

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA: INGENIERÍA DE SISTEMAS**

**Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS**

**TEMA:  
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL MÓDULO DE  
VISUALIZACIÓN Y GESTIÓN DE DATOS GEOGRÁFICOS PARA EL  
GEOPORTAL DE LA COMUNIDAD SALESIANA VERSIÓN 2.0**

**AUTORES:  
VÍCTOR HUGO COFRE GONZÁLEZ  
STALIN PATRICIO TOLEDO CUENCA**

**DIRECTOR:  
GUSTAVO ERNESTO NAVAS RUILOVA**

**Quito, mayo del 2014**

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO  
DEL TRABAJO DE TITULACIÓN**

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

-----  
Víctor Hugo Cofre González

CC: 1717162398

-----  
Stalin Patricio Toledo Cuenca

CC: 1717201253

## **AGRADECIMIENTO**

A la Universidad Politécnica Salesiana, por ser el lugar donde aprendimos y consolidamos conocimientos, por impartirnos no solo una educación superior de calidad como profesionales sino por formarnos como buenos ciudadanos con mira a siempre pensar en los demás.

A nuestro tutor de trabajo de grado Ing. Gustavo Navas, quien con su amabilidad, paciencia y dedicación ha sido un amigo y guía en el proceso de culminar con éxito nuestra carrera universitaria.

**Víctor Cofre González**

**&**

**Stalin Toledo Cuenca**

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1 .....	2
MARCO TEÓRICO.....	2
1.1. Objetivos .....	2
1.1.1. Objetivo general .....	2
1.1.2. Objetivos específicos .....	2
1.2. Alcance.....	2
1.3. Justificación.....	3
1.4. Marco teórico .....	3
1.4.1. Geoportal web .....	3
1.4.2. Metodología OOHDM .....	4
1.4.3. Herramientas de desarrollo a utilizar. ....	5
1.4.4. Arquitectura en tres capas .....	7
CAPÍTULO 2 .....	9
REQUERIMIENTOS DEL SISTEMA.....	9
2.1. Roles y tareas .....	9
2.2. Escenarios .....	10
2.3. Especificación de casos de uso .....	10
2.3.1. Especificación de casos de uso en la gestión de datos geográficos .....	11
2.3.2. Especificación de casos de uso del visualizador .....	14
2.4. Especificación de UIDs.....	19
2.4.1. UID Modulo de gestión de datos geográficos.....	19
2.4.2. UID Modulo de visualización .....	21
CAPÍTULO 3 .....	26
DISEÑO .....	26
3.1. Diseño conceptual de clases.....	27
3.2. Diseño navegacional .....	28
3.3. Diseño de contexto navegacional.....	30
3.4. Diseño de interfaz abstracta .....	32
3.5. Diagrama de clases.....	39
3.6. Modelo de la base de datos .....	42
3.7. Diccionario de base de datos.....	44

3.8. Diagrama de componentes .....	49
3.9. Diagrama de despliegue .....	50
CAPÍTULO 4 .....	51
DESARROLLO .....	51
4.1. Gestor de datos geográficos. ....	51
4.2. Visualizador .....	57
4.3. Estructura GeoJson para ingresar el área de influencia. ....	60
CAPÍTULO 5 .....	62
IMPLEMENTACIÓN Y PRUEBAS .....	62
5.1. Implementación.....	62
5.1.2. Requerimientos mínimos .....	62
5.1.3. Restauración de la base de datos .....	62
5.1.4. Carga del archivo .war en el servidor Apache Tomcat .....	63
5.2. Pruebas .....	66
CONCLUSIONES .....	70
RECOMENDACIONES .....	71
LISTA DE REFERENCIAS .....	72
ANEXOS .....	75

## ÍNDICE DE FIGURAS

Figura 1. Fases de la metodología OOHDM.....	4
Figura 2. Arquitectura en tres capas.....	8
Figura 3. Caso de uso ingreso manual de la ubicación del lugar .....	11
Figura 4. Caso de uso ingreso visual de la ubicación del lugar .....	12
Figura 5. Caso de uso ingreso de área de influencia.....	13
Figura 6. Caso de uso de búsqueda por categoría .....	14
Figura 7. Caso de uso de búsqueda por temática .....	15
Figura 8. Caso de uso de búsqueda por tipo de obra.....	16
Figura 9. Caso de uso de búsqueda por casa.....	17
Figura 10. Caso de uso de búsqueda por casa y tipo de obra.....	18
Figura 11. UID escenario ingreso manual de la ubicación del lugar .....	19
Figura 12. UID del escenario de ingreso visual de la ubicación del lugar.....	20
Figura 13. UID escenario ingreso del área de influencia.....	20
Figura 14. UID escenario búsqueda por categoría .....	21
Figura 15. UID escenario búsqueda temática .....	22
Figura 16. UID escenario búsqueda obra.....	23
Figura 17. UID escenario búsqueda casa-obra.....	24
Figura 18. UID escenario búsqueda casa-tipo-obra .....	25
Figura 19. Diseño conceptual de clases .....	27
Figura 20. Diseño navegacional de la gestión de datos geográficos.....	28
Figura 21. Diseño navegacional del visualizador .....	29
Figura 22. Diseño de contexto navegacional de gestión de datos geográficos .....	30
Figura 23. Diseño de contexto navegacional del visualizador.....	31
Figura 24. ADV's Pantalla ingreso manual de la ubicación del lugar.....	32
Figura 25. ADV's Pantalla ingreso visual de la ubicación del lugar .....	33
Figura 26. ADV's Pantalla ingreso del área de influencia.....	34
Figura 27. ADV's Pantalla búsqueda categorías.....	35
Figura 28. ADV's Pantalla búsqueda temática .....	36
Figura 29. ADV's Pantalla búsqueda tipo de obra.....	37
Figura 30. ADV's Pantalla búsqueda casa-obra .....	38
Figura 31. ADV's Pantalla búsqueda casa-tipo-obra.....	39
Figura 32. Diagrama de clases del gestor datos geográficos .....	40
Figura 33. Diagrama de clases del visualizador.....	41

Figura 34. Modelo lógico de la base de datos .....	42
Figura 35. Modelo físico de la base de datos .....	43
Figura 36. Diagrama de componentes del módulo de visualización.....	49
Figura 37. Diagrama de componentes del módulo de gestión de datos geográficos .	49
Figura 38. Diagrama de despliegue.....	50
Figura 39. Ejemplo estructura GeoJson .....	61
Figura 40. Creación de la base de datos .....	62
Figura 41. Ejecución del script postgis .....	63
Figura 42. Ejecución de script Spatial.....	63
Figura 43. Restauración de la base de datos .....	63
Figura 44. Ingreso al servidor web.....	63
Figura 45. Ingreso a administración del servidor.....	64
Figura 46. Pantalla de gestión de aplicaciones.....	64
Figura 47. Selección del archivo war .....	64
Figura 48. Archivo war seleccionado .....	65
Figura 49. Proyectos desplegados .....	65
Figura 50. Creación de grupo de hilos .....	66
Figura 51. Creación de petición HTTP .....	67
Figura 52. Creación de resultados .....	67
Figura 53. Resultados de prueba con 50 muestras .....	68
Figura 54. Resultados de prueba con 150 muestras .....	68
Figura 55. Resultados de prueba con 200 muestras .....	69

## ÍNDICE DE TABLAS

Tabla 1. Roles y tareas del módulo de gestión de datos geográficos.....	9
Tabla 2. Roles y tareas del módulo de visualización .....	9
Tabla 3. Especificación de escenario del módulo de gestión geográfica.....	10
Tabla 4. Especificación de escenario del módulo de visualización .....	10
Tabla 5. Casos de uso del ingreso manual de la ubicación del lugar .....	11
Tabla 6. Casos de uso ingreso visual de la ubicación del lugar .....	12
Tabla 7. Casos de uso ingreso de área de influencia.....	13
Tabla 8. Casos de uso búsqueda por categoría.....	14
Tabla 9. Casos de uso búsqueda por temática.....	15
Tabla 10. Casos de uso búsqueda por tipo de obra .....	16
Tabla 11. Casos de uso búsqueda por casa .....	17
Tabla 12. Casos de uso búsqueda por casa y tipo de obra .....	18
Tabla 13. Diccionario de datos tabla casa salesiana .....	44
Tabla 14. Diccionario de datos tabla tipo de obra salesiana .....	45
Tabla 15. Diccionario de datos tabla lugar.....	45
Tabla 16. Diccionario de datos tabla obras salesiana.....	46
Tabla 17. Diccionario de datos tabla estilo lugar.....	47
Tabla 18. Diccionario de datos tabla estilo beneficiario .....	47
Tabla 19. Diccionario de datos tabla foto lugar .....	48
Tabla 20. Diccionario de datos tabla beneficiario.....	48
Tabla 21. Requerimientos de software.....	62



## **RESUMEN**

Este producto desarrollará el módulo de gestión y visualizador del geoportal de la Comunidad Salesiana a partir de su versión 1.0 para incrementar requerimientos del usuario en estos módulos.

El documento consta de 5 capítulos que se detallan a continuación.

El capítulo 1 contiene información de la metodología OOADM y herramientas de desarrollo que se utilizará, así como las versiones requeridas para el desarrollo e implementación del producto.

El capítulo 2 describe los requerimientos para el sistema a desarrollar, estos en base a las necesidades que el cliente manifieste.

El capítulo 3 parte del análisis de los requerimientos para crear los diagramas de clases, base de datos, secuencia, entre otros que servirán como base para la creación del sistema y a futuro sean una herramienta para el mantenimiento de la aplicación.

El capítulo 4 explica la creación de la aplicación en base al diseño, se usará la tecnología JSF conjuntamente con primefaces.

El capítulo 5 sirve de referencia en caso de tener problemas al momento de realizar las pruebas para corregir y tener un sistema confiable, las pruebas se las realizará al producto desplegado en el servidor web mediante la herramienta JMeter.

Por último tenemos las conclusiones y recomendaciones del desarrollo del sistema.

## **ABSTRACT**

This product will develop the management module and display the geoportal of the Salesian Community from version 1.0 to increase user requirements in these modules.

The document consists of 5 chapters below.

Chapter 1 contains information on the OOHDM methodology and development tools to be used, as well as those required for the development and implementation of product releases.

Chapter 2 describes the requirements for the system to be developed, based on these requirements the client manifest.

Chapter 3 of the requirements analysis to create class diagrams, database , string, among others that serve as the basis for the creation of future system and are a tool for the maintenance of the application.

Chapter 4 explains the creation of the application based on the design, use JSF technology together with primefaces.

Chapter 5 serves as a reference in case you have problems when testing to correct and have a reliable system, tests were performed to the product displayed in the web server using the JMeter tool.

Finally we have the conclusions and recommendations of system development.

## INTRODUCCIÓN

La Inspectoría Salesiana “Sagrado Corazón de Jesús” nace el 31 de julio de 1973 gracias al Decreto del Rector Mayor. Es producto de la unificación de la Inspectoría de Quito con la de Cuenca; es así que se crea la Inspectoría del Ecuador con sede en Quito y, además se establece una Delegación para el Vicariato de Méndez.

Y a mediados del año 2002, en la calle Madrid E12-39 y Andalucía, se puso la primera piedra de la nueva sede Inspectorial, bendecida por el P. Esteban Ortiz y en presencia de numerosos miembros de la Familia Salesiana. La nueva casa está ubicada en la Calle Madrid E12-68 y Andalucía.

El 26 de febrero de 2004 se inauguró la Casa Inspectorial, con la presencia del Rector Mayor de los Salesianos, Pascual Chávez.

Actualmente existen más de 26 casas salesianas y 173 salesianos: 129 sacerdotes; 16 coadjutores; 20 salesianos clérigos y 8 novicios. (Salesianos, 2014)

La necesidad de ayudar a las personas y jóvenes tanto en el aspecto social como pastoral y educativo, como lo hizo Don Bosco, ha permitido una expansión de casas y obras salesianas en todo el Ecuador, que se encuentran en distintas provincias de las mismas que requiere un mejor conocimiento con respecto a las actividades que realiza cada una y así poder saber: los requerimientos, la autoridad a cargo, número de colaboradores, beneficiarios, el lugar o lugares donde están ubicadas, área de influencia en la cual se encuentran los beneficiarios, entre otros datos que ayudaran a poder realizar un mejor control de las actividades y fortalecer las debilidades si existieran y así también abarcar las zonas y lugares a los que aún no se han podido ayudar. Por ello surge la necesidad de un portal web que ayude a cumplir las necesidades requeridas por la Inspectoría Salesiana y toda la comunidad en general, proyecto que consta de diferentes módulos como son: módulo de visualización, módulo de gestión de datos, módulo de gestión de datos geográficos, módulo de gestión de estilos y módulo de seguridad. El módulo de visualización y gestión de datos geográficos requiere soportar varios lugares para las obras salesianas e igualmente el poder asignar varios polígonos para el área de influencia de los beneficiarios así como varias adaptaciones que mejoraran la usabilidad y funcionalidad del geoportal web.

# CAPÍTULO 1

## MARCO TEÓRICO

### 1.1. Objetivos

#### 1.1.1. Objetivo general.

Analizar, diseñar, construir e implementar el módulo de visualización y gestión de datos geográficos para el GEOPORTAL DE LA COMUNIDAD SALESIANA a partir de su versión 1.0.

#### 1.1.2. Objetivos específicos.

- Incluir en el visualizador de la comunidad salesiana los siguientes elementos: escala gráfica, simbología, latitud y longitud en el mapa.
- Manipular los datos geográficos a través del formato *GeoJSON* para mejorar la velocidad de respuesta del geoportal.
- Mostrar información geográfica la cual pueden ser: puntos, polígonos, multipolígono, etc.
- Permitir al usuario gestionar varios puntos geográficos a la vez, sean estos de lugares salesianos o beneficiarios.

### 1.2. Alcance

El resultado final de este producto contendrá el visualizador y la gestión de datos geográficos en su versión 2.0 con las mejoras sugeridas por la Inspectoría Salesiana.

**Módulo visualización:** Mostrará la información de casas y obras en un mapa digital, permitiendo realizar búsqueda más intuitiva, de fácil manejo y visualización para el usuario así como una simbología clara con la respectiva leyenda.

**Módulo de ingreso de datos geográficos:** Este módulo trabajará con un archivo “*GeoJSON*” que tendrá la estructura de los datos geográficos de un polígono o un multipolígono para el ingreso del área de influencia de los beneficiarios de determinada obra. Con respecto al ingreso del lugar donde se encuentra ubicada la obra, el sistema permitirá subir uno o varios puntos de referencia.

### **1.3. Justificación**

En la actualidad gracias a la tecnología se crean aplicaciones que ayudan a las personas, en este caso la aplicación permitirá conocer la labor que realiza la Comunidad Salesiana en diferentes partes del Ecuador, y mostrar los diferentes lugares donde actúan las obras salesianas brindando apoyo a los moradores de los distintos sectores.

El objetivo principal de la versión 2.0 es permitir el ingreso de uno o varios puntos que represente el lugar donde está ubicada la obra, también el ingreso y visualización de polígonos o multipolígono en el área de influencia del beneficiario lo que involucra retirar restricciones en la base de datos que no permiten esta funcionalidad así como también modificar el visualizador para trabajar con este tipo de datos ya que una de las limitaciones es que la librería de primefaces no permite trabajar con multipuntos y multipolígono a nivel de capa de presentación.

Tomando en cuenta la necesidad de utilizar multipolígonos y multipuntos para la correcta apreciación y almacenamientos de los datos geográficos en el módulo de gestión geográfica como en el visualizador, se ve necesario el programar estos requerimientos a nivel de controladores y servicios, así obtener la conversión de los datos para que la presentación sea transparente para el usuario en ambos módulos.

### **1.4. Marco teórico**

El marco teórico como parte fundamental permite desarrollar la teoría del producto.

#### **1.4.1. Geoportal web.**

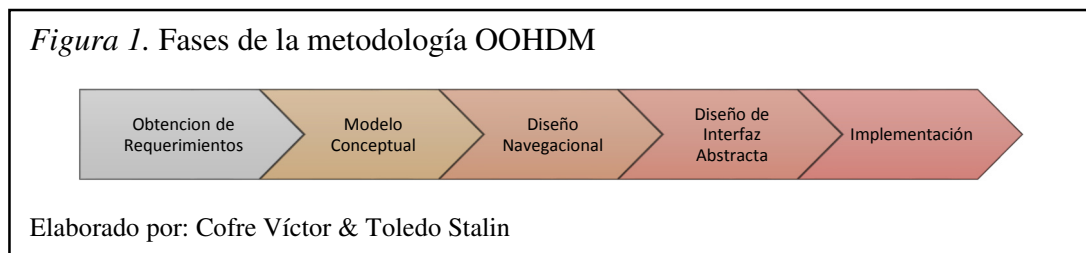
Un geoportal o portal geoespacial se presenta como el acceso vía Internet a información geográfica. La información que puede ofrecer un geoportal puede ser de lo más variada. Mediante un geoportal se utiliza la red para permitir el acceso y visualización de los datos geoespaciales, utilizando un navegador estándar de Internet, y favoreciendo la integración, interoperabilidad e intercambio de información entre las diferentes instituciones y personas.

La definición de un geoportal apunta a lograr una aplicación web que sea un punto de acceso a la información geográfica generada, y esta información sea servida mediante distintos recursos.” (Geopulse, 2013)

### 1.4.2. Metodología OOHDm.

Método de Diseño de Hipermedia Orientado a Objetos, es una metodología de desarrollo propuesta por Rossi y Schwabe para la elaboración de aplicaciones multimedia que tiene como objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia.

OOHDm propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por 5 etapas:



- **Obtención de requerimientos**

La obtención de requerimientos es una de las fases más importantes esta es el pilar con el que se empezará a trabajar, si estos están claros las características o descripción del software a crear cumplirá con las expectativas del cliente. En esta fase se obtendrán los diagramas de caso de uso.

- **Diseño conceptual**

En esta etapa se crea un modelo orientado a objetos cuyos atributos pueden ser de diferentes tipos de datos de tal manera que las clases que los contengan puedan optimizar su funcionamiento al momento de ejecutarse.

- **Diseño navegacional**

Un modelo navegacional es construido como una vista sobre el diseño conceptual, consta con un esquema de clases de navegación. El diseño de navegación es expresado en dos esquemas: el esquema de clases y de contextos navegacionales.

- **Diseño de interfaz abstracta**

Cuando se define las interfaces de la aplicación se definirán como aparecerán los objetos navegacionales y cuales activarán la navegación.

En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia. (Sevilla, 2008, pág. 13)

- **Implementación**

Al llegar a esta fase, el primer paso es definir los ítems de información que son parte del dominio del problema, además se organiza los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo se comportará. (Silva & Mercerat, 2002)

### **1.4.3. Herramientas de desarrollo a utilizar.**

Se menciona las herramientas que se usarán en la etapa de desarrollo del software y las respectivas versiones.

- **PostgreSQL 9**

Es un sistema de gestión de base de datos relacional, de código abierto y orientado a objetos, de software libre y publicado bajo la licencia BSD. Ofrece características como alta concurrencia, amplia variedad de tipos nativos, soporta tipos de datos y operaciones geométricas (PostgreSQL, 2013).

- **PostGIS 1.5**

Es una extensión de base de datos espacial para PostgreSQL. Añade soporte para objetos geográficos permitiendo consultas de ubicación para que se ejecuten en SQL. Se puede utilizar para almacenar y consultar datos espaciales (puntos, líneas, polígonos, etc). (Martínez, 2013, pág. 23)

- **Netbeans 7.1**

Netbeans es un IDE que permite desarrollo de aplicación java de escritorio, web, móvil, etc. de manera más rápida ya que cuenta con un gran número de librerías, plantillas, consejos y herramientas para facilitar el desarrollo de software aplicativo.

Es gratuito, de código abierto además soporta muchos lenguaje como Java, C / C + +, XML y HTML, PHP, JavaScript y JSP. (Corporation, 2013)

- **Quantum GIS (QGIS) 1.8**

Es un sistema de información geográfica de código libre, permite visualizar, administrar, editar, analizar los datos geográficos, tiene Licencia Pública General GNU. Se lo puede utilizar en diferentes sistemas operativos como Linux, Windows, Android, etc. (Foundation, 2013)

Las principales características son:

- Transforma el sistema de coordenadas de forma inmediata.
- Edición y selección de elementos geoespaciales.
- Edición, visualización y búsqueda de atributos.
- Herramientas de Análisis Espacial.
- Importación y exportación de datos GPS. (Fuentes, 2011)

- **JSF**

JavaServer Faces (JSF) es una tecnología basada en java para desarrollar aplicaciones web que contiene el código necesario para el manejo de eventos y organización de componentes. Al contar con componentes prediseñados permite al desarrollador concentrarse menos en la capas de presentación y dar prioridad a la capa de negocio y la capa de gestión de datos y actualmente se encuentra en contante mejora y actualización. (Geary & Horstmann, 2010, págs. 3-4)

- **Primefaces**

Es un componente diseñado para trabajar con JavaServerFaces(JSF) la ventaja es que al disponer de componentes diseñados el programador solo tiene que utilizarlos y no créalos desde cero, esto ayuda a tener una interfaz gráfica en menos tiempo y llamadas. Es de código abierto y tiene licencia de Apache License V2. (teknoloji, 2014, pág. 11)

Entre los componentes que se utilizara estarán:

- **Gmap:** Este componente está diseñado para trabajar con JSF, se puede utilizar Ajax, cabe resaltar que está construido sobre Google Maps API.



- **Markers:** Estas marcas o puntos pueden asociarse con iconos para aportar en el mejoramiento de la interfaz gráfica que se le presentara al usuario, además tienen diferentes comandos y eventos que puede ayudar al desarrollador de software a tener más variedad de efectos para mostrar la información.
- **Polygons:** Los polígonos se mostraran en el mapa para este caso se utilizaran para identificar el área de influencia de las distintas obras salesianas.
- **Info window:** Este componente muestra una pequeña ventana cuando se da clic sobre una marca que contendrá la información que el programador crea necesaria para mostrar al usuario.

- **Postgresql JDBC**

Es un controlador escrito en lenguaje java que permite mediante un conjunto de estándares e interfaces la comunicación en el propio protocolo de la red del sistema de bases de datos, que permitiendo trabajar con bases de datos independientes de la plataforma que se esté utilizando.

#### **1.4.4. Arquitectura en tres capas.**

Para la elaboración de este producto se utilizará una arquitectura en tres capas que permitirá que los datos, la lógica del negocio y presentación estén separados, lo que brinda una mayor seguridad ya que el usuario recibe los datos de forma indirecta a través del servidor. Las capas que se utilizaran son:

- **Capa de presentación o Interfaz de usuario**

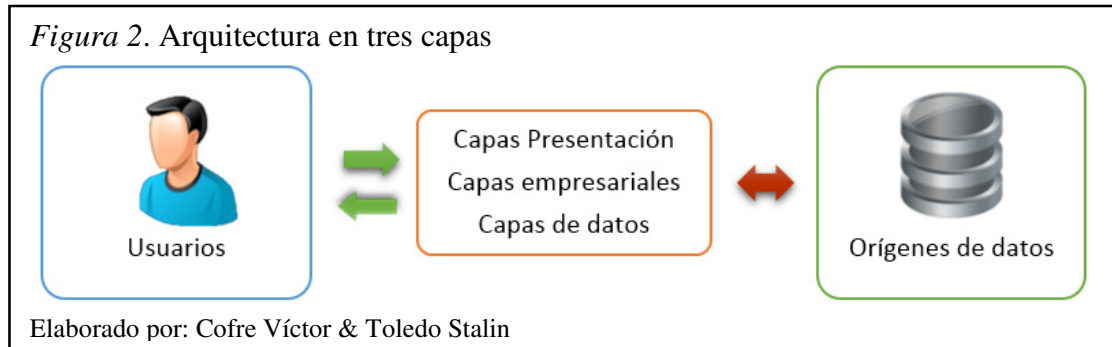
Aquí encontraremos las interfaces que comunican al usuario de una forma amigable y sencilla tanto para que el usuario ingrese la información o que se le pueda mostrar la información que él requiere de la base de datos. Cabe resaltar que las interfaces se crearan en páginas web JSF.

- **Capa de negocio o empresarial**

Para esta capa se definirán clases que contendrán métodos y atributos que permitan interactuar con la base de datos y el usuario a través de la capa de presentación y la capa de acceso a datos, además tendrá todas las reglas que deban cumplirse para que el sistema tenga un correcto funcionamiento.

- **Capa de acceso a datos**

En esta capa tendremos clases que ayudaran con la comunicación con la base de datos que en este caso será PostgreSQL 9.1 y su complemento de datos geográficos PostGIS 1.5 tanto para guardar información como para recuperarla y poderla mostrar al usuario.



## CAPÍTULO 2

### REQUERIMIENTOS DEL SISTEMA

Los requerimientos de los módulos de gestión de datos geográficos y visualizador del proyecto del geoportal de la comunidad salesiana tiene como finalidad, por un lado el ingreso de los lugares puede ser uno o varios puntos y el ingreso del área de influencia como polígonos y multipolígonos, por otro lado la visualización que trabaja con los tipos de datos geográficos ingresados.

#### 2.1. Roles y tareas

Entre los roles y tareas se recalca que funcionalmente esta característica será diseñada e implementada por el módulo de seguridad se los menciona en el documento para que se tenga en cuenta que roles y tareas deberá tener, tanto el módulo de gestión de datos geográficos como el módulo de visualización.

Tabla 1

*Roles y tareas del módulo de gestión de datos geográficos*

Roles	Tareas
<b>Administrador</b>	Ingreso de la información geográfico del lugar (uno o varios puntos) Ingreso de la información geográfico del beneficiario (polígono o multipolígono)

*Nota:* el usuario administrador tendrá acceso al módulo de gestión de datos geográficos para el ingreso o modificación.

Elaborado por: Cofre Víctor & Toledo Stalin

Tabla 2

*Roles y tareas del módulo de visualización*

Roles	Tareas
<b>Invitado</b>	Visualizar la información de las obras Salesianas mediante las distintas búsquedas disponibles.

*Nota:* el usuario invitado tendrá acceso al visualizador para ver la información de las obras en el mapa.

Elaborado por: Cofre Víctor & Toledo Stalin

## 2.2. Escenarios

Los escenarios permiten describen la funcionalidad del sistema al momento de ser utilizados, muestra los escenarios principales en el momento de su funcionamiento, para el escenario de gestión geográfica usará el rol de administrador mientras que el visualizador usará el rol de invitado. Ver roles en tabla 2.

Tabla 3

### *Especificación de escenario del módulo de gestión geográfica*

Escenario	Descripción
<b>Ingreso manual de la ubicación del lugar</b>	Permite el ingreso de los datos geográficos por coordenadas en grados decimales de cada lugar referente a la obra deseada
<b>Ingreso visual de la ubicación del lugar</b>	Permite el ingreso de los datos geográficos mediante marcas en el mapa, de cada lugar referente a la obra deseada
<b>Ingreso de área de influencia</b>	Permite el ingreso del área de influencia mediante un archivo GeoJson del lugar

Elaborado por: Cofre Víctor & Toledo Stalin

Tabla 4

### *Especificación de escenario del módulo de visualización*

Escenario	Descripción
<b>Búsqueda por categoría</b>	Permite ver la información e información geográfica de las obras mediante la selección de la casa y el tipo de obra
<b>Búsqueda por temática</b>	Permite ver la información e información geográfica del lugar mediante el ingreso de nombre del lugar
<b>Búsqueda por tipo de obra</b>	Permite ver la información e información geográfica de las obras mediante la selección del tipo de obra, la casa y obra
<b>Búsqueda por casa</b>	Permite ver la información e información geográfica de las obras mediante la selección de la casa y obra
<b>Búsqueda por casa y tipo de obra</b>	Permite ver la información e información geográfica de las obras mediante la selección del tipo de obra, la casa y obra

Elaborado por: Cofre Víctor & Toledo Stalin

## 2.3. Especificación de casos de uso

Los casos de uso identifican la relación entre los usuarios y los distintos procesos de la aplicación.

- **Especificación de casos de uso en la gestión de datos geográficos**

El caso de uso en la gestión de datos geográficos se encargará del ingreso de la información geográfica a la base de datos a través del sistema mediante el usuario administrador.

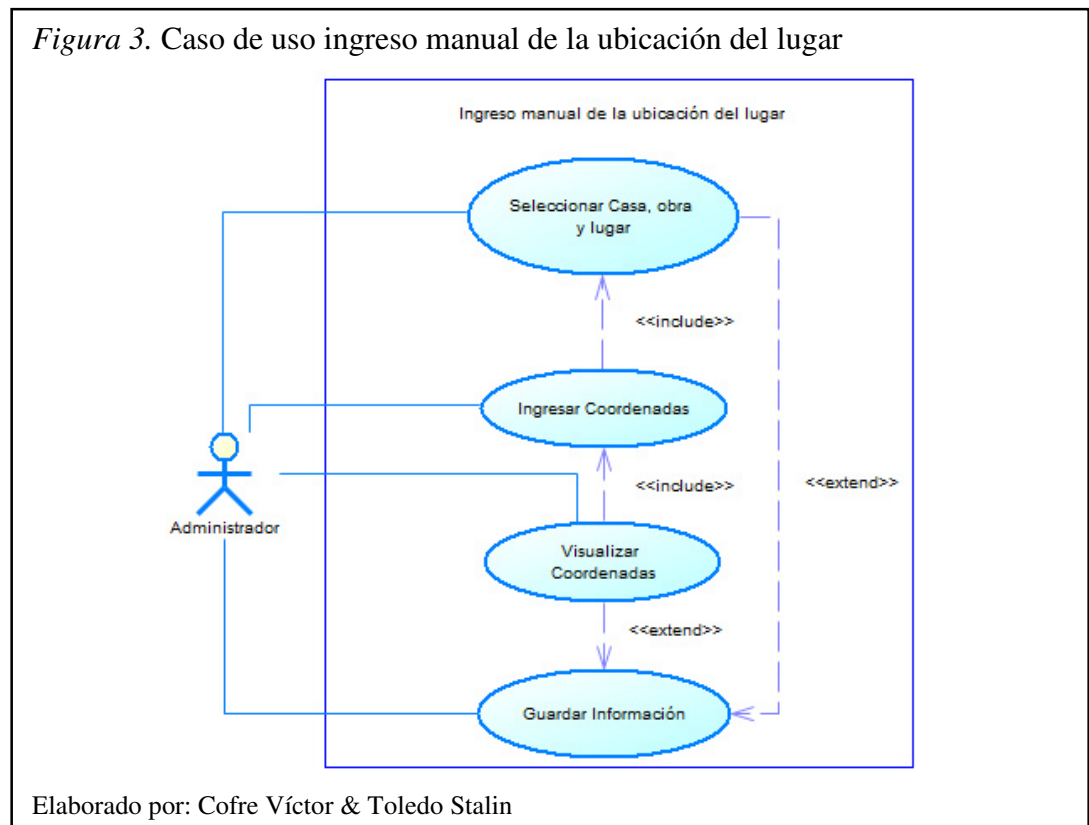
Entre los tipos de ingreso de la coordenada del lugar se encuentra el ingreso manual de la ubicación del lugar para el cual se debe establecer con anticipación la casa, obra, el lugar y la coordenada del lugar para guardar en la base de datos.

