

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
INGENIERA E INGENIERO DE SISTEMAS**

**TEMA:
INTEGRACIÓN DEL SISTEMA GEO-PORTAL SALESIANO VERSIÓN 3.0
PARA LA COMUNIDAD SALESIANA**

**AUTORES:
JULIA LIZETH SÁNCHEZ ALEMÁN
CARLOS ALBERTO YÁNEZ PONCE**

**TUTOR:
GUSTAVO ERNESTO NAVAS RUILOVA**

Quito, septiembre del 2016

CESIÓN DE DERECHOS DE AUTOR

Nosotros Julia Lizeth Sánchez Alemán y Carlos Alberto Yáñez Ponce, con documento de identificación N° 1722759360 y 1714540919, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: "INTEGRACIÓN DEL SISTEMA GEO-PORTAL SALESIANO VERSIÓN 3.0 PARA LA COMUNIDAD SALESIANA", mismo que ha sido desarrollado para optar por el título de: INGENIERO DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, septiembre del 2016



Julia Lizeth Sánchez Alemán

1722759360



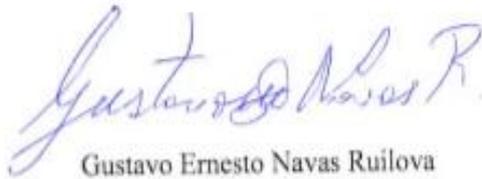
Carlos Alberto Yáñez Ponce

1714540919

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, INTEGRACIÓN DEL SISTEMA GEO-PORTAL SALESIANO VERSIÓN 3.0 PARA LA COMUNIDAD SALESIANA realizado por Julia Lizeth Sánchez Alemán y Carlos Alberto Yáñez Ponce, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, septiembre del 2016



Gustavo Ernesto Navas Ruilova

CI. 1705675625

Dedicatoria

El presente proyecto de tesis lo dedicamos de sobremanera a nuestras familias quienes nos han brindado la confianza, oportunidad, consejos y recursos en todo el tiempo de estudios y por quienes tenemos un sincero sentimiento de agradecimiento en el presente proyecto y en nuestros proyectos futuros.

A mis compañeros y amigos presentes y pasados con quienes compartimos conocimientos, alegrías y tristezas y a todas aquellas personas que durante 5 años estuvieron apoyándonos y lograron que este sueño se haga realidad.

Finalmente dedicamos el presente trabajo al grupo con el que compartimos cada una de las fases del proyecto y con el que siempre permanecerá una valiosa amistad.

Agradecimiento

Los resultados de este proyecto, están dedicados a todas aquellas personas que, de alguna forma, son parte de su culminación. Nuestros sinceros agradecimientos están dirigidos hacia el Ingeniero Gustavo Navas tutor del proyecto, quien con su valiosa ayuda, nos brindó información relevante muy cercana a la realidad de nuestras necesidades. A nuestras familias por siempre brindarnos su apoyo, tanto sentimental, como económico. Pero, principalmente nuestros agradecimientos están dirigidos hacia la institución de la que formamos parte, Universidad Politécnica Salesiana, por brindarnos una educación acorde a las necesidades laborales de nuestro país sin la cual no hubiésemos podido salir adelante.

Lizeth Sánchez Alemán, Carlos Yáñez Ponce.

Tabla de contenido

Introducción	1
1. Marco referencial	3
1.1 Problema	3
1.2 Objetivos	4
1.2.1 Objetivo general.....	4
1.2.2 Objetivos específicos.....	4
2. Marco Metodológico.....	5
2.1 Introducción	5
2.2 Desarrollo metodológico del proyecto de integración	5
2.3 Fases de la metodología XP	7
2.3.1 Fase de Planificación.....	7
2.3.1.1 Historia de Usuario.....	8
2.3.1.2 Plan de Entrega.....	11
2.3.1 Fase de Iteraciones.....	18
2.3.2 Fase de Diseño.....	19
2.3.3 Fase de Desarrollo.....	20
2.3.4 Fase de pruebas:	20
3. Marco teórico	22
3.1 Introducción	22
3.2 Módulos que intervienen en la Integración.....	23
3.2.1 Módulo gestión de datos.....	25

3.2.2 Módulo de gestión visualizador.....	25
3.2.3 Módulo de gestión estilos.....	26
3.2.4 Módulo de administración.....	26
3.2.5 Módulo de Seguridad.....	26
3.2.6 Módulo Integración.....	26
3.3 Informe de estado de proyectos.....	26
3.3.1 Proyecto de Administración.....	26
3.3.1.1 Principales aspectos del módulo de administración.....	27
3.3.1.2 Consideraciones para la integración del módulo de administración.....	28
3.3.1.3 Modelos metodológicos del módulo de administración.....	29
3.3.1.4 Diagrama de componentes.....	29
3.3.1.5 Diagrama de casos de uso.....	31
3.3.2 Módulo de Seguridad.....	33
3.3.2.1 Principales aspectos del módulo de seguridad.....	34
3.3.2.2 Consideraciones para la integración del módulo de seguridad.....	35
3.3.2.3 Modelos metodológicos del módulo de seguridad.....	36
3.3.2.4 Diagrama de casos de uso.....	36
3.3.2.5 Diagrama de secuencia.....	37
3.4 Beneficios de la integración de proyectos.....	38
3.5 Diagramas de Integración.....	38
4. Proceso Integración.....	46
4.1 Restauración módulo de seguridad.....	46

4.2 Restauración módulo administración.....	50
4.3 Comparación visual de ambos sistemas.....	53
4.4 Unificación base de datos	53
4.5 Unificación de paquetes	58
4.5.1 Paquete Servicios.....	58
4.5.2 Paquete clases.	62
4.5.3 Comparación controladores.	63
4.5.4 Paquete Utils.....	68
4.5.5 Paquete filtros.	70
4.5.6 Paquete Enumeraciones.	70
4.5.7 Paquete Constantes.	71
4.6 Comparación contenido web.....	73
4.6.1 Módulo Administración.	73
4.6.2 Módulo Gestión Datos.	76
4.6.3 Módulo Seguridad.....	79
4.6.4 Módulo Reportes.....	80
4.6.5 Módulo Edición datos geográficos.	81
4.7 Detección de errores y corrección de los mismos.....	82
5. Implementación y pruebas	90
5.1 Implementación.....	90
5.2 Características	90
5.3 Restauración de la base de datos.....	90

5.3.1 3. Configuración y carga del archivo .war en el servidor Apache Tomcat	92
5.4 Pruebas	96
Conclusiones	99
Recomendaciones	100
Referencias bibliográficas.....	101
ANEXOS	104

Índice de figuras

Figura 1. Integración Geo-Portal Salesiano iteraciones.....	19
Figura 2. Menú completo con sub módulos faltantes	20
Figura 3. Módulos que forman parte del Sistema integrado	24
Figura 4. Diagrama de componentes del módulo de visualización.....	30
Figura 5. Diagrama de componentes del módulo de gestión de datos geográficos ...	31
Figura 6. Caso de uso creación de un nuevo ítem menú.....	33
Figura 7. Caso de uso administración de perfiles	36
Figura 8. Diagrama desecuencia administración de perfiles.....	37
Figura 9. Proyectos de Integración Principales.....	39
Figura 10. Arquitectura módulo de administración	40
Figura 11. Arquitectura módulo de seguridad	41
Figura 12. Arquitectura módulo de integración	42
Figura 13. Integración de capa de datos.....	43
Figura 14. Integración a nivel funcional	44
Figura 15. Módulos Integrados	45
Figura 16. Problema con la referencia de las librerías.....	47
Figura 17, Configuración JavaServer Faces	48
Figura 18. Problema con versiones en las librerías.....	49
Figura 19. Imágenes no encontradas en el proyecto	49
Figura 20. Problema direccionamiento de páginas	50
Figura 21. Estructura carpeta Geo-Portal.....	51
Figura 22. Usuario nulo al ingresar al sistema.....	52
Figura 23. Visualización formularios.....	52
Figura 24. Comparación módulo administración vs módulo seguridad	53

Figura 25. Diseño Conceptual módulo de Administración de seguridad	55
Figura 26. Modelo conceptual de base de datos	57
Figura 27. Identificación servicios faltantes	59
Figura 28. Comparación código fuente Servicio Beneficiario.....	61
Figura 29. Implementación auditoria de sistema	62
Figura 30. Comparación código fuente Clase Lugar	63
Figura 31. Identificación controladores faltantes.....	64
Figura 32. Código fuente controlador Submenu	66
Figura 33. Instancia de LoginBean para controladores.....	67
Figura 34. Validación de permisos	67
Figura 35. Identificación utilitarios faltantes	68
Figura 36. Comparación código fuente QueryUtil.....	69
Figura 37. Comparación código fuente LoginFilter.....	70
Figura 38. Comparación código fuente ParametrosProyectoEnum	71
Figura 39. Comparación código fuente clase Mensaje	72
Figura 40. Comparación de paquetes	73
Figura 41. Submódulos faltantes Módulo Administración	75
Figura 42. Comparación EdicionSubmodul.xhtml	76
Figura 43. Sub módulos a implementar	77
Figura 44. Corrección relaciones de obra Salesiana	78
Figura 45. Carpetas y páginas que conforman el módulo de seguridad	79
Figura 46. Módulo reportes integrado.....	80
Figura 47. Módulo Edición datos geográficos integrado.....	81
Figura 48. Problema de carga de datos en sub módulo Beneficiario.....	82
Figura 49. Corrección query de consulta submódulo Beneficiario.....	83

Figura 50. Corrección Estilo tabla de datos submóduloAreaInfluencia	84
Figura 51. Mensaje de confirmación cambio de contraseña	85
Figura 52. Método remplazado controlador GeneraReporteSeguridad	86
Figura 53. Fragmento de código para filtros.....	87
Figura 54. Utilización de PrimeFaces para filtros.....	87
Figura 55. Acceso al servidor por PgAdmin.....	91
Figura 56. Creación base de datos	91
Figura 57. Restauración base de datos	92
Figura 58. Servidor apache tomcat inicializado.....	92
Figura 59. Acceso Manager Weapp.....	93
Figura 60. Gestor de aplicaciones web de tomcat.....	94
Figura 61. Carga archivo .war en el servidor	95
Figura 62. Sistema Geo-Portal salesiano deployado en el servidor.....	95
Figura 63. Parámetros de prueba de carga sobre el sistema integrado.	96
Figura 64. Reporte de resultados de carga.	98

Índice de tablas

Tabla 1. Especificación responsabilidades y roles	7
Tabla 2. Historia de Usuario 1.	9
Tabla 3. Historia de Usuario 2.	10
Tabla 4. Historia de Usuario 3	11
Tabla 5. Herramientas ambiente de implementación.....	13
Tabla 6. Plan de Entrega (Administración).....	14
Tabla 7. Plan de Entrega (Gestión Datos).....	15
Tabla 8. Plan de Entrega (Datos Geográficos).....	16
Tabla 9. Plan de entrega (reporte).....	16
Tabla 10. Plan de Entrega (Seguridad)	17
Tabla 11. Plan de Entrega (Gestión de Estilos)	18
Tabla 12. Repositorio digital tesis.....	25
Tabla 13. Herramientas Modulo de administración.....	27
Tabla 14. Caso de uso del Menú Dinámico	32
Tabla 15. Herramientas Modulo de seguridad	34
Tabla 16. Diferencias arquitectura módulos seguridad y administración	46
Tabla 17. Descripción servicios implementados.....	60
Tabla 18. Descripción controladores implementados	65
Tabla 19. Descripción clases utilitarios implementados	69
Tabla 20. Descripción Sub módulos administración	74
Tabla 21. Descripción Sub módulos gestión datos	78
Tabla 22. Descripción Sub módulos seguridad.....	80
Tabla 23. Cuadro comparación funcionalidades existentes en cada módulo.....	88
Tabla 24. Requerimientos de Software integración	90

Tabla 25. Tabla de resultados de prueba de carga	97
--	----

Índice de anexos

Anexo 1. Manual de usuario	104
Anexo 2. Script de base de datos	104

Resumen

El presente proyecto de titulación consiste en la integración e implementación de los proyectos de tesis “Análisis, diseño y construcción del módulo de seguridad e integración de los módulos de visualización y gestión de datos geográficos del Geo-Portal de la comunidad salesiana versión 2.0”, desarrollado por Carina Torres y Darwin Aldas, y “Análisis, diseño y construcción del módulo de administración del Geo-Portal de la comunidad salesiana versión 2.0”, desarrollado por Oswaldo Tutillo y Byron Sandoval, unificando las versiones de los proyectos a nivel de código fuente y herramientas de desarrollo de software, con la finalidad de solventar los problemas surgidos a raíz de la utilización por separado de cada proyecto, siguiendo los lineamientos de la metodología ágil XP (programación extrema) para la gestión metodológica del proyecto.

En el primer capítulo se realiza una breve descripción del cliente final, la comunidad Salesiana, y las actividades que se ejecutaran con el Geo-Portal, así como los beneficios que se obtendrán a partir de la integración de los proyectos.

En el segundo capítulo se describe las fases de la metodología XP que se han sido expuestos de acuerdo al cronograma establecido en el plan del proyecto.

En el tercer capítulo se menciona el procedimiento para la homologación de las versiones de los proyectos a integrar a nivel de código fuente, servidor de aplicaciones, base de datos y librerías externas.

En el cuarto capítulo se describe la implementación del producto final sobre el servidor de la Universidad Politécnica Salesiana, para realizar la actualización del Geo-Portal.

Abstract

This titling project is the integration and implementation projects thesis "Analysis, design and construction of the security module and integration of display modules and management of geographic data Geo-Portal of the Salesian community" version 2.0, developed by Carina Torres and Darwin Aldas, and "Analysis, design and construction management module Geo-Portal of the Salesian community" version 2.0, developed by Oswaldo Tutillo and Byron Sandoval, unifying versions of projects at the level of source code and development tools software, in order to solve the problems arising out of the use of each project separately, following the guidelines of the agile methodology XP (extreme programming) methodology for project management.

In the first chapter a brief description of the final customer, the Salesian community and the activities to be implemented with the Geo-Portal, and the benefits to be derived from the integration of projects that institution and its infrastructure is done, as well as an introduction to the thesis projects to be integrated.

In the second chapter the phases of the XP methodology have been incorporated for project management and have been exposed according to the schedule described in the project plan.

In the third chapter the procedure for the approval of the versions of projects to integrate source level, application server, database and external libraries were used in the development.

In the fourth chapter the implementation of the final product on the server of the Salesian Polytechnic University described, as well as the technical procedure used to update the Geo-Portal to the integrated version.

Introducción

El desarrollo de software para la comunidad Salesiana tiene una dinámica particular diferente a la que normalmente se administra para la gestión de proyectos de software, en donde, las entregas de cada requerimiento se realizan en un tiempo determinado y estrictamente vinculado entre sí, es decir, cada requerimiento se gestiona con diferentes grupos de desarrolladores que deben cumplir con un tiempo de entrega y que no comúnmente pueden depender de actividades entre sí para culminar su trabajo.

Esta dinámica de trabajo con la que se ha venido trabajando los últimos meses ha originado algunos inconvenientes en cuanto a la independización de los requerimientos, a base de módulos que estaban generando entre otros aspectos como ambigüedad y redundancia de información , utilización de herramientas de desarrollo, diferentes versiones de funciones geográficas PostGis para la gestión de mapas y desarrollo de nuevas funcionalidades que se desarrollaban sobre arquitectura diferente entre proyectos (exportación de Geo-Json).

La mencionada forma de trabajo en la que los proyectos se desarrollan para la comunidad Salesiana, no permite establecer un procedimientos base para su ejecución, motivo por el cual, se plantea establecer un sistema integrado para que pueda ser manipulado por varios grupos de trabajo y mitigar los riesgos que significa el desarrollo paralelo de los requerimientos. Considerando como punto de partida los últimos proyectos desarrollados para la comunidad Salesiana que a su vez contienen otros proyectos que se han integrado a sus módulos y que han venido trabajando normalmente y sin novedades para la institución.

Entre los principales aspectos que influyen en la problemática del desarrollo de software para comunidad Salesiana es la utilización de herramientas de desarrollo diferentes entre módulos lo que genera la instalación y mantenimiento de diferentes versiones de plataformas WEB y base de datos, esto ocasionada por los nuevos requerimientos que el cliente solicita para ser agregados al portal Salesiano.

A partir de estas premisas la planificación de una integración de módulos es necesaria para solventar los inconvenientes y novedades que van surgiendo con el pasar de los días y las nuevas necesidades de la institución y teniendo en cuenta una arquitectura que soporte cada uno de los módulos desarrollados y los nuevos que serán agregados al proyecto de integración, de tal forma que cuando se necesite migrar a una nueva infraestructura se tengan la menor cantidad de problemas de migración y homologación de herramientas de desarrollo.

1. Marco referencial

1.1 Problema

El Geo-Portal de la comunidad salesiana al momento cuenta con dos módulos desarrollados de manera independiente que permiten la gestión de los requerimientos establecidos por la institución, sin embargo, el mantener sistemas separados no permite obtener información confiable y única que ayude a la toma de decisiones confiables, es por esto que se plantea unificar los últimos módulos “Análisis, diseño y construcción del módulo de seguridad e integración de los módulos de visualización y gestión de datos geográficos del Geo-Portal de la comunidad salesiana versión 2.0”, y “Análisis, diseño y construcción del módulo de administración del Geo-Portal de la comunidad salesiana versión 2.0” en una sola herramienta y normalizar su desarrollo.

La integración de los sistemas geográficos desarrollados para la comunidad salesiana en su versión 3.0 permite a la institución adecuar la herramienta tecnológica o Geo-Portal a un sistema centralizado en un solo módulo que trabaje con todos los requerimientos funcionales y de diseño establecidos por la inspectoría Salesiana con la finalidad de unificar, normalizar y homologar el manejo de la información subministrada por los diferentes sistemas desarrollados evitando la redundancia de la información.

De esta manera se conseguirá eficacia y eficiencia en el manejo del sistema Geo-Portal salesiano versión 3.0 y estableciendo un marco de trabajo unificado para los futuros contribuidores o desarrolladores que sean parte del proyecto de la comunidad Salesiana.

1.2 Objetivos

1.2.1 Objetivo general.

Analizar, desarrollar, integrar e implementar el Geo-Portal Salesiano Versión 3.0

1.2.2 Objetivos específicos.

Unificar los últimos módulos del Geo-Portal Salesiano.

Desarrollar nuevos requerimientos establecidos por la comunidad salesiana.

Integrar la Tesis “Análisis, diseño y construcción del módulo de seguridad e integración de los módulos de visualización y gestión de datos geográficos del Geo-Portal de la comunidad salesiana versión 2.0” con la tesis “Análisis, diseño y construcción del módulo de administración del Geo-Portal de la comunidad salesiana versión 2.0” unificando versiones de librerías y base de datos que en la actualidad se encuentra unificado en un 60 %.

2. Marco Metodológico

2.1 Introducción

La metodología XP se refiere a los pasos o secuencia de varias metodologías unificadas de tal forma que sean flexibles y de fácil uso, con la finalidad de realizar un desarrollo sencillo. Este esquema metodológico tiene como fin común la simplicidad y como objetivo la satisfacción del cliente. (Martin y Newkirk, 2012).

El desarrollo ágil nos permite agregar valor al desarrollo de software por medio de pasos sencillos que permiten una interacción más cercana con el cliente y entrega de avances con retroalimentación que nos permiten controlar los cambios que el cliente solicite de una forma más organizada y que no afecte en gran cantidad los tiempos definidos mediante cronogramas.

El presente proyecto de integración está gestionado bajo la metodología ágil XP, ya que nos permite el desarrollo ágil y entregas a nivel de iteración de forma rápida y eficaz considerando la dinámica en la que se desenvuelve el proyecto con la comunidad salesiana, cada iteración en el proyecto está relacionada con cada módulo del geo-portal lo que nos permite considerar cada dependencia entre opciones de menú en una sola entrega con el cliente final.

2.2 Desarrollo metodológico del proyecto de integración

La integración del Geo-portal Salesiano en el aspecto metodológico se ha administrado bajo la metodología ágil XP (Programación Extrema) utilizando sus principales características y fases de ejecución por medio de iteraciones, definiendo sus actores principales con el objetivo de agilizar las tareas de desarrollo de software

En el siguiente apartado se define el desarrollo del proyecto de integración alineado a la metodología XP.

2.2.1 Roles.

Se han definido los siguientes roles que son participes principales del proyecto de integración con sus actividades:

Jefe del proyecto: supervisa que se cumpla el cronograma y los entregables y el monitoreo de los avances del proyecto.

Cliente: define especificaciones y requerimientos, controla en proceso de integración del sistema y define pruebas funcionales

Programador: responsable de la integración de cada uno de los módulos, comparación de código fuente de cada módulo, responsable de la generación de código fuente y responsable sobre la integridad del sistema.

Tracker: analiza los avances del proyecto, supervisa el cumplimiento de las estimaciones en cada iteración y controla las pruebas funcionales y los errores reportados.

En la tabla 1 se muestra la definición de las responsabilidades bajo el siguiente esquema entre los integrantes del proyecto:

Tabla 1.
Especificación responsabilidades y roles

Nombre	Descripción	Rol XP
Lizeth Sánchez Carlos Yáñez	Quienes desarrollan el sistema	Programador
Ing. Gustavo Navas	Quien solicita los requerimientos	Cliente
Ing. Gustavo Navas	Controla que las actividades del cronograma se desarrollen	Jefe del proyecto
Ing. Gustavo Navas	Controla pruebas y corrección de errores	Tracker

Nota: Tabla que muestra los roles que interviene en el sistema

2.3 Fases de la metodología XP

Para la administración metodológica del proyecto de integración de la comunidad Salesiana se han establecido y utilizado las principales fases de la metodología XP, teniendo en cuenta su influencia en el desarrollo del proyecto y su aportación para completar los cronogramas establecidos en el plan.

2.3.1 Fase de Planificación.

El sistema integrado tiene como finalidad ofrecer a los usuarios un sitio web para gestionar información sobre las obras salesianas, interactuando con mapas que permiten al usuario tener un acercamiento a la realidad de la infraestructura, viabilidad y acceso de las obras salesianas a lo largo del país, para contar con una visión más cercana a la realidad de cada una de estas y gestionar las necesidades de cada obra Salesiana.

En primera instancia el desarrollo de software para la comunidad Salesiana comenzó de manera independiente de acuerdo a los requerimientos y necesidades de la institución se crearon módulos separados que cumplían su objetivo sin la necesidad de interactuar entre sí, con el pasar del tiempo surgieron algunos problemas,

principalmente, la redundancia de la información y la utilización de versiones diferentes en las herramientas de desarrollo de software.

En la fase planificación se definió los requerimientos iniciales de la integración tomando en cuenta los roles de la metodología XP registrados por medio de historias de usuario, en dónde se especificaron principalmente la necesidad de integrar los 2 últimos proyectos desarrollados para la comunidad Salesiana.

2.3.1.1 Historia de Usuario.

Como se muestra en la Tabla 2, la primera historia de usuario describe la entrega de los proyectos a ser integrados, proyectos de administración y seguridad, para iniciar la planificación y elabora el cronograma de actividades de la integración.

Tabla 2.
Historia de Usuario 1.

