.edu.co/index.php/identific/article/view/2480>

### Informes

SALESIANOS, E. (2019). *Casa Inspectorial*. Obtenido de [https://salesianos.org.ec/pags/casas\\_obras.jsp?casa=Quito+-+Casa+Inspectorial](https://salesianos.org.ec/pags/casas_obras.jsp?casa=Quito+-+Casa+Inspectorial)

UPS, E. I. (2019). *Proyecto modulo financiero*. Quito.

### Metodologías ágiles

Trigás, M. (2012). *Metodología Scrum*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

### Programas Informáticos

Avalos, S. S. (30 de agosto de 2019). *Pruebas de carga con JMeter*. Recuperado el 17 de agosto de 2020, de <https://sospnt.com/blog/91-load-testing-con-jmeter>

DBC, I. C. (2019). *Operación financiera para la industria global y extendida*. Recuperado el 17 de 08 de 2020, de <http://www.axentit.com.mx/finanzas.html>

Gil, J. G. (16 de diciembre de 2018). *Qué es PostgreSQL*. Obtenido de <https://openwebinars.net/blog/que-es-postgresql/>

- javaHispano, A. (13 de marzo de 2012). *PrimeFaces vs RichFaces vs IceFaces*.  
Obtenido de <http://www.javahispano.org/ecuador/2012/3/13/richfaces-vs-primefaces-vs-icefaces.html>
- JMeter, A. (2020). *JMeter en Español*. Recuperado el 17 de agosto de 2020, de <https://jmeterenespanol.org/>
- myabcm. (19 de maro de 2018). *Conozca los tipos de software financiero y elija el mejor para usted*. Obtenido de <https://www.myabcm.com/es/blog-post/tipos-de-software-financiero/>
- netbeans. (08 de 2020). *Bienvenido a NetBeans y www.netbeans.org*. Obtenido de [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html)
- Novalys. (2019). *FUNCIONALIDADES PRINCIPALES DE POWERDESIGNER*.  
Obtenido de <https://www.powerdesigner.biz/ES/powerdesigner/powerdesigner-features.html>
- NOVATECH. (2015). *GESTIÓN FINANCIERA*. Obtenido de <https://www.grupo-novatech.com/erp/>
- ORACLE. (2020). *java*. Obtenido de [https://www.java.com/es/download/faq/whatis\\_java.xml](https://www.java.com/es/download/faq/whatis_java.xml)
- Quezada, V. (06 de octubre de 2008). *Sybase PowerDesigner: la Herramienta Líder en Modelado Empresarial*. Obtenido de <https://www.revistagadgets.com/2008/10/06/sybase-powerdesigner-la-herramienta-lder-en-modelado-empresarial/>
- Salazar, F. S. (2019). *Microsoft Software para empresas*. Obtenido de <https://lahora.com.ec/noticia/1000122347/microsoft-presentc3b3-software-para-empresas-lojanas>
- SOFINA. (2020). *Software Financiero gestiona tus créditos de forma sencilla y rápida*. Recuperado el 08 de 17 de 2020, de <http://www.sofina.club/>
- ROSERO, W. (2020). *¿Qué es el JDK?* Obtenido de <https://supertecnodroid.com/que-es-el-jdk/>

## **Tesis**

Andrango, B. Jácome, D. (julio de 2013). *Análisis, diseño, construcción e implementación del módulo de anexos del S.R.I. y gestión integradora de los módulos del sistema contable para la casa de Inspectoría Salesiana.*

(Tesis de pregrado). Universidad Politécnica Salesiana. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/5184>

## **GLOSARIO DE TÉRMINOS**

**Sistema RUA:** Sistema Contable Financiero de Casas y Obras Salesianas

**Rendiconto:** Sistema de Contabilidad Open-Source.

**Economato Inspectorial:** Departamento Financiero de la Inspectoría Salesiana Ecuador.

**Casa Inspectorial Salesiana:** Ente regulador y controlador de las Casas Salesianas en Ecuador.

**Casas y Obras Salesianas:** Lugares sin fines de lucro de acogida y ayuda social para personas de la calle o en riesgo de exclusión.