Tabla 5

*Casos de uso del ingreso manual de la ubicación del lugar*

Caso de uso # 1	Ingreso manual de la ubicación del lugar
<b>Actores</b>	Usuario Administrador
<b>Camino principal</b>	<ol style="list-style-type: none"> <li>1. Seleccionar el nombre de la casa salesiana</li> <li>2. Seleccionar el nombre de la obra salesiana</li> <li>3. Seleccionar la descripción del lugar</li> <li>4. Ingresar las coordenadas (longitud y latitud) de la marca que representara la ubicación lugar</li> <li>5. Visualizar la coordenada ingresada en el mapa</li> <li>6. Guardar la información</li> </ol>
<b>Camino secundario</b>	Al ingresar la coordenada verificar que la ubicación de la marca sea la deseada
<b>Precondiciones</b>	Para acceder se requiere estar registrado
<b>Postcondiciones</b>	Las coordenadas se almacena en la bases de datos

Elaborado por: Cofre Víctor & Toledo Stalin



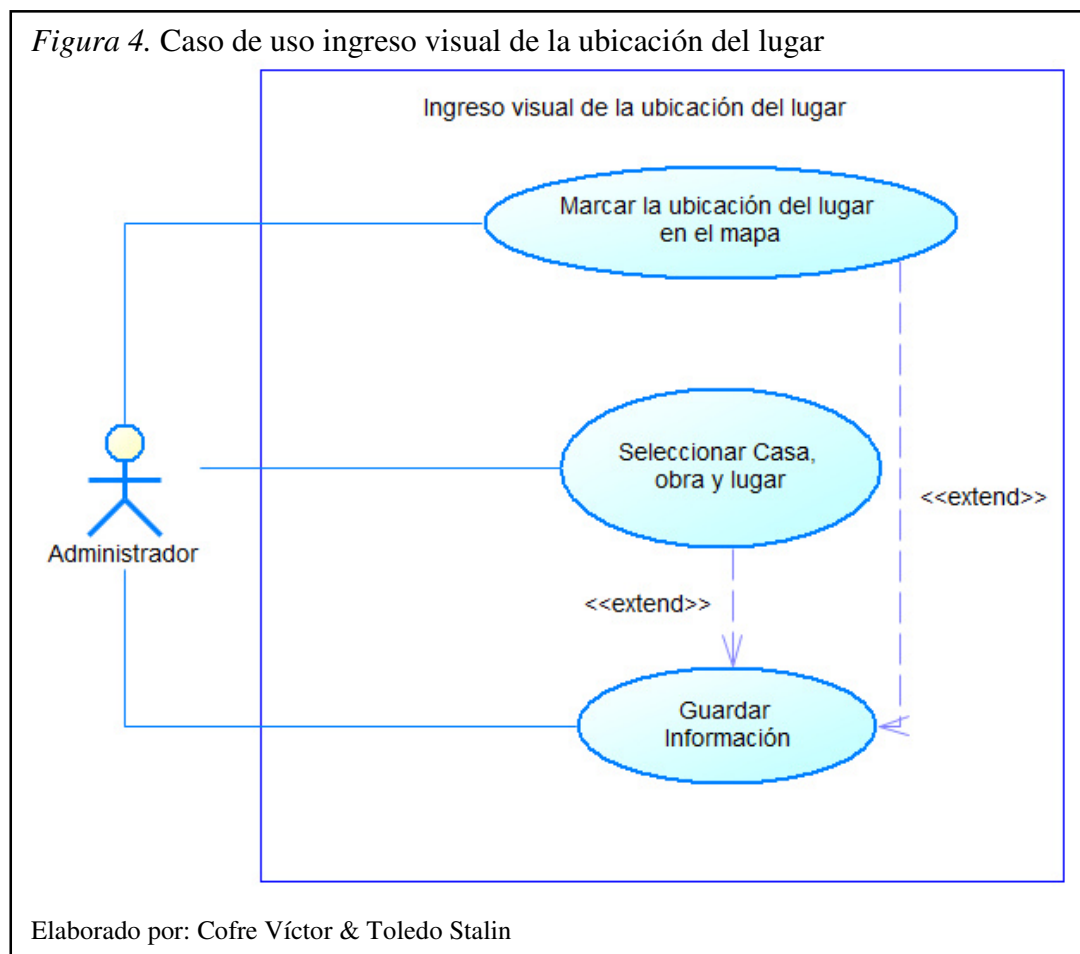
Entre los tipos de ingreso de la coordenada del lugar se encuentra el ingreso visual de la ubicación del lugar, para el cual primero deberá buscar en el mapa la ubicación del lugar y dar un clic sobre el mismo para que se agregue una marca al mapa, y así obtener la coordenada del lugar y luego se especifica la casa, obra y el lugar al que se le va asignar esa coordenada para posteriormente guardar en la base de datos.

Tabla 6

*Casos de uso ingreso visual de la ubicación del lugar*

Caso de uso # 2	Ingreso visual de la ubicación del lugar
<b>Actores</b>	Usuario Administrador
<b>Camino principal</b>	Marcar la ubicación del lugar en el mapa Seleccionar el nombre de la casa salesiana Seleccionar el nombre de la obra salesiana Seleccionar la descripción del lugar Guardar la información
<b>Camino secundario</b>	Verificar que los puntos estén correctamente ubicados
<b>Precondiciones</b>	Para acceder se requiere estar registrado
<b>Postcondiciones</b>	Las coordenadas se almacena en la bases de datos

Elaborado por: Cofre Víctor & Toledo Stalin



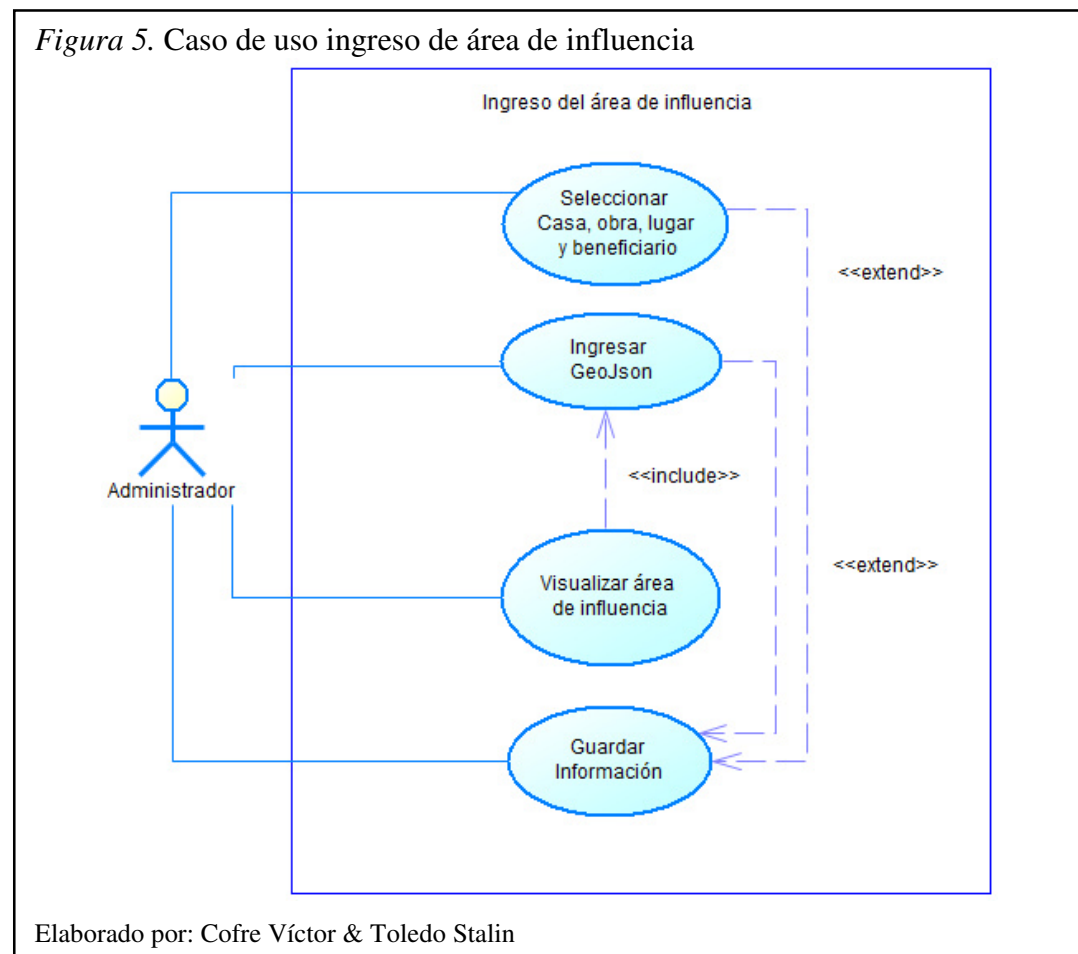
El caso de uso de ingreso de área de influencia indica que es necesario seleccionar la casa, obra, lugar y el beneficiario al que se va a asignar el archivo GeoJson con el respectivo polígono o multipolígono para guardarlo en la base de datos.

Tabla 7

Casos de uso ingreso de área de influencia

Caso de uso # 3	Ingreso de área de influencia
<b>Actores</b>	Usuario Administrador
<b>Camino principal</b>	Seleccionar el nombre de la casa salesiana Seleccionar el nombre de la obra salesiana Seleccionar la descripción del lugar Seleccionar descripción del beneficiario Seleccionar archivo GeoJson Visualizar el área de influencia en el mapa Guardar la información
<b>Camino secundario</b>	La selección de GeoJson es un archivo por beneficiario Si el archivo no se visualiza reducir el número de puntos de los polígonos
<b>Precondiciones</b>	El archivo debe ser un GeoJson El número de puntos de los polígonos no debe superar los 150
<b>Postcondiciones</b>	La información geográfica se convertirá y almacenará en un campo geométrico en la base de datos

Elaborado por: Cofre Víctor & Toledo Stalin



- **Especificación de casos de uso del visualizador**

En caso de uso del visualizador se encargara de indicar las distintas formas de búsqueda que permitirán al usuario invitado visualizar tanto la información de las obras salesianas así como sus datos geográficos.

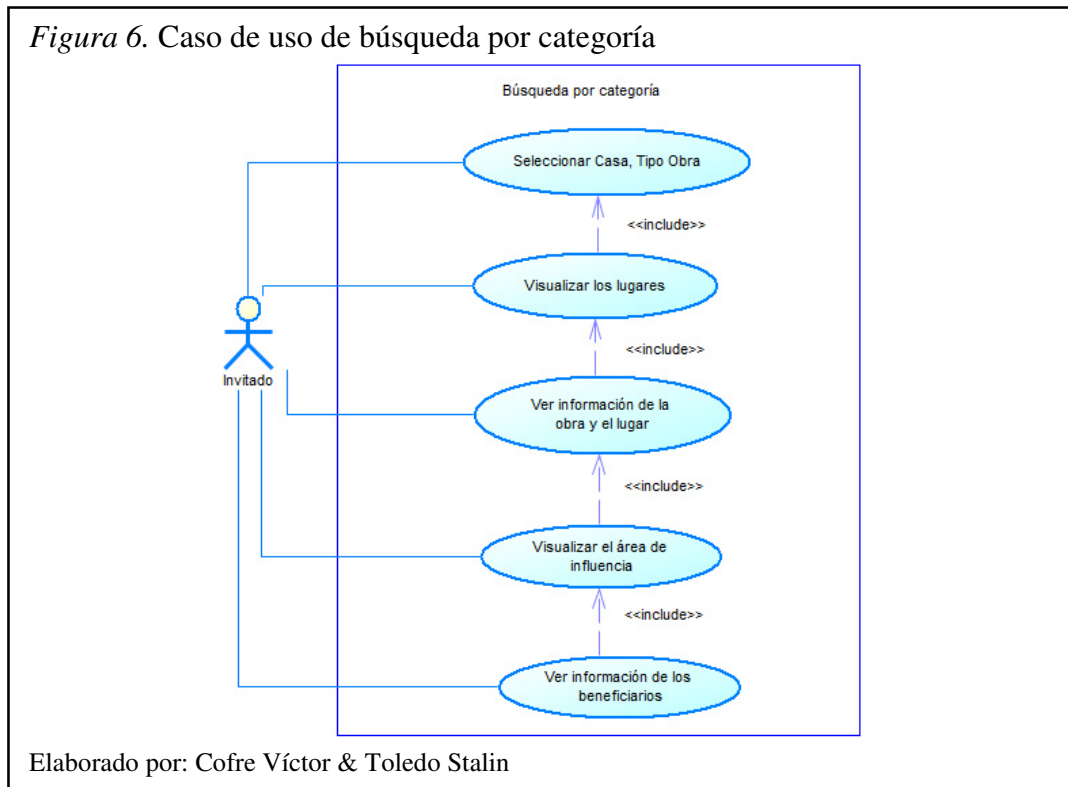
En el caso de uso de búsqueda por categoría muestra cómo se realizará la búsqueda de las distintas obras mediante la selección de la casa y el tipo de obra.

Tabla 8

*Casos de uso búsqueda por categoría*

Caso de uso # 4	Búsqueda por categoría
<b>Actores</b>	Usuario invitado
<b>Camino principal</b>	Seleccionar el nombre de la casa salesiana Seleccionar la descripción del tipo de obra salesiana Visualizar los lugares resultantes de la búsqueda Ver la información del lugar Visualizar el área de influencia Ver la información del beneficiario deseado
<b>Camino secundario</b>	Si no se ha ingresado el área de influencia se mostrara un mensaje que indique que no existe el dato geográfico
<b>Precondiciones</b>	Tener la información de las distintas obras salesiana en la base de datos
<b>Postcondiciones</b>	La búsqueda muestra los lugares de las obras salesianas y su respectiva información.

Elaborado por: Cofre Víctor & Toledo Stalin





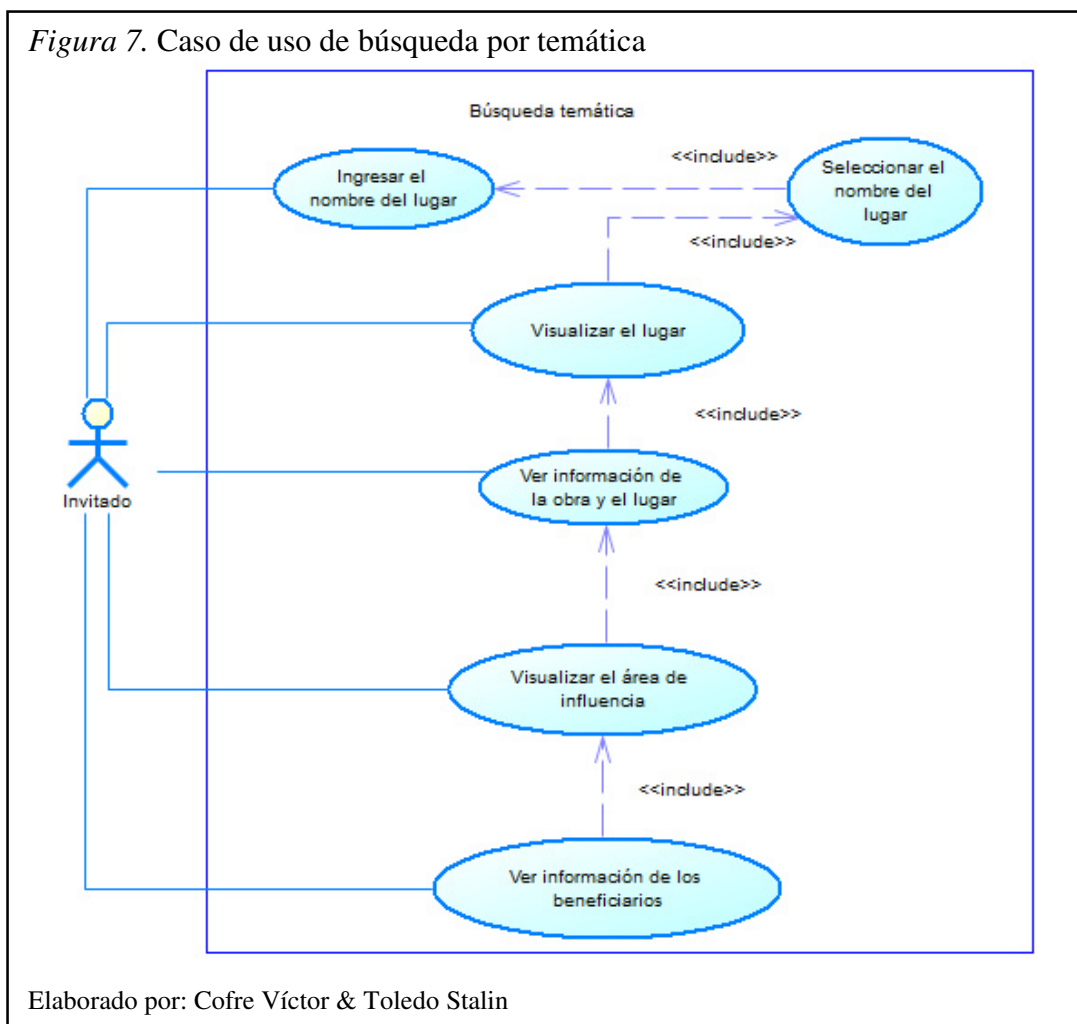
En el caso de uso de búsqueda temática, muestra cómo se realizará la búsqueda de lugares mediante el nombre o parte del nombre del lugar del que se desee observar.

Tabla 9

*Casos de uso búsqueda por temática*

Caso de uso # 5	Búsqueda temática
<b>Actores</b>	Usuario Invitado
<b>Camino principal</b>	Ingresar el nombre de lugar Seleccionar el nombre del lugar deseado Visualizar el lugar resultante de la búsqueda Ver la información del lugar Visualizar el área de influencia Ver la información del beneficiario deseado
<b>Camino secundario</b>	Si no se ha ingresado el área de influencia se mostrara un mensaje que indique que no existe el dato geográfico
<b>Precondiciones</b>	Tener la información de las distintas obras salesiana en la base de datos
<b>Postcondiciones</b>	La búsqueda muestra el lugar de la obra salesiana y su respectiva información.

Elaborado por: Cofre Víctor & Toledo Stalin



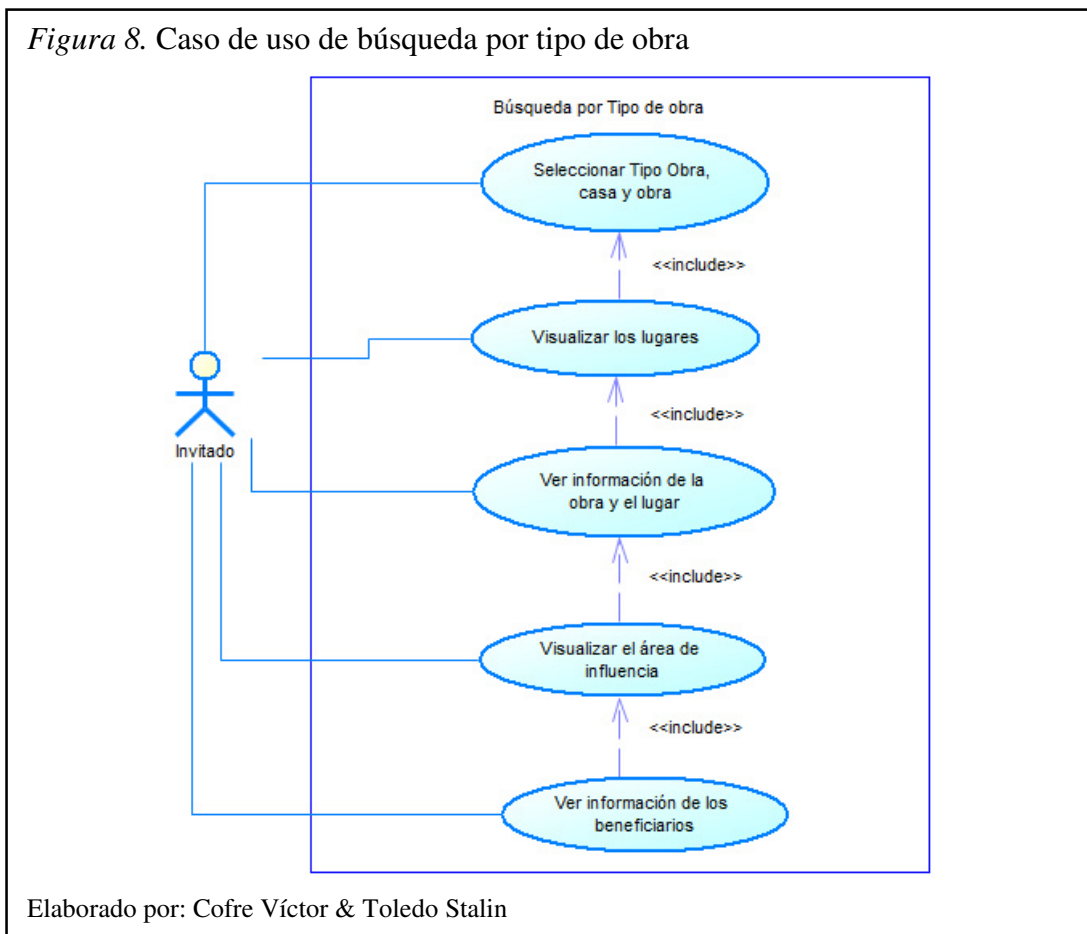
En el caso de uso de búsqueda por tipo de obra, muestra cómo se realizará la búsqueda de las distintas obras y lugares mediante la selección del tipo obra, la casa y la obra a visualizar.

Tabla 10

*Casos de uso búsqueda por tipo de obra*

Caso de uso # 6	Búsqueda por tipo de obra
<b>Actores</b>	Usuario Invitado
<b>Camino principal</b>	Seleccionar la descripción del tipo de obra salesiana Seleccionar el nombre de la casa salesiana Seleccionar el nombre de la obra salesiana Visualizar los lugares resultantes de la búsqueda Ver la información del lugar Visualizar el área de influencia Ver la información del beneficiario deseado
<b>Camino secundario</b>	Se pueden seleccionar varios tipos de obras salesianas Se pueden seleccionar varias casas salesianas Se pueden seleccionar varias obras salesianas Si no se ha ingresado el área de influencia se mostrara un mensaje que indique que no existe el dato geográfico
<b>Precondiciones</b>	Tener la información de las distintas obras salesiana en la base de datos
<b>Postcondiciones</b>	La búsqueda muestra los lugares de las obras salesianas y su respectiva información.

Elaborado por: Cofre Víctor & Toledo Stalin



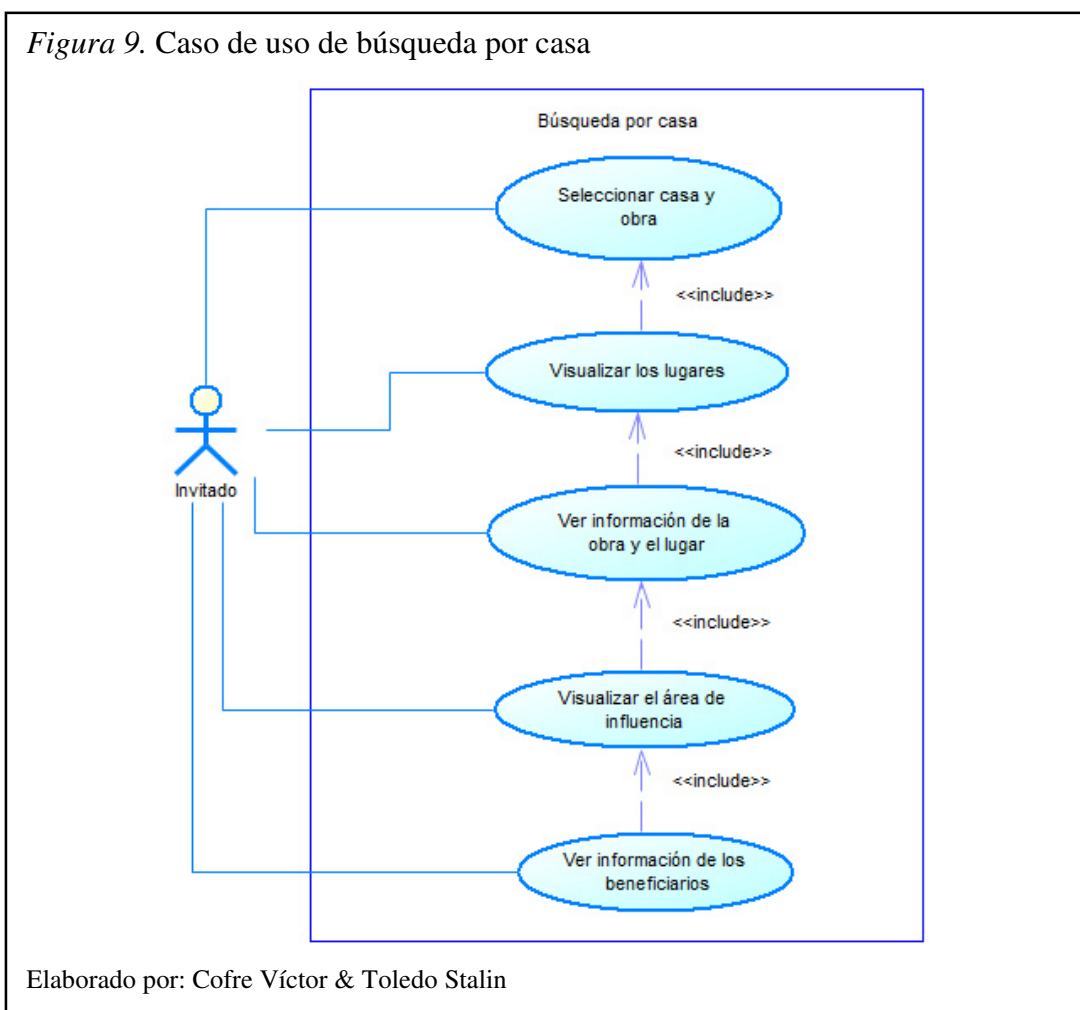
En el caso de uso de búsqueda por casa, muestra cómo se realizará la búsqueda de las distintas obras mediante la selección de la casa(s) y la obra(s).

Tabla 11

Casos de uso búsqueda por casa

Caso de uso # 7	Búsqueda por casa
<b>Actores</b>	Usuario Invitado
<b>Camino principal</b>	Seleccionar el nombre de la casa salesiana Seleccionar el nombre de la obra salesiana Visualizar los lugares resultantes de la búsqueda Ver la información del lugar Visualizar el área de influencia Ver la información del beneficiario deseado
<b>Camino secundario</b>	Se pueden seleccionar varias casas salesianas Se pueden seleccionar varias obras salesianas Si no se ha ingresado el área de influencia se mostrara un mensaje que indique que no existe el dato geográfico
<b>Precondiciones</b>	Tener la información de las distintas obras salesiana en la base de datos
<b>Postcondiciones</b>	La búsqueda muestra los lugares de las obras salesianas y su respectiva información.

Elaborado por: Cofre Víctor & Toledo Stalin



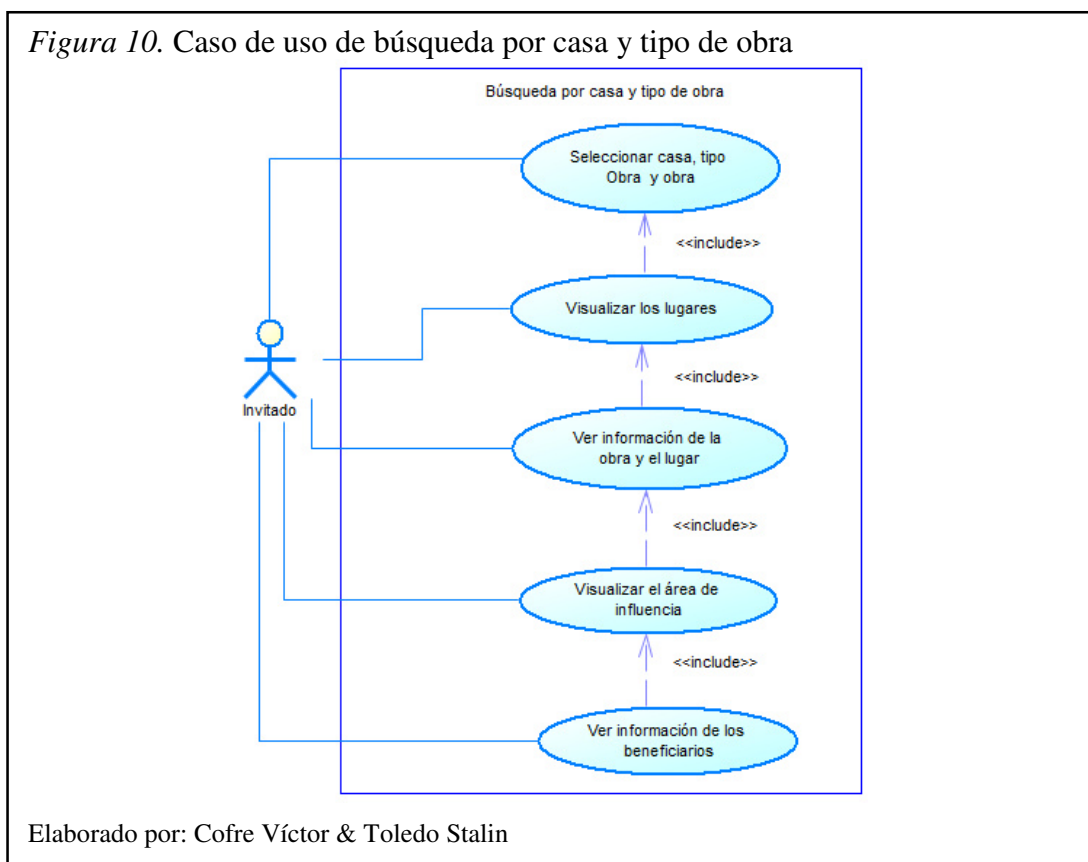
En el caso de uso de Búsqueda por casa y tipo de obra, muestra cómo se realizará la búsqueda de las distintas obras mediante la selección de la casa, el tipo de obra y la obra que se quiere visualizar.