Historia de Usuario	
Número: 1	Usuario: Ing. Gustavo Navas
Nombre historia: Restauración de módulos de administración y seguridad.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada: 1
Programador responsable: Lizeth Sanchez & Carlos Yanez	
<p>Descripción:</p> <p>Se requiere la restauración de cada uno de los módulos en el mismo ambiente en el que fueron desarrollados, para validar las funcionalidades comunes que existen entre ellos y lo que sería necesario implementar en el módulo de integración.</p>	
<p>Observaciones:</p> <p>Se realiza la entrega de los proyectos a nivel de código fuente de administración y seguridad para el análisis y planificación de la integración.</p>	

Nota: Tabla que describe las especificaciones que se debe tener en la historia de usuario 1

Como se muestra en la Tabla 3, la historia de usuario 2 describe la entrega del plan del proyecto de forma física y digital, en donde, se describe la planificación del proyecto de integración para su posterior aprobación en el consejo de Carrera.

Tabla 3.
Historia de Usuario 2.

Historia de Usuario	
Número: 2	Usuario: Ing. Gustavo Navas
Nombre historia: Entrega del plan del proyecto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada: 1
Programador responsable: Lizeth Sanchez & Carlos Yanez	
Descripción: Se entrega el plan del proyecto de integración con cronograma de actividades para cada entregable, y se definen fechas de entrega para ingresar a consejo universitario	
Observaciones: Se realiza la entrega del plan de proyecto para la aprobación a consejo de carrera con cronograma coordinado con el tutor del producto tecnológico.	

Nota: Tabla que describe las especificaciones que se debe tener en la historia de usuario 2

Una vez recolectados los requerimientos a partir de las historias de usuario anteriores, se realizó una exploración del estatus de los proyectos a ser integrados, en lo que respecta a versiones y herramientas de desarrollo, con la finalidad de planificar la ejecución de la integración con un escenario claro y que permita un cumplimiento óptimo del proyecto de integración descrito en la tabla 4.

Tabla 4.
Historia de Usuario 3

Historia de Usuario	
Número: 1	Usuario: Ing. Gustavo Navas
Nombre historia: Integración del Geo-Portal Salesiano.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada: 1
Programador responsable: Lizeth Sanchez & Carlos Yanez	
<p>Descripción:</p> <p>Se requiere la integración de los módulos de administración y seguridad del Geo-Portal Salesiano con la finalidad de obtener un producto único y administrable que suministre la información correcta para la gestión de la comunidad Salesiana.</p>	
<p>Observaciones:</p> <p>Se identificó las diferencias entre los proyecto y se realizó la planificación para la integración de cada módulo del sistema.</p>	

Nota: Tabla que describe las especificaciones que se debe tener en la historia de usuario 3

2.3.1.2 Plan de Entrega.

Una vez obtenidos los requerimientos del cliente y analizado el estatus de los proyectos a integrar se procedió a planificar los entregables del proyecto por medio de un cronograma de alineado al plan del proyecto en su fase de ejecución, estimando el esfuerzo necesario para culminar todas la tareas que son necesarias para completar satisfactoriamente la integración.

Para completar satisfactoriamente cada una de las entregas se definió la funcionalidad de los requerimientos acordados con el cliente y se procedió a la

integración de cada uno de los módulos del proyecto, esto hasta completar la primera iteración, teniendo en cuenta, que el requerimiento se refiere en su totalidad a la unificación de los módulos; esta fase no contemplo cambios en los mismos, por lo que, el proceso de planificación no requirió demasiado tiempo y esfuerzo.

En la planificación, se acordó la entrega de los módulos por iteraciones, organizados de acuerdo al menú principal del perfil de administrador que tiene habilitadas todas las opciones del portal.

Se definió un plan de entrega de los módulos a integrar de acuerdo al cronograma establecido y que describe cada una de las opciones del Geo-Portal que se denominaron entregables del proyecto bajo el siguiente esquema:

1ra Entrega:

El proceso de integración inicio con la restauración de los módulos de seguridad y administración. Para su implementación se creó el ambiente sobre el que fue desarrollado cada uno para validar las funcionalidades y planificar el proceso de integración.

2da Entrega:

A partir de la primera entrega se definió el ambiente para la implementación del proyecto integrado, este se desarrolló con las especificaciones técnicas definidas en la planificación y que deberán ser implementadas en el servidor de producción, tal como se especifica en la Tabla 5:

Tabla 5.
Herramientas ambiente de implementación

Herramienta	Versión
Netbeans	7.1
Apache	7.5.0
PostgreSQL	9.1.9
PostGIS	1.5
Java JDK	7

Nota: Tabla que describe las herramientas que se utilizarán para la implementación del módulo integrado

De acuerdo a lo analizado, la segunda entrega se refirió a las clases vinculadas entre los módulos de seguridad y de administración, es decir, la unificación de código fuente de los paquetes base que contienen conexiones, librerías y controladores comunes entre proyectos.

3ra Entrega:

Como tercera entrega tenemos el módulo de Administración que contiene las funcionalidades de edición y parametrización relacionadas con permisos de usuario sobre las opciones del sistema. Como se muestra en la tabla 6.

Tabla 6.
Plan de Entrega (Administración)

Modulo	Opción	Menú
Administración	Auditoria del sistema	
	Edición modulo	
	Edición Permisos Menú	
	Edición de Persona	
	Edición Rol	
	Edición Sub-modulo	
	Edición Usuario	
Tiempo Estimado:	2 Semanas	

Nota: Tabla que describe los objetivos que se debe alcanzar en el entregable 3

4ta Entrega:

En la cuarta entrega tenemos las opciones de Gestión de Datos relacionadas con la parametrización de la información manejada por el portal tal como se muestra en la tabla 7.

Tabla 7.
Plan de Entrega (Gestión Datos)

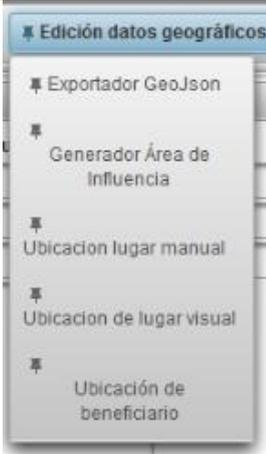
Modulo	Opción	Menú
Gestión de Datos	Área de influencia	
	Beneficiario	
	Casa Salesiana	
	Colaborador	
	Foto	
	Lugar	
	Obra Salesiana	
	Tipo Colaborador	
	Tipo Obra Salesiana	
Tiempo Estimado:	2 Semanas	

Nota: Tabla que describe los objetivos que se debe alcanzar en el entregable 4

5ta Entrega:

En la quinta entrega tenemos las opciones de Edición de Datos Geográficos en donde se ingresa y visualiza la información geográfica y se generan archivos con datos geométricos como se observa en la tabla 8.

Tabla 8.
Plan de Entrega (Datos Geográficos)

Modulo	Opción	Menú
Edición de Datos Geográficos	Exportar Geo-JSON	
	Generador Área de influencia	
	Ubicación Lugar Manual	
	Ubicación Lugar Visual	
	Ubicación de Beneficiario	
Tiempo Estimado:	3 Semanas	

Nota: Tabla que describe los objetivos que se debe alcanzar en el entregable 5

6ta Entrega:

En la sexta entrega tenemos las opciones de reportes, en donde, se visualiza los informes de sistema y auditoria de usuarios como se muestra en la tabla 9.

Tabla 9.
Plan de entrega (reporte)

Modulo	Opción	Menú
Reportes	Reportes del sistema	
	Reporte Auditoria	
Tiempo Estimado:	2 Semanas	

Nota: Tabla que describe los objetivos que se debe alcanzar en el entregable 6

7ma Entrega:

En la séptima entrega tenemos las opciones de Seguridad en donde se administra la seguridad de accesos a las opciones del sistema como se muestra en la tabla 10.

Tabla 10.
Plan de Entrega (Seguridad)

Modulo	Opción	Menú
Seguridad	Perfil	
	Cambio contraseña	
Tiempo Estimado:	2 Semanas	

Nota: Tabla que describe los objetivos que se debe alcanzar en el entregable 7

8va Entrega:

En la 8va entrega tenemos las opciones de Gestión de Estilos y Visualizador, en donde se maneja la información relacionada con mapas de áreas de influencia como se muestra en la tabla 11.

Tabla 11.
Plan de Entrega (Gestión de Estilos)

Módulos	Opción	Menú
Gestión Estilos	Gestión Estilos Visualizador	
Visualizador	Visualizador	
Tiempo Estimado:	3 Semanas	

Nota: Tabla que describe los objetivos que se debe alcanzar en el entregable 8

En base al plan de entrega se estimó el esfuerzo necesario para completar cada uno de los módulos del proyecto de integración en un tiempo determinado que tomo alrededor de 21 semanas entre los 2 desarrolladores asignados, de acuerdo al plan de entrega, se realizó las reuniones para cada avance e ir receptando inquietudes y observaciones.

2.3.1 Fase de Iteraciones.

En la fase de iteración se abordó cada entregable de acuerdo al plan establecido en la fase de planificación y considerando como una iteración cada módulo del proyecto, la dinámica en la que se desarrolló las iteraciones de la integración del Geo-Portal Salesiano se describe a continuación tal como se muestra en la figura 1

Integración Geo-Portal Salesiano iteraciones



Figura 1. Integración Geo-Portal Salesiano iteraciones

Describe los módulos a ser integrados y el flujo de desarrollo para la implementación.

Elaborado por: Lizeth Sánchez & Carlos Yáñez

2.3.2 Fase de Diseño.

En la integración del sistema el diseño únicamente se definirá la ubicación de los sub módulos faltantes en el menú del sistema y las herramientas estándar que se utilizará para el desarrollo de cada una de las páginas html, ya que el diseño de interfaces en su totalidad será el que ya fue definido por los módulos de seguridad y administración.

Se define como herramienta estándar el uso de primafaces ya que en el módulo de administración se implementó las funcionalidades del mismo en algunas de sus interfaces como el uso de filtros en las tablas de consulta.

En la figura 2 se muestra la distribución que tendrán los sub módulos faltantes en el menú del sistema.

Menú completo con sub módulos faltantes

Administración	Gestion de Datos	Edicion Datos Geograficos	Reportes	Seguridad	Gestion de Estilos
<ul style="list-style-type: none">• Auditoria del Sistema• Edicion Modulo<ul style="list-style-type: none">• Edicion Permisos Menu• Edicion Persona<ul style="list-style-type: none">• Edicion Rol<ul style="list-style-type: none">• Edicion Submodulo• Edicion Usuario	<ul style="list-style-type: none">• Beneficiario• Casa Salesiana• Colaborador<ul style="list-style-type: none">• Foto• Lugar• Obra Salesiana<ul style="list-style-type: none">• Tipo Colaborador• Tipo Obra Salesiana	<ul style="list-style-type: none">• Exportar GeoJSON• Generador Area de Influencia• Ubicacion lugar manual• Ubicacion lugar visual• Ubicacion de beneficiario	<ul style="list-style-type: none">• Reportes del sistema• Reportes Auditoria	<ul style="list-style-type: none">• Perfil• Cambio Contraseña	<ul style="list-style-type: none">• Gestion Estilos Visualizador

Figura 2. Menú completo con sub módulos faltantes

Muestra el menú del sistema totalmente integrado

Elaborado por: Lizeth Sánchez & Carlos Yáñez

2.3.3 Fase de Desarrollo.

En la fase de desarrollo se define el proceso que se empleará para la unificación de cada uno de los paquetes que conforman el sistema.

Como parte del proceso de desarrollo se elaboró un esquema de diagramas que permiten controlar todos los aspectos relacionados con la integración de los sistemas y obtener una base metodológica sobre la que se desarrolló el proyecto.

2.3.4 Fase de pruebas:

Una vez culminado el desarrollo de la integración se iniciará la fase de pruebas para cada una de las opciones del portal sobre la arquitectura establecida en la fase de planificación para validar el correcto funcionamiento de las mismas.

Las pruebas que se realizarán en el proceso de integración se definen a continuación:

- **Pruebas de funcionalidad:** las pruebas de funcionalidad se realizarán al finalizar la unificación de los componentes de cada uno de los sub módulos que conforman el sistema para validar su correcto funcionamiento. No se podrá continuar con la unificación de los componentes del siguiente sub módulo sin haber finalizado con el actual proceso.
- **Pruebas de aceptación.** Una vez concluidas las pruebas de funcionamiento de cada uno de los sub módulos del sistema se realizará una prueba de aceptación con el usuario final del sistema para validar las funcionalidades del mismo y determinar cambios que se pueden producir o la corrección de errores que se notifiquen.

Una vez certificado junto al cliente la integración, se inició el proceso de implementación en el ambiente de producción coordinando los accesos y métodos utilizados para la instalación, los mismos que se definieron bajo un esquema seguro que no interfiera con la ejecución de los procesos del servidor por medio de SSH e IP's privadas.

3. Marco teórico

3.1 Introducción

La comunidad Salesiana viene desarrollando proyectos de vinculación con la sociedad hace un largo tiempo que permiten a personas de escasos recursos educarse en un ambiente colaborativo para aportar a la sociedad ecuatoriana, la comunidad tiene inaugurados algunas obras que alojan a una gran cantidad de personas alrededor del país y que requieren una administración adecuada y barata para su gestión, teniendo en cuenta la misión de la institución sin fines de lucro y con un gran aporte sobre las familias de escasos recursos.

Con el objetivo de cumplir con las necesidades de la comunidad Salesiana, se ha desarrollado un sistema que permita llevar un control de las obras que han sido creadas y automatizar procesos que permitan mejorar el almacenamiento de este tipo de información, por medio de un Geo-Portal, diseñado exclusivamente para la administración y gestión de las obras creadas y que en el futuro se seguirán inaugurando.

El mencionado sistema permite conocer toda la información del servicio social que brinda la Universidad y las diferentes actividades que realizan, por medio de mapas interactivos que permiten identificar de manera sencilla cada una de los lugares en donde se encuentran.

El desarrollo de los módulos para la comunidad Salesiana se ha venido desarrollando de forma separada y de acuerdo a los requerimientos que en un instante determinado se ha necesitado incorporar, a partir de esta premisa, el desarrollo individual de cada módulo de forma separada ha ocasionado que la información ingresada en cada

proceso sea ambigua en algunos aspectos, teniendo problemas de fidelidad de información.

A partir de estas inconvenientes se ha visto la necesidad de unificar los módulos del Geo-Portal, para obtener una información única y segura relacionada con las actividades de la comunidad, tanto a nivel de seguridad de la información como en el aspecto de gestión de la misma.

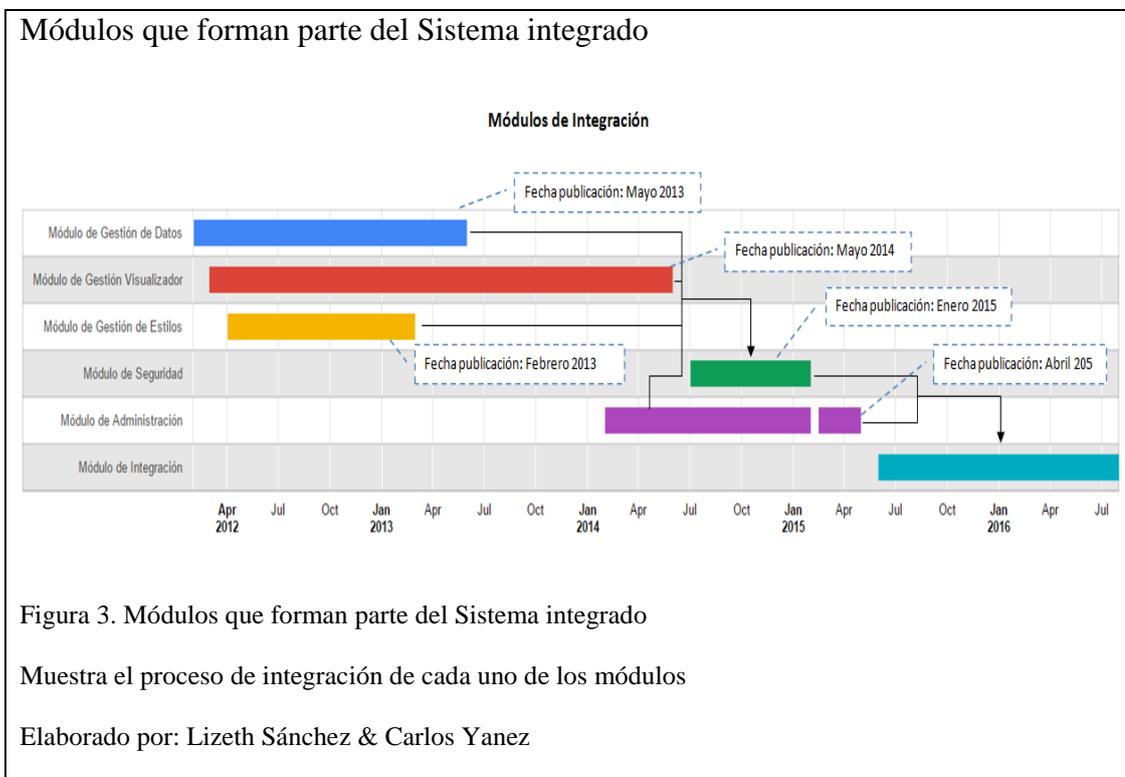
La integración de los módulos del Geo-Portal ayudara a la comunidad también en el aspecto económico y de infraestructura tecnológica ya que el mantener varias funcionalidades por separado genera costos de mantenimiento mayores que el tener una sola versión unificada.

3.2 Módulos que intervienen en la Integración

La integración de cada una de los módulos del Geo-Portal Salesiano dará como resultado un sistema completo que permita manejar funcionalidades tanto de administración, seguridad, gestión de mapas, visualización entre otros, así como la planificación de nuevos requerimientos con una arquitectura única que permita la incorporación de nuevos procesos en beneficio de la comunidad Salesiana, la integración del Geo-Portal Salesiano unifica versiones anteriores que han sido incorporadas tanto en el proyecto de seguridad como en el proyecto de administración.

En la figura 3 se muestra cada uno de los módulos que han sido integrados en un intervalo de tiempo especificando las fechas de finalización de cada uno de ellos; los módulos de gestión de datos, gestión de visualizador, gestión de estilos y aproximadamente un 60% del módulo de administración resaltado de color morado fueron integrados en el módulo de seguridad. Por lo tanto en el módulo de

integración se incorporó únicamente los módulos de seguridad y el 40% restante del módulo de administración.



La documentación de cada uno de los módulos se encuentra en el repositorio digital de la universidad donde se puede encontrar información detallada del desarrollo de los mismos, el acceso a cada uno de ellos se muestra en la tabla 12.

Tabla 12.

Repositorio digital tesis

Módulo	Integrantes	URL	Fecha Publicación
Gestión de datos	Fabrizio Mullo - Andrea Moya	http://dspace.ups.edu.ec/handle/123456789/4905	Mayo 2013
Gestión Visualizador	Víctor Cofre - Stalyn Toledo	http://dspace.ups.edu.ec/handle/123456789/6835	Mayo 2014
Gestión estilos	Gabriela Eras - Cristian Arcos	http://dspace.ups.edu.ec/handle/123456789/4341	Febrero 2013
Seguridad	Darwin Aldas - Carina Torres	http://dspace.ups.edu.ec/handle/123456789/7581	Enero 2015
Administración	Oswaldo Tutillo - Byron Sandoval	http://dspace.ups.edu.ec/handle/123456789/10123	Abril 2015

Nota: Describe cada uno de los módulos integrados y el acceso a la documentación de los mismos

3.2.1 Módulo gestión de datos.

Este módulo abarca la parametrización de la información que es manejada en todo el entorno de la aplicación como son creación, edición, eliminación y visualización de casas, obras salesianas, beneficiarios, colaboradores, fotos, lugares, tipos de obra salesiana y tipo de colaboradores.

3.2.2 Módulo de gestión visualizador.

Este módulo permite la gestión de la información de las casas y obras salesianas por medio de mapas interactivos que permiten la visualización geográfica de las mismas, utilizando georreferenciación de puntos y polígonos que pueden ser manipulados por el usuario permitiendo una interfaz más interactiva.

3.2.3 Módulo de gestión estilos.

Este módulo permite controlar los estilos asignados a un lugar, obra salesiana o beneficiario, como la personalización de iconos y colores.

3.2.4 Módulo de administración.

Este módulo permite la administración del sistema en lo que respecta a la auditoria, creación y edición y eliminación de módulos, sub-módulos, usuario, personas, permisos de menú y roles que serán definidos para controlar los accesos al sistema.

3.2.5 Módulo de Seguridad.

Este módulo nos permite la administración de usuarios, perfiles y sesiones, para de esta manera, restringir el acceso a los procesos o funcionalidades del sistema dependiendo del tipo de rol que ha sido asignado.

Se consideran aspectos claves la autenticación, que se refiere a la validación de la identidad del usuario utilizando credenciales de ingreso (nombre de usuario y contraseña) y a la autorización de acceso que define los permisos a la navegabilidad en el portal, es decir, las opciones a las que el usuario puede ingresar.

3.2.6 Módulo Integración.

El módulo de integración unifica todos los módulos anteriormente mencionados con la finalidad de obtener un sistema completo que maneje cada una de las funcionalidades previamente desarrolladas.

3.3 Informe de estado de proyectos

3.3.1 Proyecto de Administración.

El proyecto del módulo de administración se refiere a la gestión de los recursos de los que disponga el Geo-Portal, es decir, el desarrollo de sub-módulos que permitan la parametrización, creación actualización y eliminación, de registros de información involucrados con las áreas de influencia de la comunidad Salesiana, que serán implementadas en los visualizadores y reportes del sistema geográfico, así como la creación de archivos en formato GEO-JSON para la exportación de información geográfica hacia otros sistemas de información o bases de datos geográficas.

La arquitectura bajo la cual está definido el proyecto de administración se detalla en la tabla 13:

Tabla 13.

Herramientas Modulo de administración

Patrón de arquitectura MVC (modelo, vista, controlador)	
Sistema Operativo	Centos versión 5.9
Base de Datos	PostgreSQL versión 9.1.9
Datos Espaciales	PostGIS versión 1.5
Servidor Web	Apache 2.2.24
Lenguaje de Desarrollo	Java 7

Nota: Muestra las herramientas que se utilizaron para la restauración del módulo de administración

3.3.1.1 Principales aspectos del módulo de administración.

Las principales funcionalidades que se encuentran implementadas en el proyecto de administración nos ayudaron a definir una arquitectura que permita un desarrollo acorde a las necesidades de la comunidad salesiana y que no afecte a las funcionalidades en ese momento implementadas. A continuación una breve descripción de los aspectos principales del módulo de administración:

Una parte esencial del proyecto de Administración es la optimización del manejo de la información Geo-gráfica a través del formato de intercambio de datos geográficos GEO-JSON y que tiene como principal funcionalidad la creación dinámica de archivos planos que contienen referencias a las ubicaciones de las áreas de influencia de la comunidad salesiana, esto con la finalidad de agilizar y automatizar la exportación de los datos en un formato legible desde otros sistemas de información y bases de datos geográficas, que permitan ofrecer mayor información y desarrollo de futuros requerimientos.

La creación del menú dinámico desde la capa de negocio permite optimizar el acceso de acuerdo a los perfiles de usuario, facilitando la tarea de asignación de perfiles y opciones desde el menú principal, extraídos directamente desde la base de datos e implementados en la capa de visualización sin mayor esfuerzo.