Tabla 12

Casos de uso búsqueda por casa y tipo de obra

Caso de uso # 8	Búsqueda por casa y tipo de obra
<b>Actores</b>	Usuario Invitado
<b>Camino principal</b>	Seleccionar el nombre de la casa salesiana Seleccionar la descripción del tipo de obra salesiana Seleccionar el nombre de la obra salesiana Visualizar los lugares resultantes de la búsqueda Ver la información del lugar Visualizar el área de influencia Ver la información del beneficiario deseado
<b>Camino secundario</b>	Se pueden seleccionar varias casas salesianas Se pueden seleccionar varios tipos de obras salesianas Se pueden seleccionar varias obras salesianas Si no se ha ingresado el área de influencia se mostrara un mensaje que indique que no existe el dato geográfico
<b>Precondiciones</b>	Tener la información de las distintas obras salesiana en la base de datos
<b>Postcondiciones</b>	La búsqueda muestra los lugares de las obras salesianas y su respectiva información.

Elaborado por: Cofre Víctor & Toledo Stalin

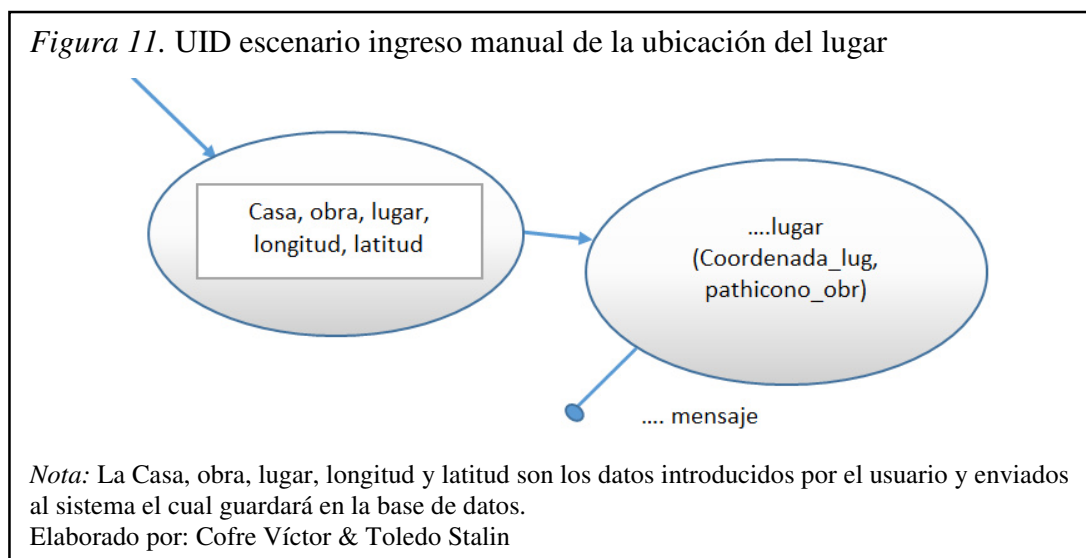


## 2.4. Especificación de UIDs

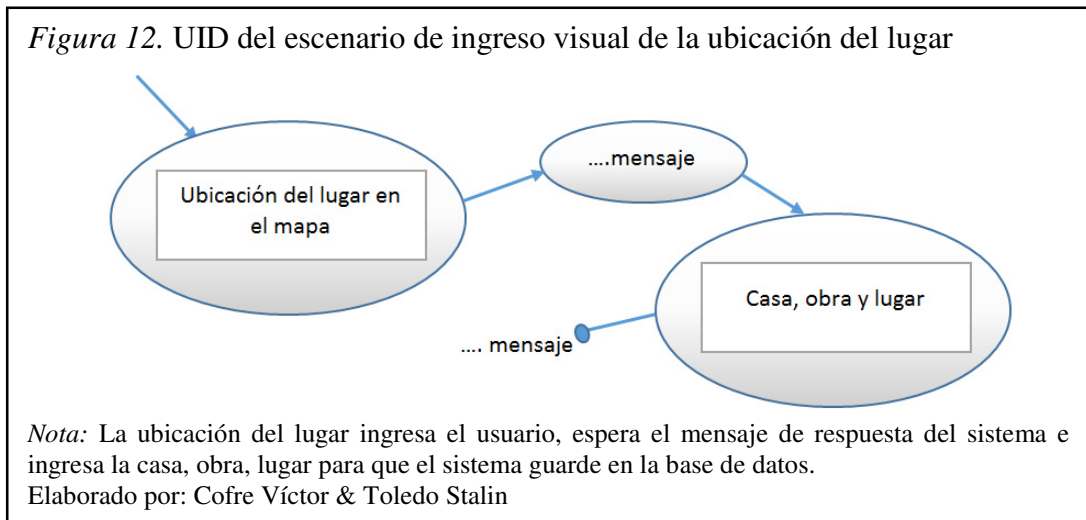
UID (User Interaction Diagrams) o en español (Diagramas de interacción de usuario) con este diagrama se representa la interacción entre el usuario y el sistema. Este diagrama describe el intercambio de información entre el sistema y el usuario sin tomar en cuenta los aspectos específicos de la interfaz del usuario y se parte de un caso de uso para generar el UID. (Vilain, Schwabe, & Sieckenius de Souza, 2000)

- **UID Modulo de gestión de datos geográficos**

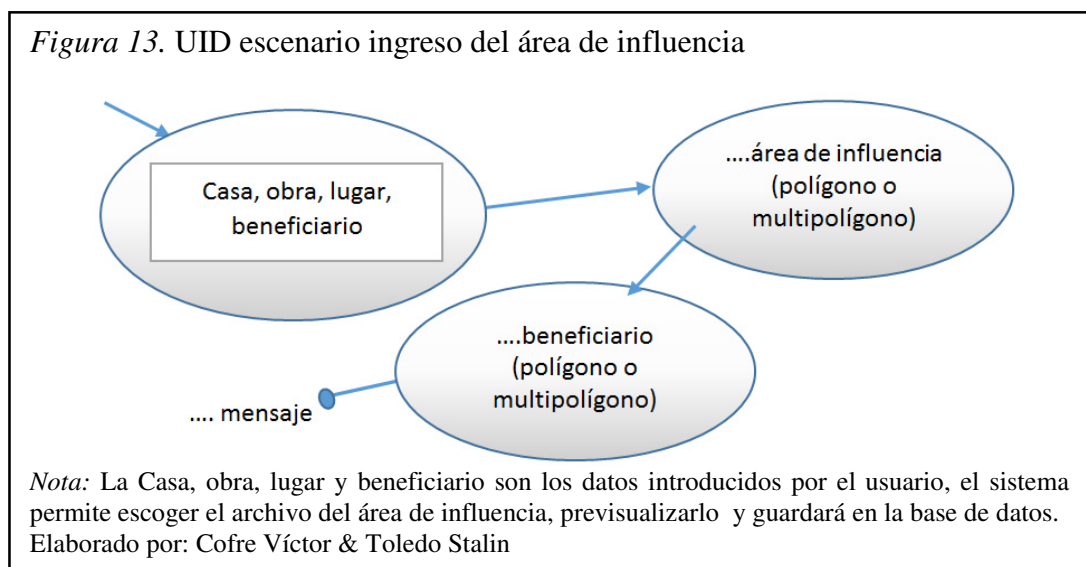
El UID del escenario de ingreso manual de la ubicación del lugar muestra el intercambio de información entre el usuario y el sistema desde el momento que el usuario ingresa datos, hace una pre visualización del lugar en el mapa hasta que estos datos son guardados en la base de datos.



El UID del escenario de ingreso visual de la ubicación del lugar muestra el intercambio de información entre el usuario y el sistema desde el momento de escoger el lugar donde se encuentra ubicada la obra en el mapa, el ingreso de datos hasta que estos sean guardados en la base de datos.

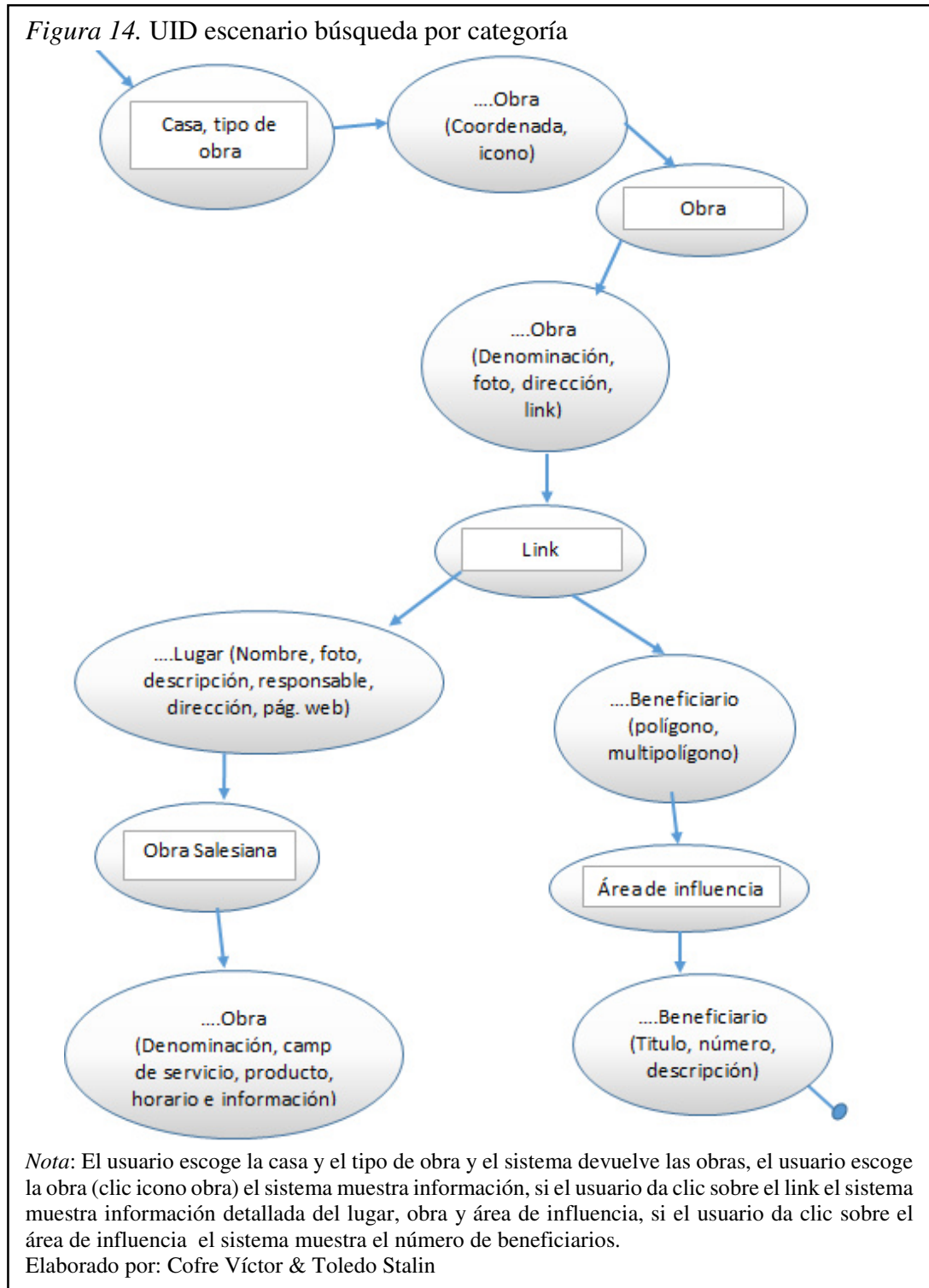


El UID del escenario de ingreso del área de influencia, muestra el intercambio de información entre el usuario y el sistema desde el momento de asignar el área de influencia que tendrá la obra hasta el momento de insertar los datos en la base de datos.



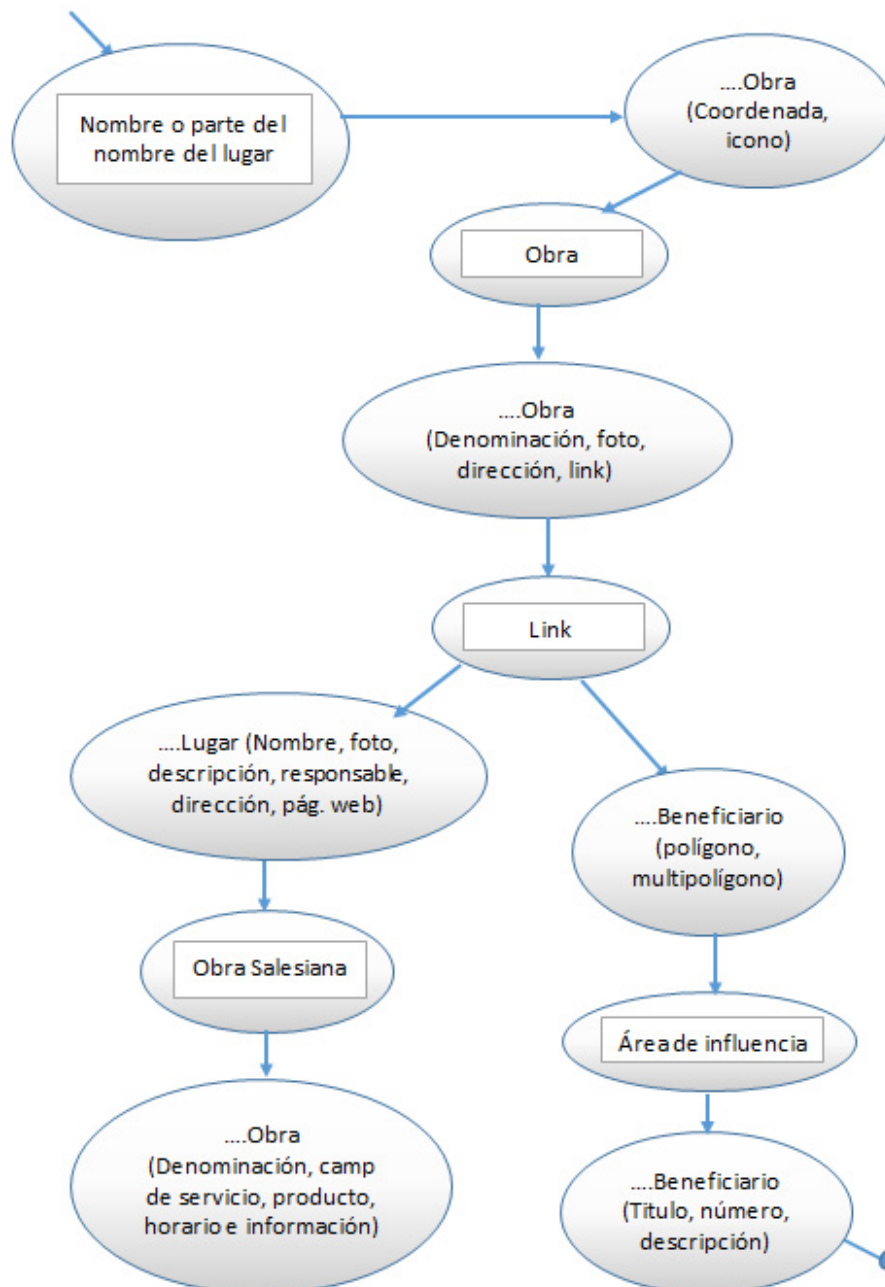
- **UID Modulo de visualización**

El UID del escenario de búsqueda por categoría, muestra el intercambio de información desde el momento que el usuario seleccione la casa y el tipo de obra que quiere ver hasta que el sistema muestre la información.



El UID del escenario de búsqueda temática, muestra el intercambio de información desde el momento que el usuario ingresa el nombre del lugar o parte del nombre hasta que el sistema muestre la información.

Figura 15. UID escenario búsqueda temática



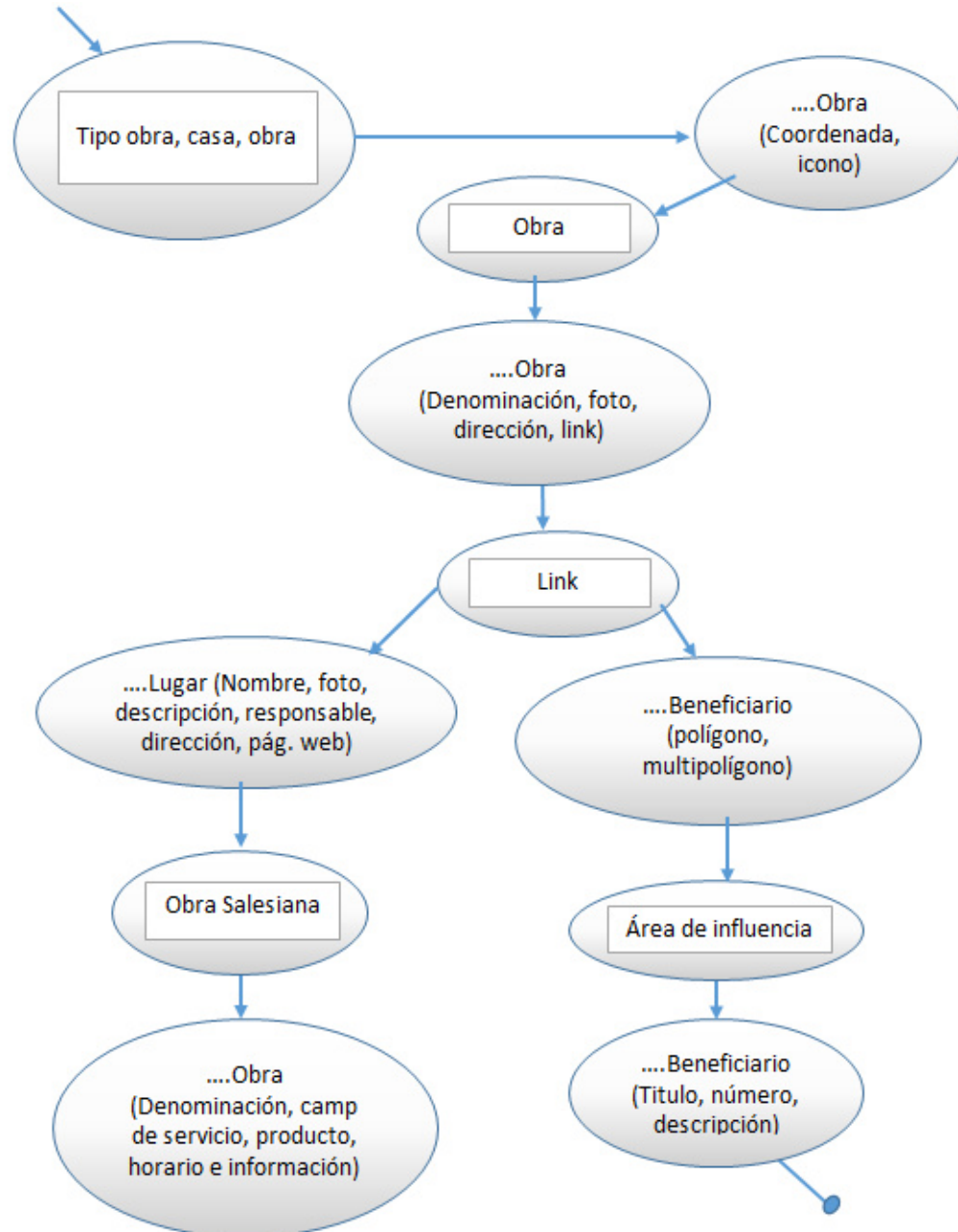
*Nota:* El usuario ingresa el nombre del lugar y el sistema devuelve las obras que están en ese lugar, el usuario escoge la obra (clic icono obra) el sistema muestra información, si el usuario da clic sobre el link el sistema muestra información detallada del lugar, obra y área de influencia, si el usuario da clic sobre el área de influencia el sistema muestra el número de beneficiarios.

Elaborado por: Cofre Víctor & Toledo Stalin



El UID del escenario de búsqueda obra, muestra el intercambio de información desde el momento que el usuario selecciona en el árbol dinámico el tipo de obra, la casa y la obra que quiere ver hasta que el sistema muestre la información.

Figura 16. UID escenario búsqueda obra

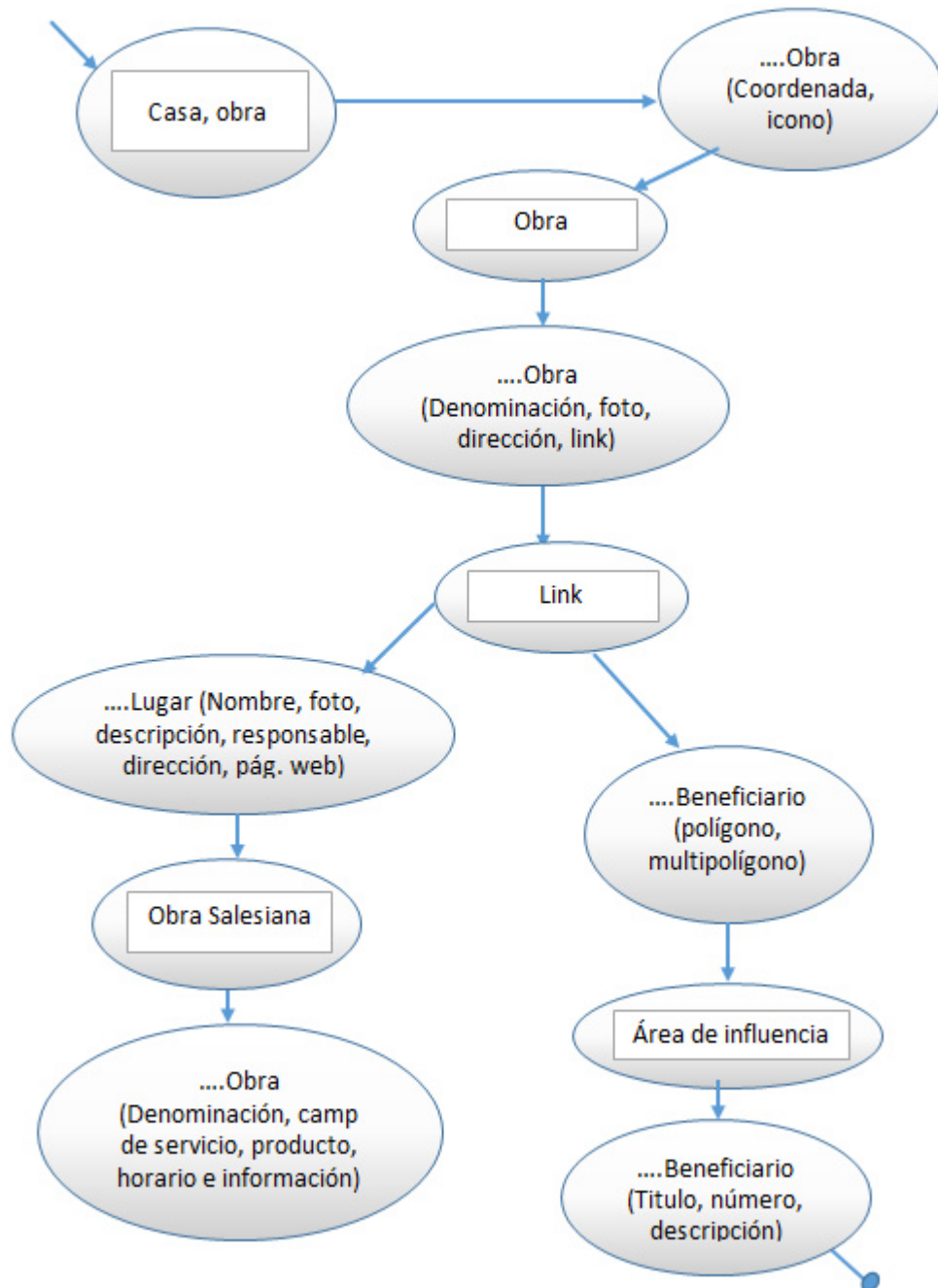


Nota: El usuario selecciona en el árbol dinámico el tipo de obra la casa y la obra y el sistema devuelve las obras que están en ese lugar, el usuario escoge la obra (clic icono obra) el sistema muestra información, si el usuario da clic sobre el link el sistema muestra información detallada del lugar, obra y área de influencia, si el usuario da clic sobre el área de influencia el sistema muestra el número de beneficiarios.

Elaborado por: Cofre Víctor & Toledo Stalin

El UID del escenario de búsqueda casa-obra, muestra el intercambio de información desde el momento que el usuario selecciona en el árbol dinámico la casa y la obra que quiere ver hasta que el sistema muestre la información.

Figura 17. UID escenario búsqueda casa-obra

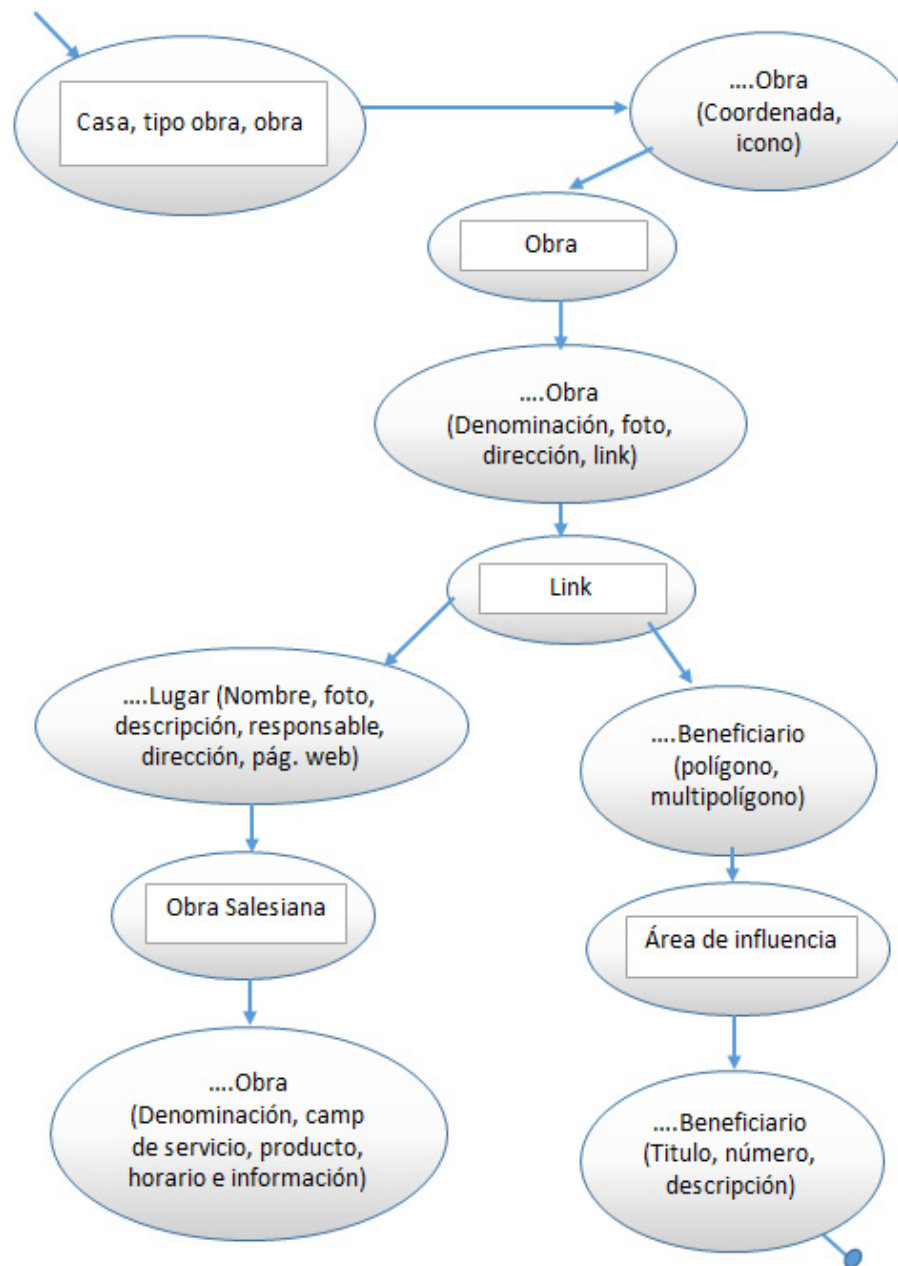


*Nota:* El usuario selecciona en el árbol dinámico el tipo de obra la casa y la obra y el sistema devuelve las obras que están en ese lugar, el usuario escoge la obra (clic icono obra) el sistema muestra información, si el usuario da clic sobre el link el sistema muestra información detallada del lugar, obra y área de influencia, si el usuario da clic sobre el área de influencia el sistema muestra el número de beneficiarios.

Elaborado por: Cofre Víctor & Toledo Stalin

El UID del escenario de búsqueda casa-tipo-obra, muestra el intercambio de información entre el usuario y el sistema desde el momento que selecciona en el árbol dinámico la casa, el tipo de obra y la obra que quiere ver hasta observar los datos de la obra, lugar y beneficiario.

Figura 18. UID escenario búsqueda casa-tipo-obra



*Nota:* El usuario selecciona en el árbol dinámico la casa, el tipo de obra y la obra y el sistema devuelve las obras que están en ese lugar, el usuario escoge la obra (clic icono obra) el sistema muestra información, si el usuario da clic sobre el link el sistema muestra información detallada del lugar, obra y área de influencia, si el usuario da clic sobre el área de influencia el sistema muestra el número de beneficiarios.