La gestión del área de influencia de una obra salesiana es una funcionalidad que nos permite por medio de herramientas JavaScript graficar un área en un mapa y obtener las coordenadas que posteriormente serán almacenados en formato geográfico en la base de datos PostgreSQL, facilitando el ingreso de información a nivel de coordenadas de manera interactiva en el Mapa del portal.

3.3.1.2 Consideraciones para la integración del módulo de administración.

A partir de la exploración de las principales funcionalidades del módulo de administración, se establece que la integración de los proyectos requerirá conservar los principales procesos o funcionalidades como son la de generación de archivos en formato GEO-JSON, Menú dinámico y herramientas para gráficos de interactivos de

áreas de influencia y adaptarlo en un 100% a las nuevas plataformas de desarrollo, por lo que la nueva arquitectura deberá adaptarse y ofrecer la misma funcionalidad con la menor cantidad de cambios en el código fuente, considerando las versiones de Post-GIS y el servidor Apache Tomcat.

3.3.1.3 Modelos metodológicos del módulo de administración.

El desarrollo del módulo de administración se basa en la definición de un marco de trabajo metodológico basado en diagramas de caso de uso y componentes que edifican el desarrollo de software y permiten una comprensión ilustrada del sistema geográfico.

Los principales diagramas que definen al módulo de administración se describen a continuación.

3.3.1.4 Diagrama de componentes.

Los componentes sobre los cuales se encuentra desarrollado el módulo de administración conservan la arquitectura sobre las que fueron creados desde la versión 1 elaborado por Víctor Cofre y Stalin Toledo y que para la integración únicamente requirió una actualización de las versiones.

Como se puede apreciar en la Figura 4 el módulo de visualización utiliza librerías externas que permiten optimizar la vista de las páginas enriqueciendo el formato de los componentes WEB y que se integran al Servidor y al navegador, permitiendo en conjunto, trabajar con la información de la base de datos de una manera rápida, estos componentes requirieron la actualización de sus versiones en la integración.

Diagrama de componentes del módulo de visualización.

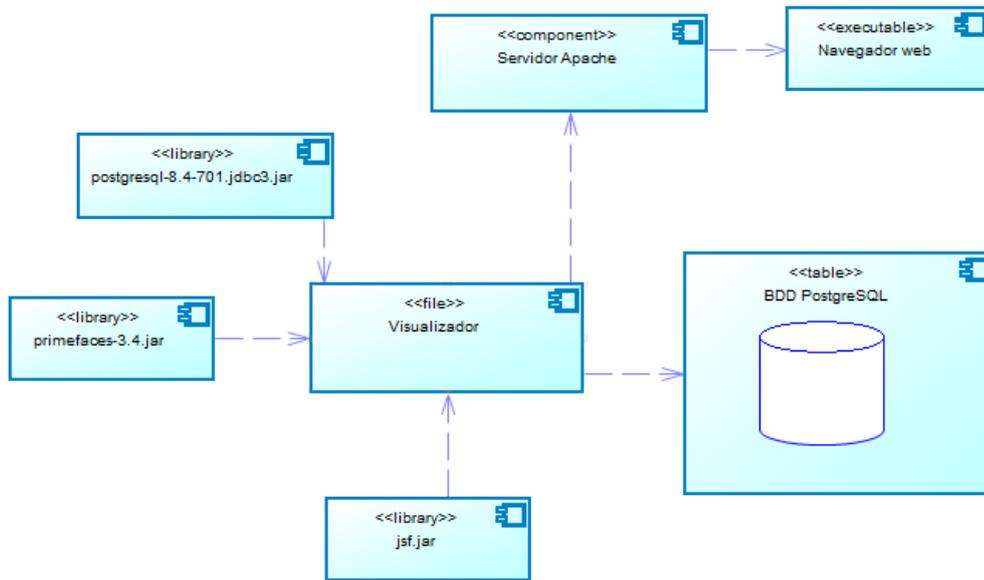


Figura 4. Diagrama de componentes del módulo de visualización.

Muestra el diagrama de componentes que utiliza el sistema

Elaborado por: Cofre Víctor & Toledo Stalin

Como se puede apreciar en la Figura 5, el módulo de gestión de datos geográficos incorpora en su desarrollo librerías externas que permiten optimizar la visualización de la información y a su vez se facilitan el manejo de los datos geográficos minimizando los tiempos de desarrollo, al igual que el módulo de visualización, estas herramientas se integran con facilidad al Servidor y Navegador WEB.

Diagrama de componentes del módulo de gestión de datos geográficos

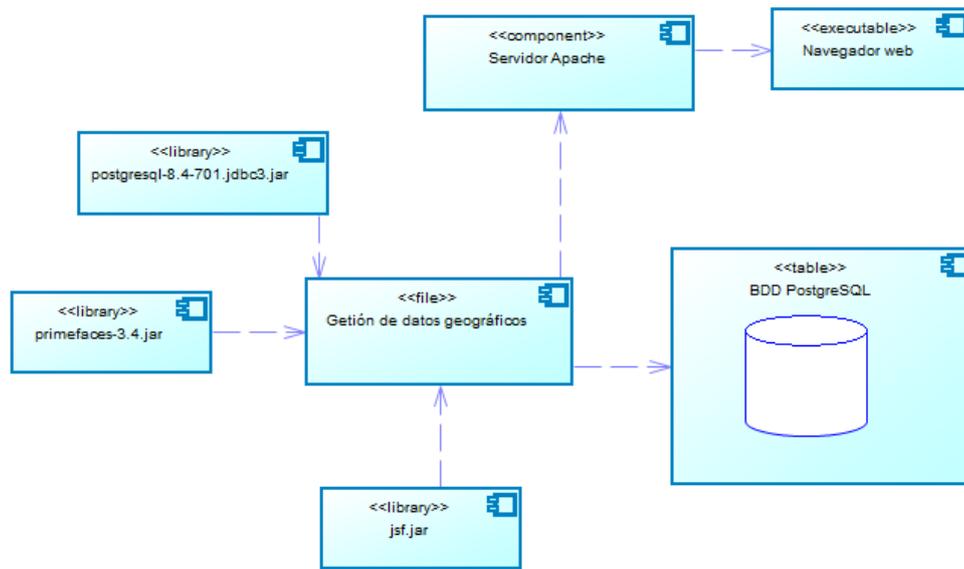


Figura 5. Diagrama de componentes del módulo de gestión de datos geográficos

Muestra el diagrama de componentes del módulo de datos geográficos

Elaborado por: Cofre Víctor & Toledo Stalin

3.3.1.5 Diagrama de casos de uso.

Los principales casos de uso en el módulo de administración definen la funcionalidad básica de la parametrización de información en cada módulo del proyecto de administración, relacionados con el acceso por perfiles al sistema geográfico.

Como se puede apreciar en la Tabla 14, el procedimiento para la creación de una nueva opción en el menú principal requiere como precondition, tener el perfil de administrador y tener un conocimiento técnico y funcional de la aplicación, este caso de uso describe el procedimiento para la creación de nuevas opciones para el menú principal.

Tabla 14.
Caso de uso del Menú Dinámico

Caso de Uso # 1	Creación de un nuevo ítem para el menú dinámico
Actores	Usuario Administrador
Camino Principal	<ol style="list-style-type: none"> 1. Crear el ítem del menú 2. Escoger el perfil a cual será visible este submenú 3. Brindar los permisos que tendrá el perfil escogido para ese submenú 4. Guardar la información
Precondiciones	Para crear el submenú tiene que tener perfil de administrador
Post-condiciones	El submenú se registra en la Bases de datos

Nota: Muestra el caso de uso para la creación de un nuevo módulo y sub módulo

Como se muestra en la Figura 6, el caso de uso de creación de una nueva opción en el menú principal de forma visual requiere de un actor con permisos de administrador, el cual realiza el procedimiento y guarda la información en la base de datos.

Caso de uso creación de un nuevo ítem menú

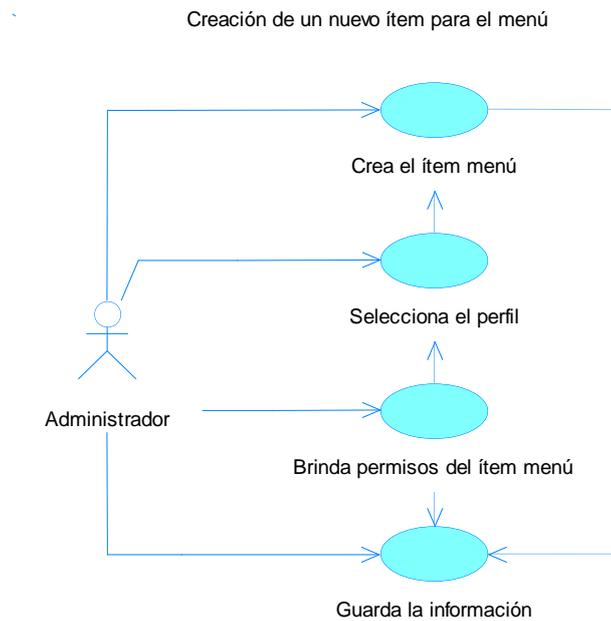


Figura 6. Caso de uso creación de un nuevo ítem menú

Muestra el caso de uso para la creación de un nuevo ítem en el menú del sistema

Elaborado por: Oswaldo Tutillo & Byron Sandoval

3.3.2 Módulo de Seguridad.

El proyecto del módulo de seguridad se refiere a la administración de los perfiles de usuario y el esquema de protección de información con la finalidad de controlar el acceso a las opciones del Geo-Portal Salesiano, enfocado en la autorización, autenticación y auditoria de accesos, mediante el cual se permitirá conocer quien administrara los recursos del portal y los eventos o alteraciones en la información.

Por medio de esta funcionalidad se permite conocer las diferentes actividades que los usuarios realizan sobre la información en las bases de datos, para de esta manera, controlar los accesos a la misma, esta funcionalidad de seguridad se encuentra

agregada en cada inserción actualización o creación de información en todos los módulos del Geo-Portal Salesiano conteniendo en si la integración de anteriores proyectos(módulo de visualización y gestión de datos) y que con respecto al módulo de administración tiene un pequeño desfase de versiones, ocasionado por el desarrollo en paralelo de ambos requerimientos

El proyecto del módulo de seguridad consta de la siguiente arquitectura de software como se muestra en la tabla 15:

Tabla 15.
Herramientas Modulo de seguridad

Patrón de arquitectura MVC (modelo, vista, controlador)	
JSF(Java Server Faces)	Versión 2.0
Primfaces	Versión3.3
JDBC (Java DatabaseConectivity)	Conector con la base de datos
Apache Tomcat	Versión 7.5.0
PostgreSQL	Versión 9.1
PostGis	Versión 1

Nota: Muestra las herramientas que intervienen en la restauración del módulo de seguridad

3.3.2.1 Principales aspectos del módulo de seguridad.

El módulo de seguridad tiene como objetivo principal la gestión de la seguridad y la autorización de los accesos a los módulos del Geo-Portal Salesiano, funcionalidades que están agregadas en cada módulo y que generan un log de actividades implementado mediante reportes en PDF, las principales funcionalidades del módulo de seguridad son:

La autenticación del Geo-Portal Salesiano permite controlar el acceso a los módulos por perfil de usuario por medio de usuario y contraseña encriptado que garantiza la identidad y privacidad de la información

La autorización complementa la creación del menú dinámico desarrollado en el módulo de administración y permite definir los accesos a los módulos restringiendo las labores de actualización, creación o eliminación y visualización.

La auditoría de accesos permite identificar los usuarios que han modificado información en la base de datos mediante un historial de transacciones que define actualizaciones, eliminaciones y creación de registros de información.

3.3.2.2 Consideraciones para la integración del módulo de seguridad.

A partir de la identificación de las principales funcionalidades del proyecto de seguridad, se establece que la integración requerirá una actualización del código fuente y librerías utilizadas en el desarrollo del módulo, es decir, de acuerdo a las versiones de las herramientas de desarrollo fue necesario cambiar una parte del código fuente para acoplarlo a las nuevas arquitecturas a nivel de reportes y auditoria o log de información, este proceso al estar relacionado a nivel de capa de negocio facilita la modificación del código fuente.

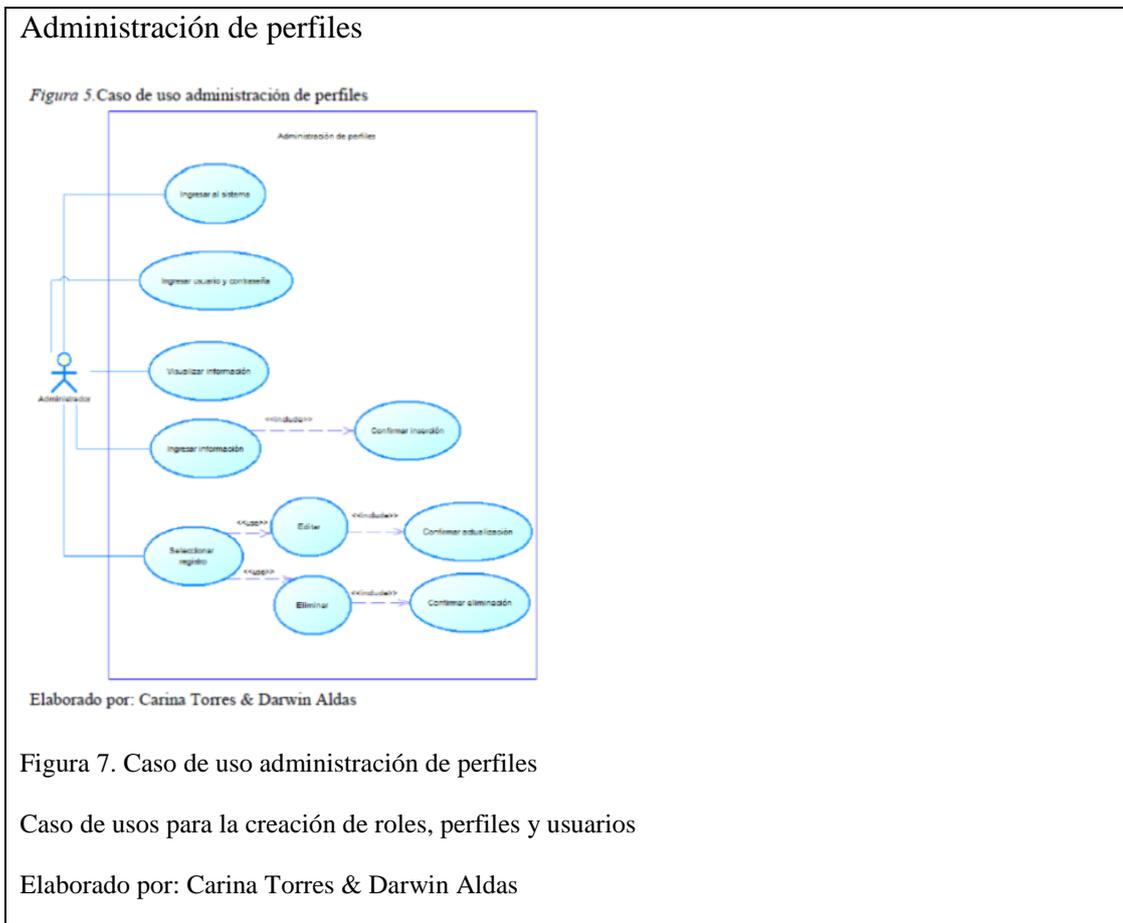
En cuanto a la visualización de la información el proyecto contenía una versión diferente de código HTML y vistas por lo que se debió cambiar estas funcionalidades con las del módulo de administración para homologar las vistas de cada página que forman parte del Geo-Portal Salesiano.

3.3.2.3 Modelos metodológicos del módulo de seguridad.

La metodología utilizada para el módulo de seguridad está basada en las buenas prácticas de OOADM, por medio de en una arquitectura de 3 capas que centra su implementación en un conjunto de prototipos de visualización a continuación describimos los más importantes.

3.3.2.4 Diagrama de casos de uso.

En el caso de uso presentado a continuación por medio de la Figura 7, se detalla como el usuario administra los perfiles e ingresa información para ser insertada en la base de datos, con su respectivo camino secundario.



3.3.2.5 Diagrama de secuencia.

La administración de perfiles permite la gestión de la seguridad de los módulos del geo-portal salesiano para que cada usuario tenga accesos a la información que necesita.

A continuación se describe la administración de perfiles por medio de un diagrama de secuencia en la figura 8.

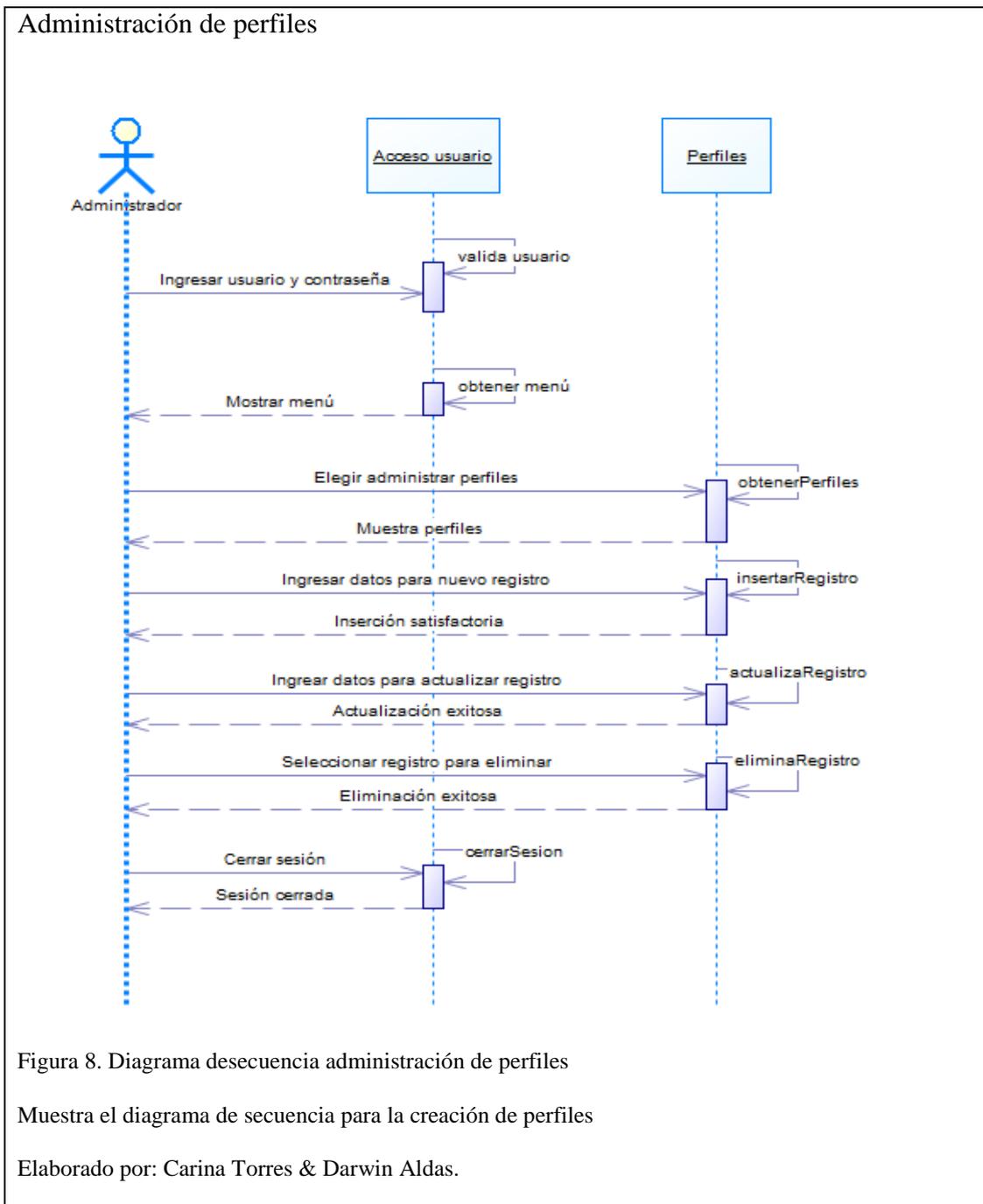


Figura 8. Diagrama de secuencia administración de perfiles

Muestra el diagrama de secuencia para la creación de perfiles

Elaborado por: Carina Torres & Darwin Aldas.

3.4 Beneficios de la integración de proyectos

El desarrollo de proyectos de software implica principalmente satisfacer una necesidad determinada plasmada en requerimientos, que tienen como finalidad, establecer una solución que pueda satisfacer en un gran porcentaje las necesidades del cliente final considerando las limitantes que se tengan a nivel de costos, infraestructura y tiempo.

Desde el punto de vista del desarrollo de software para la comunidad Salesiana, el planteamiento de una solución adecuada que soporte la dinámica de ejecución de los diferentes proyectos a ella asignados, generaran un beneficio que aporte valor a la institución y que permitan soportar futuros requerimientos que seguirán formando parte del sistema unificado, teniendo en cuenta que en futuros desarrollos se requerirá una actualización del proyecto, la integración permitirá la migración de todos los módulos en un solo proyecto, beneficiando a la institución en tiempos de entrega y ejecución de nuevos requerimientos de forma eficaz y confiable.

3.5 Diagramas de Integración

3.5.1 Módulos integración.

En la Figura 9 se describe los principales aspectos de la integración del Geo-Portal Salesiano, como son la Autenticación y Autorización en la parte de seguridad, y la generación de Geo-JSON y perfiles en la parte de Administración, considerándolos como elementos principales de cada proyecto y que marcan la diferencia entre cada uno.

Proyectos de Integración Principales

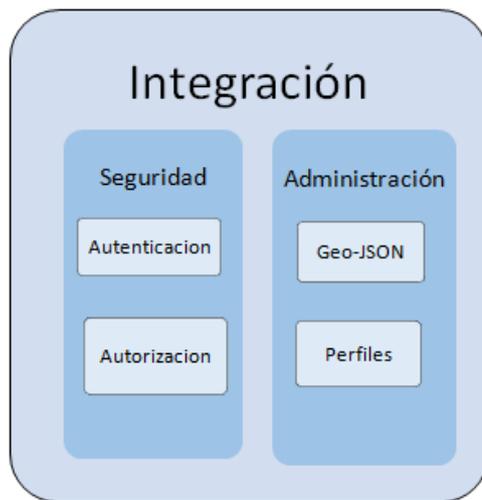


Figura 9. Proyectos de Integración Principales

Describe los procesos más destacados que serán integrados de cada uno de los sub módulos

Elaborado por: Lizeth Sánchez & Carlos Yáñez

3.5.2 Integración a nivel de arquitectura.

A nivel de arquitectura la integración del Geo-Portal Salesiano implica la normalización de las herramientas sobre las que trabaja el sistema, es decir, servidor WEB, base de datos, lenguaje de programación (FRAMEWORK), en una arquitectura que soporte las funcionalidades desarrolladas en los 2 proyectos a integrar, proyecto de seguridad y proyecto de administración.

La figura 10 muestra la arquitectura del módulo de administración sobre el que fue desarrollado el proyecto.