Elaborado por: Cofre Víctor & Toledo Stalin

## **CAPÍTULO 3**

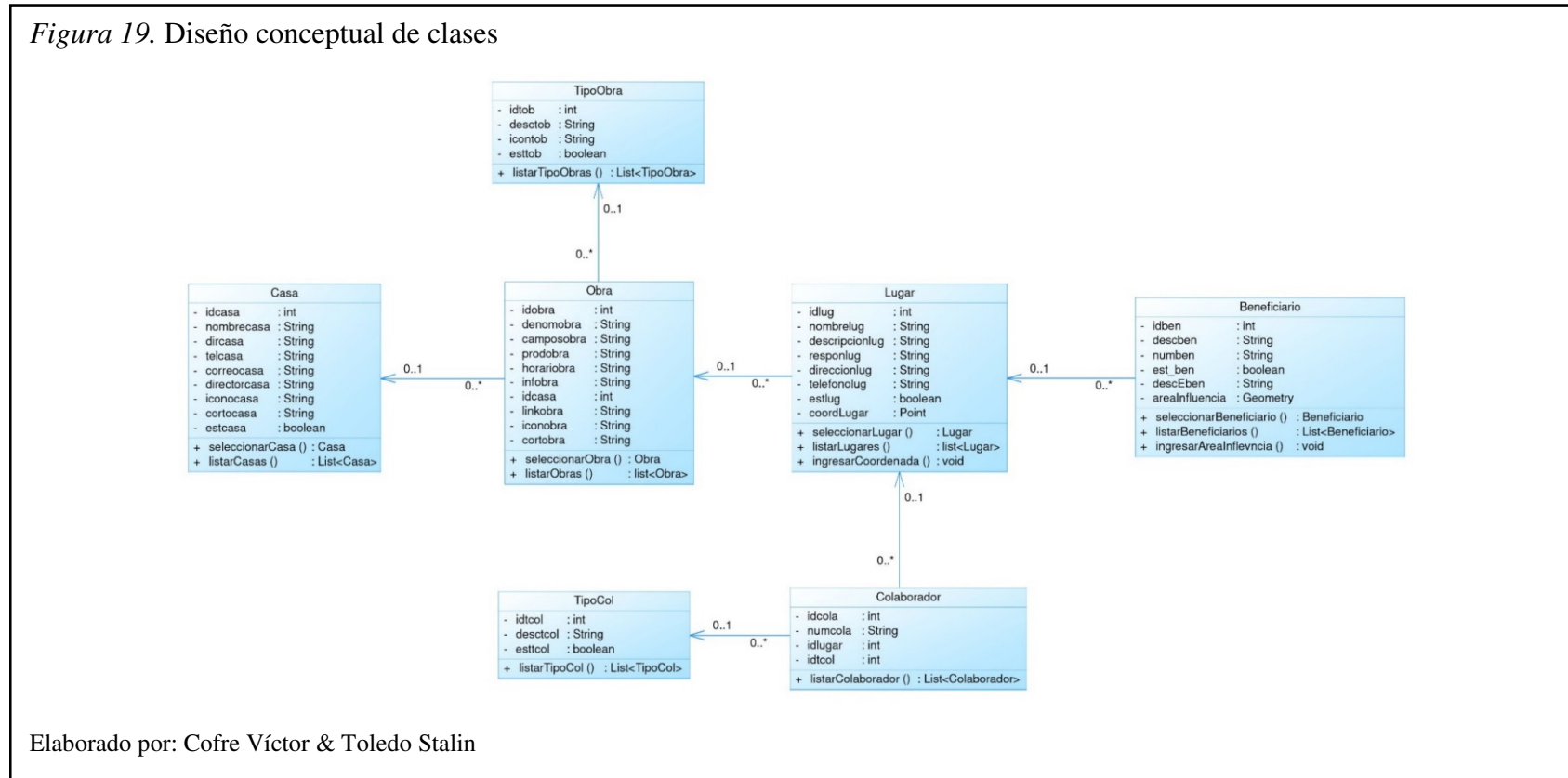
### **DISEÑO**

En este capítulo se diseñará los componentes para la aplicación como los diagramas de clases, de base de datos, de secuencia entre otros. Para esto se parte de los requerimientos del software que son analizados y así obtener una descripción de la estructura interna del software que servirá como base para su construcción. Este diseño debe proporcionar una completa idea de lo que es el software, enfocado en el manejo de datos, funcionalidad y comportamiento desde el punto de vista de la implementación.

También se considera importante que al momento del diseño de la interfaz sea amigable, fácil de usar para que el usuario pueda familiarizarse con la aplicación lo más rápido posible.

### 3.1. Diseño conceptual de clases

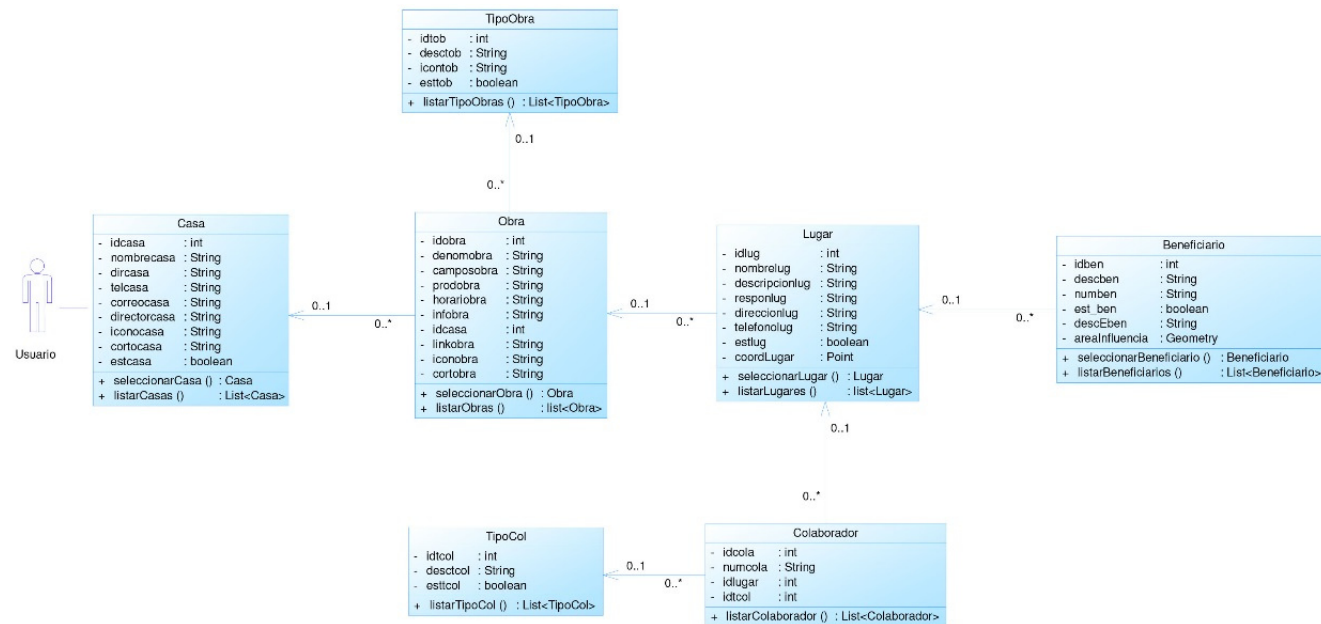
El esquema conceptual representa las clases, relaciones y colaboraciones que el sistema de gestión geográfica y el visualizador tendrán, debido a que el proyecto usa jsf el diagrama de clases representa las clases que interactuarán en el proyecto como entidades.



### 3.2. Diseño navegacional

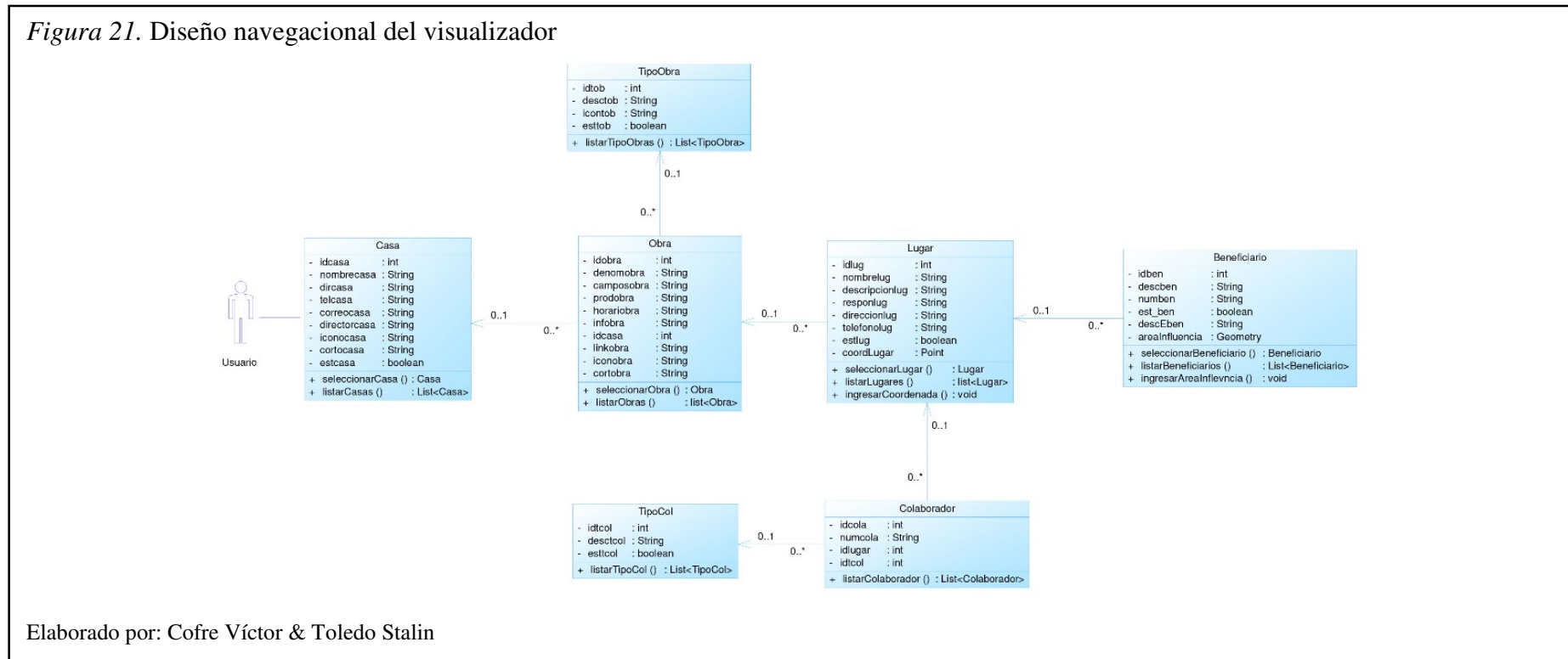
El modelo navegacional perteneciente al gestor de datos geográficos, permite ver los objetos que serán navegados por el usuario y cuál es la relación existente entre los mismos y los atributos que los componen mediante la definición de nodos y enlaces. También se puede observar en qué contexto el usuario navegara por los diferentes objetos al momento de ingresar el dato geográfico correspondiente al área de influencia como los puntos de ubicación de los distintos lugares.

Figura 20. Diseño navegacional de la gestión de datos geográficos



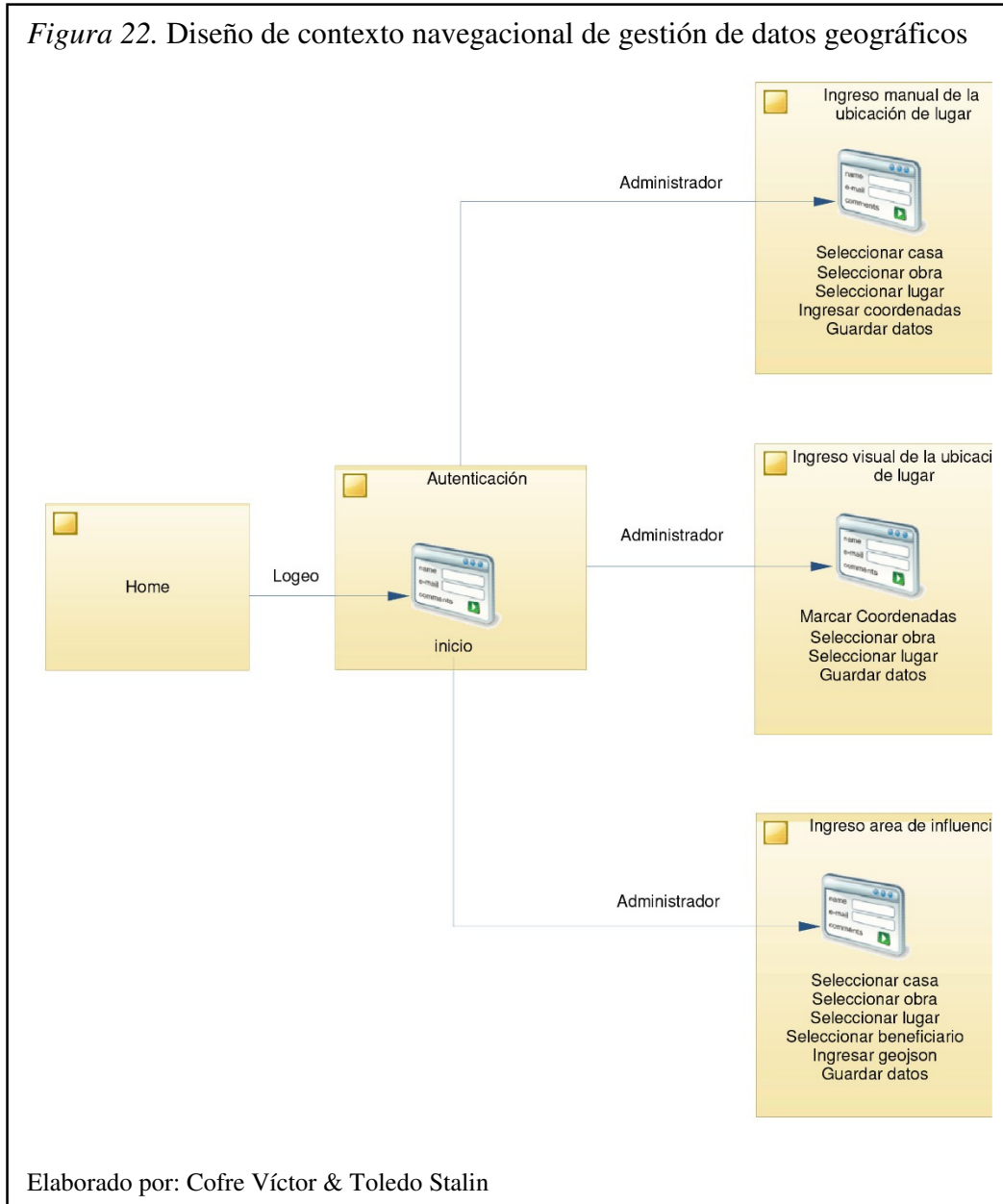
Elaborado por: Cofre Víctor & Toledo Stalin

El modelo navegacional perteneciente al visualizador, permite ver los objetos que serán navegados por el usuario y cuál es la relación existente entre los mismos y los atributos que los componen mediante la definición de nodos y enlaces. También se puede observar en qué contexto el usuario navegara por los diferentes objetos al momento de visualizar los distintos tipos de búsqueda de la información de las distintas Obras Salesianas y los lugares en donde se ubican y la respectiva información e información geográfica.



### 3.3. Diseño de contexto navegacional

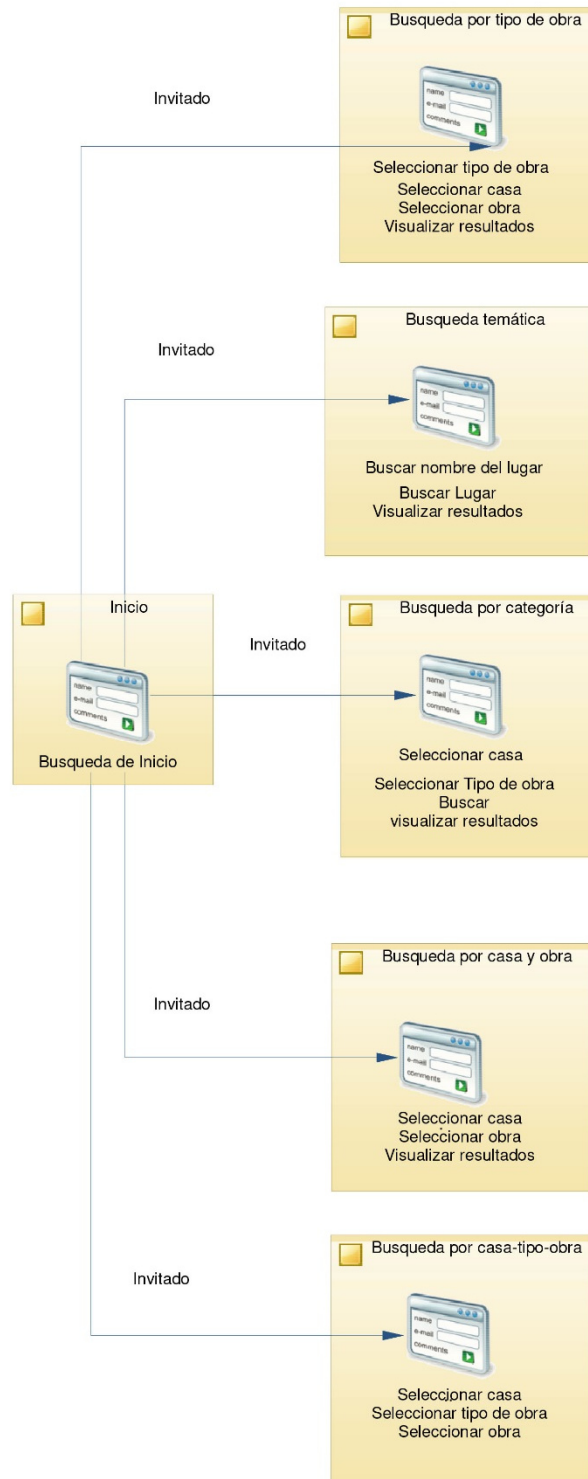
El diagrama muestra las maneras en la que se puede acceder a los distintos nodos existentes en el módulo de gestión de datos geográficos.





El diagrama muestra las maneras en la que se puede acceder a los distintos nodos existentes en el módulo de visualización.

Figura 23. Diseño de contexto navegacional del visualizador



Elaborado por: Cofre Víctor & Toledo Stalin

### 3.4. Diseño de interfaz abstracta

ADV son interfaces representadas por diferentes elementos con los que interactuará el usuario, además contienen los elementos claves que visualizará el usuario al acceder a las diferentes pantallas.

- **Módulo de gestión de datos geográficos**

Sirve para definir la interfaz del usuario para la aplicación hipermedia cabe recalcar que en esta parte el usuario tendrá que ser administrador para acceder a estas páginas. El diseño de esta interfaz se utilizará para el ingreso del lugar dando la opción al usuario que digite la longitud y latitud del lugar.

Figura 24. ADV's Pantalla ingreso manual de la ubicación del lugar

The screenshot shows a web browser window titled "Geoportal Salesiano". The address bar contains "http://ide.ups.edu.ec:8080/TesisP/Asigna". Below the address bar is a banner area. A navigation menu includes "Home", "Ubicación lugar", "Ubicación beneficiario", and "Logout". Below the menu is a prompt: "INGRESE LA COORDENADA DEL LUGAR QUE SE MOSTRARA EN EL MAPA". The main form area contains three dropdown menus labeled "Id Casa", "Id Obra", and "Id Lugar", each with a "Seleccione" option. Below these are input fields for "Longitud" and "Latitud" with the prompts "Ingresar longitud" and "Ingresar latitud". There are two buttons: "Agregar punto" and "GUARGAR". Below the form is a map area with a "Map Location" label. At the bottom of the page, there is a "Connected" status bar and a globe icon.

*Nota:* En el diseño de interfaz de la pantalla de ingreso manual de la ubicación del lugar se crea combos para casa, obra y lugar para que el usuario tenga facilidad al ingresar los datos, además tiene input para el ingreso de longitud y latitud. Cuenta con un botón para agregar la marca en el mapa que representará las coordenadas ingresadas y un botón que guarda la información en la base de datos.

Elaborado por: Cofre Víctor & Toledo Stalin

El diseño de esta interfaz se utilizará para el ingreso del lugar dando la opción al usuario que busque la ubicación en el mapa digital.

Figura 25. ADV's Pantalla ingreso visual de la ubicación del lugar

The wireframe shows a web browser window titled "Geoportal Salesiano" with the URL "http://ide.ups.edu.ec:8080/TesisP/Asigna". The interface includes a navigation bar with "Home", "Ubicación lugar", "Ubicación beneficiario", and "Logout". Below this is a prompt "SELECCIONE UN LUGAR EN EL MAPA" and a map area labeled "Map Location" showing "1st Street" and "2nd Street". The form contains three dropdown menus: "Id Casa" (Selecione casa), "Id Obra" (Selecione obra), and "Id Lugar" (Selecione lugar), followed by a "GUARGAR" button. A placeholder for an image is labeled "Imagen". The status bar at the bottom shows "Connected".

*Nota:* En el diseño de interfaz de la pantalla de ingreso visual de la ubicación del lugar se dispone de un mapa en el cual el usuario podrá ingresar la ubicación del lugar buscando en el mapa cuando ya tenga ubicado el lugar podrá poner la marca en el mapa después seleccionará en los combos la casa, obra y lugar. Cuenta con un botón que guarda la información en la base de datos.  
Elaborado por: Cofre Víctor & Toledo Stalin

El diseño de esta interfaz se utilizará para el ingreso del área de influencia dando la opción al usuario que prevismalize el área en el mapa digital.

Figura 26. ADV's Pantalla ingreso del área de influencia

Geoportal Salesiano

http://ide.ups.edu.ec:8080/TesisP/Asigna

Banner

Home Ubicación lugar Ubicación beneficiario Logout

SUBIR SHAPE

Id Casa Seleccione casa | ▾

Id Obra Seleccione obra | ▾

Id Lugar Seleccione lugar | ▾

Id Beneficiario Seleccione beneficiario | ▾

Seleccionar geojson Subir Cancelar

INGRESAR

Imagen

Connected

*Nota:* En el diseño de interfaz de la pantalla de ingreso del área de influencia se crea combos para casa, obra, lugar y el beneficiario para que el usuario tenga facilidad al ingresar los datos, además tiene botones para la selección del archivo que contiene el área de influencia, otro botón que muestra la visualización previa del área de influencia y un botón para cancelar el ingreso del archivo. Cuenta con un botón que guarda la información en la base de datos si el área de influencia es correcta.

Elaborado por: Cofre Víctor & Toledo Stalin

- **Módulo de visualización**

En el diseño de interfaces de este módulo cuenta con una tabla que contiene los iconos que la obra puede tener con su descripción, todo los usuarios podrán acceder a estas pantallas sin necesidad de estar registrados.

En esta pantalla de búsqueda el usuario podrá ver las obras que están en determinada casa y tipo de obra.

*Figura 27. ADV's Pantalla búsqueda categorías*

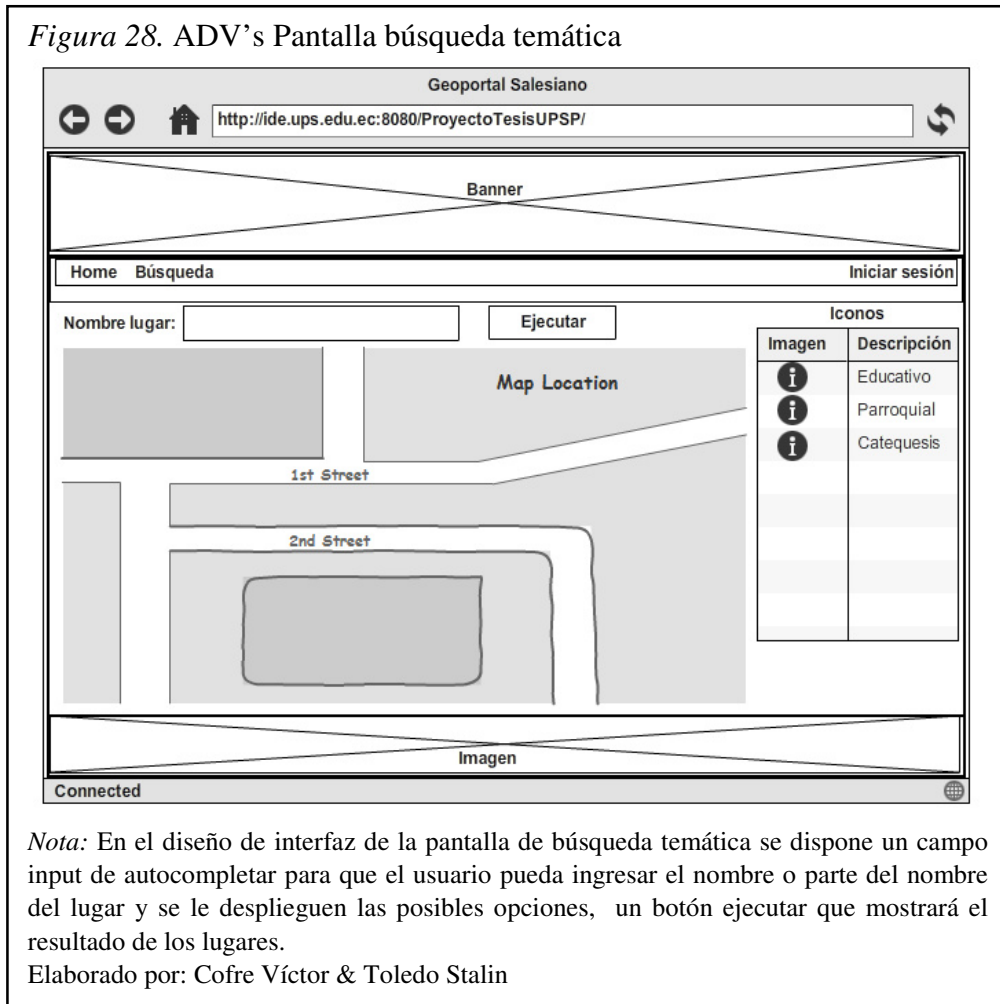


*Nota:* En el diseño de interfaz de la pantalla de búsqueda por categoría se crea combos para que el usuario pueda escoger la casa y dependiendo de la casa que escoja se cargara los elementos para el tipo de obra, además tiene botón ejecutar que mostrará la ubicación de la obra en el mapa, además cuenta con una tabla que contiene los iconos que la obra y su descripción.

Elaborado por: Cofre Víctor & Toledo Stalin

En esta pantalla de búsqueda el usuario podrá ver las obras ingresando el nombre del lugar o parte del nombre.

Figura 28. ADV's Pantalla búsqueda temática

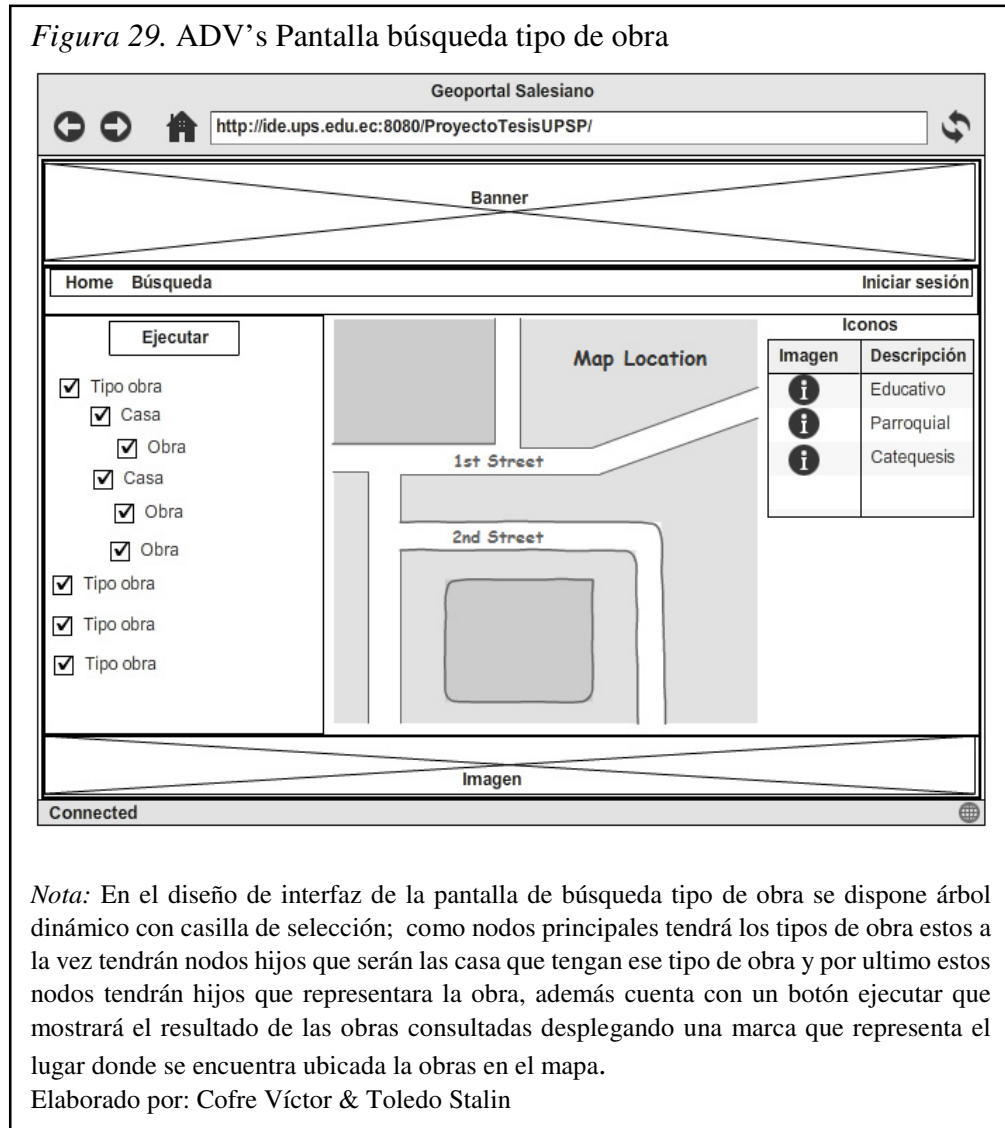


*Nota:* En el diseño de interfaz de la pantalla de búsqueda temática se dispone un campo input de autocompletar para que el usuario pueda ingresar el nombre o parte del nombre del lugar y se le desplieguen las posibles opciones, un botón ejecutar que mostrará el resultado de los lugares.