Arquitectura módulo de administración

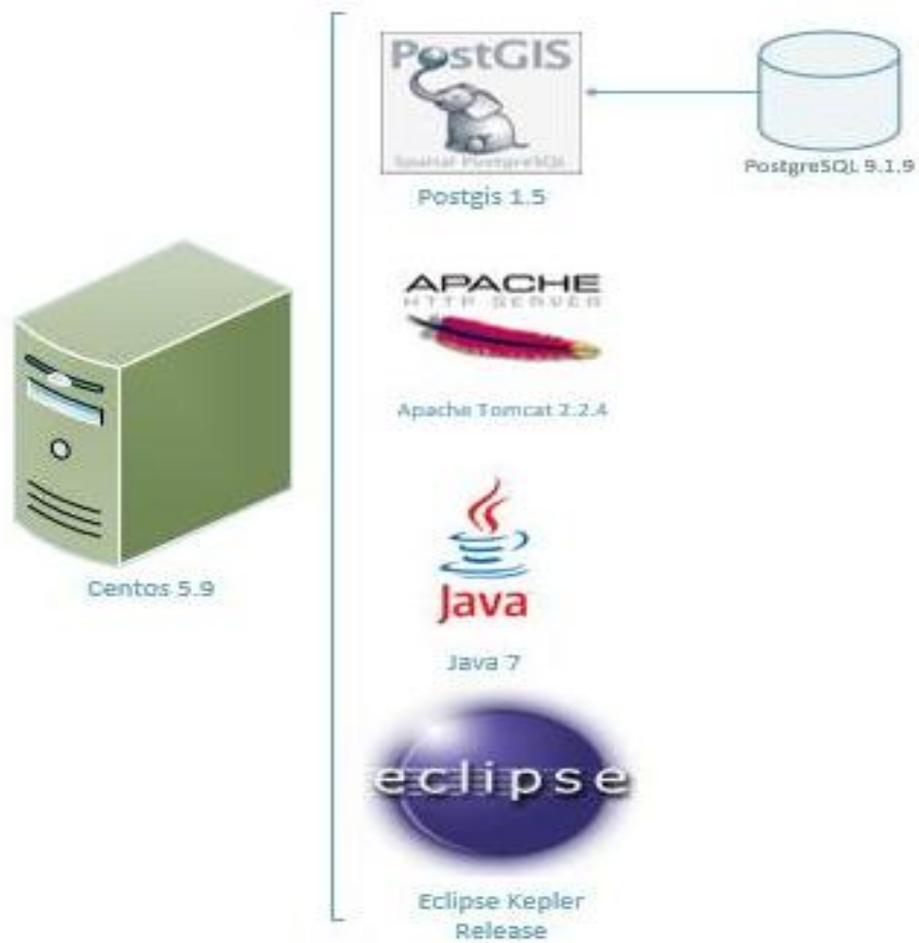


Figura 10. Arquitectura módulo de administración

Muestra la arquitectura que se utilizó para la restauración del módulo de administración

Elaborado por: Lizeth Sánchez & Carlos Yáñez

La figura 11 muestra la arquitectura del módulo de seguridad sobre el que fue desarrollado el proyecto.

Arquitectura módulo de seguridad

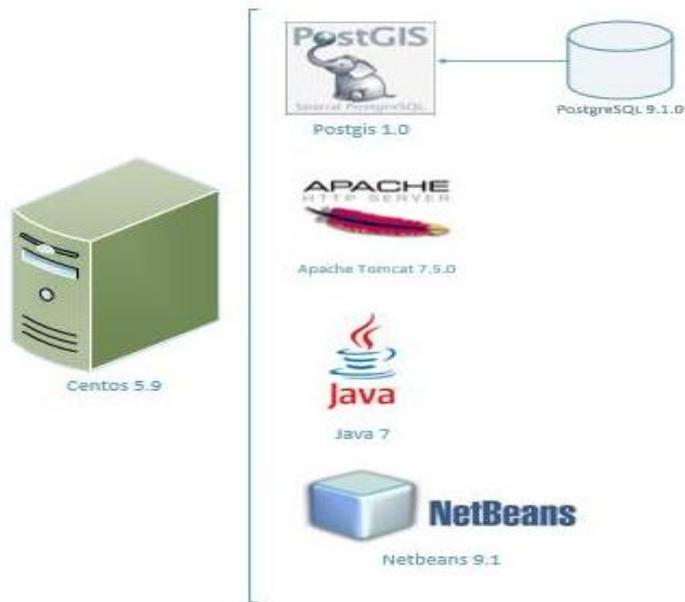


Figura 11. Arquitectura módulo de seguridad

Muestra la arquitectura utilizada para la restauración del módulo de seguridad

Elaborado por: Lizeth Sánchez & Carlos Yáñez

La implementación del módulo de integración se basa en las arquitecturas anteriormente descritas; se definió la utilización de las versiones más actuales de cada una de las herramientas, en cuanto al IDE de desarrollo se utilizó Eclipse teniendo en cuenta su facilidad de instalación y soporte sobre las herramientas para el desarrollo.

La figura 12 describe la integración de la arquitectura para el proyecto del Geo-Portal Salesiano.

Arquitectura módulo de integración

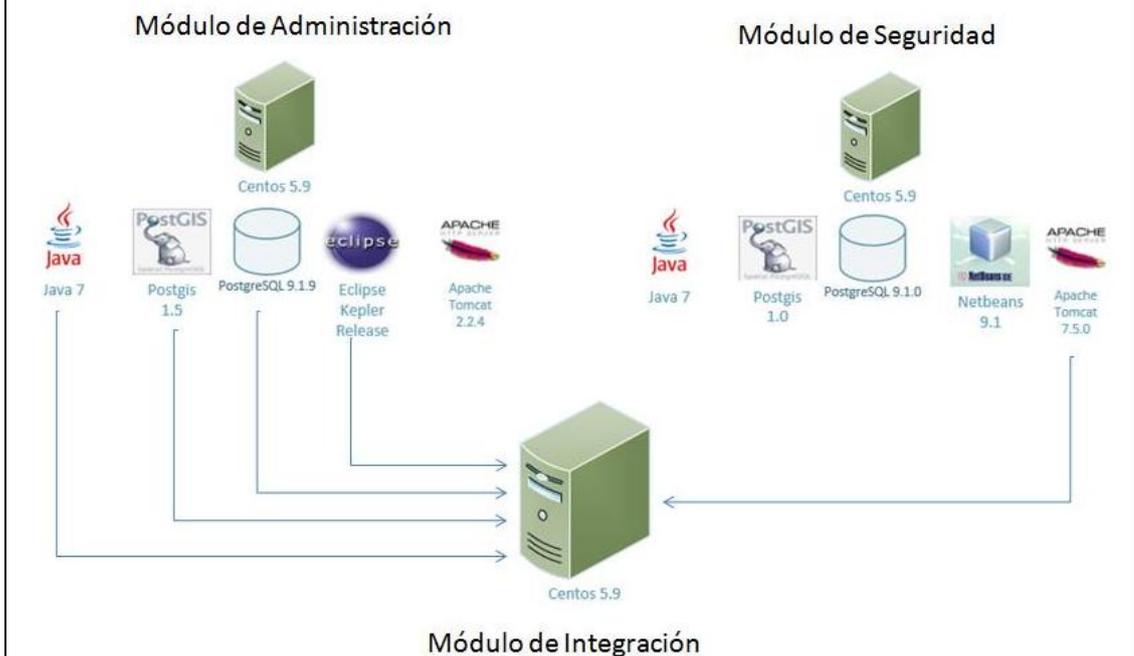


Figura 12. Arquitectura módulo de integración

Muestra la arquitectura que se utilizará para el módulo integrado

Elaborado por: Lizeth Sánchez & Carlos Yáñez

3.5.3 Integración de capa de datos.

Para la integración a nivel de la capa de datos fue necesaria la ejecución del siguiente proceso:

3.5.3.1 Restauración base de datos modulo seguridad.

Se utilizó la versión del módulo de seguridad ya que fue el esquema de base de datos con el que iniciaron el desarrollo de ambos módulos

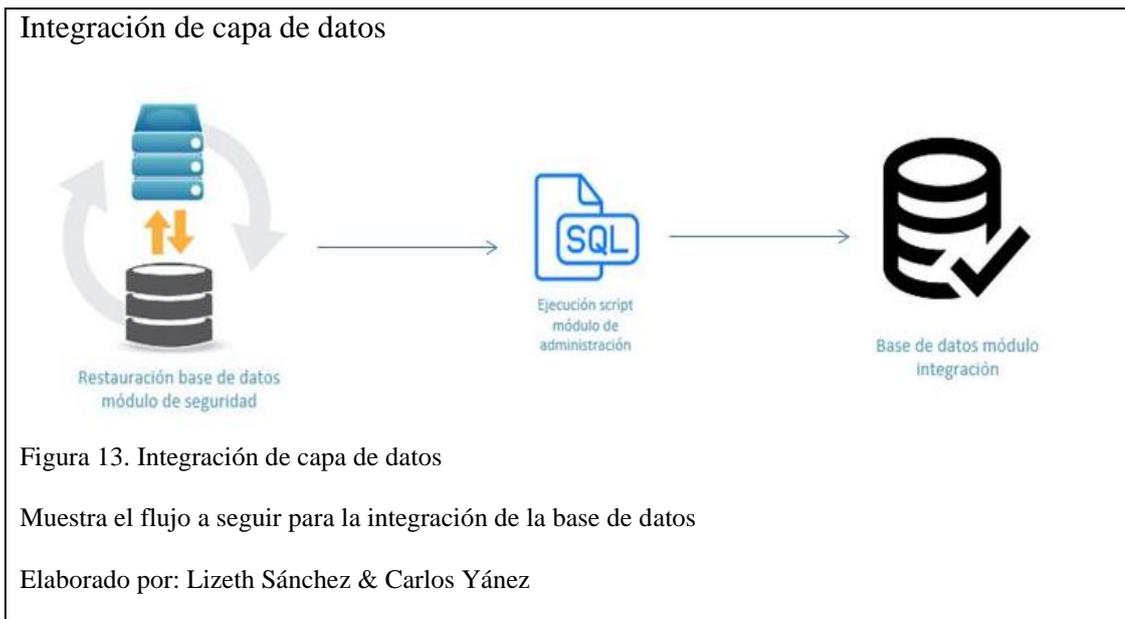
3.5.3.2 Edición Script módulo de administración.

El módulo de administración contenía modificaciones al esquema de base de datos original el cual esta implementado mediante un script que realiza todo el proceso, el mismo que se encuentra en el [Anexo 1](#). Fue necesaria la modificación del script ya

que era necesario agregar en las tablas los módulos, sub módulos y permisos necesarios para la visualización en el menú dinámico desarrollado por este módulo.

3.5.3.3 Ejecución del script.

Finalmente para obtener la unificación de la base de datos se ejecutó el script modificado anteriormente.



3.5.4 Integración a nivel funcional.

A nivel funcional la integración de los proyectos de seguridad y administración implica la normalización de procedimientos o métodos para almacenar información en la base de datos, objetos con los que el usuario interactúa (DataGrid, Combos de selección, botones, etc.), reportes y visualización (Normalización de colores, estilos, fuentes)

Estos permiten homologar la experiencia que el usuario tenga con cada proyecto integrado y crear un esquema único e intuitivo en las diferentes opciones del sistema como se muestra en la figura 14.

Integración a nivel funcional

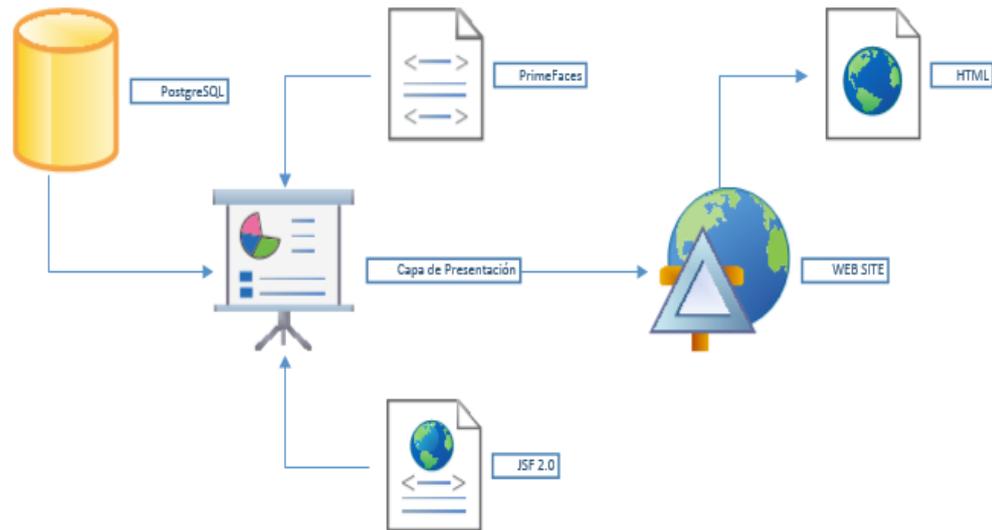


Figura 14. Integración a nivel funcional

Muestra el flujo de las funcionalidades que serán integradas en el módulo final

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Concluido el proceso de comparación y unificación de los componentes de cada uno de los sub módulos se procederá a integrar cada uno de ellos en el proyecto final logrando así la unificación de los paquetes que conforman el sistema en su totalidad. La figura 15 muestra los módulos que serán integrados en el proyecto final.

Módulos Integrados

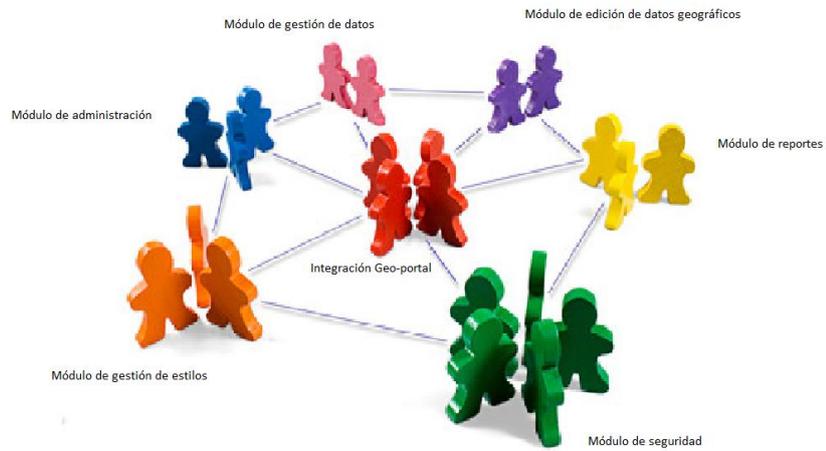


Figura 15. Módulos Integrados

Representación de los módulos integrados en el proyecto final

Elaborado por: Lizeth Sánchez & Carlos Yáñez

4. Proceso Integración

El proceso de integración inicia principalmente con la restauración de los dos últimos módulos del Geo-Portal Salesiano, ambos sistemas se encontraban desarrollados en diferente arquitectura, por lo que fue necesaria la creación de un ambiente de restauración para cada una con las siguientes especificaciones.

Las características principales del ambiente de desarrollo de cada uno de los módulos se especifican en la Tabla 16.

Tabla 16.

Diferencias arquitectura módulos seguridad y administración

Módulo Administración	Módulo Seguridad
Eclipse Kepler Release	Netbeans 7.1
Apache 2.2.24	Apache 7.5.0
PostgreSQL 9.1.9	PostgreSQL 9.1.9
PostGIS 1.5	PostGIS 1.5
Java 7	Java 7

Nota: muestra las diferencias que existen de arquitecturas entre los módulos integrados

4.1 Restauración módulo de seguridad

Para la restauración del sistema del módulo de seguridad se creó un ambiente con una arquitectura similar a la que se utilizó para el desarrollo de esta versión. Las especificaciones se describen en la Tabla 16 que fueron mencionadas anteriormente.

Durante el proceso de restauración se encontraron ciertas incidencias que se fueron resolviendo en el transcurso; una de ellas fue la referencia a librerías, para solucionar la misma fue necesaria la edición de las rutas del paquete de archivos que utilizaba el proyecto, tal como se muestra en la Figura 16.

Problema con la referencia de las librerías

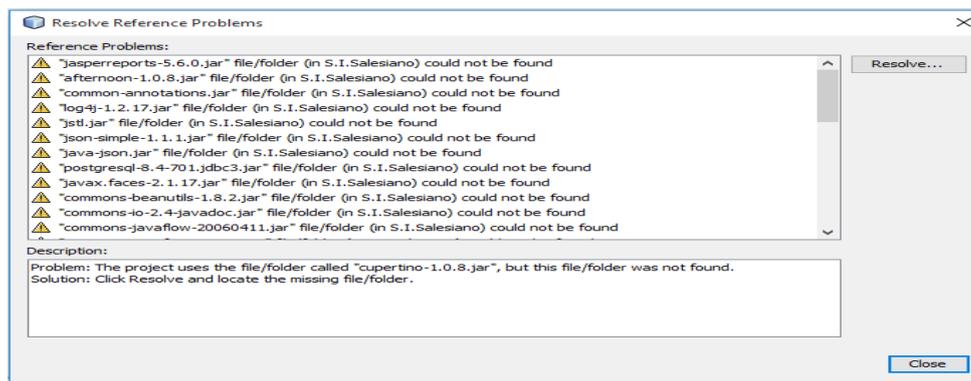


Figura 16. Problema con la referencia de las librerías

Muestra los inconvenientes que se tuvo en la referencia de librerías en la restauración del módulo de administración

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Además se configuró algunas características necesarias en el IDE Netbeans para la restauración del sistema. La Figura 17 muestra la configuración de JavaServer Faces 2.0 que es la versión utilizada para la creación del proyecto web de este módulo.

Configuración JavaServer Faces

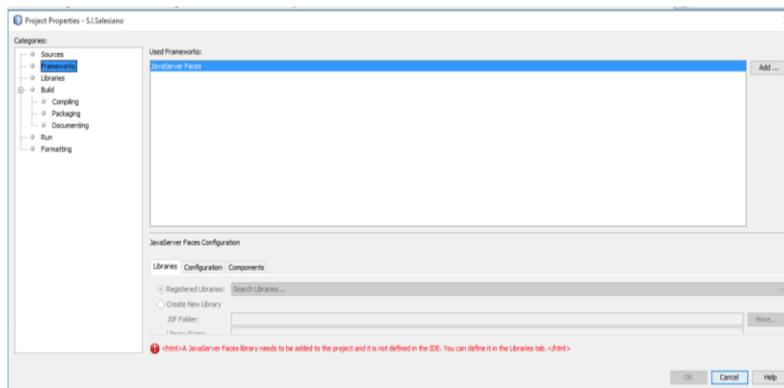


Figura 17, Configuración JavaServer Faces

Muestra la configuración que se realizó en el IDE de netbeans para el uso de java server faces

Elaborado por: Lizeth Sánchez & Carlos Yáñez

En la compilación del sistema se encontraron operaciones con problemas de funcionamiento y de versiones de librerías que no se pudieron resolver en la versión actual. La generación del reporte de auditoría fue uno de los principales problemas que se presentó en la restauración ya que la versión de la librería de jasperReport estaba descontinuada y ciertos métodos ya no existían en esta versión tal como se muestra en la Figura 18.

Problema con versiones en las librerías

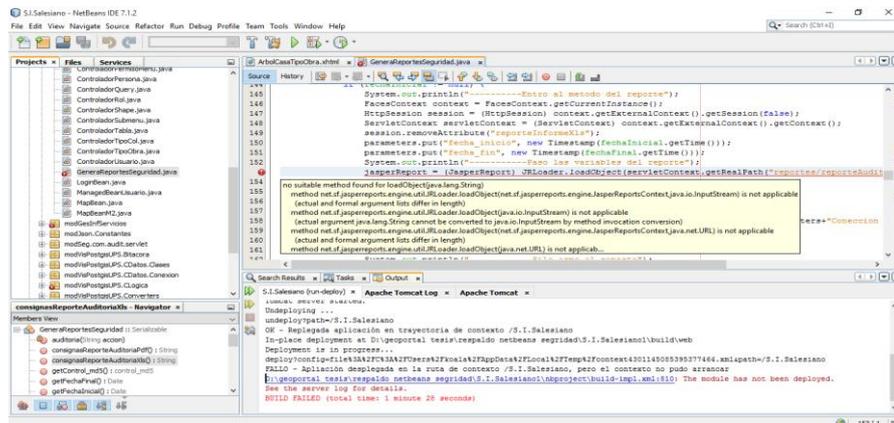


Figura 18. Problema con versiones en las librerías

Muestra los conflictos que surgieron de versiones de librerías en el proceso de restauración

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Otro de los principales problemas fue las rutas de ciertos iconos e imágenes a las que hacían referencia las paginas HTML, están no se encontraban en el proyecto tal como se muestra en la Figura 19.

Imágenes no encontradas en el proyecto



Figura 19. Imágenes no encontradas en el proyecto

Problema de referencias a imágenes que forman parte del proyecto

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Algunos métodos utilizados para direccionamientos de páginas también presentaron dificultades ocasionando problemas como el cierre de sesiones tal como se muestra en la Figura 20.



4.2 Restauración módulo administración

De igual forma que en el módulo de seguridad, fue necesaria la creación de una arquitectura similar a la que se utilizó para su implementación en el desarrollo de tesis. Estas especificaciones se describen en la Tabla 11 que fue mencionada anteriormente.

Para la restauración de este módulo existen ciertas especificaciones que se deben tomar en cuenta para el correcto funcionamiento del mismo, están se encuentran explicadas en el documento de tesis de este módulo. Entre estas se encuentra la creación de una carpeta que contiene información relevante para el funcionamiento del reporte de sistema y la generación de archivos JSON, la ruta en la que se debe crear esta información se explica en la Figura 21.

Estructura carpeta Geo-Portal

```
C:\geoportal\geoportal>tree /F
Listado de rutas de carpetas
El número de serie del volumen es 00000014 3EDC:9113
C:.
|-- imagenes
|   |-- FotosLugaresSalesianos
|   |   |-- Hydrangeas.jpg
|   |   |-- Penguins.jpg
|-- jasper
|   |-- reportCasa.jasper
|   |-- reportCasa.jrxml
|   |-- reportObra.jasper
|   |-- reportObra.jrxml
|-- pdf
|   |-- background.png
|   |-- bannerpdf.png
|   |-- logoUPS.png
|   |-- media-no-image-81e2649f06729d443e75a9ce76cf33ed.gif
|   |-- noImage.jpg
|-- tmp
|   |-- 5.jpg
|   |-- 9.JPG
C:\geoportal\geoportal>
```

Figura 21. Estructura carpeta Geo-Portal

Muestra la estructura de la carpeta de contenidos necesarios para la aplicación del sistema

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Otras de las especificaciones que describen los autores de este módulo es la ejecución de una serie de scripts en la base de datos que contienen inserciones de los nuevos módulos que fueron implementados, perfiles de usuario y ediciones de cierta información, los mismos que se encuentran en el [Anexo 1](#) .

Una vez concluida esta etapa de restauración se encontraron problemas de visualización de las pantallas debido a errores en las consultas y métodos; como se puede observar en la figura 22 el usuario logeado se mostraba como nulo; estos problemas se resolvieron en la versión integrada.

Usuario nulo al ingresar al sistema



Figura 22. Usuario nulo al ingresar al sistema

Se observa un problema al obtener el usuario de la sesión del sistema

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Otro problema de visualización que se encontró en este módulo fue la falta de estilos en ciertas pantallas como las tablas de edición y consultas en el módulo de beneficiarios tal como se muestra en la Figura 23; estos estilos fueron mejorados en la versión integrada.

Visualización formularios



Figura 23. Visualización formularios

Visualiza la alteración del estilo de las tablas

Elaborado por: Lizeth Sánchez & Carlos Yáñez

el diseño conceptual muestra la representación de las clases con las relaciones tal como se puede observar en la figura 25.

Diseño Conceptual módulo de Administración de seguridad

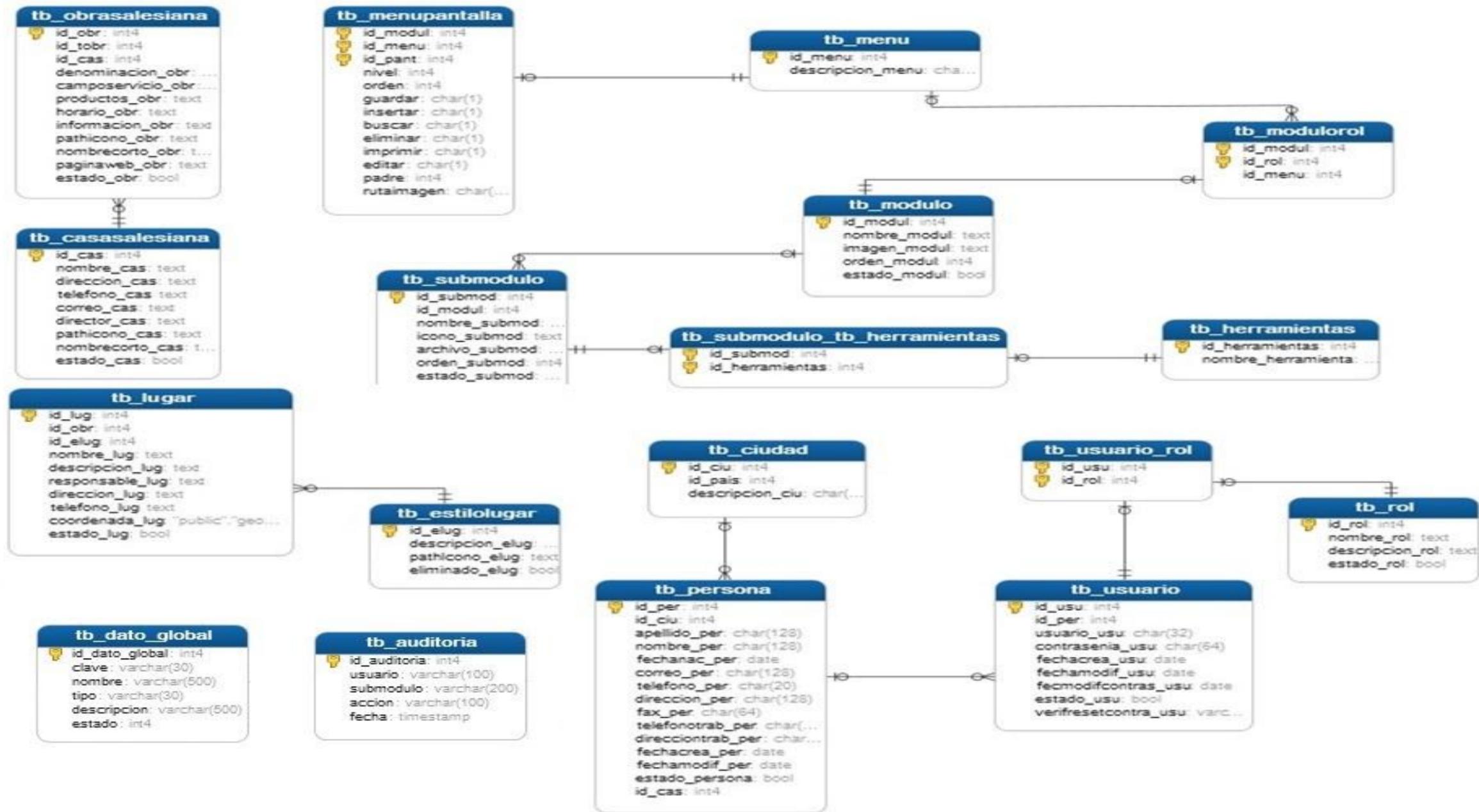


Figura 25. Diseño Conceptual módulo de Administración de seguridad

Muestra el diseñador conceptual de la base de datos del geo portal salesiano

Elaborado por: Carina Torres y Darwin Aldas

El siguiente modelo de la base de datos permite visualizar la estructura con la que se inició el proceso de integración y lo que se incrementó para el desarrollo de la versión del módulo de administración del Geo-Portal Salesiano.

En la Figura 26 se especifica el modelo conceptual utilizado para el desarrollo del módulo de seguridad; al cual se ha integrado las nuevas tablas necesarias para el correcto funcionamiento del módulo de administración, las mismas están diferenciadas en colores.

Modelo conceptual de base de datos

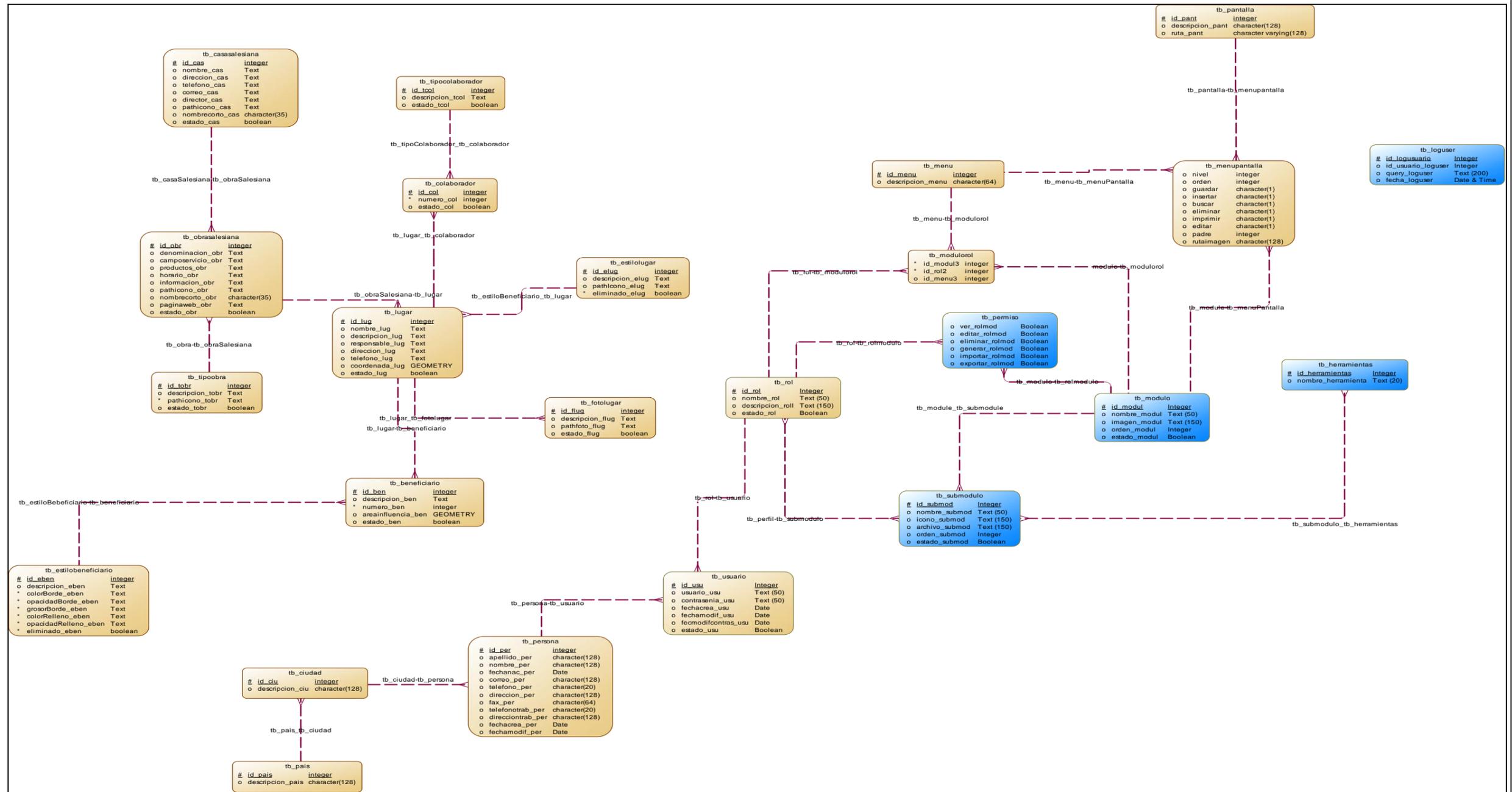


Figura 26. Modelo conceptual de base de datos

Muestra la estructura de la base de datos

Elaborado por: Sandoval Byron & Tutillo

4.5 Unificación de paquetes

En esta fase se pretende organizar cada una de las clases, servicios, controladores, páginas HTML, entre otros, que forman parte de los módulos.

El módulo de administración se conservó para proceder a la integración del sistema así que se adaptó la estructura de paquetes que se utiliza en esta versión para proceder a la copia y comparación de código fuente de las clases involucradas en los procesos del sistema.

4.5.1 Paquete Servicios

El paquete de servicios contiene los procesos necesarios para la ejecución de queries de consulta, actualización, edición y eliminación de registros y las configuraciones necesarias para establecer la conexión con la base de datos del sistema.

Para obtener la unificación del paquete de servicios se inició copiando las clases que no tenían en común los módulos necesarios para las implementaciones del módulo de seguridad.

En la figura 27 se muestra resaltados con círculos rojos las clases servicios faltantes que fueron implementadas en el módulo de integración.

Identificación servicios faltantes

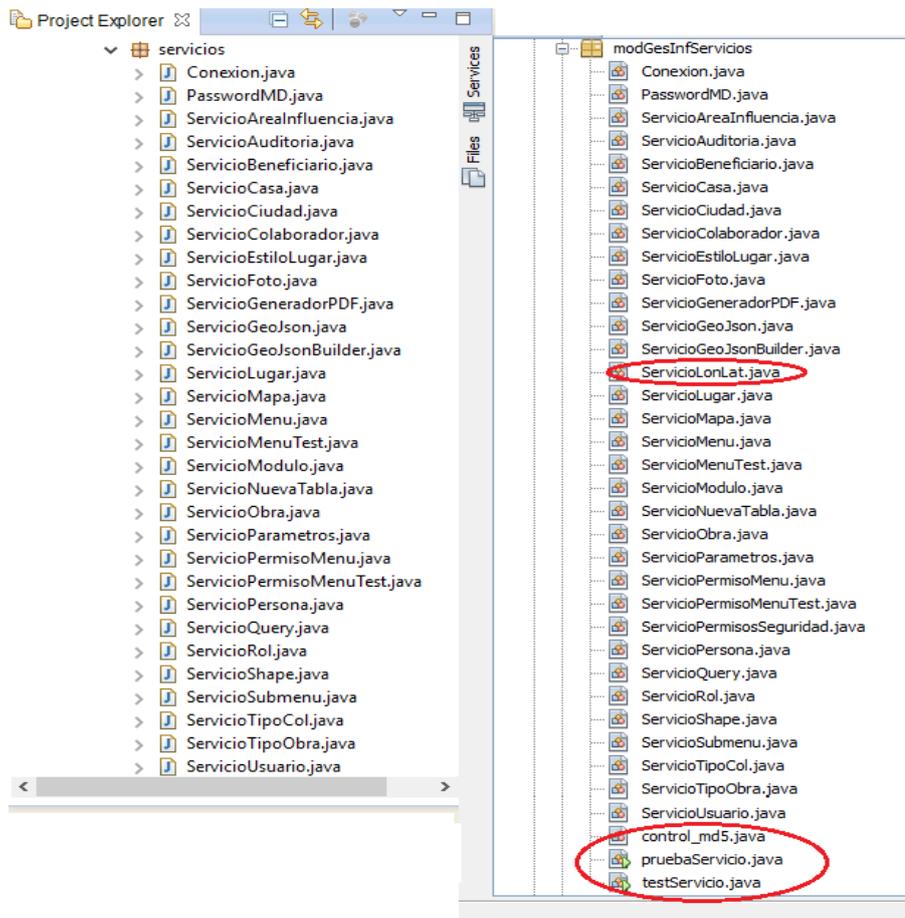


Figura 27. Identificación servicios faltantes

Muestra la comparación visual de los servicios faltantes en el módulo

Elaborado por: Carlos Yáñez & Lizeth Sánchez

La tabla 17 describe los servicios faltantes del módulo de edición de datos geográficos, con sus respectivas funcionalidades, los mismos que fueron integrados en el proyecto final.

Tabla 17.

Descripción servicios implementados

Clase	Descripción
ServicioLonLat.java	Implementa métodos para la obtención, edición y consulta de información o datos geográficos utilizados en el sub modulo generador área de influencia.
Control_md5.java	Implementa el método de cifrado o encriptación utilizados en la acceso al sistema.

Nota: describe la clase servicio que se implementaron el módulo integrado

Para la integración de las clases de servicio comunes entre los módulos se realizó una comparación de cada uno de ellos con ayuda de la herramienta WinMerge para encontrar las diferencias en el código fuente y proceder a su reemplazo o agregación con la versión más actual.

En la figura 28 se puede observar la implementación de permisos mediante la clase LoginBean que es uno de los procesos realizados por el módulo de administración.

Comparación código fuente Servicio Beneficiario

The image shows a WinMerge window comparing two versions of a Java source file. The left pane shows the original code, and the right pane shows the modified code. The differences are highlighted in yellow. The code defines a class with static fields and a static method 'cargarDatos()' that interacts with a database via JDBC. The changes include the addition of a 'loginBean' field and the replacement of 'ArrayList' with 'LinkedList' for the 'datos' collection.

```
private static List<Beneficiario> datos;
private static List<Beneficiario> beneficiarios;
private static List<Permiso> permisos;
private static Permiso permiso;
// ManagedProperties injection from another Bean
@ManagedProperty(value = "#{loginBean}")
private LoginBean loginBean;

private static void cargarDatos() {
    datos = new ArrayList<Beneficiario>();
    LOG.info("ENTRO AL CARGADATOS");
    try {
        ResultSet rs;
        rs = ConsultarTodo();
        ResultSetMetaData rsm = rs.getMe
        int columnIndex = rsm.getColu
        System.out.println("*****");
        while (rs.next()) {
            datos.add(new Beneficiario(rs.getInt(1),
            rs.getInt(2), rs.getBoolean(7),
            //
            datos.add(new Beneficiario
            rs.getInt(2), rs.getBoolean(7)));
        }
        rs.close();
    }
    catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

private static void cargarDatos() {
    datos = new LinkedList<Beneficiario>();
    LOG.info("ENTRO AL CARGADATOS");
    try {
        ResultSet rs;
        rs = ConsultarTodo();
        while (rs.next()) {
            datos.add(new Beneficiario(rs.getInt(1),
            rs.getInt(2), rs.getBoolean(7)));
        }
        rs.close();
    }
    catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}
```

Figura 28. Comparación código fuente Servicio Beneficiario

Muestra las diferencias encontradas en una de las clases de servicio del sistema.

Elaborado por: Carlos Yáñez & Lizeth Sánchez

En el módulo de seguridad se desarrolló la auditoria del sistema tanto para métodos de actualización, inserción y eliminación de información, la misma que se agregó en los servicios tal como se muestra en la figura 29.

Implementación auditoria de sistema

```
lLat.java control_md5.java ServicioArealInfluencia.java ServicioAuditoria.java ServicioBeneficiario.java ✕  
  
public static void insertar(Beneficiario datosb, Integer usuarioSession) throws Exception {  
    cargarDatos();  
    if ((buscarBen(datosb.getIdben()) == null)) {  
        String sentencia = "insert into tb_beneficiario (descripcion_ben,numero_ben,id_lug,id_eben,estado_ben) values ("  
            + datosb.getDescben()  
            + ","  
            + Integer.parseInt(datosb.getNumben())  
            + ","  
            + datosb.getIdlug()  
            + ",1,true)";  
        Conexion con = new Conexion();  
        con.Ejecutar(sentencia);  
        ServicioAuditoria.guardarAuditoria(usuarioSession, SubmoduloEnum.SUBMODULO_BENEFICIARIO.getValue(),  
            AccionModuloEnum.INSERTAR_DATO.getValue());  
    } else {  
        throw new Exception("El codigo del beneficiario ya existe");  
    }  
}
```

Figura 29. Implementación auditoria de sistema

Muestra la implementación del método para guardar la auditoria del sistema

Elaborado por: Carlos Yáñez & Lizeth Sánchez

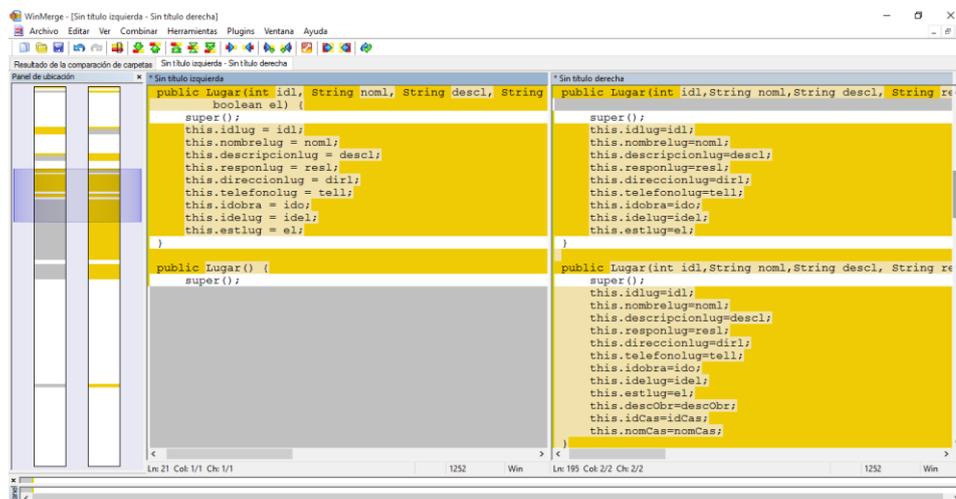
4.5.2 Paquete clases.

El paquete de clases especifica las clases POJO que hacen referencia a cada una de las tablas de la base de datos.

Este paquete en ambos módulos contenía el mismo número de clases así que se procedió a la comparación del código fuente de cada una de ellas.

En la figura 30 muestra la comparación del código fuente de la clase lugar; existe un nuevo constructor desarrollado en el módulo de administración para mejorar el proceso de consulta de información mostrada en cada uno de los sub módulos.

Comparación código fuente Clase Lugar



The image shows a WinMerge window comparing two versions of the 'Lugar' class source code. The left pane shows the original code, and the right pane shows a modified version. The modified code includes additional attributes like 'descObr', 'idCas', and 'nomCas'.

```
public Lugar(int idl, String noml, String descl, String re...
boolean el) {
    super();
    this.idlug = idl;
    this.nombrelug = noml;
    this.descripcionlug = descl;
    this.responlug = resl;
    this.direccionlug = dir1;
    this.telefonolug = tell;
    this.idobra = ido;
    this.idelug = idel;
    this.estlug = el;
}

public Lugar() {
    super();
}

public Lugar(int idl, String noml, String descl, String re...
    super();
    this.idlug=idl;
    this.nombrelug=noml;
    this.descripcionlug=descl;
    this.responlug=resl;
    this.direccionlug=dir1;
    this.telefonolug=tell;
    this.idobra=ido;
    this.idelug=idel;
    this.estlug=el;
    this.descObr=descObr;
    this.idCas=idCas;
    this.nomCas=nomCas;
}
```

Figura 30. Comparación código fuente Clase Lugar

Muestra la comparación de código fuente de una de las clases del sistema

Elaborado por: Carlos Yáñez & Lizeth Sánchez

4.5.3 Comparación controladores.

El paquete de controladores contiene los métodos necesarios para la administración de cada módulo; la lógica de negocio.

El módulo de seguridad tiene implementados controladores necesarios para el manejo de seguridades y creación de reportes; se procedió a la copia de estos controladores al módulo integrado.

La Figura 31 muestra los controles que fueron implementados para el manejo de seguridades en el módulo de integración; los mismos se encuentran resaltados por círculos de color rojo.

Identificación controladores faltantes

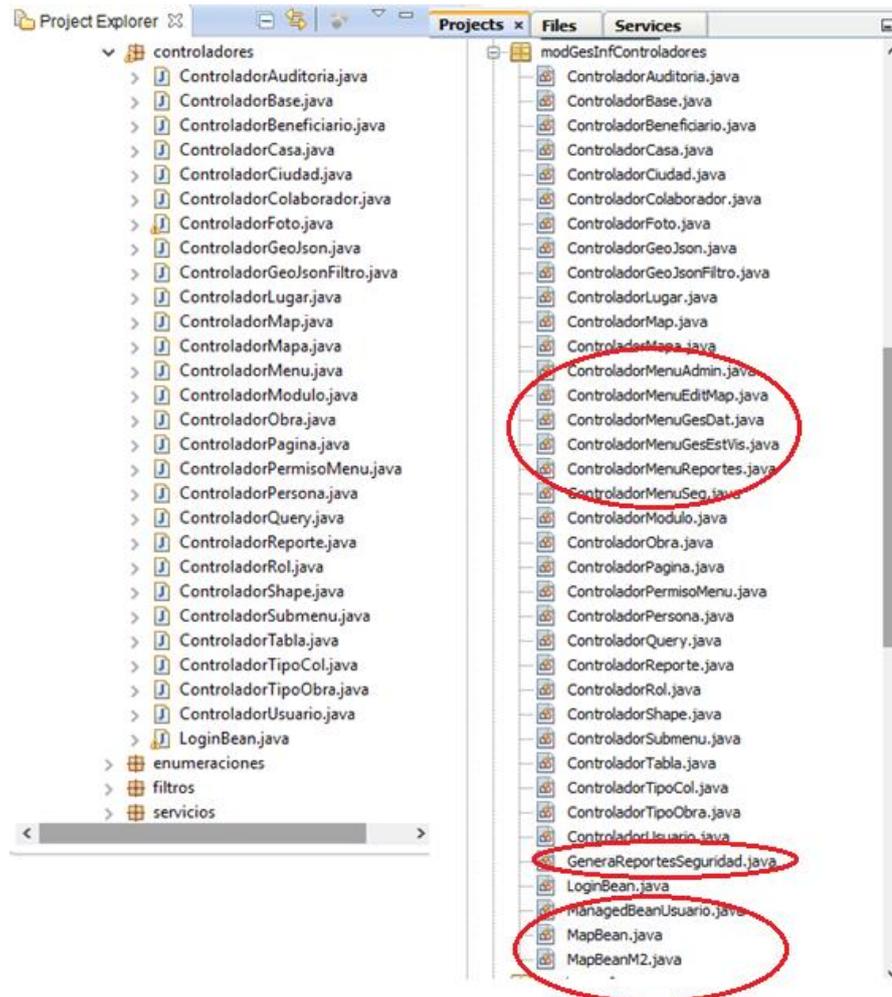


Figura 31. Identificación controladores faltantes

Muestra la diferencia visual de clases no existentes en el módulo integrado

Elaborado por: Carlos Yáñez & Lizeth Sánchez

En la tabla 18 se describe los controladores faltantes en el módulo de administración que fueron agregados en el proyecto final.