Elaborado por: Cofre Víctor & Toledo Stalin

En esta pantalla de búsqueda el usuario podrá ver las obras escogiendo en el árbol dinámico el tipo de obra, nombre de la casa y nombre de la obra, puede ser una o varias selecciones a la vez.

Figura 29. ADV's Pantalla búsqueda tipo de obra

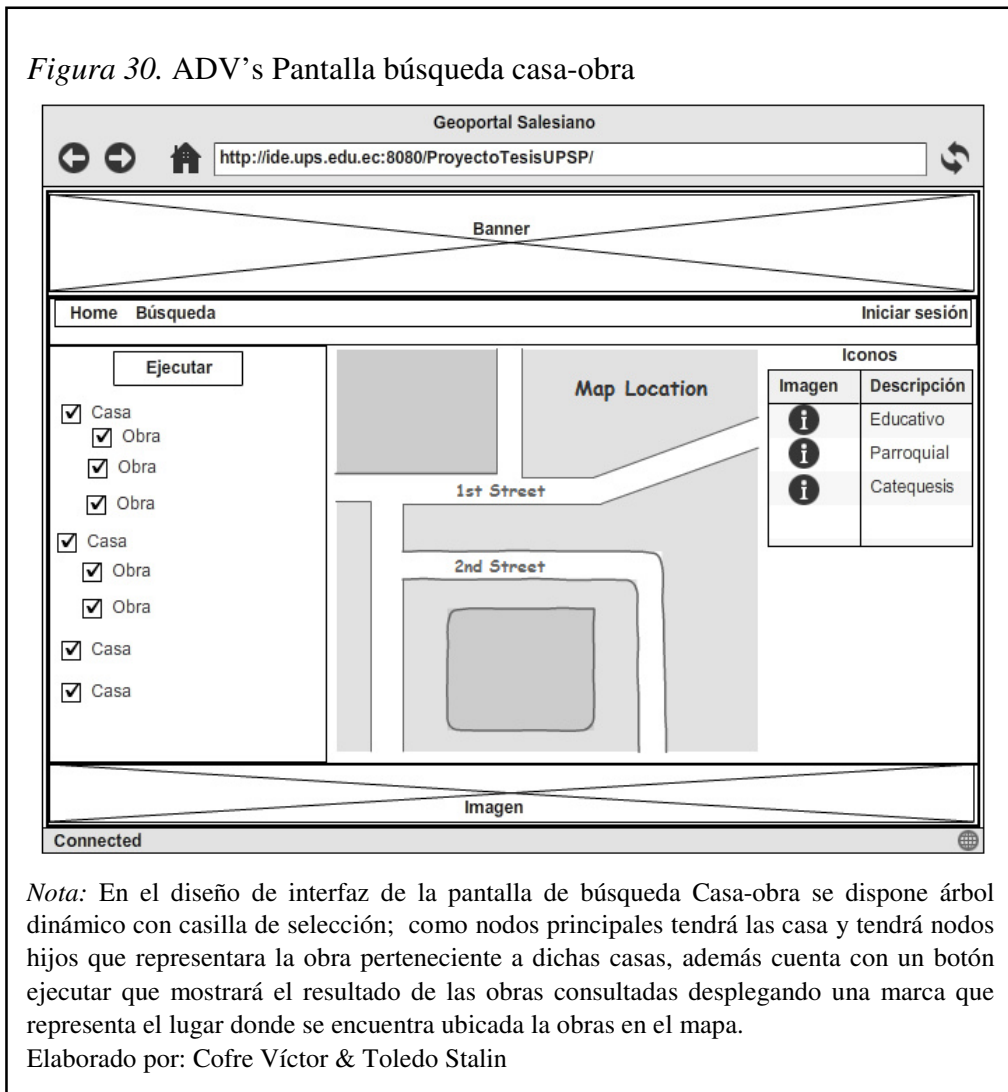


*Nota:* En el diseño de interfaz de la pantalla de búsqueda tipo de obra se dispone árbol dinámico con casilla de selección; como nodos principales tendrá los tipos de obra estos a la vez tendrán nodos hijos que serán las casa que tengan ese tipo de obra y por ultimo estos nodos tendrán hijos que representara la obra, además cuenta con un botón ejecutar que mostrará el resultado de las obras consultadas desplegando una marca que representa el lugar donde se encuentra ubicada la obras en el mapa.

Elaborado por: Cofre Víctor & Toledo Stalin

En esta pantalla de búsqueda el usuario podrá ver las obras escogiendo en el árbol dinámico el nombre de la casa y nombre de la obra, puede ser una o varias selecciones a la vez.

Figura 30. ADV's Pantalla búsqueda casa-obra



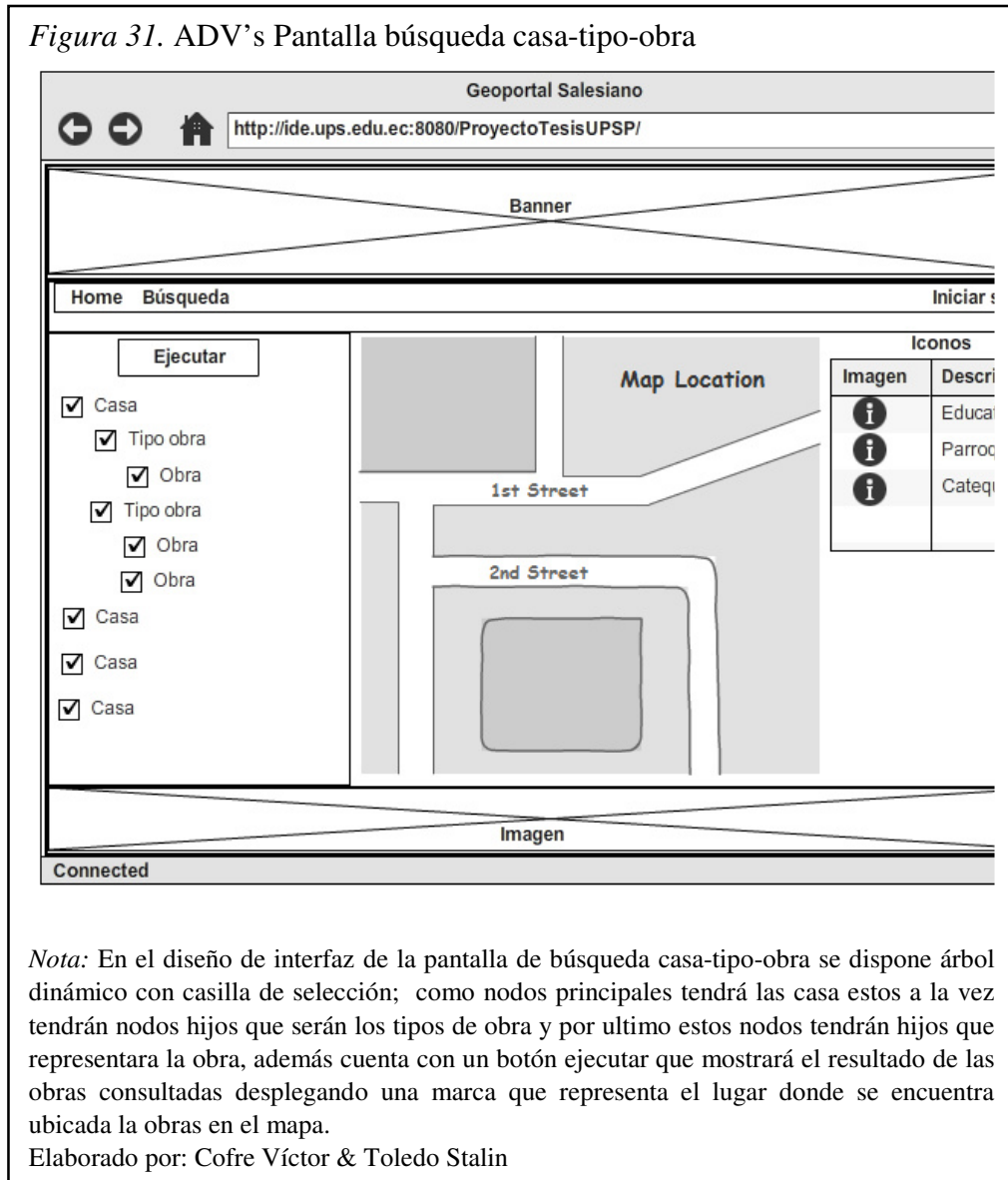
*Nota:* En el diseño de interfaz de la pantalla de búsqueda Casa-obra se dispone árbol dinámico con casilla de selección; como nodos principales tendrá las casa y tendrá nodos hijos que representara la obra perteneciente a dichas casas, además cuenta con un botón ejecutar que mostrará el resultado de las obras consultadas desplegando una marca que representa el lugar donde se encuentra ubicada la obras en el mapa.

Elaborado por: Cofre Víctor & Toledo Stalin



En esta pantalla de búsqueda el usuario podrá ver las obras escogiendo en el árbol dinámico el nombre de la casa, el tipo de obra, y nombre de la obra, puede ser una o varias selecciones a la vez.

Figura 31. ADV's Pantalla búsqueda casa-tipo-obra



### 3.5. Diagrama de clases

Los diagramas de clases siguiente muestran las entidades, controladores y servicios que la aplicación utilizara basado cada controlador en base a los ADVs y los servicios según los requerimientos que cada controlador.

- Diagrama de clases del gestor datos geográficos

A partir de la figura 19. Diseño conceptual de clases y los diagramas ADV's se obtuvo el diagrama de clases definitivo que cuenta con las entidades, controladores y servicios para el sistema.

Figura 32. Diagrama de clases del gestor datos geográficos

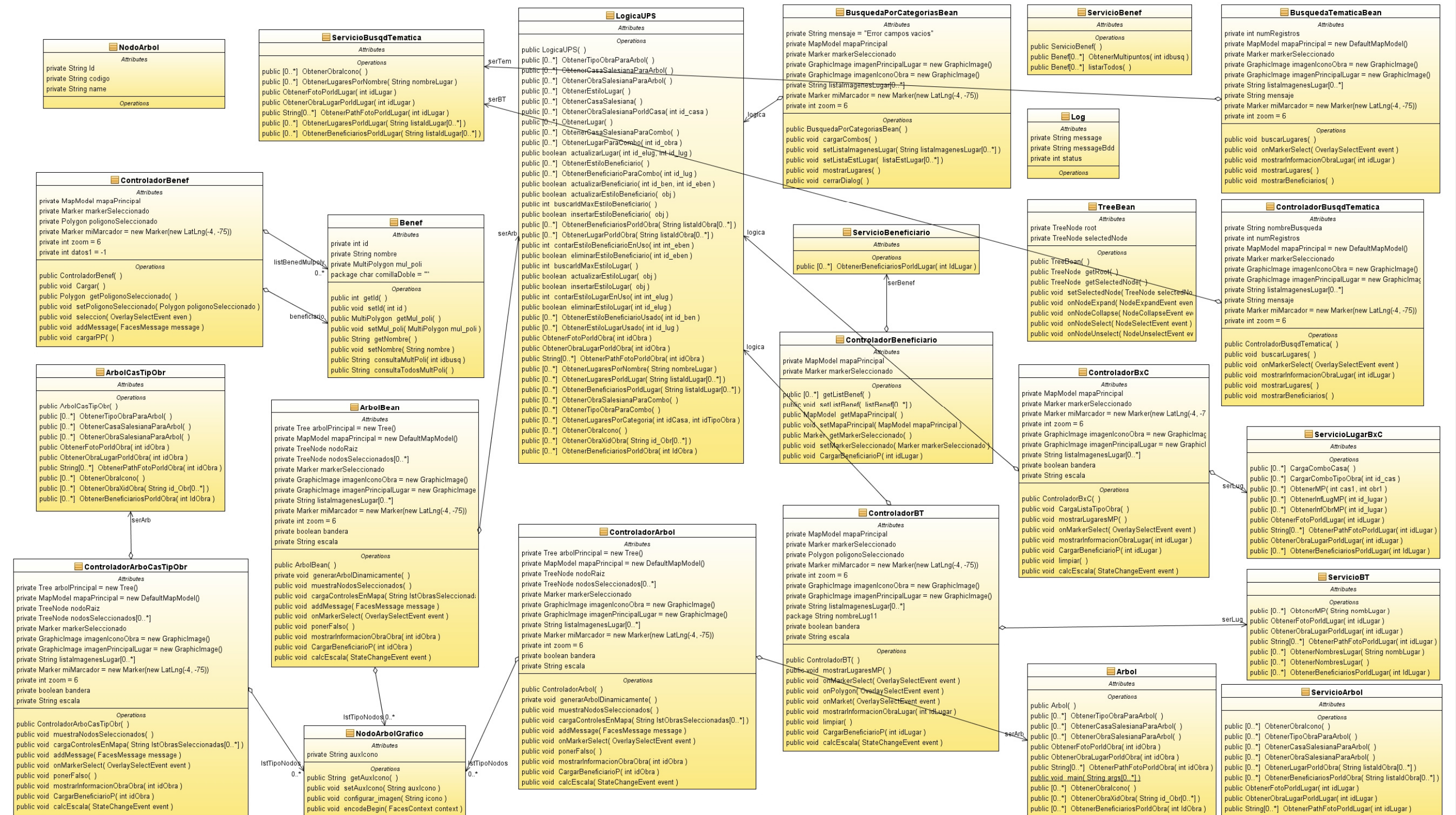


Elaborado por: Cofre Víctor & Toledo Stalin

• **Diagrama de clases del visualizador**

A partir de la figura 19. Diseño conceptual de clases y los diagramas ADV's se obtuvo el diagrama de clases definitivo que cuenta con las entidades, controladores y servicios para el sistema.

Figura 33. Diagrama de clases del visualizador

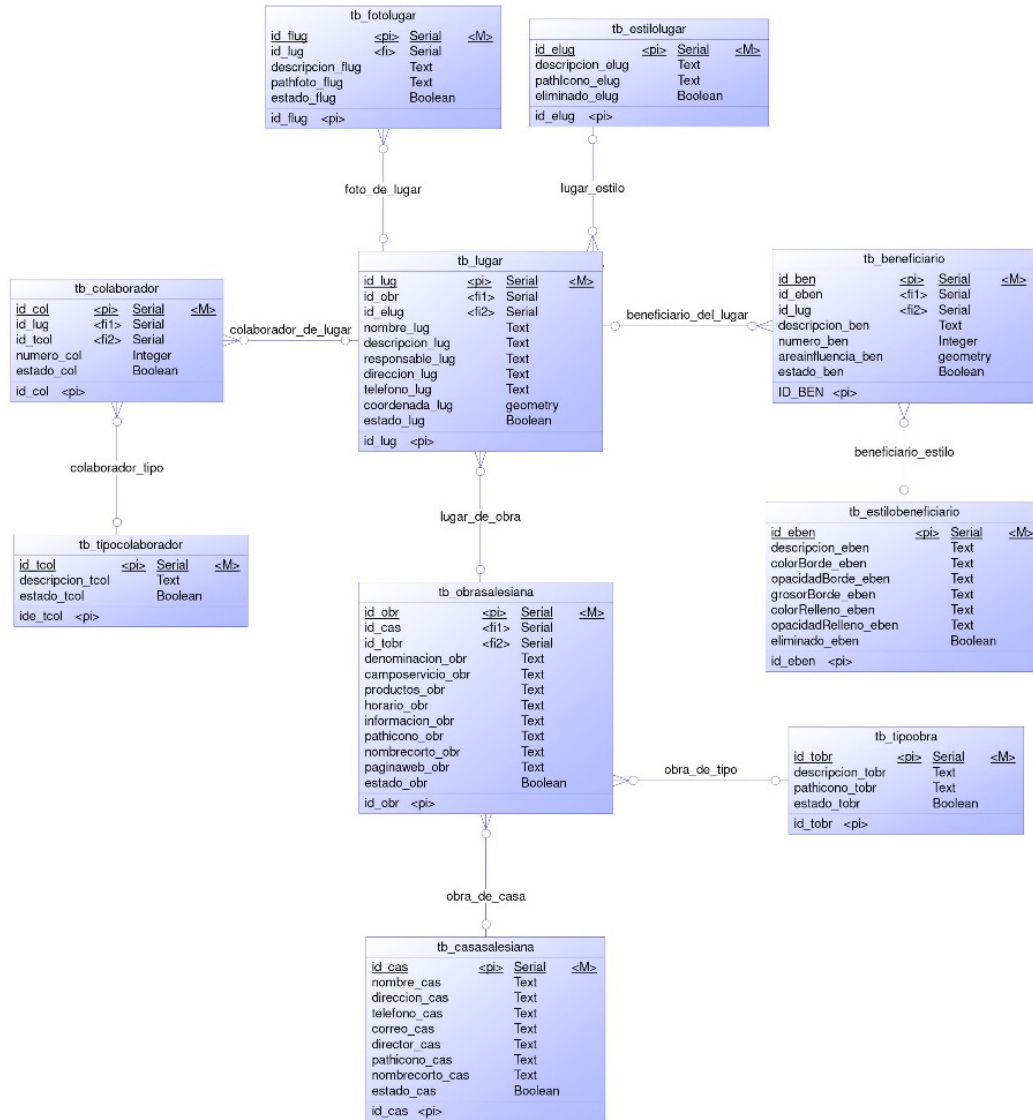


Elaborado por: Cofre Víctor & Toledo Stalin

### 3.6. Modelo de la base de datos

El modelo lógico de la base de datos permite ver la estructura que se va a usar en el ingreso y consulta de la información de las obras salesianas y que servirá tanto para el módulo de gestión de datos geográficos como para el visualizador.

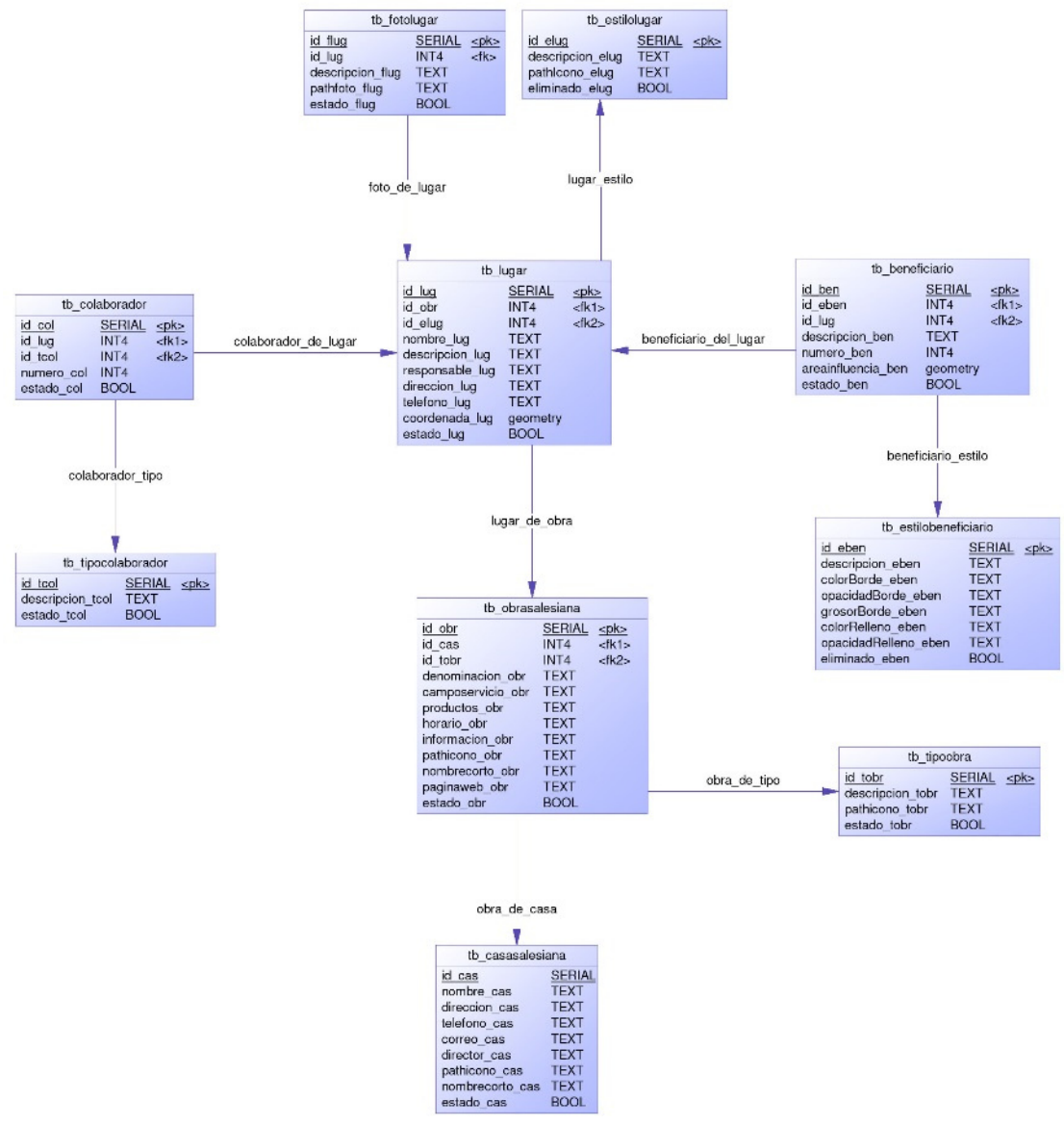
Figura 34. Modelo lógico de la base de datos



Elaborado por: Cofre Víctor & Toledo Stalin

El modelo físico define las relaciones y la estructura de almacenamiento que utilizara en la base de datos para tener un acceso rápido, ordenado y volviendo a la base de datos completamente eficiente.

Figura 35. Modelo físico de la base de datos



Elaborado por: Cofre Víctor & Toledo Stalin

### 3.7. Diccionario de base de datos

En esta sección se muestran los cuadros de las tablas con la descripción de los campos con los que cuenta la base de datos incluyendo información como el nombre de los campos, el tipo de dato y la descripción del tipo de información que dicho campo almacenará.

Uno de los aspectos importantes es que se inició con la base de datos de la versión 1.0 del Geoportal de las Casas Salesianas y se realizó pequeñas modificaciones en la tabla lugar y beneficiario, que contenían restricciones sobre los datos geográficos que podían guardarse en las mismas uno con las coordenadas de los lugares y otra con el área de influencia de los beneficiarios que solo permitía trabajar con puntos y polígonos, con esos cambios podrá trabajar con multipuntos y multipolígonos. La tabla Casa Salesiana contiene un conjunto de datos que representa las características que deberá tener cada casa que almacena el sistema.

Tabla 13

*Diccionario de datos tabla casa salesiana*

<b>Nombre</b>	tb_casasalesiana			
<b>Descripción</b>	Almacena la información básica de las Casas Salesianas			
<b>Primary Key</b>	id_cas			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_cas	serial	Yes	Identifica a la casa a la cual hace referencia
	nombre_cas	text	Yes	Nombre de la casa salesiana
	direccion_cas	text	Yes	Dirección o ubicación de la casa salesiana.
	telefono_cas	text	No	Teléfono de la casa salesiana.
	correo_cas	text	Yes	Correo de la casa salesiana
	director_cas	text	Yes	Director o persona responsable de la casa salesiana.
	pathicono_cas	text	Yes	Es el directorio o path donde se encuentran almacenados todos los iconos disponibles para las casas salesianas.
	nombrecorto_cas	text	Yes	Es una abreviación del nombre de la casa salesiana por motivo de diseño de interfaz.
	estado_cas	boolean	No	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin

La tabla tipo de obra salesiana contiene un conjunto de datos que representa los diferentes tipos a los cuales puede pertenecer una determinada obra.

Tabla 14

*Diccionario de datos tabla tipo de obra salesiana*

<b>Nombre</b>	tb_tipoobra			
<b>Descripción</b>	Almacena la información básica del tipo de obra			
<b>Primary Key</b>	id_tobr			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_tobr	serial	Yes	Identificación única del tipo de obra
	descripcion_tobr	text	Yes	Descripción del tipo de obra
	pathicono_tobr	text	Yes	Se especifica el directorio en el cual se guardarán los iconos disponibles para los tipos de obra.
	estado_tobr	boolean	No	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin

La tabla lugar contiene un conjunto de datos que definirá a un determinado lugar.

Tabla 15

*Diccionario de datos tabla lugar*

<b>Nombre</b>	tb_lugar			
<b>Descripción</b>	Almacena la información del lugar con énfasis en las coordenadas para la obra salesiana asociada a este.			
<b>Primary Key</b>	id_lug			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_lug	serial	Yes	Identificador único del lugar
FK	id_obra	integer	Yes	Representa al indicador de la obra
FK	id_elug	integer	Yes	Representa al indicador del estilo de un lugar.
	nombre_lug	text	Yes	Es el nombre del lugar.
	descripcion_lug	text	Yes	Es la descripción del servicio de la obra
	responsable_lug	text	Yes	Responsable de la obra asociada a este lugar.
	direccion_lug	text	Yes	Dirección del lugar.
	telefono_lug	text	Yes	Número telefónico del lugar.
	coordenada_lug	geometry (puntos y multipuntos)	Yes	Es la ubicación geográfica del lugar
	estado_lug	boolean	No	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin

La tabla obras salesiana contiene un conjunto de datos que definirá a una obra.

Tabla 16

*Diccionario de datos tabla obras salesiana*

<b>Nombre</b>	tb_obrasalesiana			
<b>Descripción</b>	Almacena la información de las Obras Salesianas			
<b>Primary Key</b>	id_obr			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_obr	serial	Yes	Identificador único de la obra salesiana a la cual hace referencia
FK	id_tobr	integer	Yes	Representa al indicador del tipo de obra
FK	id_cas	integer	Yes	Representa al indicador de la casa
	denominacion_obr	text	Yes	Es el nombre de la obra salesiana.
	camposervicio_obr	text	Yes	Es el área de servicio de la obra salesiana
	productos_obr	text	Yes	Son los servicios que ofrece la obra salesiana a la comunidad.
	horario_obr	text	Yes	Es el horario de atención de la obra salesiana.
	informacion_obr	text	Yes	Es la descripción, historia e información de la obra salesiana.
	pathicono_obr	text	Yes	Se especifica el directorio en el cual se guardaran los iconos disponibles para las obras salesianas.
	nombrecorto_obr	text	Yes	Es una abreviación del nombre de la obra salesiana
	paginaweb_obr	text	No	Es la dirección web de la obra salesiana
	estado_obr	boolean	No	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin



La tabla estilo lugar contiene un conjunto de datos que especifica el icono que tendrá un lugar.

Tabla 17

*Diccionario de datos tabla estilo lugar*

<b>Nombre</b>	tb_estiloLugar			
<b>Descripción</b>	Almacena la información básica del estilo del lugar			
<b>Primary Key</b>	id_elug			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_elug	serial	Yes	Identificación única del estilo del lugar
	descripcion_elug	text	Yes	Descripción o nombre del estilo del lugar
	pathIcono_elug	text	Yes	Se especifica el directorio en el cual se guardaran los iconos disponibles para los estilos del lugar.
	eliminado_elug	boolean	Yes	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin

La tabla estilo beneficiario contiene un conjunto de datos que definirá el aspecto que tendrá el área de influencia como el color, grosor del borde, etc.

Tabla 18

*Diccionario de datos tabla estilo beneficiario*

<b>Nombre</b>	tb_estilobeneficiario			
<b>Descripción</b>	Almacena la información de los diferentes estilos que tendrá el beneficiario			
<b>Primary Key</b>	id_eben			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_eben	serial	Yes	Identificador único del estilo que tendrá un beneficiario.
	descripcion_eben	text	Yes	Es una pequeña frase que indica cuales son los beneficiarios
	colorBorde_eben	text	Yes	Es el color que tendrá el borde del área de influencia
	opacidadBorde_eben	text	Yes	Es la intensidad de color que tendrá el borde del área de influencia
	grosorBorde_eben	text	Yes	Es el grosor de la línea de borde del área de influencia.
	colorRelleno_eben	text	Yes	Es el color que tendrá el área de influencia.
	opacidadRelleno_eben	text	Yes	Es la intensidad de color que tendrá el área de influencia.
	eliminado_eben	boolean	Yes	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin

La tabla foto lugar contiene un conjunto de datos que ayuda en la asignación de imágenes para los distintos lugares.

Tabla 19

*Diccionario de datos tabla foto lugar*

<b>Nombre</b>	tb_fotolugar			
<b>Descripción</b>	Almacena la información de fotografías que se asignara al lugar			
<b>Primary Key</b>	id_flug			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_flug	serial	Yes	Identificación única de la foto que se le asigna a un lugar
FK	id_lug	integer	Yes	Representa al indicador del lugar
	descripcion_flug	text	Yes	Es la descripción o nombre de una foto.
	pathfoto_flug	text	Yes	Se especifica el directorio en el cual se guardaran las fotos del lugar.
	estado_flug	boolean	No	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin

La tabla beneficiario contiene un conjunto de datos que especifica el número de beneficiarios que tendrá un área de influencia.