Tabla 18.

Descripción controladores implementados

Clase	Descripción
ControladorMenuAdmin.java	Construye el menú de administración de acuerdo a los permisos del usuario generados en el control LoginBean.java.
ControladorMenuEditMap.java	Construye el menú de edición de mapas de acuerdo a los permisos del usuario generados en el control LoginBean.java.
ControladorMenuGesDat.java	Construye el menú de gestión de datos de acuerdo a los permisos del usuario generados en el control LoginBean.java.
ControladorMenuGesEstVis.java	Construye el menú de gestión de estilos visualizador de acuerdo a los permisos del usuario generados en el control LoginBean.java
ControladorMenuReportes.java	Construye el menú de reportes de acuerdo a los permisos del usuario generados en el control LoginBean.java
ControladorMenuSeg.java	Construye el menú de seguridad de acuerdo a los permisos del usuario generados en el control LoginBean.java
GeneraReportesSeguridad.java	Contiene los métodos necesarios para la generación del reporte de auditoria.
ManageBeanUsuario.java	Contiene los atributos para la creación

	de permisos.
MapBean.java	Administración de los datos geográficos para la visualización del mapa.

Nota: describe los controladores que fueron incluidos en el módulo integrado

Los controladores comunes de ambos módulos tuvieron un mayor grado de complejidad en la comparación del código fuente debido a la reutilización de los mismos métodos y la adaptación de los mismos para las necesidades de cada módulo.

En la figura 32 nos muestra la implementación de permisos y roles desarrollada por el módulo de seguridad.

Código fuente controlador Submenu

```

@ViewScoped
public class ControladorSubmenu extends ControladorBase {
    private static final long serialVersionUID = 88809450;
    private final static Logger LOG = Logger.getLogger(ControladorSubmenu.class);

    public List<Submenu> submenus;
    public Submenu submenu;
    private ModeTypeEnum mode;

    public List<Modulo> moduloList;

    @PostConstruct
    public void init() {
        try {
            mode = ModeTypeEnum.CREAR;
            submenus = ServicioSubmenu.recuperarTodos();
            moduloList = ServicioModulo.recuperarTodos();
            submenu = new Submenu();
        } catch (Exception e) {
            FacesMessageUtil.addErrorMessage(Mensaje.ERROR);
        }
    }
}

```

```

public class ControladorSubmenu extends ControladorBase {
    private static final long serialVersionUID = 88809450;
    private final static Logger LOG = Logger.getLogger(ControladorSubmenu.class);

    public List<Submenu> submenus;
    public List<Rol> roles;
    public List<String> rolesSeleccionados;
    public Submenu submenu;
    public List<Modulo> moduloList;
    @ManagedProperty(value = "#{loginBean}")
    private LoginBean loginBean;
    private boolean accedePath;
    private Date fechaReg;
    private control_md5 control_md5;
    private ModeTypeEnum mode;

    @PostConstruct
    public void init() {
        try {
            mode = ModeTypeEnum.CREAR;
            submenus = ServicioSubmenu.recuperarTodos();
            moduloList = ServicioModulo.recuperarTodos();
            roles = ServicioRol.recuperarTodos();
            submenu = new Submenu();
        } catch (Exception e) {
            FacesMessageUtil.addErrorMessage(Mensaje.ERROR);
        }
    }
}

```

Figura 32. Código fuente controlador Submenu

Comparación visual código fuente controlar implementado en el módulo integrado

Elaborado por: Carlos Yáñez & Lizeth Sánchez

Para la implementación de los permisos se agregaron instancias de la clase LoginBean.java en todos los controladores de este paquete que permiten obtener la

información del inicio de sesión del usuario, utilizando notaciones JSF como se describe el figura 33.

Instancia de LoginBean para controladores.

```
ladorC...  Controlador...  Controlador...  Controlador...  ControladorL...  »17
// ManagedProperties injection from another Bean
@ManagedProperty(value = "#{loginBean}")
private LoginBean loginBean;
private boolean accedePath;
private Date fechaReg;
```

Figura 33. Instancia de LoginBean para controladores.

Implementación clase loginBean en los controladores del sistema

Elaborado por: Carlos Yáñez & Lizeth Sánchez

El proceso de actualización, eliminación e inserción de información está ligado a una validación que es ejecutada por los controladores y servicios del módulo de seguridad que permiten que el usuario registrado pueda modificar la base de datos, siempre y cuando tenga el perfil indicado para esas tareas, la figura 34 muestra el fragmento de código implementado.

Validación de permisos.

```
public void validaEliminaRegistros() {
    try {
        String pathPaginasSeg = "/gestionDatos/DatosLugar.jsf";
        control_md5 = new control_md5();
        boolean resultadoValor = ServicioPermisosSeguridad.consultarPermisoEliminar(loginBean.getNombre(),
            getControl_md5().md5(loginBean.getClave()), pathPaginasSeg);
        if (resultadoValor == true) {
            elimina();
        } else {
            FacesContext.getCurrentInstance().addMessage(
                null,
                new FacesMessage(FacesMessage.SEVERITY_ERROR,
                    "Permiso Denegado", ""));
        }
    } catch (Exception e) {
    }
}
```

Figura 34. Validación de permisos

Implementación método para manejo del permisos de sistema

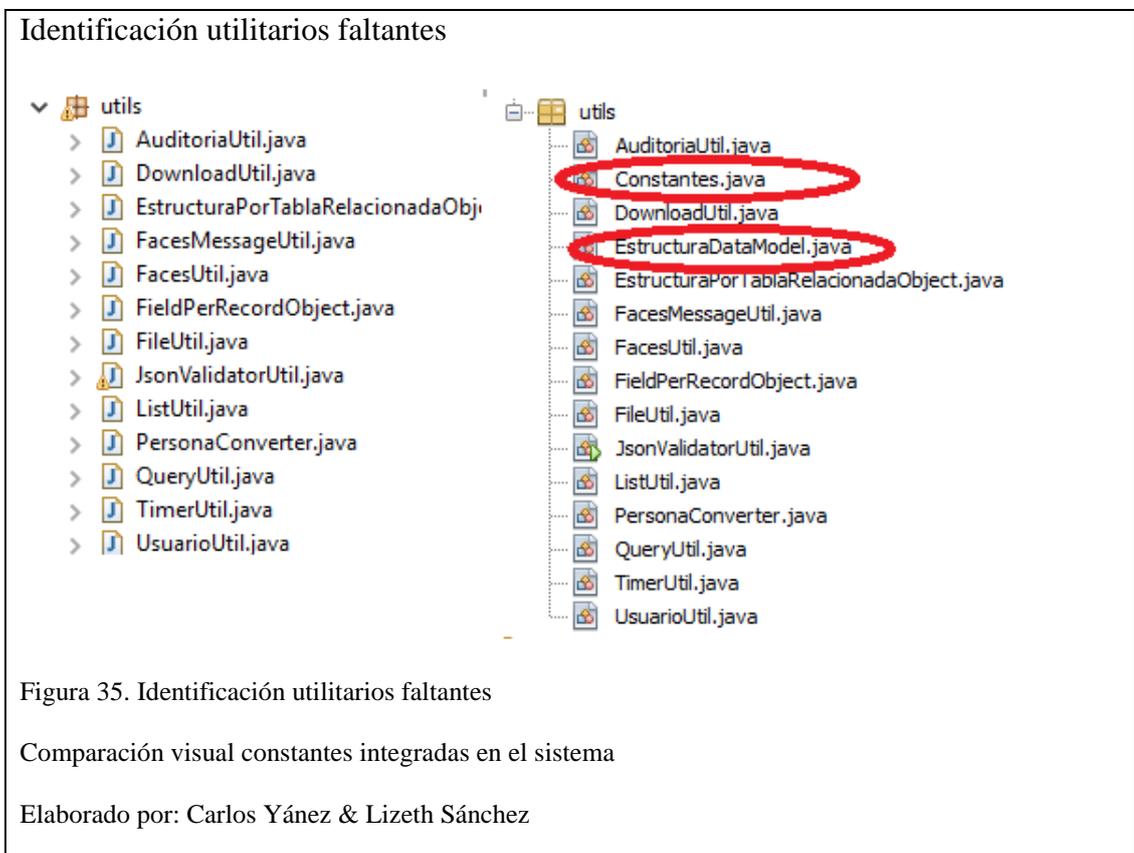
Elaborado por: Carlos Yáñez & Lizeth Sánchez

4.5.4 Paquete Utils.

En este paquete se especifica métodos y procesos en común que son utilizados por varios sub módulos.

La identificación de las clases utilitarios que no tenían en común ambos módulos se realizó de forma visual para proceder a copiarlos en la versión integrada.

Las clases faltantes en el paquete de utils se muestran en la Figura 35 resaltados por círculos de color rojo.



En la tabla 19 se describe las clases utilitario que se agregaron en el proyecto final de la integración.

Tabla 19.

Descripción clases utilitarios implementados

Clase	Descripción
Constantes.java	Almacena valores constantes que se utilizan en todo el proyecto.
EstructuraDataModel.java	Métodos necesarios para el manejo de objetos.

Nota: describe la clase utilitario que fueron implementadas en el módulo integrado

En la figura 36 se muestra la comparación de código fuente del utilitario Query; existe una mejora del método **crearStringCamposParaSelectGeojson** necesario para la creación de la cadena de consulta formada para la generación de JSON en el módulo de administracion.

Comparación código fuente QueryUtil

Figure 36 shows a side-by-side comparison of the `crearStringCamposParaSelectGeojson` method in a code editor. The left pane shows the original code, and the right pane shows the updated code. The updated code includes additional logic to handle different data types (e.g., `getTipoDatoReal()`) and to format the query string correctly for JSON generation.

Figura 36. Comparación código fuente QueryUtil

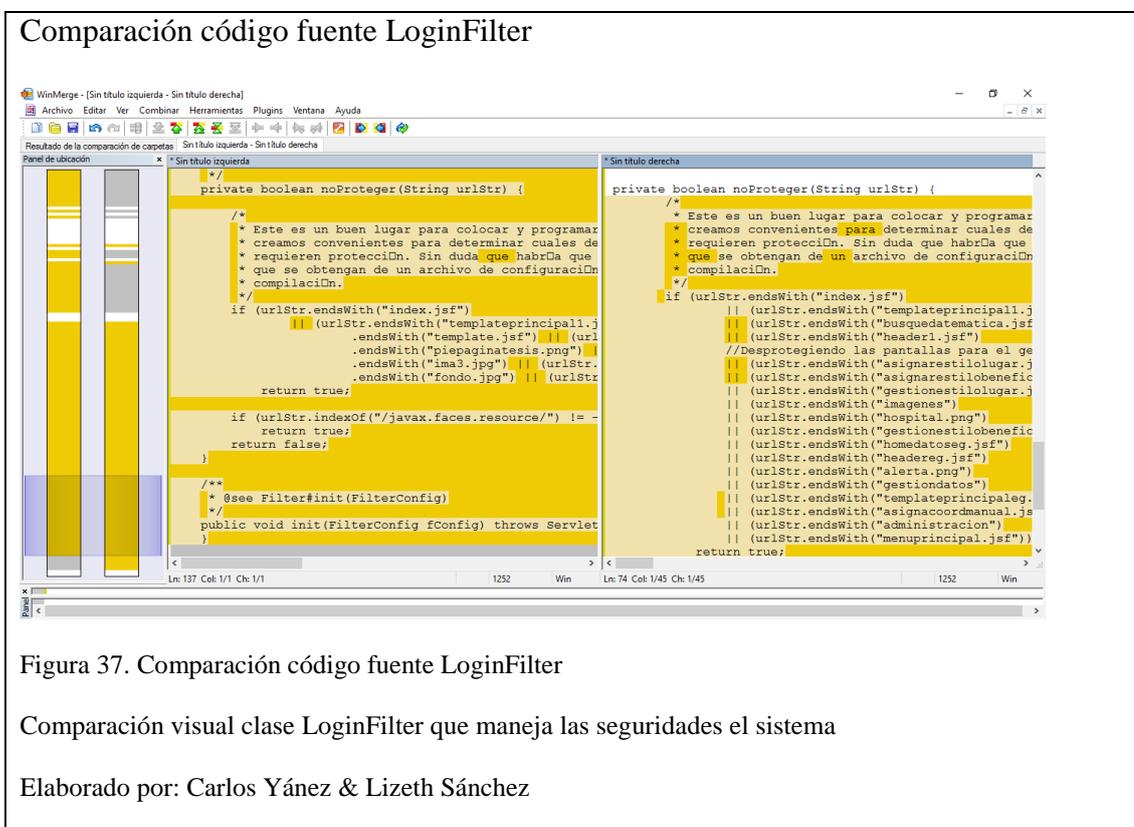
Comparación visual clases utilitarios integradas en el sistema

Elaborado por: Carlos Yáñez & Lizeth Sánchez

4.5.5 Paquete filtros.

En este paquete solo existe la clase LoginFilter para ambas versiones así que se procedió a la comparación de código fuente para actualizar sus métodos con las versiones más recientes.

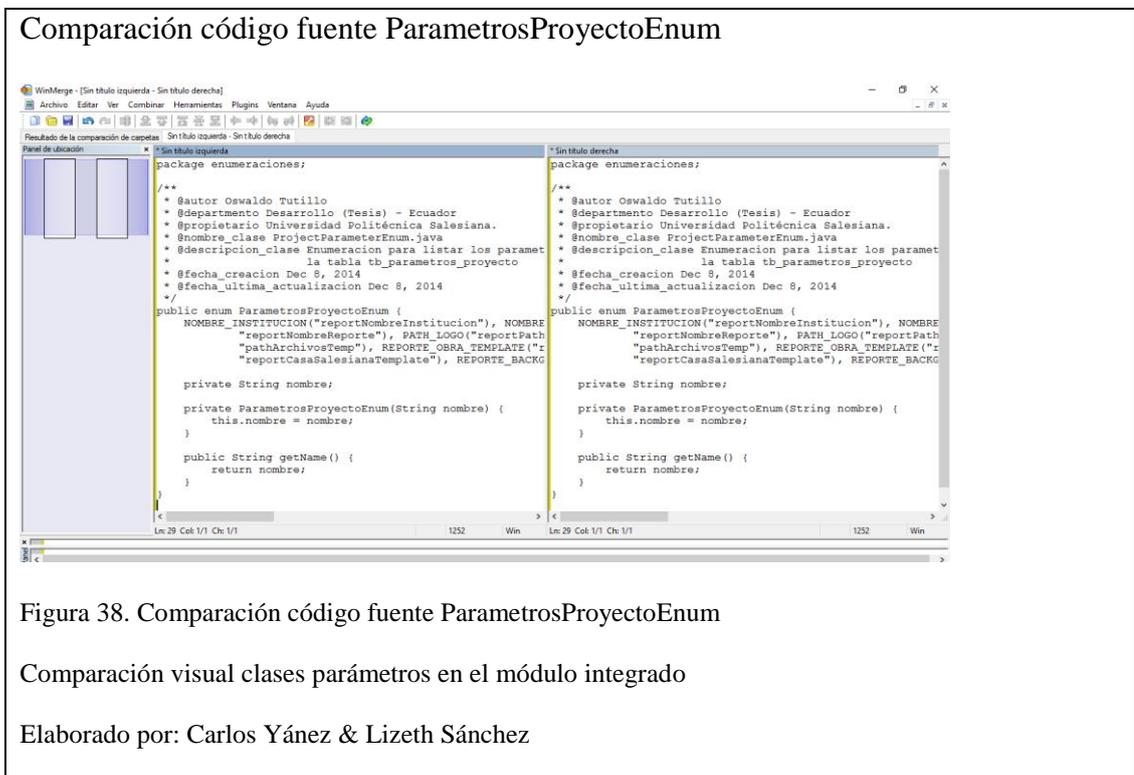
En la figura 37 se muestra las mejoras realizadas para la clase LoginFilter necesaria para el control de acceso a los recursos del sistema dependiendo de perfiles de usuario.



4.5.6 Paquete Enumeraciones.

En este paquete no fue necesario realizar ningún cambio ya que existe en ambos módulos las mismas clases y la misma versión de cada uno de sus métodos.

En la figura 38 se muestra la comparación realizada de la clase ParametrosProyectoEnum con la herramienta WinMerge; en la misma se observa que no se ha identificado diferencias.

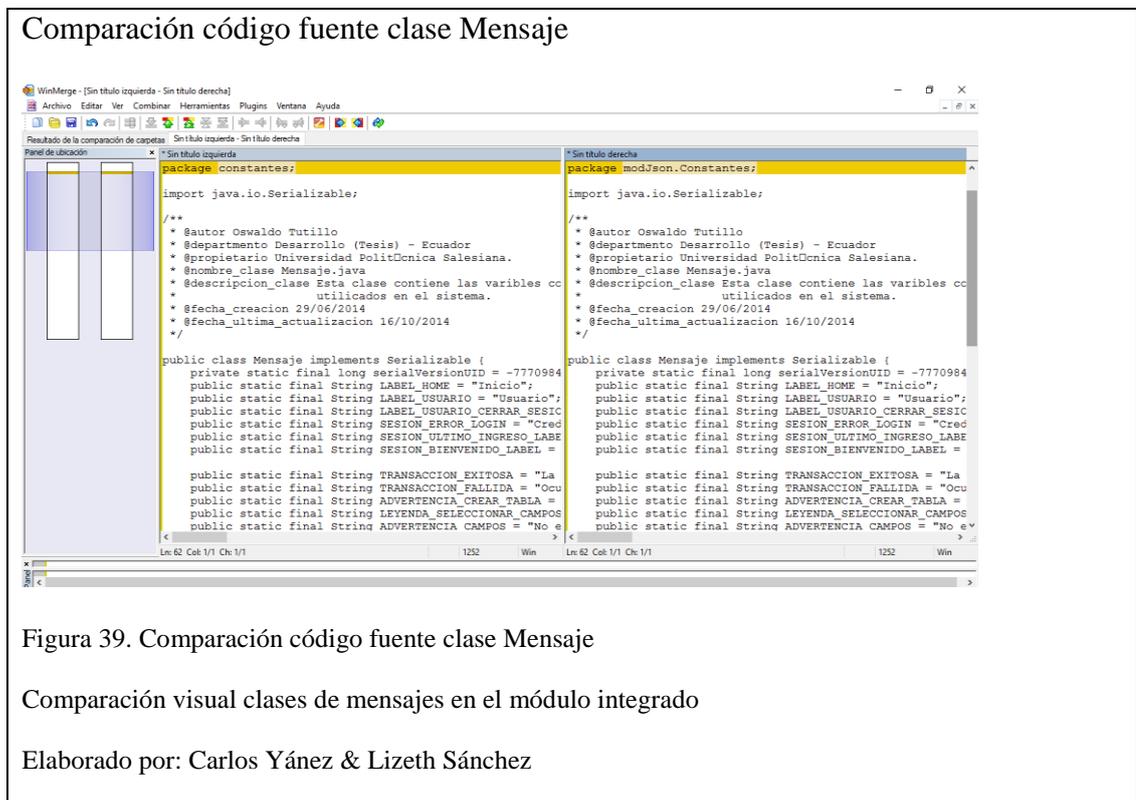


4.5.7 Paquete Constantes.

Este paquete almacena todas las constantes que se han utilizado en el desarrollo tanto del módulo de seguridad como en el módulo de administración para evitar la quema de código.

Al igual que el paquete Enumeraciones no fue necesario realizar ningún cambio en el mismo ya que ambas versiones no existen diferencias ni en sus clases ni en sus métodos.

En la figura 39 se muestra la comparación realizada de la clase Mensaje con la herramienta WinMerge; en la misma se observa que no se ha identificado diferencias.



Para adaptar la estructura de paquetes usada en ambos módulos se procedió a copiar paquetes faltantes que se encontraban en el módulo de seguridad necesarios para procesos que ejecuta el módulo como la generación del reporte de sistema.

En la Figura 40 se resaltan los paquetes del módulo de seguridad que fueron implementados en la integración para el correcto funcionamiento de la generación del reporte de sistema

Comparación de paquetes

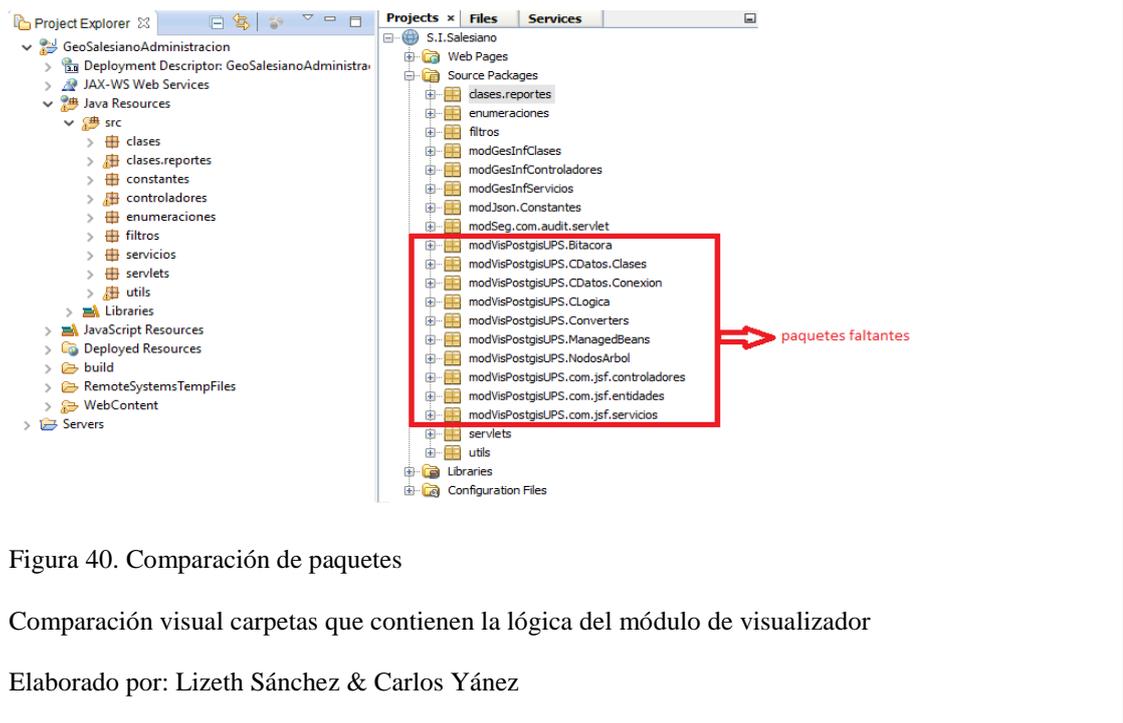


Figura 40. Comparación de paquetes

Comparación visual carpetas que contienen la lógica del módulo de visualizador

Elaborado por: Lizeth Sánchez & Carlos Yáñez

4.6 Comparación contenido web

4.6.1 Módulo Administración.

Para la implementación del módulo de integración se reutilizó la funcionalidad ya existente del módulo de administración que fue el encargado de la implementación y actualización de los sub módulos que lo conforman y se precedió a copiar las páginas que existían en el módulo de seguridad que no estaban en ambas versiones.

Las páginas de los sub módulos faltantes que fueron copiados del módulo de seguridad se describen en la Tabla 20.

Tabla 20.