Tabla 20

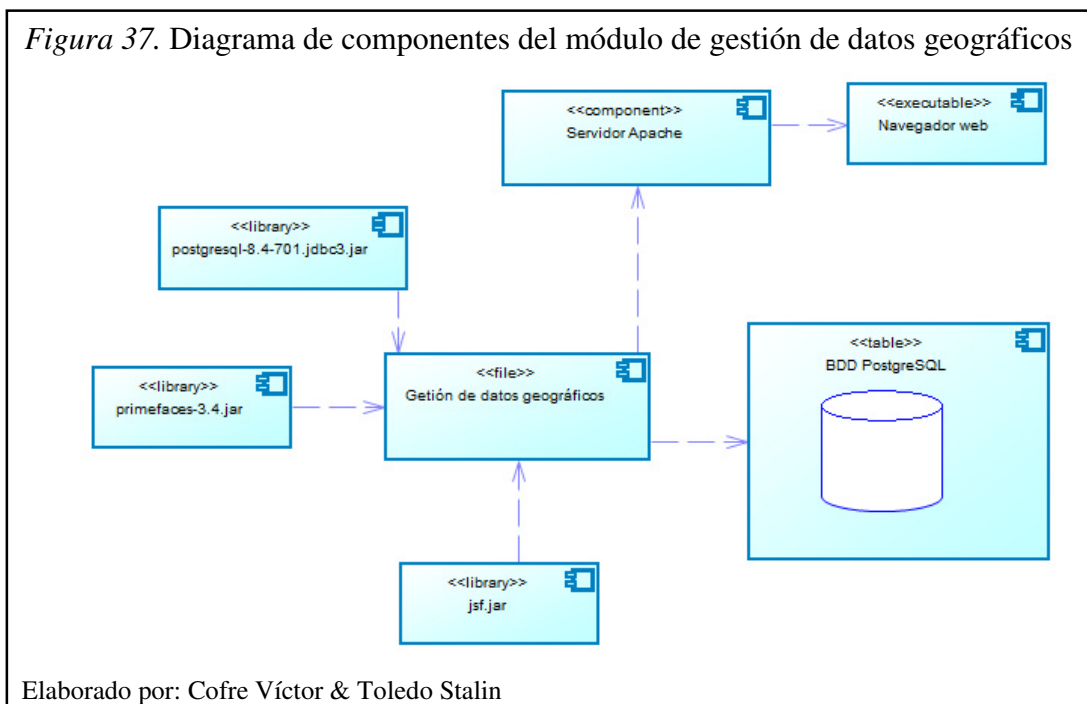
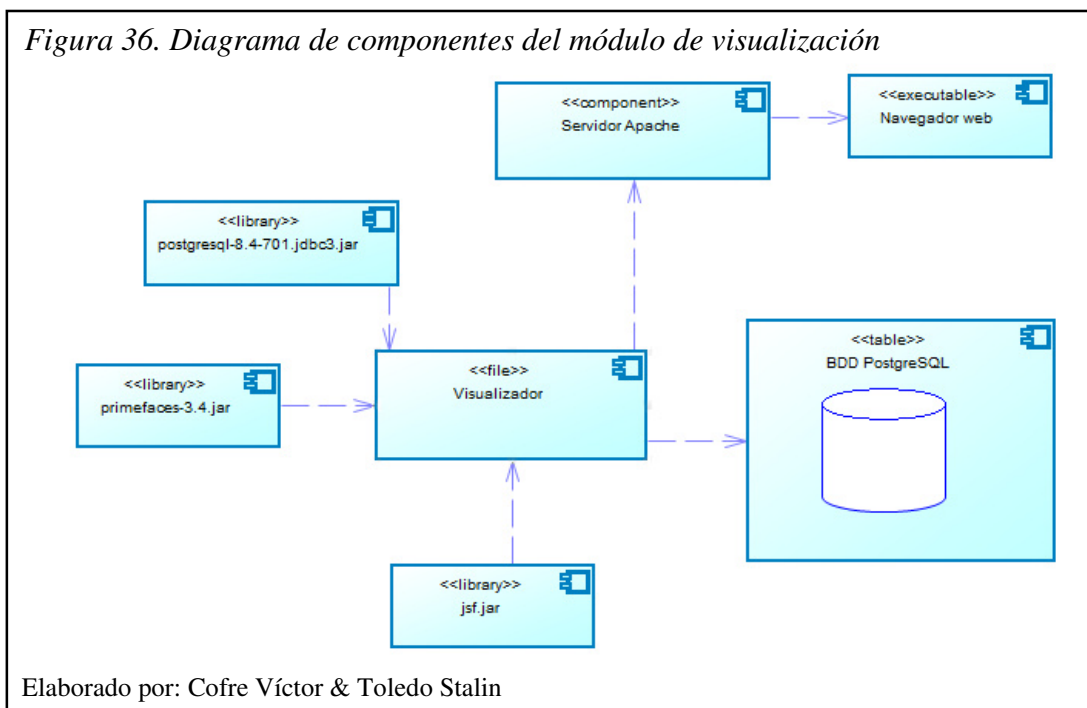
*Diccionario de datos tabla beneficiario*

<b>Nombre</b>	tb_beneficiario			
<b>Descripción</b>	Almacena la información del lugar con énfasis en las coordenadas para la obra salesiana asociada a este			
<b>Primary Key</b>	id_ben			
<b>Key</b>	<b>Column Name</b>	<b>Data Type</b>	<b>Not Null</b>	<b>Descripción</b>
PK	id_ben	serial	Yes	Identificador único del beneficiario
FK	id_eben	integer	Yes	Representa al indicador del estilo del beneficiario
FK	id_lug	integer	Yes	Representa al indicador del lugar
	descripcion_ben	text	Yes	Es una pequeña frase que indica cuales son los beneficiarios
	numero_ben	integer	Yes	Es el número de beneficiarios del área de influencia.
	areainfluencia_ben	geometry (Polígono multipolígono)	Yes	Es el área de influencia representada a través de un polígono multipolígono para cada beneficiario.
	estado_ben	boolean	No	Almacenará un true o false

Elaborado por: Cofre Víctor & Toledo Stalin

### 3.8. Diagrama de componentes

El módulo de visualización y gestor de datos geográficos del Geoportal Salesiano está construido con ayuda de las herramientas libres, el diagrama de componentes muestra los componentes físicos del sistema y las relaciones entre los componentes (elementos de software) que proporcionan o consumen mediante las interfaces. Entre los componentes de software tenemos: Servidor Apache, Java, BDD Postgres, Primefaces, JSF.

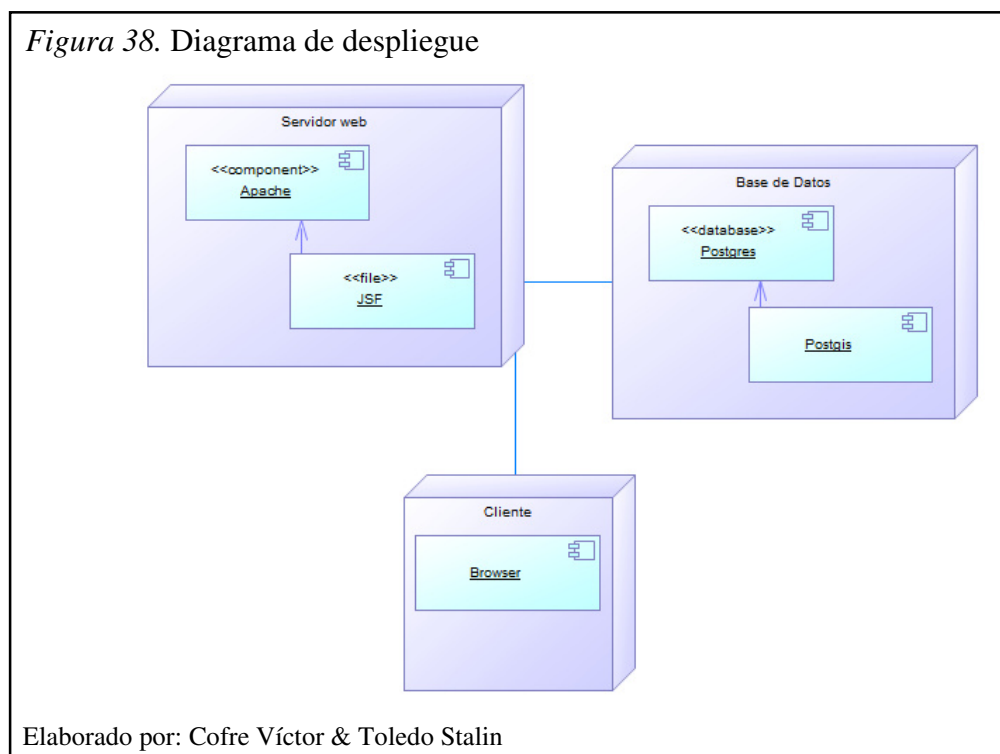


Descripción de los componentes de software:

- **Servidor Apache:** Es un servidor web HTTP de código abierto, multiplataforma que se utiliza para alojar la presente aplicación y que pueda ser utilizada a través de internet.
- **Java:** Lenguaje de programación que trabaja con librerías como primefaces, postgres, jsf, etc que se utiliza para el desarrollo de la aplicación.
- **PostgreSQL:** Almacena la información que la aplicación consumirá para mostrar al usuario la información requerida.
- **Primefaces:** Es una librería que contiene muchos componentes que ayudan en el diseño de una interfaz agradable al usuario y que facilita el desarrollo de aplicaciones web.
- **JSF(Java Server Faces):** Es un tecnología y framework para aplicaciones Java que facilita el desarrollo de interfaces de usuario y dispone de una gran cantidad de componentes.

### 3.9. Diagrama de despliegue

Este diagrama modela la arquitectura cuando se ejecuta el sistema, muestra la configuración de los elementos de hardware (nodos) que contendrá el sistema con sus relaciones.



## CAPÍTULO 4

### DESARROLLO

En esta etapa se desarrolla el código del sistema ofreciéndole la funcionalidad que el usuario espera, aquí solo se expondrá partes importantes del código que se crea conveniente explicar.

#### 4.1. Gestor de datos geográficos

- **Ingreso manual de la ubicación del lugar**

El siguiente fragmento de código método agregarPunto agrega la longitud y latitud al mapa con la finalidad de tener una pre-visualización antes de almacenar en la base de datos, mediante la instancia de un objeto Market colocándolo sobre implementación de un objeto DefaultMapModel que es el objeto que contienen los componente que se visualizan en un mapa digital en la librería de primefaces.

```
public void agregarPunto() {
    emptyModel = new DefaultMapModel();
    pts1 = new LatLng(getLat(), getLng());
    listLatLng.add(pts1);
    for (LatLng ll : listLatLng) {
        marker = new Marker(ll);
        emptyModel.addOverlay(marker);
        centery = marker.getLatLng().getLat();//Latitud
        centerx = marker.getLatLng().getLng();//Longitud
        zoom = 8;//Zoom
    }
}
```

En el método insertarMp se utiliza para guardar los datos del lugar sea un punto o varios puntos, donde el objeto datosm contendrá el id del lugar, el string mp contendrá las coordenadas pertenecientes al lugar y la sentencia Update actualizará los datos en la base.

```
public static void insertarMp(Mapa datosm,String mp) throws Exception{
    cargarDatos();
    String sentencia="UPDATE tb_lugar set coordenada_lug=GeometryFromText" +
        "("+mp+",4326)+"where id_lug="+datosm.getIdl()+";";
    con=new Conexion();
    String mensaje=con.Ejecutar(sentencia);
    con.cerrarConex();
}
```

El siguiente código del método verDatos2 crea la estructura del multipunto que contendrá las coordenadas y el tipo de dato geométrico para poder ingresar a la base de datos. Primero se agrega el encabezado MULTIPOINT en el string dt2, mediante el for se recorre una lista de puntos para agregar las coordenadas al string dt2.

```
// Genera el multipunto
public void verDatos2() {
    dt2 = "MULTIPOINT( ";
    for (int i = 0; i < listLatLng.size(); i++) {
        if (i < listLatLng.size() - 1) {
            dt2 = dt2 + listLatLng.get(i).getLng() + " "
                + listLatLng.get(i).getLat() + ",";
        } else if (i == listLatLng.size() - 1) {
            dt2 = dt2 + listLatLng.get(i).getLng() + " "
                + listLatLng.get(i).getLat();
        }
    }
    dt2 = dt2 + ")";
}
```

Un aspecto importante es el complemento gmap, este tendrá una estructura estándar como se muestra a continuación. Además una de las ventajas que tiene es que trabaja con variables que ayudan a que el mapa sea dinámico. Mayor información en el manual de usuario de Primefaces. (teknoloji, 2014)

```
<p:gmap id="mapIM" center="{mapBeanM2.centery},{mapBeanM2.centerx}"
    zoom="{mapBeanM2.zoom}"
    type="HYBRID" style="width:960px;height:300px"
    model="{mapBeanM2.emptyModel}" />
```

- **Ingreso visual de la ubicación del lugar**

Para este ingreso lo primero que hay que tomar en cuenta es que cuando el usuario da un clic sobre el mapa se dibujará una marca en el mismo por lo que necesitamos tener un atributo extra al momento de definir el componente gmap este atributo es onPonitClick que se ejecuta cuando se hace un clic en el mapa.

OnPonitClick es el evento que permite agregar las marcas cuando se presiona sobre el mapa digital.

```
<p:gmap id="gmap" center="{mapBean.centery},{mapBean.centerx}"
    zoom="{mapBean.zoom}" type="HYBRID"
    style="width:940px;height:300px" model="{mapBean.emptyModel}"
    onPonitClick="handlePonitClick(event);" widgetVar="map" />
```

Se necesita un script para obtener las coordenadas del lugar cada vez que el usuario de un clic en el mapa. Mediante código del script se interactúa con el mapa digital y así obtener las coordenadas en el lugar que se dio el clic con la función `handlePointClick(event)`, mientras que la función `markerAddComplete()` hace una confirmación para agregar la marca al mapa y la función `cancel()` el insertar la marca en el mapa.

```

<script type="text/javascript">
var currentMarker = null;
function handlePointClick(event) {
    if(currentMarker === null) {
        document.getElementById('lat').value = event.latLng.lat();
        document.getElementById('lng').value = event.latLng.lng();
        currentMarker = new google.maps.Marker({
            position:new google.maps.LatLng(event.latLng.lat(),
                event.latLng.lng())
        });
        map.addOverlay(currentMarker);
        dlg.show();
    }
}
function markerAddComplete() {
    currentMarker = null;
    dlg.hide();
}
function cancel() {
    dlg.hide();
    currentMarker.setMap(null);
    currentMarker = null;
    return false;
}
</script>

```

Este método `addMarker` interactúa con las variables de longitud y latitud que posteriormente serán guardados en la base de datos, también cabe destacar que encontramos una lista de puntos que es donde guardaremos las coordenadas si es que el usuario desea guardar más de un punto en el lugar.

```

public void addMarker(ActionEvent actionEvent) {
    marker = new Marker(new LatLng(lat, lng));
    pts1= new LatLng(lat, lng);
    listLatLng.add(pts1);
    emptyModel.addOverlay(marker);
    addMessage(new FacesMessage(FacesMessage.SEVERITY_INFO,
        "Marker Added", "Lat:" + pts1.getLat() + ", Lng:"
            + pts1.getLng()));
}

```

- **Ingreso de área de influencia**

El siguiente fragmento de código muestra el método para subir el archivo GeoJson al servidor para así ser procesado y posteriormente almacenado. El método principalmente hace uso del objeto `FileOutputStream` y al estar el archivo en el servidor permite la respectiva manipulación para obtener la vista previa de la geometría en un mapa virtual.

```
public void escribirEnDisco(byte[] contenido, String nombre){
    path="/home/tomcat/apache-tomcat-6.0.32/webapps/TesisP/sh/";
    FileOutputStream fos = null;
    File f1 = new File(path+nombre);
    path=f1.getAbsolutePath();
    nomb=nombre;
    try {
        fos = new FileOutputStream(f1);
        fos.write(contenido);
        fos.flush();
        verMapa();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Para obtener el o los datos geométricos el archivo GeoJSON debe ser subido al servidor con anterioridad con lo cual mediante el uso del objeto `FileInputStream`, procedemos a abrir el archivo y así obtener en una sola fila de texto el contenido. Obtenido el contenido y almacenado en un elemento `String` se prosigue a procesar la estructura del GeoJSON (Ver. Figura 39. Ejemplo estructura GeoJson) y crear la geometría que se almacenara y permitirá la previsualización.

```
public String convertirGeoJson(){
    String contGeoJson="";
    try{
// Abrimos el archivo
        FileInputStream fstream = new FileInputStream(path);
        // Creamos el objeto de entrada
        DataInputStream entrada = new DataInputStream(fstream);
        // Creamos el Buffer de Lectura
        BufferedReader buffer = new BufferedReader(new
        InputStreamReader(entrada));
        String strLinea;
        // Leer el archivo linea por linea
        while ((strLinea = buffer.readLine()) != null) {
            // Imprimimos la línea por pantalla
            contGeoJson= contGeoJson+strLinea;
        }
        // Cerramos el archivo
        entrada.close();
    }catch (Exception e){ //Catch de excepciones
        System.err.println("Ocurrio un error: " + e.getMessage());
    }
}
```



```

    }
    // inicio de la conversion
    if (contGeoJson.indexOf("MultiPolygon") != -1){
    //1 Buscar [ [ [ [
    int pInicial=contGeoJson.indexOf("[ [ [ [");
    //2 Buscar ] } ] ]
    int pFinal=contGeoJson.indexOf(" ] } ] ]");
    //3 Obtener datos geometricos quitando un corchete del principio y
    uno del final
    contGeoJson=contGeoJson.substring(pInicial+2, pFinal-2);
    //4 Reemplazar ], [
    contGeoJson = contGeoJson.replace(" ], [ ", ", ");
    //5 Reemplazar , (coma con espacio despues)
    contGeoJson = contGeoJson.replace(", ", " ");
    //6 reemplazar [ por ( y ] por )
    contGeoJson = contGeoJson.replace("[", "(");
    contGeoJson = contGeoJson.replace("]", ")");
    contGeoJson = contGeoJson.replace("( ", "(");
    contGeoJson = contGeoJson.replace(" )", ")");
    //7 agregar type al text
    contGeoJson = "MULTIPOLYGON"+contGeoJson;
    } else{
    //1 Buscar [ [ [
    int pInicial=contGeoJson.indexOf("[ [ [");
    //2 Buscar ] } ] ]
    int pFinal=contGeoJson.indexOf(" ] } ] ]");
    //3 Obtener datos geométricos quitando un corchete del principio y
    uno del final
    contGeoJson=contGeoJson.substring(pInicial+2, pFinal-2);
    //4 Reemplazar ], [
    contGeoJson = contGeoJson.replace(" ], [ ", ", ");
    //5 Reemplazar , (coma con espacio despues)
    contGeoJson = contGeoJson.replace(", ", " ");
    //6 reemplazar [ por ( y ] por )
    contGeoJson = contGeoJson.replace("[", "(");
    contGeoJson = contGeoJson.replace("]", ")");
    contGeoJson = contGeoJson.replace("( ", "(");
    contGeoJson = contGeoJson.replace(" )", ")");
    //7 agregar type al text
    contGeoJson = "POLYGON"+contGeoJson;
    if (contGeoJson.indexOf("))) } }, {"") != -1) {
contGeoJson = contGeoJson.substring(0, contGeoJson.indexOf("))) }
}, {"") + 2);
    }
    }
    return contGeoJson;
}
}

```

El método actual separa los polígonos en base a la característica requeridas usando la arquitectura de polígonos del dato geométrico, los polígonos se encuentran dentro de un paréntesis “(” y cada polígono tiene sus datos dentro de 2 corchetes “(”.

```

public static List<String> sepPolig(Sting strg) {
    String datos = strg;
    List<String> poligonos = new ArrayList<String>();
    if (datos.indexOf("(((") != -1) {
        datos = datos.substring(datos.indexOf("(((") + 3,
            datos.length() - 3);
    }
}

```

```

        while (datos.indexOf(","), "(" != -1) {
poligonos.add(datos.substring(0, datos.indexOf(","), "("));
        datos = datos.substring(datos.indexOf(","), "(" +
5);
        }
        poligonos.add(datos);
    } else {
        datos = datos.substring(datos.indexOf("(") + 2,
            datos.length() - 2);
        //System.out.println("****dato: "+datos);
        poligonos.add(datos);
    }
    return poligonos;
}
}

```

El método sepCoord obtiene las coordenadas que contiene cada polígono.

Cada polígono tiene las coordenadas de los distintos puntos que forman la geometría y estos se encuentran separados unos de otros con comas “,”.

```

public static List<String> sepCoord(String polig) {
    String dato = polig;
    List<String> coord = new ArrayList<String>();
    while (dato.indexOf(",") != -1) {
        coord.add(dato.substring(0, polig.indexOf(",")));
        dato = dato.substring(dato.indexOf(",") + 1);
    }
    coord.add(dato);
    return coord;
}
}

```

En el fragmento de código se obtiene los polígonos de la geometría y se los almacena en un objeto de tipo polygon que pertenece a la librería de primefaces y permitirá la visualización de los datos geométricos mediante el componente gmap.

```

public static List<Polygon> recuperarPoligonos(String miDato) {
    String datos = miDato;
    List<String> poligonos = new ArrayList<String>();
    List<String> coord = new ArrayList<String>();
    List<Polygon> polygons = new ArrayList<Polygon>();
    List<LatLng> coordXY = new ArrayList<LatLng>();
    Polygon polygon;
    poligonos = sepPolig(datos);
    for (String str : poligonos) {
        coord = (ArrayList<String>) sepCoord(str);
        coordXY = new ArrayList<LatLng>();
        for (String co : coord) {
            coordXY.add(convCoord(co));
        }
        polygon = new Polygon();
        polygon.setStrokeColor("#FF9900");
        polygon.setFillColor("#FF9900");
        polygon.setStrokeOpacity(0.4);
        polygon.setFillOpacity(0.7);
        for (LatLng coordenadas : coordXY) {
            polygon.getPaths().add(coordenadas);
        }
    }
}
}

```

```

    }
    polygons.add(polygon);
}return polygons;
}

```

El ingreso del dato geométrico (polígono o multipolígono) se realiza mediante el uso de la sentencia del complemento postgis. La sentencia permite almacenar el campo geométrico con el SRID 4326 requerido y establecido en la tabla de la base de datos.

```

public static void insertarGeometria(int id, String geometria)
throws Exception{
    if (true) {
String sentencia="UPDATE tb_beneficiario SET areainfluencia_ben =
(select st_geomfromtext('"+geometria+"',4326)) WHERE id_ben =
'"+id+"';";
    con=new Conexion();
String mensaje=con.Ejecutar(sentencia);
con.cerrarConex();
    } else {
    throw new Exception("error al ingresar GEOMETRIA");
    }
}

```

## 4.2. Visualizador

Con respecto al visualizador se expondrá el código importante que permitirá obtener la funcionalidad deseada de la aplicación.

Para esta parte del visualizador el complemento Gmap tiene atributos nuevos como:

- FitBounds: calcula el zoom del mapa de acuerdo a las geometrías y marcas pintadas.
- Ajax: permite ejecutar un método y actualizar componentes.
- GmapInfoWindow: es una ventana de información que puede contener datos e imágenes y es desplegado cuando se da clic sobre una marca o icono.

Por último se trabaja con variable para controlar el zoom y las coordenadas con el fin de que el mapa sea dinámico al momento de hacer las búsquedas.

```

<p:gmap id="mapaPrincipal1"
center="#{controladorBxC.miMarcador.latlng.lat},#{controladorBxC.miMarcador.latlng.lng}"
model="#{controladorBxC.mapaPrincipal}" zoom="#{controladorBxC.zoom}" fitBounds="true"
type="HYBRID" style="height:557px;width: 995px;padding: 0;margin: 0" >
<p:ajax event="stateChange" listener="#{controladorBxC.calcEscala}"
update=":frmEscala:txtEscala"/>
<p:ajax event="overlaySelect" listener="#{controladorBxC.onMarkerSelect}"
update=":frmDatosObraLugar:dlgDatosLugar,:frmDatosObraLugar:dlgDatosBeneficiario,
:frmBusqueda:growl" />

<p:gmapInfoWindow id="infoWindowLugar" rendered="#{controladorBxC.bandera}"
maxWidth="100">
<div align="center">
<table align="center">
<tr>
<td align="center" width="200">
<h:outputText value="#{controladorBxC.markerSeleccionado.title}"

```

Este método obtiene todos los lugares que coincidan con la búsqueda enviada por el usuario y ubica los iconos de las obras en el mapa. Primero se obtienen los lugares y se los almacena en una lista. Mediante el primer for se recorre la lista y se obtiene la latitud y longitud donde se centrara el mapa, con el segundo for se descompone la coordenada del multipunto para obtener la longitud y latitud de cada punto y se agrega al mapa con determinado zoom.

```

public void mostrarLugaresMP () {
    try {
        lstLugarMult =
serLug.ObtenerMP (casaSeleccionado.getId_cas (), tipoObraSeleccionado.g
etId_tobr ());
        mapaPrincipal=new DefaultMapModel ();
        for (Lugar lugar : lstLugarMult) {
            MultiPoint punt= new MultiPoint ();
            punt=lugar.getCoordenada_lugMP ();
Marker marcador = new Marker(new
LatLng(lugar.getCoordenada_lugMP ().getFirstPoint ().getY (),
lugar.getCoordenada_lugMP ().getFirstPoint ().getX ()), lugar.getNombre_
lug (), lugar, lugar.getPathIcono_elug ());
            for(int
i=0;i<lugar.getCoordenada_lugMP ().numPoints ();i++){
                punt.getPoint (i);
                marcador = new Marker (new
LatLng(punt.getPoint (i).getY (), punt.getPoint (i).getX ()),
lugar.getNombre_lug (), lugar, lugar.getPathIcono_elug ());
                marcador.setIcon (lugar.getPathIcono_elug ());
                marcador.setTitle (lugar.getNombre_lug ());
                mapaPrincipal.addOverlay (marcador);
                miMarcador=marcador;
                zoom=16;
            }} catch (Exception e) {
                FacesContext.getCurrentInstance ().addMessage (null, new
FacesMessage (FacesMessage.SEVERITY_ERROR, "Error", "Al cargar los
puntos en el mapa"));
            } limpiar ();
        }
    }
}

```

Este método es llamado cuando se da un clic sobre el icono o el área de influencia que se encuentra en el mapa si es el primer caso desplegara el componente **gmapInfoWindow** para mostrar información de la obra, en el segundo caso despliega un componente **dialog** que mostrara información de los beneficiarios de esa obra.

En el evento lo único que se hace es cargar los objetos obtenidos y mostrar dicha información.

```

public void onMarkerSelect(OverlaySelectEvent event) {
    try {
        if (event.getOverlay().getClass().equals(Marker.class)) {
            bandera=true;
            markerSeleccionado = (Marker) event.getOverlay();
            lugarSeleccionado = (Lugar) markerSeleccionado.getData();
            foto=serLug.ObtenerFotoPorIdLugar(lugarSeleccionado.getId_lug());
            imagenPrincipalLugar.setUrl(foto.getPathFoto_flg());
            mostrarInformacionObraLugar(lugarSeleccionado.getId_lug());
            mapaPrincipal.getPolygons().clear();
            CargarBeneficiarioP(lugarSeleccionado.getId_lug());
        }else if(event.getOverlay().getClass().equals(Polygon.class)){
            bandera=false;
            beneficiarioSeleccionado = (Beneficiario) event.getOverlay().getData();
            RequestContext requestContext = RequestContext.getCurrentInstance();
            requestContext.execute("dlgBeneficiario.show()");
        }
    } catch (Exception e) {
        bandera=false;
    }
}
}

```

En este método se obtiene la información que la aplicación desplegara de la obra y del beneficiario.

```

public void mostrarInformacionObraLugar(int idLugar) {
    lugarPrincipal = serLug.ObtenerObraLugarPorIdLugar(idLugar);
    imagenIconoObra.setUrl(lugarPrincipal.getObraSalesiana().getPathIcono_obr());
    listaImágenesLugar = serLug.ObtenerPathFotoPorIdLugar(idLugar);
    CargarBeneficiarioP(lugarSeleccionado.getId_lug());
}

```

En este metodo se obtentra la información perteneciente al beneficiario asi como la información con la que agregara el poligono o multipoligono al mapa como el color (setStrokeColor), grosor de la linea del borde (setStrokeWeight), etc. Los if actuan como clasificadores tratando a los objetos dependiendo del campo geométrico que contengan si el multipolígono o polígono.

```

public void CargarBeneficiarioP(int idLugar){
    try {
        listBenef=serLug.ObtenerBeneficiariosPorIdLugar(idLugar);
        for (Beneficiario beneficiario : listBenef) {
            Polygon polygon = new Polygon();
            MultiPolygon mpolygon = new MultiPolygon();
            if (beneficiario.getAreaInfluencia_ben() != null) {
                polygon.setData(beneficiario);
                polygon.setStrokeColor(beneficiario.getEstiloBeneficiario().getColorBorde_eben());
                polygon.setStrokeOpacity(Double.parseDouble(beneficiario.getEstiloBeneficiario().getOpacidadBorde_eben()));
                polygon.setStrokeWeight(Integer.parseInt(beneficiario.getEstiloBeneficiario().getGrosorBorde_eben()));
                polygon.setFillColor(beneficiario.getEstiloBeneficiario().getColorRelleno_eben());
                polygon.setFillOpacity(Double.parseDouble(beneficiario.getEstiloBeneficiario().getOpacidadRelleno_eben()));
            }
        }
    }
}

```

```

        for (int i = 0; i <
beneficiario.getAreaInfluencia_ben().numPoints(); i++) {
polygon.getPaths().add(new
LatLng(beneficiario.getAreaInfluencia_ben().getPoint(i).getY(),
beneficiario.getAreaInfluencia_ben().getPoint(i).getX()));
}
mapaPrincipal.addOverlay(polygon);
}
}
if (beneficiario.getAreaMP() != null) {
mpolygon=beneficiario.getAreaMP();
Beneficiario benTemp = beneficiario;
for(int j=0;j<mpolygon.numPolygons();j++){
benTemp.setAreaInfluencia_ben(mpolygon.getPolygon(j));
Polygon polygon1 = new Polygon();
polygon1.setData(benTemp);
polygon1.setStrokeColor(beneficiario.getEstiloBeneficiario().getColorBorde_eben());
polygon1.setStrokeOpacity(Double.parseDouble(beneficiario.getEstiloBeneficiario().getOpacidadBorde_eben()));
polygon1.setStrokeWeight(Integer.parseInt(beneficiario.getEstiloBeneficiario().getGrosorBorde_eben()));
polygon1.setFillColor(beneficiario.getEstiloBeneficiario().getColorRelleno_eben());
polygon1.setFillOpacity(Double.parseDouble(beneficiario.getEstiloBeneficiario().getOpacidadRelleno_eben()));
for (int i = 0; i <
beneficiario.getAreaInfluencia_ben().numPoints(); i++) {
polygon1.getPaths().add(new
LatLng(beneficiario.getAreaInfluencia_ben().getPoint(i).getY(),
beneficiario.getAreaInfluencia_ben().getPoint(i).getX()));
}
mapaPrincipal.addOverlay(polygon1);
}
}
}
} catch (Exception e) {
FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "Lamentamos informarle",
"Que aun no existen datos de los beneficiarios
de este lugar "));
}
}
}

```

### 4.3. Estructura GeoJson para ingresar el área de influencia

El archivo GeoJson que procesa el actual gestor de datos geográficos tiene la estructura de tipo “FeatureCollection” dada por la herramienta Quantum GIS, la cual contiene en su interior un array de “features”. Cada Feature del array se encuentra entre llaves “{ }” y lo primero que se identifica es el tipo mediante las líneas de código "type": "Feature" en el cual se observa que es de tipo feature el cual tendrá un id, propiedades y la estructura geometry.

El ingreso del área de influencia requiere que la estructura geometry se cumpla y por necesidad exista en el archivo GeoJson caso contrario no se podrá procesar.

La estructura geométrica se identifica por el atributo `geometry` que contiene el `feature` y los datos requeridos que se encuentran dentro de 2 llaves “{}” en las cuales se encuentran los datos geométricos.

La estructura `geometry` contendrá los datos gráficos entre 2 llaves “{}” en los que se indica el tipo de geometría que para el sistema puede ser polígono o multipolígono y las coordenadas respectivas para graficar dicha geometría.

Las coordenadas están formadas estructuralmente de la siguiente forma:

- Geometría Multipolygon: [Geometría Polygon, Geometría Polygon,...] – Array de Geometrías Polygon.
- Geometría Polygon: [Polígono hueco] – Array que consta de un polígono principal y opcionalmente uno o varios huecos que para el sistema actual los huecos no se requieren ni se procesara.
- Polígono, Hueco: [Punto, Punto...] – Los Polígonos y Huecos son arrays de puntos que definen un área. El polígono indica un área a graficar mientras que el punto un área a eliminar del gráfico.
- Punto: [longitud, latitud] – Array de dos elementos primero la longitud seguido por la latitud.

El sistema actual permite el ingreso del área de influencia individual por cada beneficiario por lo cual el archivo GeoJson solo deberá contener un `features` en el `FeatureCollection` el cual solo contendrá el multipolígono o polígono a ingresar.

*Figura 39. Ejemplo estructura GeoJson*

```

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature", "id": 0, "properties": { "zon_admin": "Los Chillos", "agu1": 0 },
      "geometry": { "type": "MultiPolygon",
        "coordinates": [ [ [ [ [ -78.462453245065859, -0.294689454077642 ],
        [ -78.557641191064562, -0.355832545405266 ],
        [ -78.435920333927655, -0.230074945908217 ],
        [ -78.430532843222764, -0.242929121238072 ],
        [ -78.418644438533903, -0.256551550307235 ],
        [ -78.462453245065859, -0.294689454077642 ] ] ],
        [ [ [ [ -78.677572343751507, 0.041462485547604 ],
        [ -78.677859727928777, 0.054054495089884 ],
        [ -78.666061449648197, 0.082628275712663 ],
        [ -78.644278621176412, 0.119301536374499 ],
        [ -78.551758876535274, 0.059767679224806 ],
        [ -78.677572343751507, 0.041462485547604 ] ] ] ] ] ]
    }
  ]
}

```

Elaborado por: Cofre Víctor & Toledo Stalin

## CAPÍTULO 5

### IMPLEMENTACIÓN Y PRUEBAS

En esta etapa se realiza la instalación de la aplicación en el servidor y las pruebas que verifican su funcionamiento.

#### 5.1. Implementación

La presente aplicación se configuró en el servidor HP ProLiant ML110 G7 designado por el CIMA-UPS, que consta con la versión del servidor web y base de datos requerida para la implementación del software.

##### 5.1.2. Requerimientos mínimos.

Para el funcionamiento del sistema con una carga mínima de rendimiento se necesitan de los siguientes requisitos a nivel de software:

Tabla 21

#### *Requerimientos de software*

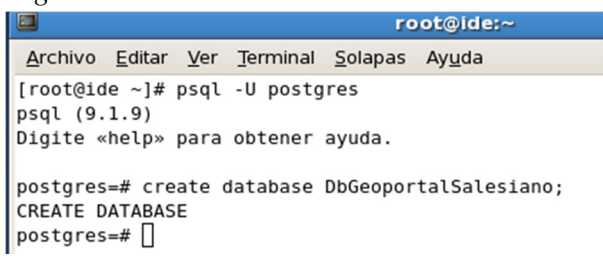
Especificaciones de software	
Sistema Operativo	Centos versión 5.9
Base de Datos	PostgreSQL versión 9.1.9
Datos Espaciales	PostGIS versión 1.5
Servidor Web	Apache Tomcat 6.0.32
Lenguaje de Desarrollo	Java 7

Elaborado por: Cofre Víctor & Toledo Stalin

##### 5.1.3. Restauración de la base de datos.

Primero se deberá crear una base de datos mediante la ejecución de los siguientes comandos en el terminal y se obtiene la confirmación de que se ha creado la base de datos.

*Figura 40. Creación de la base de datos*



```
root@ide:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@ide ~]# psql -U postgres
psql (9.1.9)
Digite <help> para obtener ayuda.

postgres=# create database DbGeoportalSalesiano;
CREATE DATABASE
postgres=#
```

Elaborado por: Cofre Víctor & Toledo Stalin



A continuación se ejecuta el script PostGis y el script Spatial esto se realiza para el manejo de datos espaciales en el motor de base de datos y la restauración de la misma.

Figura 41. Ejecución del script postgis



```
root@ide:/usr/pgsql-9.1/share/contrib/postgis-1.5
Archivo Editar Ver Terminal Solapas Ayuda
[root@ide postgis-1.5]# pwd
/usr/pgsql-9.1/share/contrib/postgis-1.5
[root@ide postgis-1.5]# psql -U postgres -d dbgeoportalsalesiano -f postgis.sql
```

Elaborado por: Cofre Víctor & Toledo Stalin

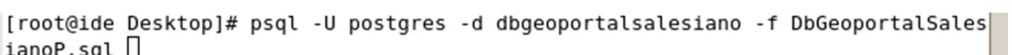
Figura 42. Ejecución de script Spatial



```
root@ide:/usr/pgsql-9.1/share/contrib/postgis-1.5
Archivo Editar Ver Terminal Solapas Ayuda
[root@ide postgis-1.5]# pwd
/usr/pgsql-9.1/share/contrib/postgis-1.5
[root@ide postgis-1.5]# psql -U postgres -d dbgeoportalsalesiano -f spatial_ref_sys.sql
```

Elaborado por: Cofre Víctor & Toledo Stalin

Figura 43. Restauración de la base de datos