Descripción Sub módulos administración

Página XHtml	Descripción
EdicionRol.xhtml	Muestra la pantalla de administración de roles utilizados en el sistema.
EdicionPersona.xhtml	Muestra la pantalla de administración de personas utilizados en el sistema.
EdicionUsuario.xhtml	Muestra la pantalla de administración de usuarios utilizados en el sistema.
Auditoria.xhtml	Muestra la pantalla para el control de accesos y transacciones al sistema.

Nota: describe las páginas web que fueron implementadas del módulo de administración.

En la Figura 41 se muestra diferenciadas por círculos rojos las páginas de los sub módulos que fueron identificadas que no existen en ambas versiones y que se procedieron a implementar en el proyecto integrado.

Sub módulos faltantes Módulo Administración

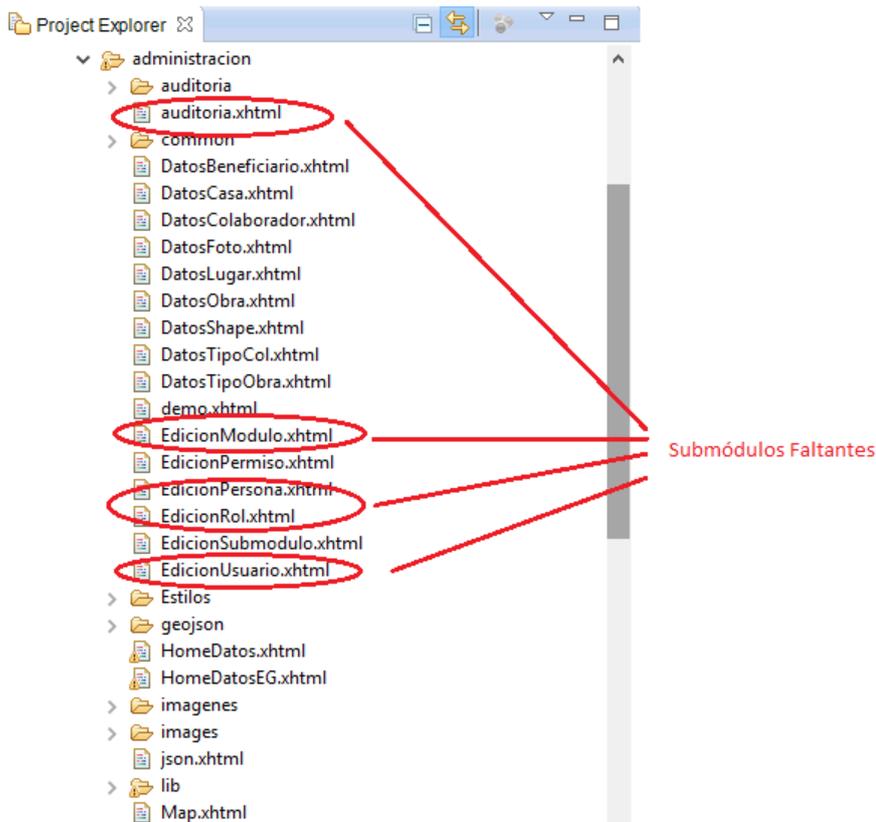


Figura 41. Submódulos faltantes Módulo Administración

Muestra de forma visual las páginas html implementadas en el módulo integrado

Elaborado por: Carlos Yáñez & Lizeth Sánchez

Con las páginas en común de ambos módulos se procedió hacer la comparación del código fuente; se encontró solo diferencias de estilos y filtros que se agregaron para una mejor visualización de las opciones.

En la Figura 42 se muestra ciertas correcciones de estilos que se debió realizar en la página web EdicionSubmodul.xhtml para mejorar la visualización de las tablas de consulta y edición.

Comparación EdicionSubmodul.xhtml

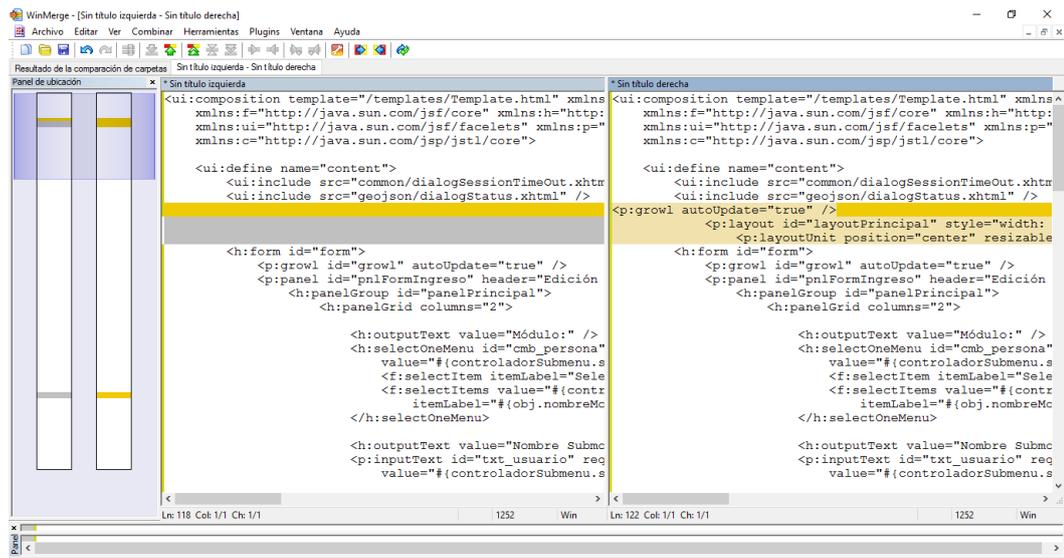


Figura 42. Comparación EdicionSubmodul.xhtml

Comparación visual páginas html módulo de administración

Elaborado por: Carlos Yáñez & Lizeth Sánchez

Una vez finalizado el proceso de comparación del módulo de Administración se procedió a realizar pruebas respectivas para verificar el correcto funcionamiento en cada uno de los procesos.

4.6.2 Módulo Gestión Datos.

Para la integración del módulo se identificó las páginas de los sub módulos que no tenían en común ambas versiones. Se procedió a copiar las páginas del módulo de seguridad para implementar funcionalidades que se actualizaron en el mismo.

En la figura 43 se muestra resaltadas las páginas de los sub módulos que fueron necesarios implementar en la versión integrada.

Sub módulos a implementar.

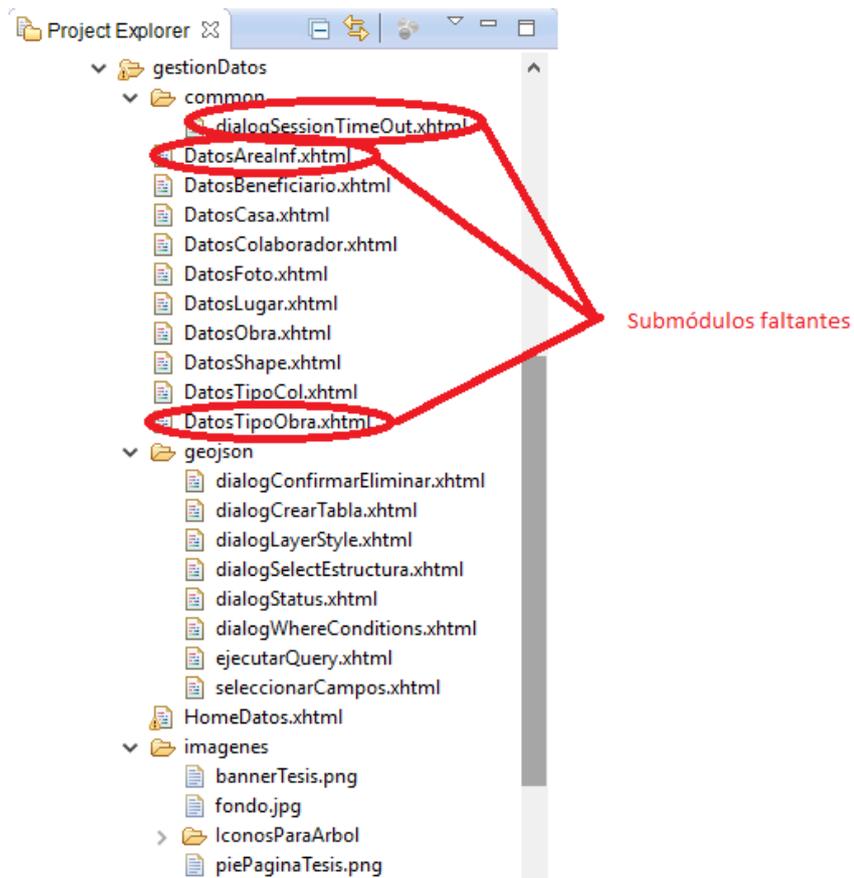


Figura 43. Sub módulos a implementar

Visualización páginas html a implementar en el módulo de gestión de datos

Elaborado por: Carlos Yáñez & Lizeth Sánchez

Los sub módulos faltantes en gestión de datos que fueron agregados en el proyecto de integración se describen en la tabla 21

Tabla 21.

Descripción Sub módulos gestión datos

Página XHTML	Descripción
DialogoSessionTimeOut.xhtml	Contiene el tiempo de duración de la sesión de usuario registrado en el sistema.
DatosAreaInf.xhtml	Contiene la pantalla del sub modulo área de influencia
DatosTipoObra.xhtml	Contiene la pantalla del sub modulo área de Tipo de Obra.

Nota: Descripción de las páginas html implementadas en el módulo de gestión de datos

A realizar la comparación del código fuente no se encontraron diferencias. Al iniciar con las respectivas pruebas de funcionamiento se encontró algunos errores en los queries de consulta que se realizaba para la carga de cierta información; estos fueron corregidos en la versión integrada para su correcto funcionamiento.

En la figura 44 se muestra la corrección de las relaciones entre tablas en la consulta de datos de la obra salesiana.

Corrección relaciones de obra Salesiana

```

public static ResultSet ConsultarTodo() throws Exception {
    ResultSet rs = null;
    String sentencia = "select obr. *, ca.nombre_cas, tobr.descripcion_tobr from tb_obraSalesiana obr ,tb_casasalesiana ca,"
    + " tb_tipoobra tobr "
    + " where estado_obr=true "
    + " and obr.id_cas = ca.id_cas "
    + " and obr.id_tobr = tobr.id_tobr ";
    try {
        Conexion clsCon = new Conexion();
        rs = clsCon.Consulta(sentencia);
    } catch (Exception e) {
        throw new Exception("..ERROR EN ConsultarTodo" + e);
    }
    return rs;
}
    
```

Figura 44. Corrección relaciones de obra Salesiana

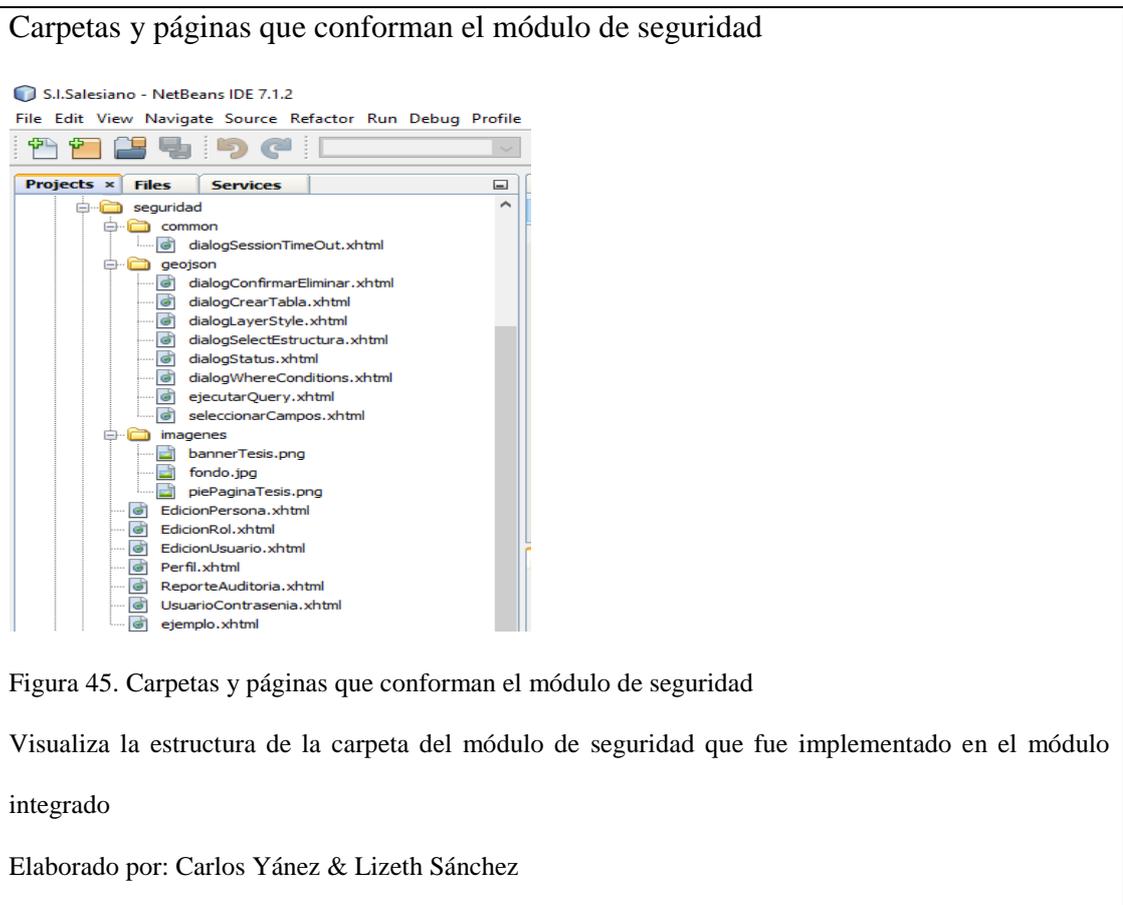
Visualización consulta de obras y casas corregida por problemas en la visualización de datos

Elaborado por: Carlos Yáñez & Lizeth Sánchez

4.6.3 Módulo Seguridad.

Este módulo se lo implemento por completo en la versión integrada ya que solo existía en una de las versiones de tesis, así que se procedió a copiar cada una de las páginas y carpetas que lo conformaban.

En la figura 45 se muestra la estructura del módulo de seguridad que fue implementado en la versión integrada.



Las opciones correspondientes al módulo de seguridad que fueron implementadas en el proyecto final se describen en la tabla 22.

Tabla 22.

Descripción Sub módulos seguridad

Sub modulo	Descripción
Perfil	Permite la creación, edición y eliminación de los permisos de usuario.
Cambio de Contraseña	Permite la actualización de la contraseña del usuario registrado en el sistema.

Nota: describe el contenido de las páginas HTML implementadas en el módulo de seguridad

4.6.4 Módulo Reportes.

El reporte de Auditoria se desarrolló en el módulo de seguridad así que fue implementado por completo en la versión integrada. Se procedió a la copia de páginas y carpetas referentes al módulo.

Para concluir el proceso de integración del módulo se realizó las pruebas respectivas de los procesos y su correcto funcionamiento. El módulo de seguridad integrado con los sub módulos de cada una de las versiones se muestra en la figura 46.



4.6.5 Módulo Edición datos geográficos.

Para el módulo de edición de datos geográficos fue necesario implementar los submódulos de Exportador GeoJson y Generador área de influencia

Estos submódulos se implementaron por el módulo de seguridad, siendo estos submódulos nuevos se procedió a copiar todo el contenido necesario para su funcionamiento.

Una vez concluida la copia de carpetas y páginas necesarias se realizó las pruebas de funcionamiento de cada uno de los procesos del módulo completo para verificar posibles errores o problemas de integración. El módulo integrado en su totalidad se muestra en la figura 47.



4.7 Detección de errores y corrección de los mismos

Al iniciar con las respectivas pruebas de cada uno de los procesos que existen en el sistema se encontró ciertos errores en los queries de consulta que se realizaba para la carga de cierta información; estos fueron corregidos en la versión integrada para su correcto funcionamiento.

Uno de los métodos de consulta que fue necesario corregir fue la consulta de beneficiarios ya que en la pantalla de administración de beneficiarios los datos estaban visualizándose como vacíos tal como se muestra en la figura 48.

Problema de carga de datos en sub módulo Beneficiario

The screenshot shows a web browser window with the URL `localhost:8080/S.I.Salesiano/gestionDatos/DatosBeneficiario.jsf`. The page header includes the logo for 'SALESIANOS ECUADOR' and a navigation menu. The main content area displays a table titled 'DATOS_BENEFICIARIOS' with 14 rows. The table has columns for 'Lugar', 'Id Beneficiario', 'Id Estilo Beneficiario', 'Descripción', and 'Número'. The first row shows a beneficiary with 'Id Beneficiario' 51 and 'Número' 0. The second row shows a beneficiary with 'Id Beneficiario' 3 and 'Número' 783. The third row shows a beneficiary with 'Id Beneficiario' 71 and 'Número' 900. The fourth row shows a beneficiary with 'Id Beneficiario' 59 and 'Número' 1000. The 'Descripción' column contains text such as 'Estudiantes de carreras que requieren prácticas de campo, Universidad Politécnica Salesiana Paute', '753 estudiantes mayores de 15 años y 30 estudiantes de Danza formados por alumnos y exalumnos, Cantón Cayambe', 'Clientes de varias parroquias', and 'Población local v xii'. The 'Número' column contains the values 0, 783, 900, and 1000. The table also includes 'SELECCIONAR' and 'ELIMINAR' buttons for each row. A red arrow points to the empty 'Id Beneficiario' and 'Id Estilo Beneficiario' fields in the first row, with the text 'Campos Vacios' written in red next to it.

Lugar	Id Beneficiario	Id Estilo Beneficiario	Descripción	Número		
	51		Estudiantes de carreras que requieren prácticas de campo, Universidad Politécnica Salesiana Paute	0	SELECCIONAR	ELIMINAR
	3		753 estudiantes mayores de 15 años y 30 estudiantes de Danza formados por alumnos y exalumnos, Cantón Cayambe	783	SELECCIONAR	ELIMINAR
	71		Clientes de varias parroquias	900	SELECCIONAR	ELIMINAR
	59		Población local v xii	1000	SELECCIONAR	ELIMINAR

UNIVERSIDAD POLITÉCNICA SALESIANA
Todos los derechos reservados.

Figura 48. Problema de carga de datos en sub módulo Beneficiario

Muestra datos que no están visualizándose en la tabla de consulta

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Fue necesario implementar en el método de consulta una corrección en las restricciones que se realizaban para obtener la información de la tabla de beneficiarios; el mismo se muestra en la figura 49.

Corrección query de consulta submódulo Beneficiario

```
public static ResultSetConsultarTodo() {  
  
    ResultSets = null;  
  
    String sentencia = "select  
  
n.id_ben,n.id_eben,n.id_lug,n.descripcion_ben,n.numero_ben,n.areainfluenci  
a_ben,n.estado_ben ,teb.descripcion_eben, l.nombre_lug, l.id_obr  
,o.id_casfromtb_beneficiario,tb_estilobeneficiarioteb,tb_lugarl,tb_obrasalesi  
anao,tb_casasalesiana c "  
  
+ " where n.id_lug=l.id_lug "  
  
        + " and l.id_obr=o.id_obr "  
  
        + " and o.id_cas=c.id_cas "  
  
        + " and n.estado_ben=true "  
  
        + " and n.id_eben=teb.id_eben;";  
  
    try {  
  
        ConexionclsCon = newConexion();  
  
        rs = clsCon.Consulta(sentencia);  
  
    } catch (Exception ex) {  
  
        System.out.println(ex.getLocalizedMessage());  
  
    }  
  
    return;  
  
}
```

Figura 49. Corrección query de consulta submódulo Beneficiario

Muestra la corrección del slect de consulta de información de obras y casa salesiana

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Además se realizó mejoras en el diseño del sistema ya que en el proceso de traslado de clases a la versión integrada ciertas tablas y estilos se distorsionaron.

En la figura 50 se muestra la corrección de estilos en la pantalla de administración de sub módulos que fue necesario implementar en la versión integrada.

Corrección Estilo tabla de datos submóduloAreaInfluencia



Figura 50. Corrección Estilo tabla de datos submóduloAreaInfluencia

Visualización estilos tablas con problemas en el proceso de integración

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Se implementó en el sub-módulo cambio de contraseña un mensaje de confirmación al realizar el proceso con éxito. El mensaje que se agregó para su validación se muestra en la Figura 51

Mensaje de confirmación cambio de contraseña

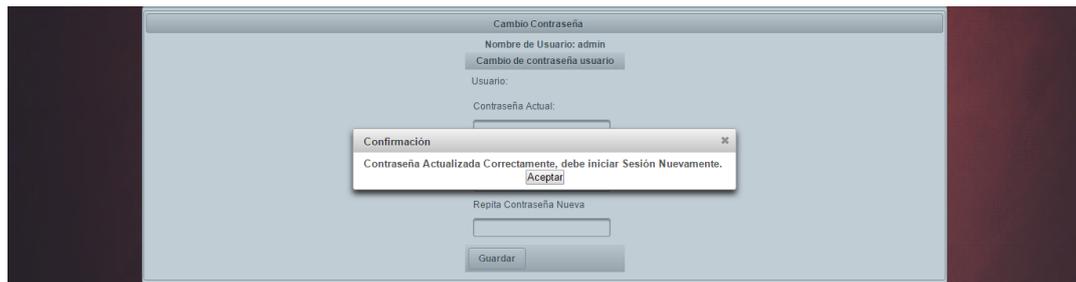


Figura 51. Mensaje de confirmación cambio de contraseña

Visualización ventana emergente de confirmación de cambios de contraseña

Elaborado por: Lizeth Sánchez & Carlos Yáñez

En la implementación del reporte de auditoría se tuvo problemas con la versión de la librería de JASPERREPORT ya que generó conflictos con el otro reporte utiliza que utiliza una versión más actual del mismo. Se actualizo la versión de la librería jasperreports_5.6.0 para ambos reportes.

Con la actualización de la librería se cambió un método del controlador GeneraReporteSeguridad que no existía en la versión actual tal como se puede observar en la Figura 52.

Método reemplazado controlador GeneraReporteSeguridad

```
WinMerge - [Sin título izquierda - Sin título derecha]
Archivo  Editor  Ver  Combinar  Herramientas  Plugins  Ventana  Ayuda
Resultado de la comparación de carpetas  Sin título izquierda - Sin título derecha

"Sin título izquierda"
= null;
ap ();
ll) {
xt = FacesContext.getCurrentInstance();
n = (HttpSession) context.getExternalContext().getSession(false);
vletContext = (ServletContext) context.getExternalContext().getCont
ibute("reporteInforme");
cha_inicio", new Timestamp(fechaInicial.getTime()));
cha_fin", new Timestamp(fechaFinal.getTime()));
ondo", servletContext.getRealPath("/imagenes/logoFondo.png"));
 JasperReport jroader.loadObjectFromFile(servletContext.getRealPath("reporteReporte");
nManager.runReportToPdf(jasperReport, parameters, con);
("Variable reporte "+jasperReport+"Variable Parametros " +parameter
te("reporteInforme", fichero);

funciona el reporte " + ex);
funciona el reporte " + e);

Ln: 206 Col: 14/74 Ch: 14/14      1252 Win

"Sin título derecha"
= null;
ap ();
ll) {
xt = FacesContext.getCurrentInstance();
n = (HttpSession) context.getExternalContext().getSession(false);
vletContext = (ServletContext) context.getExternalContext().getCont
ibute("reporteInforme");
cha_inicio", new Timestamp(fechaInicial.getTime()));
cha_fin", new Timestamp(fechaFinal.getTime()));
ondo", servletContext.getRealPath("/imagenes/logoFondo.png"));
 JasperReport jroader.loadObject(servletContext.getRealPath("reporteReporte");
nManager.runReportToPdf(jasperReport, parameters, con);
("Variable reporte "+jasperReport+"Variable Parametros " +parameter
te("reporteInforme", fichero);

funciona el reporte " + ex);
funciona el reporte " + e);

Ln: 133 Col: 27/38 Ch: 27/38      1252 Win
```

Figura 52. Método reemplazado controlador GeneraReporteSeguridad

Comparación visual librería jasper report

Elaborado por: Lizeth Sánchez & Carlos Yáñez

El módulo de administración implemento mediante la librería PrimeFaces la opción de filtros en las tablas de datos que permiten la selección y búsqueda de forma más óptima, esta opción no estaba incluida en el módulo de seguridad, por lo que fue necesario incorporar esta funcionalidad en la tabla de datos faltantes.

Se agregó mediante código la funcionalidad correspondiente en el módulo de administración en la capa de visualización (Paginas JSF) como se muestra en la figura 53.

Fragmento de código para filtros

```
114 <p:dataTable id="tabla" value="#{controladorColaborador.tdc}" style="width:100%"
115 var="ColabItem" border="1" paginator="true" rows="10"
116 paginatorTemplate="{CurrentPageReport} {FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink}
117 {LastPageLink} {RowsPerPageDropdown}"
118 rowsPerPageTemplate="15,20,30">
119 <f:facet name="header">
120 <h:outputText value="D A T O S _ C O L A B O R A D O R E S" />
121 </f:facet>
122 <p:column filterBy="#{ColabItem.nomLug}">
123 <f:facet name="header">
124 <h:outputText value="Id Lugar" />
125 </f:facet>
126 <h:outputText value="#{ColabItem.nomLug}" />
127 </p:column>
128 <p:column>
129 <f:facet name="header">
130 <h:outputText value="Id Colaborador" />
131 </f:facet>
132 <h:outputText value="#{ColabItem.idcola}" />
133 </p:column>
134 <p:column>
135 <f:facet name="header">
136 <h:outputText value="Número" />
137 </f:facet>
138 <h:outputText value="#{ColabItem.numcola}" />
139 </p:column>
140 <p:column filterBy="#{ColabItem.descCol}">
141 <f:facet name="header">
142 <h:outputText value="Tipo Colaborador" />
143 </f:facet>
```

Figura 53. Fragmento de código para filtros

Muestra el código html incluido en la visualización de tablas los módulos

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Una vez agregado el fragmento de código se habilito la opción de filtros en los sub-módulos del módulo de integración, la figura 54 muestra la funcionalidad implementada.

Utilización de PrimeFaces para filtros



Figura 54. Utilización de PrimeFaces para filtros

Visualización filtros implementados en los sub módulos

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Para mostrar de manera detallada las funcionalidades de cada una de las versiones y los cambios que se produjo en cada una de las mismas, se realizó un cuadro comparativo entre el módulo de seguridad y el módulo de administración, con la finalidad de obtener un sistema totalmente integrado con las versiones actuales de los procesos que realiza el Geo-Portal Salesiano.

Cada uno de los cambios que se realizaron en los módulos del sistema se detalla de manera general en la Tabla 23.

Tabla 23.

Cuadro comparación funcionalidades existentes en cada módulo

Módulos	Tesis Administración	Tesis Seguridad
Administración	<p>Se conservó en su totalidad el sub módulo :</p> <ul style="list-style-type: none"> • Auditoría de Sistema 	<p>Versión actualizada de los sub módulos :</p> <ul style="list-style-type: none"> • Edición Usuario • Edición Módulo • Edición Sub módulo • Edición Rol • Edición Permiso Menú • Edición Persona
Gestión de datos	<p>Se conservó los sub módulos pero fue necesaria la corrección de ciertos métodos debido a errores encontrados en las consultas SQL de los datos.</p>	<p>No existían cambios.</p>

Seguridad	Se conservó por completo cada uno de los sub módulos que lo conforman.	No existía
Reportes	Reporte de auditoría se conservó en su totalidad ya que no existía en la otra versión.	Reporte de sistema se conservó en su totalidad ya que no existía en la otra versión
Edición datos geográficos	Se conservó los sub módulos que existían en el mismo.	Sub módulos nuevos: <ul style="list-style-type: none"> • Exportador GeoJson • Generador área de influencia.
Visualizador	Se conservó por completo cada uno de los sub módulos	No existía
Gestión estilos	Se conservó por completo cada uno de los sub módulos	No existía

Nota: describe las funcionalidades que se implementaron y las que se conservaron en el proceso de integración

5. Implementación y pruebas

5.1 Implementación

En este capítulo se describe el proceso de implementación del sistema Geo-Portal salesiano integrado en su totalidad en el servidor que fue designado por la universidad.

5.2 Características

A continuación en la Tabla 24 se especifica los requerimientos a nivel de software necesarios para el correcto funcionamiento del sistema.

Tabla 24.
Requerimientos de Software integración

Especificaciones de software	
Sistema Operativo	Centos versión 5.9
Base de Datos	PostgreSQL versión 9.1
Datos Espaciales	PostGIS versión 1.5
Servidor Web	Apache 7.5.0
Lenguaje de Desarrollo	Java 7

Nota: describe la arquitectura usada para la implementación del módulo de integración

5.3 Restauración de la base de datos

La restauración de la base de datos se realizó mediante el cliente PgAdmin para lo que fue necesario conectarnos remotamente al servidor y configurar la dirección IP en el archivo pg_hba.conf para permitir el acceso.

En la figura 55 se muestra la conexión desde el cliente PgAdmin a la base de datos del servidor.

Acceso al servidor por PgAdmin



Figura 55. Acceso al servidor por PgAdmin

Visualiza la conexión al servidor de base de datos del servidor

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Una vez realizada la conexión se procedió a la creación de la base de datos con la template postgis necesaria para los datos geográficos que se manejan en el sistema; la creación del mismo se muestra en la Figura 56.

Creación base de datos

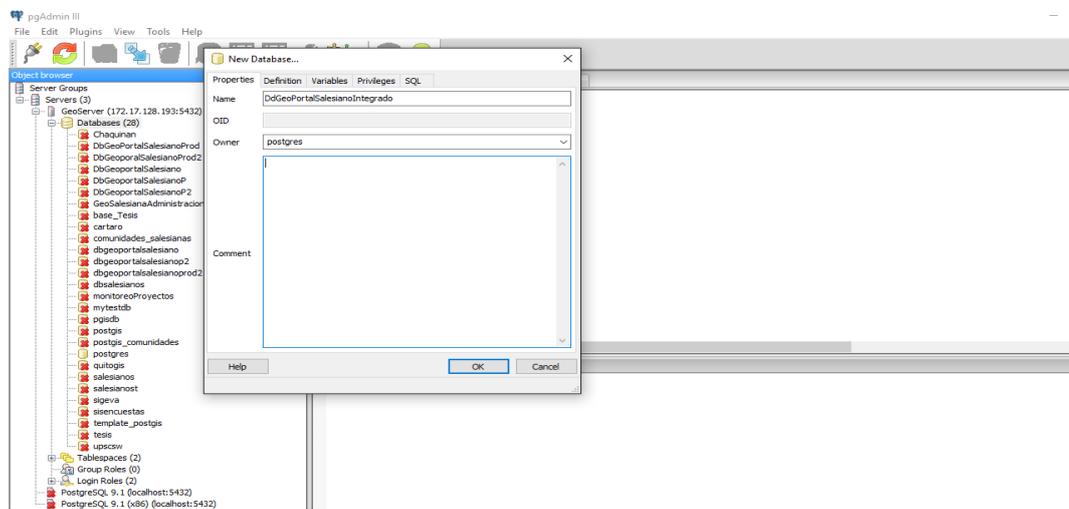


Figura 56. Creación base de datos

Creación base de datos para la implementación en el servidor

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Para la restauración de la base de datos se ejecutó la línea de comando que se muestra en la Figura 57; esta se realizó a través de la herramienta Putty para la ejecución del backup de la base de datos integrada.

Restauración base de datos

```
[root@ide data]# psql -U postgres -h localhost -d "DbGeoPortalSalesianoIntegrado" -f /opt/tmp/administracion.sql
```

Figura 57. Restauración base de datos

Comando sql utilizado para la restauración de la base de datos en el servidor

Elaborado por: Lizeth Sánchez & Carlos Yáñez

5.3.1 3. Configuración y carga del archivo .war en el servidor Apache Tomcat.

Una vez iniciado el servicio del servidor apache tomcat ingresamos la dirección <http://ide.ups.edu.ec:8080> en el navegador web para verificar que este se encuentre levantado tal como se muestra en la figura 58.

Servidor apache tomcat inicializado

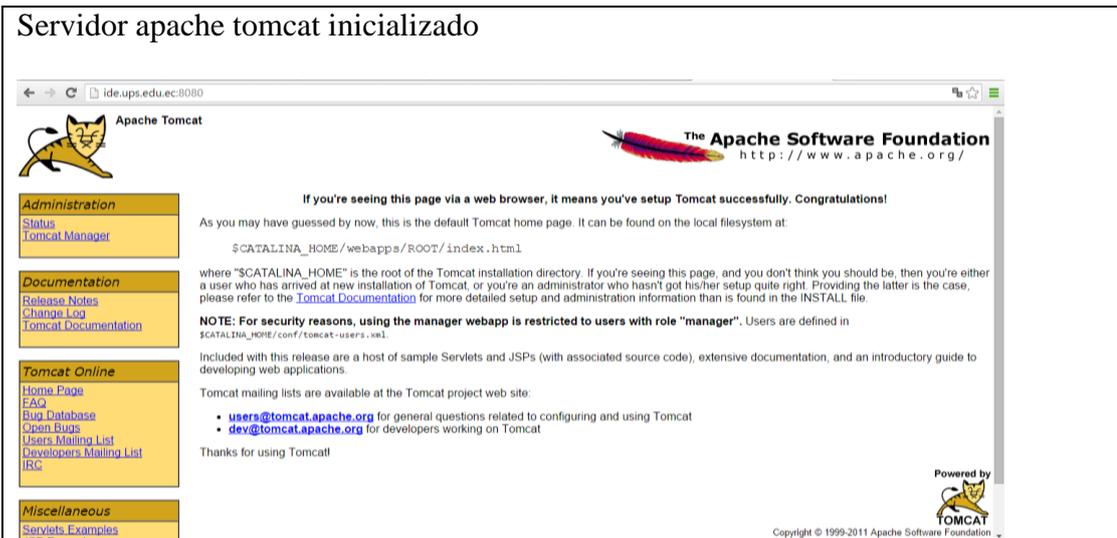


Figura 58. Servidor apache tomcat inicializado

Visualiza el servidor apache tomcat iniciado desde el servidor

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Se debe ingresar en la opción Manager Webapp del servidor el cual administra todas las aplicaciones que se entran deployadas en el servidor. Para acceder digitamos el usuario y contraseña que se encuentran configurados en el servidor información que fue proporcionada por la universidad tal como se muestra en la figura 59.

Acceso Manager Weapp

Figura 59. Acceso Manager Weapp.

Inicio de sesión con el usuario de administración en el servidor apache tomcat

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Se desplegará el gestor de aplicaciones web del servidor y se puede observar los proyectos cargados que han sido configurados en el mismo como se puede observar en la figura 60.

Gestor de aplicaciones web de tomcat

Mensaje:

Gestor

Listar Aplicaciones Ayuda HTML de Gestor Ayuda de Gestor Estado de Servidor

Aplicaciones

Trayectoria	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar > <input type="text" value="30"/> minutos
/CSW		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar > <input type="text" value="30"/> minutos
/EdicionGrafica		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar > <input type="text" value="30"/> minutos
/ProyectoTesisUPSP		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar > <input type="text" value="30"/> minutos
/QuitoGis		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar > <input type="text" value="30"/> minutos
/Salesianos/ModuloEdicionGrafica		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar > <input type="text" value="30"/> minutos
/Seguridad		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar > <input type="text" value="30"/> minutos

Figura 60. Gestor de aplicaciones web de tomcat

Muestra las aplicaciones levantadas en el servidor apache

Elaborado por: Lizeth Sánchez & Carlos Yáñez

En la parte inferior de la página se encuentra la sección de Deploy donde cargaremos nuestro proyecto por medio de un archivo .war.

Para realizar la subida del archivo, se debe pulsar el botón browse que se encuentra en esta sección de la página y seleccionar el archivo .war que se ha generado del proyecto del módulo integrado que se desea subir al servidor y finalmente presionar el botón deploy tal como se muestra en la figura 61.

Carga archivo .war en el servidor

The screenshot shows the Tomcat Manager web interface. At the top, there is a table of applications:

Trayectoria	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/manager	Tomcat Manager Application	true	2	Expirar sesiones sin trabajar > 30 minutos
/temp		false	0	Arrancar Parar Recargar Replegar
/wsp7		false	0	Arrancar Parar Recargar Replegar

Below the table, the 'Desplegar' section is active, showing a form to deploy a WAR file. The 'Archivo WAR a desplegar' section shows a selected file: 'GeoSalesiano...stracion.war'. The 'Información de Servidor' section at the bottom provides system details:

Versión de Tomcat	Versión JVM	Vendedor JVM	Nombre de SO	Versión de SO	Arquitectura de SO
Apache Tomcat/6.0.32	1.7.0_75-mockbuild_2015_01_20_18_28-b00	Oracle Corporation	Linux	2.6.18-348.3.1.el5.centos.plus	amd64

Figura 61. Carga archivo .war en el servidor

Visualización carga archivo .war en el servidor apache

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Una vez finalizado el proceso de deployment se mostrará en la parte superior de la página el sistema con un link de acceso y el estado en el que se encuentra. En la figura 62 se observa el proyecto de integración en estado inactivo.

Sistema Geo-Portal salesiano deployado en el servidor

The screenshot shows the Tomcat Manager web interface displaying the 'Aplicaciones' section. The table below lists the deployed applications:

Trayectoria	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/CSW		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/EdicionGrafica		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/GeoSalesianoAdministracion	GeoSalesianoAdministracion	false	0	Arrancar Parar Recargar Replegar
/ProyectoTesisUPSP		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/QuitoGis		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/Salesianos/ModuloEdicionGrafica		true	0	Arrancar Parar Recargar Replegar
/Seguridad		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/TesisP	TesisP	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/Visualizacion		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos

Figura 62. Sistema Geo-Portal salesiano deployado en el servidor.

Muestra el sistema que fue implementado en el servidor para ser ejecutado

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Para poder ejecutar el sistema simplemente se debe pulsar en el link que se encuentra junto al nombre del sistema. Una vez realizado el proceso de implementación se procesó hacer pruebas de funcionamiento en todo el sistema para verificar el correcto funcionamiento de los procesos que se realizan en cada uno de los sub módulos.

Para un mejor manejo del sistema se encuentra en el [Anexo 2](#) el manual de usuario desarrollado por el módulo de seguridad y el de administración.

5.4 Pruebas

En la fase de pruebas del proyecto de integración se ejecutó test de rendimiento de la aplicación web en el servidor por medio de la herramienta JMETER en donde se evaluó los siguientes aspectos, como se muestra en la figura 63.

Parámetros de prueba de carga sobre el sistema integrado

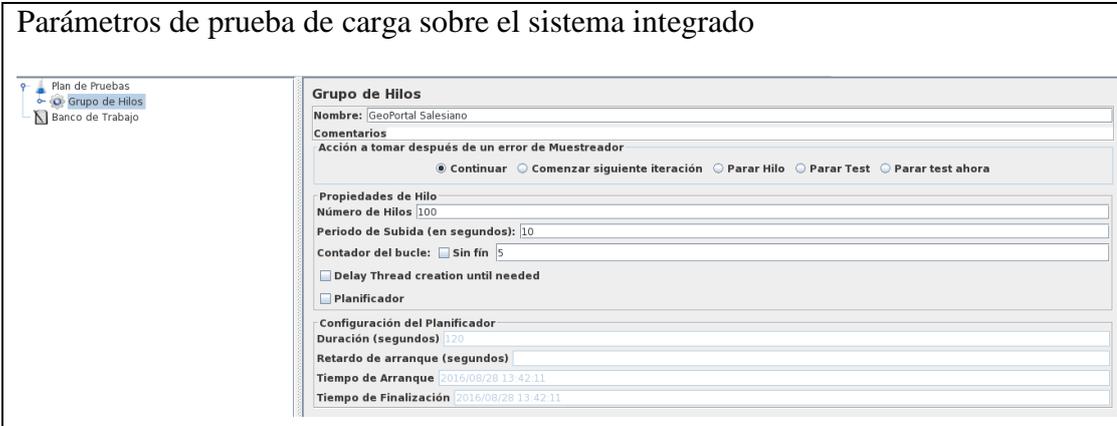


Figura 63. Parámetros de prueba de carga sobre el sistema integrado.

Muestra los parámetros que intervienen en la prueba de carga sobre el sistema integrado

Elaborado por: Lizeth Sánchez & Carlos Yáñez

En donde se establece un total de 100 usuarios que serán incrementados cada 10 segundos que serán evaluados sobre la ruta de inicio de la aplicación /Tesis/index.jsf.

El tiempo de ejecución está establecido en 3 minutos y se obtuvo como resultados lo detallado en la tabla 25

Tabla 25.
Tabla de resultados de prueba de carga

Resultados de pruebas de carga	
Sistema Operativo	Centos versión 5.9
Servidor Web	Apache 7.5.0
Lenguaje de Desarrollo	Java 7
Tiempo de respuesta	2.4 segundos
Promedio transferencia	2808 bytes
Valor mínimo	2327 bytes
Valor Máximo	3506 bytes
Desviación estándar	375.77
Porcentaje de Error	0.00%

Nota: Describe la tabla de resultados de las pruebas de carga del módulo de integración

En la figura 64 se puede apreciar el reporte generado por la herramienta JMETER después de efectuar la prueba de carga

Reporte de resultados de carga

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: Errors Success

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
HTTP Requ...	10	2808	2327	3506	375.77	0.00%	2.4/sec
TOTAL	10	2808	2327	3506	375.77	0.00%	2.4/sec

Figura 64. Reporte de resultados de carga.

Muestra el reporte de resultados generado por la herramienta JMETER

Elaborado por: Lizeth Sánchez & Carlos Yáñez

Conclusiones

- La implementación del Geo-Portal salesiano permitió la obtención de información unificada de los módulos de administración y seguridad con la finalidad de entregar información coherente de forma eficaz y eficiente, evitando así la redundancia de información a nivel de funcionalidad como de base de datos
- La implementación de un Framework de seguridad en el sistema permite manejar de mejor manera los permisos a los recursos que dispone el sistema mediante el uso de perfiles, roles y usuarios.
- El proceso de integración permitió obtener una base de datos consistente a las funcionalidades implementadas en los módulos unificados.

Recomendaciones

- El uso de un versionador de código permitirá trabajar con la misma fuente entre varias personas, el mismo que será mucho más manejable; además permitiría conservar información detallada de historial y versiones del código fuente. Otro de sus beneficios de uso es que permite compartir elementos entre varios proyectos y recuperar información que haya sido eliminada accidentalmente.
- La implementación de JPA (Java Persistence API) para la ejecución de queries de consulta mejoraría el tiempo de desarrollo del proyecto y la eliminación de errores en tiempo de ejecución. Además el mantenimiento al sistema sería de una forma más fácil y mejorada.
- El uso del API de Google para la gestión de mapas es una alternativa que ayuda a simplificar el desarrollo y mejorar la interacción con el usuario, sin embargo es recomendable parametrizar el uso de recursos que provienen del servidor, ya que está en constante cambio y puede generar problemas de acceso a las herramientas

Referencias bibliográficas

- Ambler, S. (2001). *Agile Modeling and eXtreme Programming (XP)*. Recuperado de <http://agilemodeling.com/essays/agileModelingXP.htm>
- Beck, K. y Andres, C. (2000). A Development Episode. En O'Hagan, D. (Ed.), *Extreme Programming Explained* (pp. 142). Canada: Addison-Wesley.
- Cohn, M. (2007). *Differences Between Scrum and Extreme Programming*.
Recuperado de <https://www.mountangoatsoftware.com/blog/differences-between-scrum-and-extreme-programming>
- Cortez, C. y Molina, V. (2013). *Metodologías ágiles Programación Xtrema*.
Recuperado de <http://es.slideshare.net/LisPater1/metodologias-agiles-xp>
- Echeverry, L. y Delgado, L. (2007). *Caso práctico de la metodología ágil XP al desarrollo de software* (Tesis de pregrado, Universidad Tecnológica de Pereira, Pereira, Colombia). Recuperado de <http://recursosbiblioteca.utp.edu.co/dspace/bitstream/handle/11059/794/0053E18cp.pdf;jsessionid=E8540F44077D81047066ADF97BB719C2?sequence=1>
- Fowler, M. (2005). *The New Methodology*. Recuperado de <http://www.martinfowler.com/articles/newMethodology.html>
- Gutiérrez, J. y Escalona, M. (2006). *Pruebas del sistema en programación extrema*.
Recuperado de http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf

- Hutagalung, W. (2006). *Extreme Programming*. Recuperado de <http://www.umsl.edu/~sauterv/analysis/f06Papers/Hutagalung/>
- Islam, K. (2013). Agile Methodologies. En Islam, K. (Ed.), *Agile Methodology for Developing & Measuring Learning: Training Development and Measuring Learning* (pp. 77). Bloomington: AuthorHouse.
- Jeffries, R. (2001). *What is Extreme Programming*. Recuperado de <http://ronjeffries.com/xprog/book/whatisxp/>
- Letelier, P. y Penades, C. (2006). *Metodologías ágiles para el desarrollo de software*. Recuperado de <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- O'Reilly, A. (2003). *Five Lessons You Should Learn from Extreme Programming*. Recuperado de <http://www.onlamp.com/pub/a/onlamp/2003/07/31/extremeprogramming.htm>
- 1
- Pressman, R. (2010). *Ingeniería de Software Un Enfoque Práctico*. Connecticut: McGraw-Hill Interamericana Editores S.A.
- Portillo, T. (2012). *Consultor en Sistemas de Información*. Recuperado de <http://www.tayronportilloconsultant.profusehost.net/esp/index.html>
- Rosenberg, D. y Stephens, M. (2003). The Big Picture. En Appleman, D. (Ed.), *Extreme Programming Refactored* (pp. 400). United States: Apress.
- Shore, J. y Warden, S. (2008). How to be Agile?. En O'Brien, M. (Ed.), *The Art of Agile Development* (pp. 417). United States of America: O'Really Media.

Wells, D. (2013). *Extreme Programming: A gentle introduction*. Recuperado de
<http://www.extremeprogramming.org/>

ANEXOS

Anexo 1. Manual de usuario

El manual de usuario del geo-portal salesiano se encuentra localizado en la ruta
./Anexos/manualUsuario.docx

Anexo 2. Script de base de datos

El script de base de datos para la nueva estructura se encuentra localizado en la ruta
./Anexos/script_base_GeoSalesianaAdministracion.sql