```
[root@ide Desktop]# psql -U postgres -d dbgeoportalsalesiano -f DbGeoportalSalesianoP.sql
```

Elaborado por: Cofre Víctor & Toledo Stalin

### 5.1.4 Carga del archivo .war en el servidor Apache Tomcat.

Abrimos un navegador web y digitamos `http://ide.ups.edu.ec:8080` para poder ingresar al servidor de Apache Tomcat.

Figura 44. Ingreso al servidor web

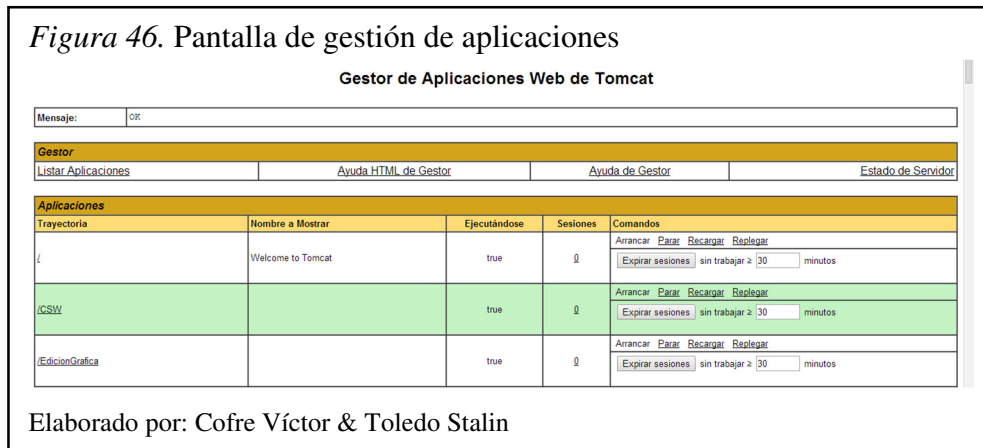


Elaborado por: Cofre Víctor & Toledo Stalin

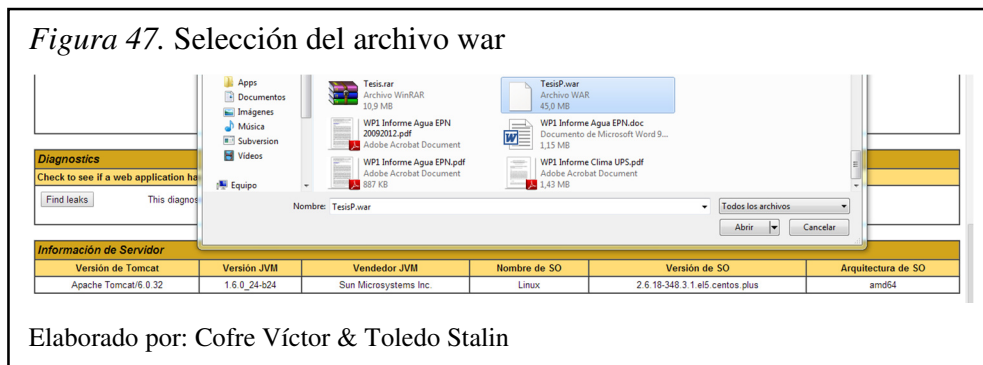
Accedemos a la característica de administración de Tomcat dando un clic en “Tomcat Manager” y se inicia sesión.



Se desplegara el gestor de aplicaciones web del servidor y se puede observar los proyectos cargados en el mismo.



En la parte inferior de la pantalla la sección de desplegar que contiene la subsección “Archivo WAR a desplegar” en la que se puede seleccionar el archivo.



A continuación se despliega el archivo.

**Figura 48. Archivo war seleccionado**

**Desplegar**  
Desplegar directorio o archivo WAR localizado en servidor

Traectoria de Contexto (opcional):   
 URL de archivo de Configuración XML:   
 URL de WAR o Directorio:

**Archivo WAR a desplegar**

Seleccione archivo WAR a cargar:  TesisP war

**Diagnostics**  
Check to see if a web application has caused a memory leak on stop, reload or undeploy

This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

**Información de Servidor**

Versión de Tomcat	Versión JVM	Vendedor JVM	Nombre de SO	Versión de SO	Arquitectura de SO
Apache Tomcat/6.0.32	1.6.0_24-b24	Sun Microsystems Inc.	Linux	2.6.18-348.3.1.el5.centos.plus	amd64

Copyright © 1999-2011, Apache Software Foundation

Elaborado por: Cofre Víctor & Toledo Stalin

Al desplegar el proyecto en el archivo WAR se podrá observar que se encuentra en la lista de aplicaciones del servidor.

**Figura 49. Proyectos desplegados**

Aplicación	Contexto	Estado	Acciones
/Seguridad		true	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/Tesis	Tesis	true	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/TesisP	TesisP	true	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos

Elaborado por: Cofre Víctor & Toledo Stalin

## 5.2. Pruebas

Las pruebas de rendimiento que se realizaron al gestor de datos geográficos como al visualizador ambos ya correctamente instalados en el servidor de CIMA mediante la herramienta Apache-JMeter-2.11.

Las pruebas se realizaron desde un equipo de escritorio con las siguientes características:

- Procesador: Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz (4 CPUs), ~3.2GHz
- Memoria: 4096MB RAM
- Sistema Operativo: Windows 8 Pro 64-bit

Los procedimientos para realizar el plan de pruebas fueron:

- Crear un nuevo grupo de hilos
  - Poner un nombre a grupo de hilos
  - Especificar la acción a realizar en caso de tener un error
  - Indicar el número de hilos que se usara y el periodo de subida que tendrán los mismos
  - Colocar el valor del contador del bucle

*Figura 50.* Creación de grupo de hilos



The screenshot shows the 'Grupo de Hilos' (Group of Threads) configuration window in Apache JMeter. It includes the following fields and options:

- Nombre:** Grupo de Hilos
- Comentarios:** (empty text area)
- Acción a tomar después de un error de Muestreador:** Continuar (selected), Comenzar siguiente iteración, Parar Hilo, Parar Test, Parar test ahora
- Propiedades de Hilo:**
  - Número de Hilos:** 150
  - Periodo de Subida (en segundos):** 1
  - Contador del bucle:** Sin fin (unchecked), 1

Elaborado por: Cofre Víctor & Toledo Stalin

- Crear la petición http e indicar el nombre del servidor o la IP, indicar el puerto que se utiliza y la ruta del proyecto.
  - Nombre del servidor: ide.ups.edu.ec
  - Puerto: el puerto actual que usa el servidor de apache utilizado es el 8080
  - La ruta: aquí se ingresa la ruta del gestor de datos geográficos (/TesisP) como la del visualizador (/ProyectoTesisUPSP) respectivamente en cada prueba

Figura 51. Creación de petición HTTP

**Petición HTTP**

Nombre: Petición HTTP

Comentarios

Servidor Web

Nombre de Servidor o IP: ide.ups.edu.ec Puerto: 8080 Timeo Conex

Petición HTTP

Implementación HTTP: Protocolo: Método: GET Codificación del contenido:

Ruta: /TesisP

Redirigir Automáticamente  Seguir Redirecciones  Utilizar KeepAlive  Usar 'multipart/form-data' para HTTP POST  Cabece

Elaborado por: Cofre Víctor & Toledo Stalin

- Crear las vistas de resultados:
  - Ver resultados en árbol
  - Ver árbol de resultados

Figura 52. Creación de resultados

Apache JMeter (2.11 r1554548)

Archivo Editar Search Lanzar Opciones Ayuda

Plan de Pruebas

Grupo de Hilos

Petición HTTP

Ver Resultados en Árbol

Ver Árbol de Resultados

Banco de Trabajo

**Ver Resultados en Árbol**

Nombre: Ver Resultados en Árbol

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Muestra # Tiempo de comie... Nombre d

Elaborado por: Cofre Víctor & Toledo Stalin

- Ejecutar el plan de pruebas con un número de muestra de 50 usuarios, en el cual se observa el estado, los bytes, la latencia y el número de errores.

Figura 53. Resultados de prueba con 50 muestras

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra	Estado	Bytes	Latency
24	12:15:06.905	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
25	12:15:06.916	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
26	12:15:06.931	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
27	12:15:06.962	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
28	12:15:06.978	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
29	12:15:06.993	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
30	12:15:07.025	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
31	12:15:07.040	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
32	12:15:07.056	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
33	12:15:07.075	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
34	12:15:07.103	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
35	12:15:07.119	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
36	12:15:07.135	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
37	12:15:07.167	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
38	12:15:07.182	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
39	12:15:07.197	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
40	12:15:07.213	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
41	12:15:07.244	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
42	12:15:07.260	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
43	12:15:07.275	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	2
44	12:15:07.291	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
45	12:15:07.324	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
46	12:15:07.339	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
47	12:15:07.354	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
48	12:15:07.385	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
49	12:15:07.400	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
50	12:15:07.416	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2

Scroll automatically?  Child samples? **No. de Muestras 50** **Última Muestra 3** **Media 2** **Desviación 0**

Elaborado por: Cofre Víctor & Toledo Stalin

- Para probar al sistema con un número de peticiones mayor se realizó nuevamente pero con una muestra de 150 y de 200 usuarios.

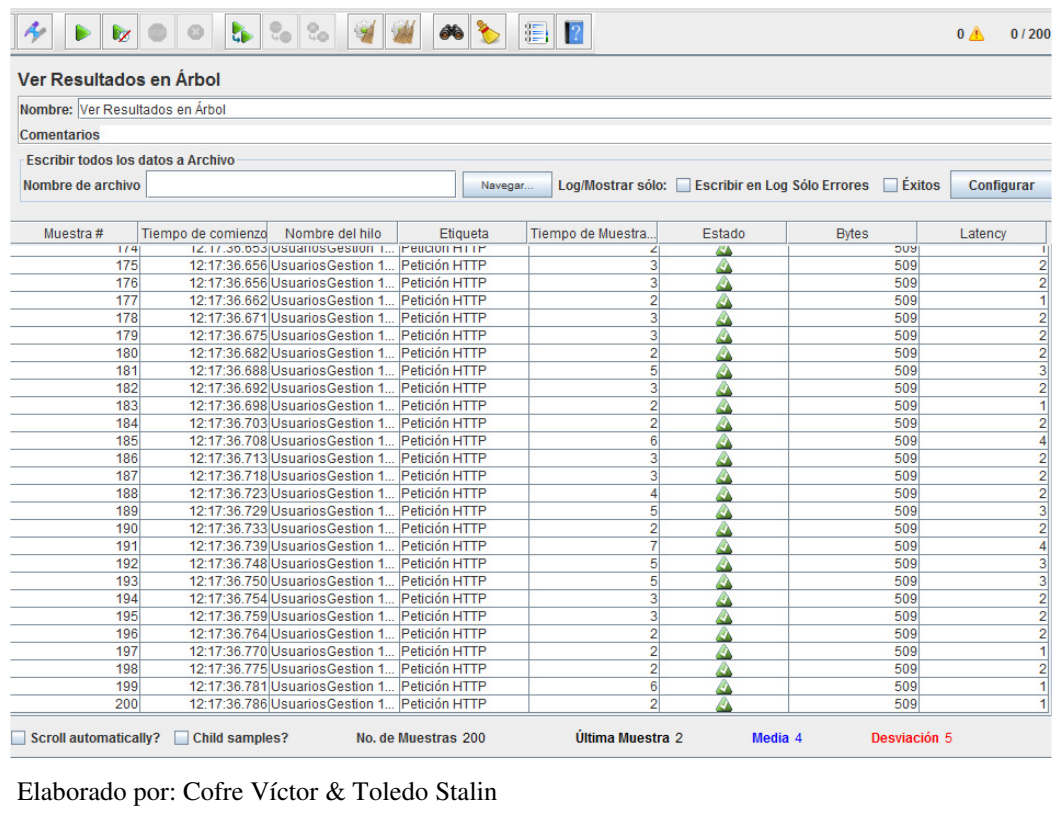
Figura 54. Resultados de prueba con 150 muestras

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra	Estado	Bytes	Latency
124	12:16:42.732	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
125	12:16:42.736	UsuariosGestion 1...	Petición HTTP	5	Exitos	509	3
126	12:16:42.741	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
127	12:16:42.749	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
128	12:16:42.756	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
129	12:16:42.764	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
130	12:16:42.769	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
131	12:16:42.776	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
132	12:16:42.783	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
133	12:16:42.790	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
134	12:16:42.795	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
135	12:16:42.802	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
136	12:16:42.813	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
137	12:16:42.818	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
138	12:16:42.826	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
139	12:16:42.833	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
140	12:16:42.839	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
141	12:16:42.846	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
142	12:16:42.853	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
143	12:16:42.863	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
144	12:16:42.867	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
145	12:16:42.875	UsuariosGestion 1...	Petición HTTP	6	Exitos	509	4
146	12:16:42.880	UsuariosGestion 1...	Petición HTTP	3	Exitos	509	2
147	12:16:42.888	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
148	12:16:42.895	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1
149	12:16:42.901	UsuariosGestion 1...	Petición HTTP	6	Exitos	509	2
150	12:16:42.907	UsuariosGestion 1...	Petición HTTP	2	Exitos	509	1

Scroll automatically?  Child samples? **No. de Muestras 150** **Última Muestra 2** **Media 3** **Desviación 1**

Elaborado por: Cofre Víctor & Toledo Stalin

Figura 55. Resultados de prueba con 200 muestras



Después de realizar las respectivas pruebas se puede aclarar que el sistema soporta los 150 usuarios aunque al aumentar el número el tiempo de carga va siendo mayor, se toma en cuenta que actualmente existen alrededor de 26 a 30 casas salesianas lo cual indica que por cada casa podrán conectarse un usuario y aún si se conectaran 3 usuarios al mismo instante por cada casa el sistema soportaría la carga satisfactoriamente.

## CONCLUSIONES

- La implementación de los nuevos requerimientos para el módulo visualizador y el módulo de gestión de datos geográficos mejora significativamente la usabilidad y funcionalidad, permitiendo al usuario manejarlos de manera fácil e intuitiva.
- La implementación de la escala gráfica y los respectivos iconos de simbología permiten al usuario un mejor entendimiento y facilidad de búsqueda al usar el sistema.
- La visualización de las coordenadas (longitud y latitud) que se implementó en el módulo gestión de datos geográficos para el ingreso de coordenadas del lugar a la que se desea asignar evitan el ingreso erróneo de la coordenada.
- La estructuración de los archivos geométricos en base al formato GeoJSON facilita su entendimiento, portabilidad y carga en el servidor, además es un formato muy difundido y está siendo usado en varios sistemas de georeferenciación.
- Al agregar el soporte de multipartes al visualizador permite subir multipolígonos y multipuntos lo cual es fundamental para la información de la Comunidad Salesiana, existen varios lugares en los que una obra interviene e igualmente existen varios beneficiarios que solo se pueden representar de una manera óptima con multipoligonos en el mapa digital.
- El soporte de multipartes al módulo de gestión de datos geográficos permite subir multipolígonos y multipuntos para el requerimiento de mostrar en el visualizador dichos datos requiere que primero se los ingrese por el módulo de gestión de datos geográficos y se los almacene en la base de datos.



## RECOMENDACIONES

- Se recomienda para el uso del módulo de visualización disponer una velocidad de transmisión de datos mínima de 1MB, así se visualizará los resultados de una manera más rápida, el uso de menor velocidad implica una visualización de las imágenes con retardo.
- Realizar la actualización de los datos geográficos como de los lugares de acuerdo al crecimiento de las obras salesianas y así el sistema se mantenga actualizado constantemente.
- Se requiere que el polígono contenido en el archivo GeoJson no supere los 150 puntos ya que el sistema no permitirá visualizarlo debido a la limitación del componente.
- Se recomienda tener en cuenta una próxima versión tras revisar los avances en las versiones tanto de primefaces como de postgis, mismas que están en constante crecimiento y permitiría el uso de menor código y recursos.

## LISTA DE REFERENCIAS

- Corporation, O. (2013). *NetBeans Platform*. Recuperado el 03 de julio de 2013, de <https://netbeans.org/features/index.html>
- Foundation, O. S. (2013). *Quantum GIS*. Recuperado el 28 de junio de 2013, de <http://www.ensayosgratis.com/Temas-Variados/Qtumgis/108948.html>
- Fuentes, H. (14 de marzo de 2011). *GeoCivil*. Obtenido de <http://geocivil.blogspot.com/2011/03/quantum-gis-sig-opensource.html>
- Geary, D., & Horstmann, C. (2010). *Core Java Server Faces 3rd edition*. Prentice hall.
- Geoupse. (01 de octubre de 2013). *Geoportal*. Obtenido de Geoportal UPSE: <http://201.218.63.172/geoupse/>
- Martínez, J. (2013). *PostGIS 2 Análisis Espacial Avanzado*. USA: San Bernardino, CA.
- PostgreSQL, E. G. (27 de junio de 2013). *PostgreSQL*. Recuperado el 28 de 06 de 2013, de <http://www.postgresql.org/about/>
- Salesianos. (2014). *Quienes Somos*. Obtenido de Salesianos Ecuador: <http://www.salesianos.org.ec>
- Sevilla, G. (abril de 2008). Desarrollo de un tutorial para la enseñanza de ensamblaje de computadores personales a nivel básico. Quito.
- Silva, D. A., & Mercerat, B. (29 de enero de 2002). *Construyendo aplicaciones web con una metodología de diseño*. Recuperado el 03 de julio de 2013, de [http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)
- teknoloji, p. (2014). *PrimeFaces User's Guide*.
- Vilain, P., Schwabe, D., & Sieckenius de Souza, C. (octubre de 2000). *TECWEB(Web Engineering Laboratory)*. Obtenido de [http://www.tecweb.inf.puc-rio.br/navigation/context/o\\_158de19d@1?p=o\\_PatriciaVilain\\_48b9](http://www.tecweb.inf.puc-rio.br/navigation/context/o_158de19d@1?p=o_PatriciaVilain_48b9)

## GLOSARIO DE TÉRMINOS

**ADVs:** (Vista de Datos Abstracta) especifica la organización y comportamiento de la interfaz.

**BDD:** Una base de datos distribuida es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran distribuidas en diferentes espacios lógicos e interconectados por una red de comunicaciones.

**Datos geoespaciales:** Representan información sobre la ubicación física y la forma de objetos geométricos.

**GEOJSON:** Es un formato para la clasificación de una variedad de estructuras de datos geográficos. GeoJSON soporta los siguientes tipos de geometría: Point, LineString, polígono, multipunto, MultiLineString, MultiPolygon.

**GPS:** Sistema de posicionamiento global, está constituido de 24 satélites en orbitas sobre el planeta y utiliza triangulación para determinar la posición y tiene una precisión en metros.

**HTTP:** Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la world wide web.

**JSF:** Java Server Faces es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java.

**Multipolígono:** Es un conjunto de polígonos.

**Multipunto:** Es un conjunto de puntos donde cada uno tiene asociada una coordenada.

**MVC:** El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

**OOHDM:** Método de Diseño Hipermedia Orientado Objeto: Metodología orientada a objetos para el diseño de aplicaciones web hipermedia.

**Polígono:** Es un conjunto de segmentos que encierra una región y tiene asociada las coordenadas de los puntos de cada segmento que lo forma.

**Postgresql:** Es un sistema de gestión de base de datos relacional, de código abierto y orientado a objetos.

**Primefaces:** Es una librería de componentes visuales de código abierto utilizado con Java Server Faces (JSF) ayuda en la creación de aplicaciones web.

**Punto:** Es la representación de la coordenada terrestres medidas por longitud y latitud.

**Script:** Es un pequeño programa que puede estar en un documento plano o en la misma página JSF.

**SRID:** Identificadores de referencia espacial. El SRID corresponde a un sistema de referencia espacial basado en el elipsoide concreto usado para la creación de mapas de tierra plana o de tierra redonda.

**UID:** Diagramas de interacción de usuario describe el intercambio de información entre el sistema y el usuario.

**UML:** Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software.

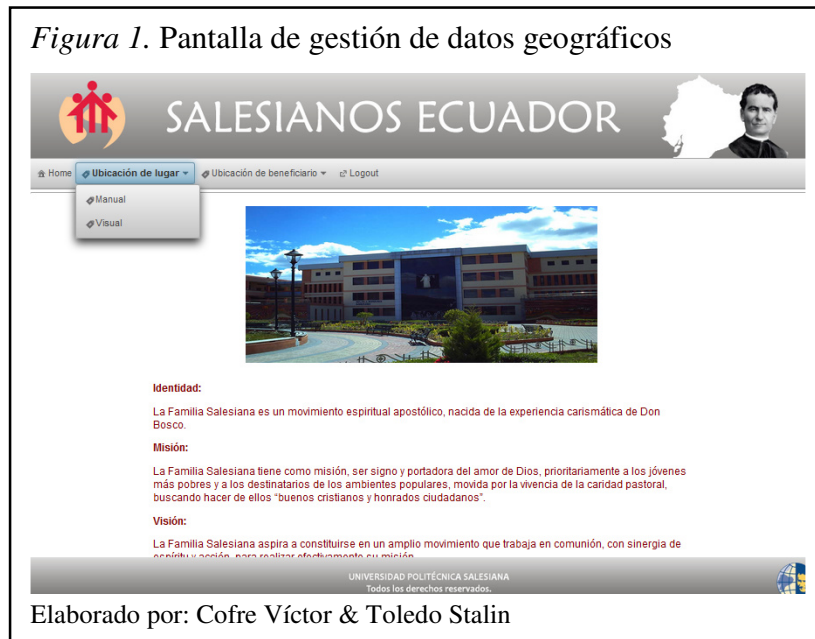
**WAR:** Es un archivo de aplicación web que contiene JavaServer Pages, servlets, clases Java, archivos XML.

## ANEXOS

### Anexo 1. Manual de usuario

- **Pantallas reales Gestión de datos geográficos**

La pantalla respectiva de módulo de gestión geográfica consta de los menús correspondientes tanto para el ingreso de los datos de ubicación del lugar como de la ubicación del beneficiario (área de influencia).



La pantalla de ingreso manual de las coordenadas del lugar permite mediante el botón agregar punto marcar en el mapa los distintos lugares de la obra, mientras se esté especificado la obra a la que se agregaran los mismos.



El botón agregar punto inserta varios puntos en el caso de que la obra tenga más de un lugar y el botón de guardar almacena la información.



La pantalla de ingreso de la ubicación de lugar mediante un mapa digital visualmente marca los distintos lugares dando clic sobre el mapa, permitiendo saber la longitud y latitud del lugar en donde se presionó el clic y preguntado si deseamos agregar el punto.



Igualmente el ingreso visual de lugar marca varios lugares en caso de que la obra lo requiera.

**Figura 5. Pantalla de ingreso de visual de ubicación de lugar con varios puntos**



The screenshot shows the SALESIANOS ECUADOR web interface. At the top, there is a navigation bar with the logo and the text 'SALESIANOS ECUADOR'. Below the navigation bar, there is a header with 'SELECCIONE UN LUGAR EN EL MAPA'. The main content area features a Google Maps view of Ecuador with a red pin placed on the map. A notification box in the top right corner displays 'Marker Added' along with coordinates: 'Lat: -2.0294826641872965' and 'Lng: -79.74902361631393'. Below the map, there are three dropdown menus for 'ID Casa', 'ID Obra', and 'ID Lugar', all currently set to '--Seleccione una casa--', '--Seleccione una obra--', and '--Seleccione un lugar--' respectively. At the bottom of the form, there are 'Limpiar' and 'GUARDAR' buttons.

Elaborado por: Cofre Víctor & Toledo Stalin

A continuación se indica la obra a la que se le asigna los puntos de lugar y se procede a almacenar mediante el botón guardar.

**Figura 6. Selección del lugar a almacenar**



This screenshot shows the same web interface as Figure 5, but with the dropdown menus populated with specific location names. The 'ID Casa' dropdown is set to 'CASA SALESIANA BEATO MIGUEL RUA DE ZUMBAHUA', the 'ID Obra' dropdown is set to 'CENTRO COMUNITARIO DE GUANGAJE', and the 'ID Lugar' dropdown is set to 'CENTRO COMUNITARIO DE GUANGAJE'. The 'GUARDAR' button is highlighted in blue, indicating it is the active action.

Elaborado por: Cofre Víctor & Toledo Stalin

La pantalla para ingresar la ubicación del beneficiario (polígonos) permite ingresar un archivo GeoJson que contenga la geometría requerida para el beneficiario de la obra seleccionado.

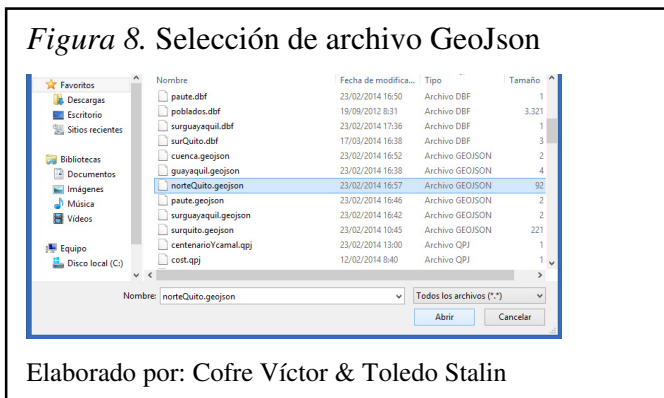
**Figura 7. Pantalla de ingreso de beneficiario**



Elaborado por: Cofre Víctor & Toledo Stalin

El componente solamente permite el ingreso de archivos GeoJson.

**Figura 8. Selección de archivo GeoJson**



Elaborado por: Cofre Víctor & Toledo Stalin

El archivo que se seleccione debe ser subido al servidor para que se visualice si es la geometría deseada se muestra su nombre en el componente y de ser incorrecto el tipo de archivo se pintara de rojo y evitara la subida del mismo.

**Figura 9. Archivo GeoJson seleccionado**



Elaborado por: Cofre Víctor & Toledo Stalin



El archivo subir al servidor mediante el botón subir del componente carga una vista previa del contenido que tenía el archivo GeoJson escogido para así verificar que se ha cargado correctamente.

Figura 10. Visualización del contenido del archivo GeoJson

ID Casa: QUITO-EL GIRÓN

ID Obra: DISPENSARIO PARROQUIAL MARÍA AUXILIADORA

ID Lugar: DISPENSARIO PARROQUIAL MARÍA AUXILIADORA

ID Beneficiario: Brinda atención medica y odontológica personalizada a personas que la requieran como niños, adolescentes, adultos y adultos mayores.

+ Seleccionar archivo geojson   Subir   Cancelar

Elaborado por: Cofre Víctor & Toledo Stalin

Correctamente seguro que lo que se ha cargado es lo deseado se procede a guardar mediante el botón ingresar.

Figura 11. Selección de beneficiario e ingreso de datos

SALESIANOS ECUADOR

Home   Ubicación de lugar   Ubicación de beneficiario   Logout

ID Lugar: DISPENSARIO PARROQUIAL MARÍA AUXILIADORA

ID Beneficiario: Brinda atención medica y odontológica personalizada a personas que la requieran como niños, adolescentes, adultos y adultos mayores.

+ Seleccionar archivo geojson   Subir   Cancelar

INGRESAR

Elaborado por: Cofre Víctor & Toledo Stalin

- **Pantallas reales Visualizador**

El visualizador maneja una interfaz intuitiva y estándar en todos los estilos de búsqueda para una mejor visualización de los resultados tanto el panel izquierdo (opciones de búsqueda) como el panel derecho (iconos) se despliegan y permite desplegarlos dependiendo la necesidad del usuario.

La pantalla de búsqueda por categoría realiza la búsqueda mediante el ingreso de la casa Salesiana y especificando el tipo de obra, ambos campos obligatorios, si se desea buscar por casa solamente existen otros tipos de búsqueda que lo permitirán.

Figura 12. Pantalla de Búsqueda por categoría



Elaborado por: Cofre Víctor & Toledo Stalin

La pantalla de búsqueda temática presenta un campo que ingresa el nombre del lugar o parte del nombre para seleccionar lo que se dese encontrar mediante el autocompletado que se crea en base a los lugares almacenados.

Seleccionado el lugar mediante el botón buscar se observa el resultado.



La pantalla para la búsqueda por tipo de obra muestra en el panel izquierdo un árbol de los tipos de obras existentes los cuales en su interior igualmente contienen un sub árbol los nombres de la casas que tenga las obras del tipo desplegado, para realizar la búsqueda deseada se puede seleccionar el check de los datos que se dese ver tanto como tipo, casa u obra y al presionar el botón ejecutar todos los resultados se verán en el mapa.



La pantalla de búsqueda por casa y obra igualmente consta de un árbol y su funcionamiento es el mismo que contenga el árbol de búsqueda en el sistema, tan solo se requiere marcar lo que se requiera buscar ya sea casa(s) u obra(s) y se ejecuta la búsqueda.



La pantalla de búsqueda por casa, tipo de obra y obra permite observar en el árbol primero las casas, las cuales en su interior contiene el tipo de obra existentes en esa casa y por último las obras correspondientes permitiendo marcar los elementos que se desea observar.



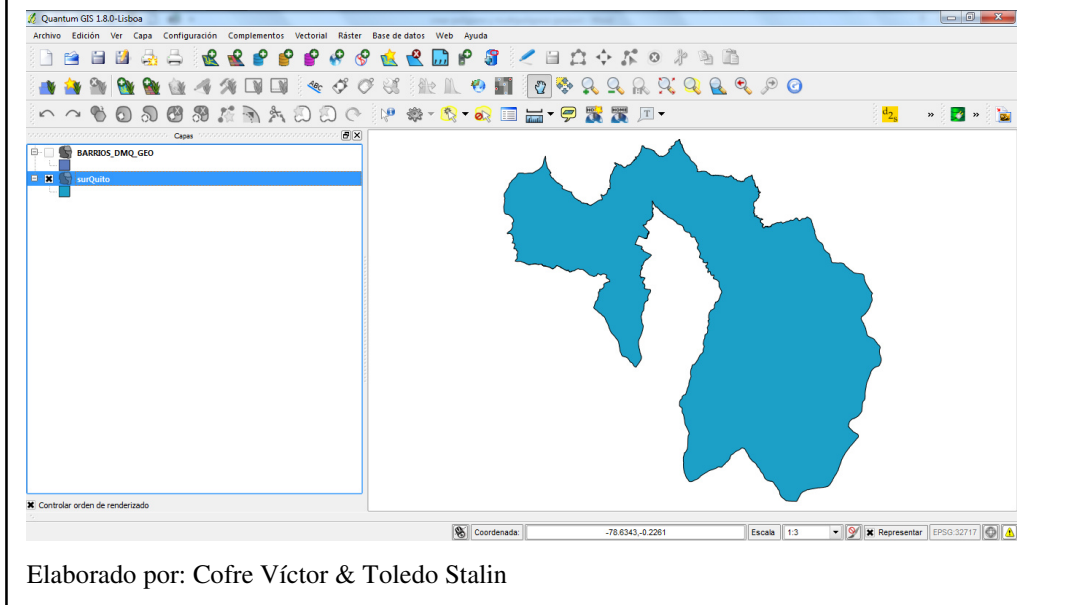
## Anexo 2. Generar archivo GeoJson utilizando Quantum GIS

Para generar un archivo GeoJson existen una variedad de aplicaciones, pero la que se utiliza fue Quantum GIS debido a que es un software de licencia libre y su manejo es sencillo, y permite generar el archivo GeoJson a partir de cualquier shapefile.

- **Polígonos**

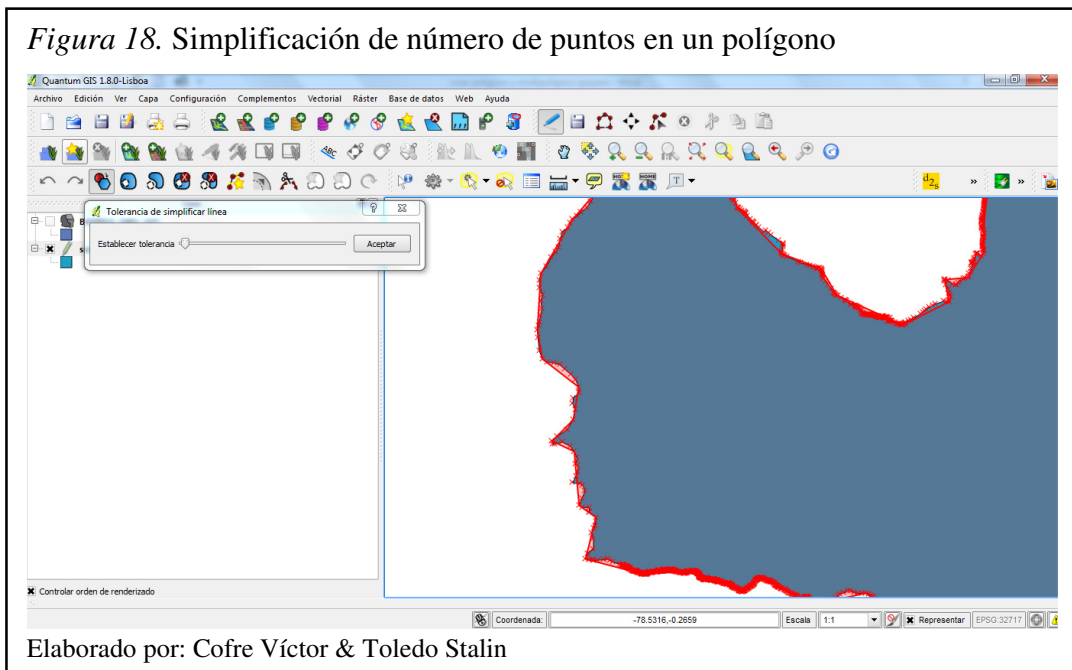
Se debe cargar la capa de la cual se desea generar el GeoJson en el Quantum GIS.

*Figura 17. Capa con geometría en Quantum Gis 1.0.8 – Lisboa*

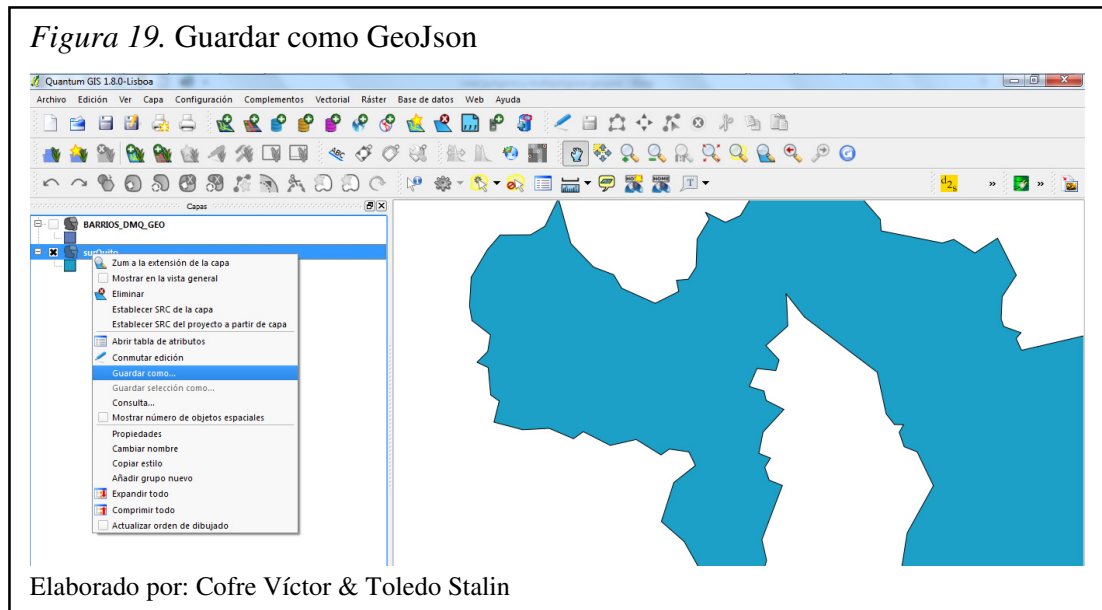


Simplificamos el número de puntos del polígono mediante el uso de la herramienta simplificar.

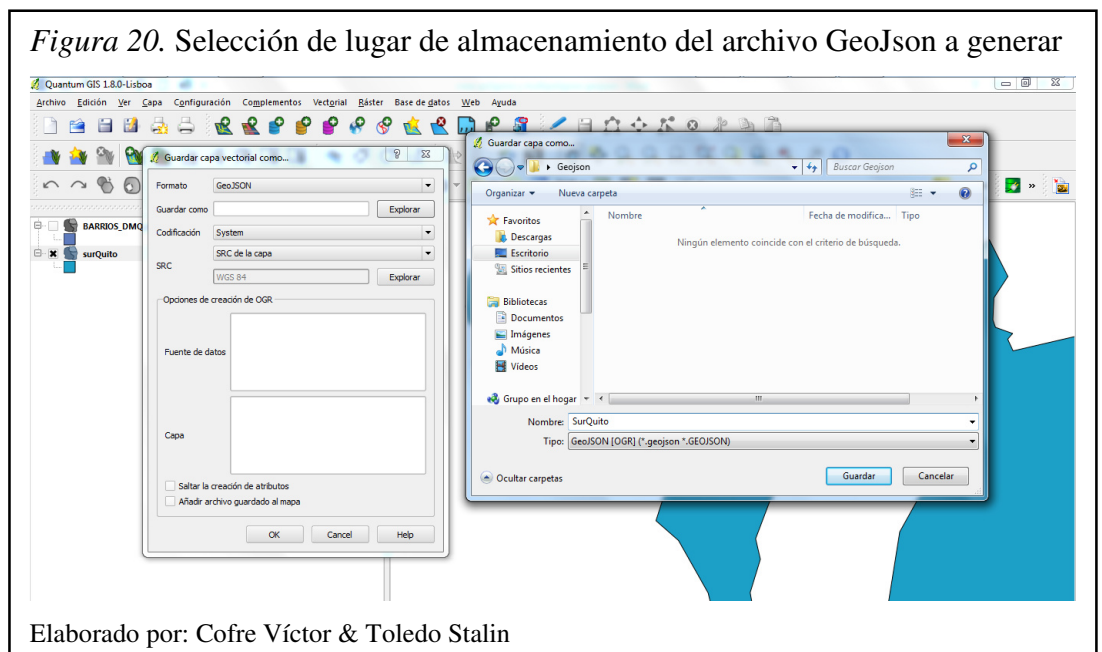
*Figura 18. Simplificación de número de puntos en un polígono*



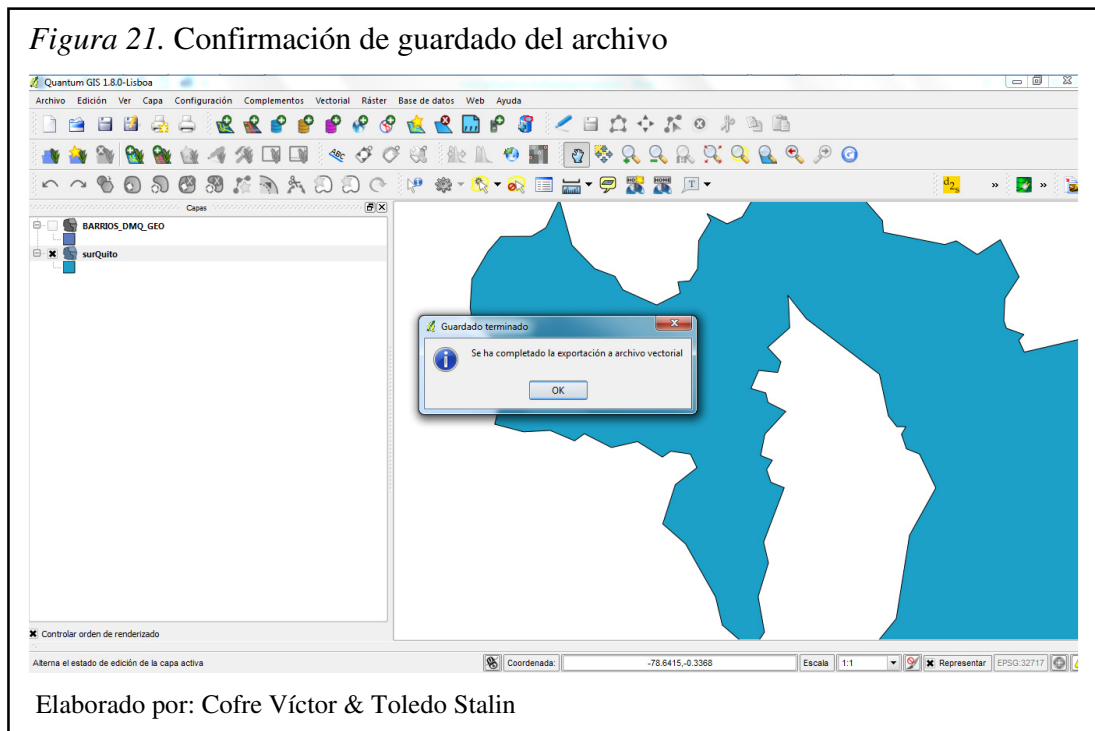
Guardar los cambios y proceder a generar el GeoJson de la capa requerida dando clic en guardar como.



Seleccionar el formato GeoJson, el lugar en donde se guardara el archivo generado e indicar el nombre que se le asignara, verificar que la codificación se mantenga en system y que el campo SRC se encuentre con la selección de WGS-84 y presionar el botón OK.



Al generarlo con éxito se presentara la pantalla de guardado con éxito.



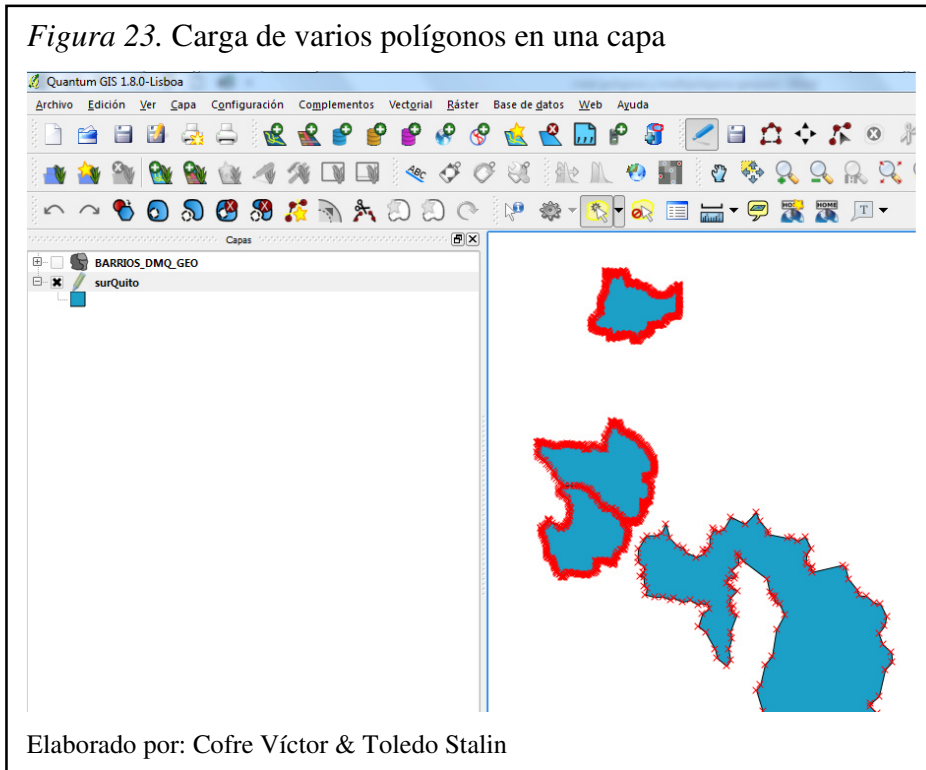
El archivo generado se puede abrir con cualquier editor de texto y se debe verificar el tipo de elemento geométrico (Polygon) y que contenga las coordenadas en grados longitud y latitud.



- **Multipoligonos**

Para la generación de una geometría de tipo multipolígono se requiere tener los polígonos cargados en una sola capa.

*Figura 23. Carga de varios polígonos en una capa*



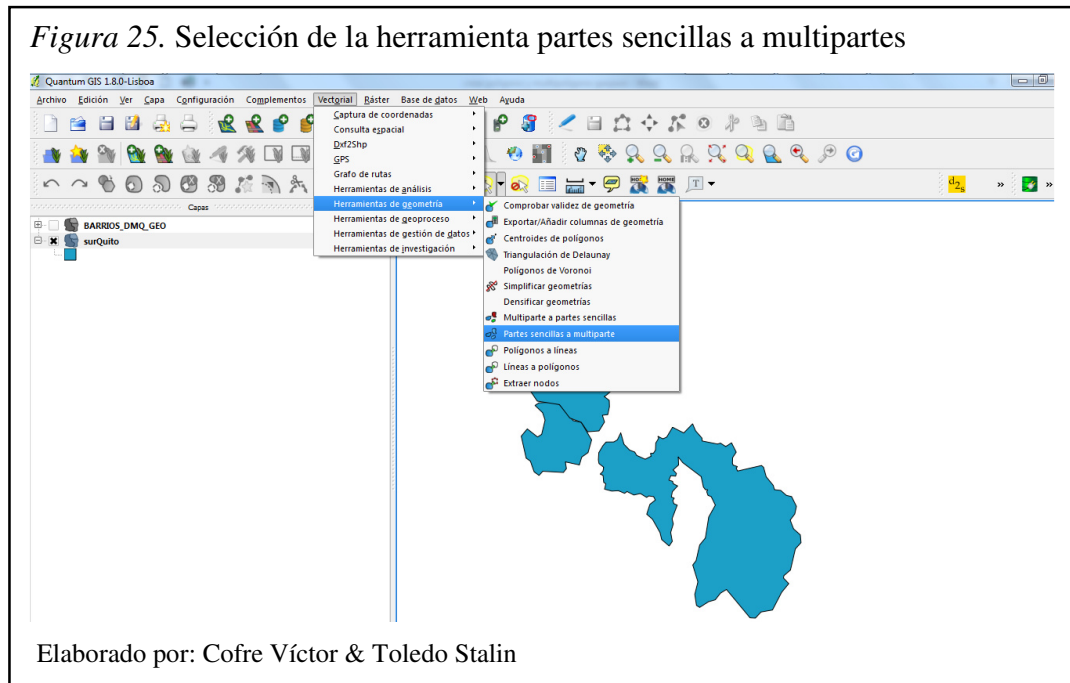
Se debe simplificar el número de puntos mediante la herramienta simplificar en cada polígono uno por uno.

*Figura 24. Simplificación de número de puntos en los polígonos*

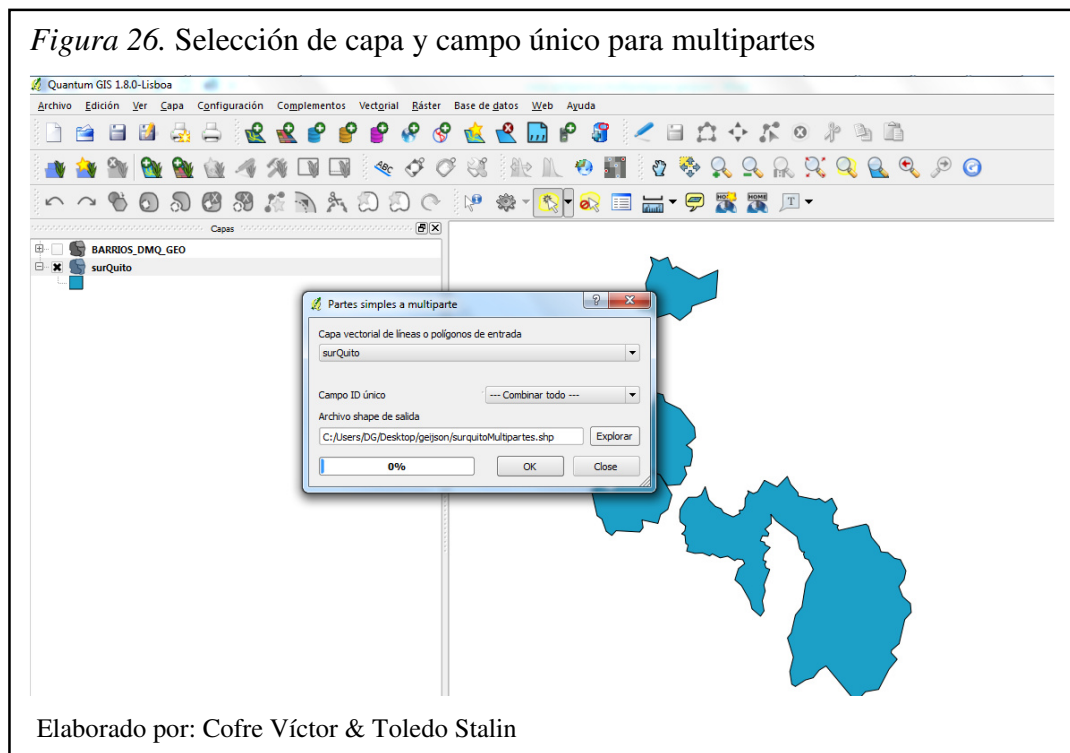




Para crear un multipolígono con varios polígonos se usa la herramienta “Partes sencillas a multipartes” que se encuentra en las Herramientas de geometría del menú Vector de la barra de herramientas de Quantum GIS.

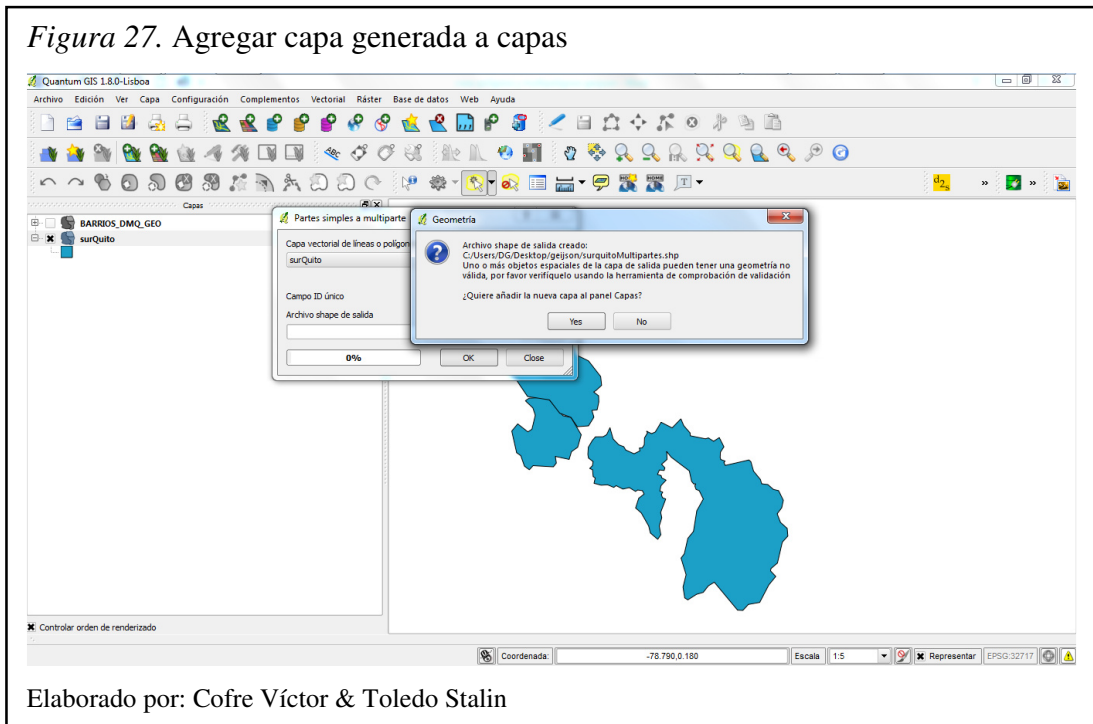


Seleccionar la capa a la que se desee generar la multiparte, asignar un lugar y nombre para guardar y confirmar que el campo ID único sea “--- Combinar Todo ---”.



Se presiona OK y se agrega la capa generada al panel de capas.

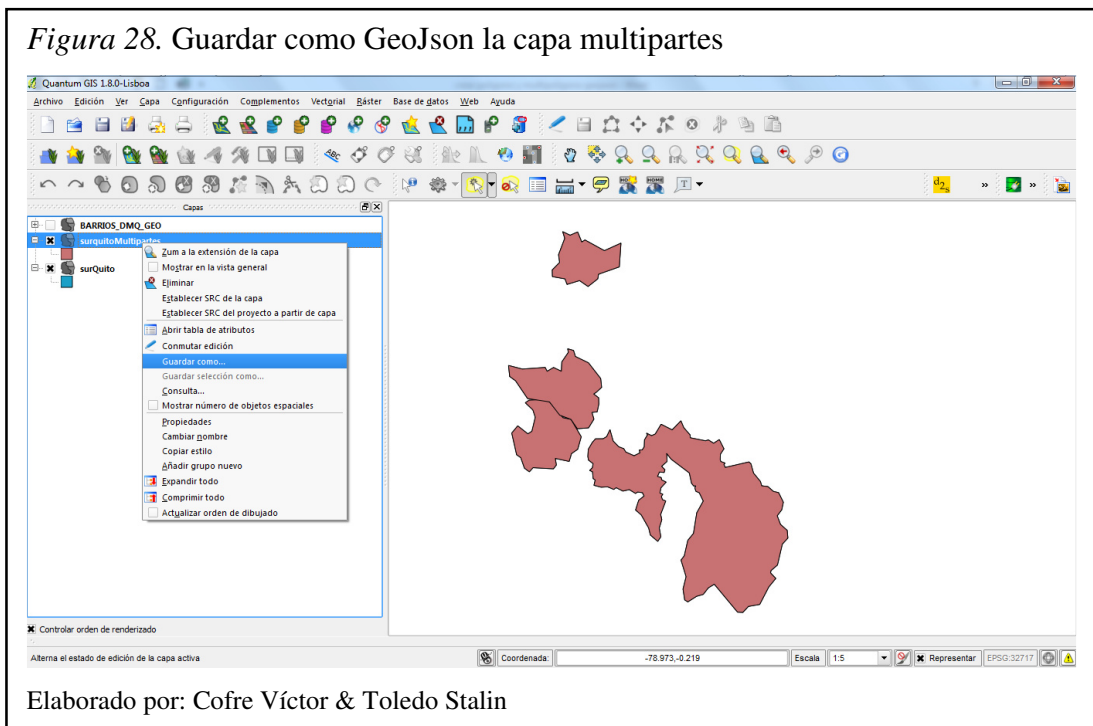
Figura 27. Agregar capa generada a capas



Elaborado por: Cofre Víctor & Toledo Stalin

Seleccionar la nueva capa y presionar guardar como.

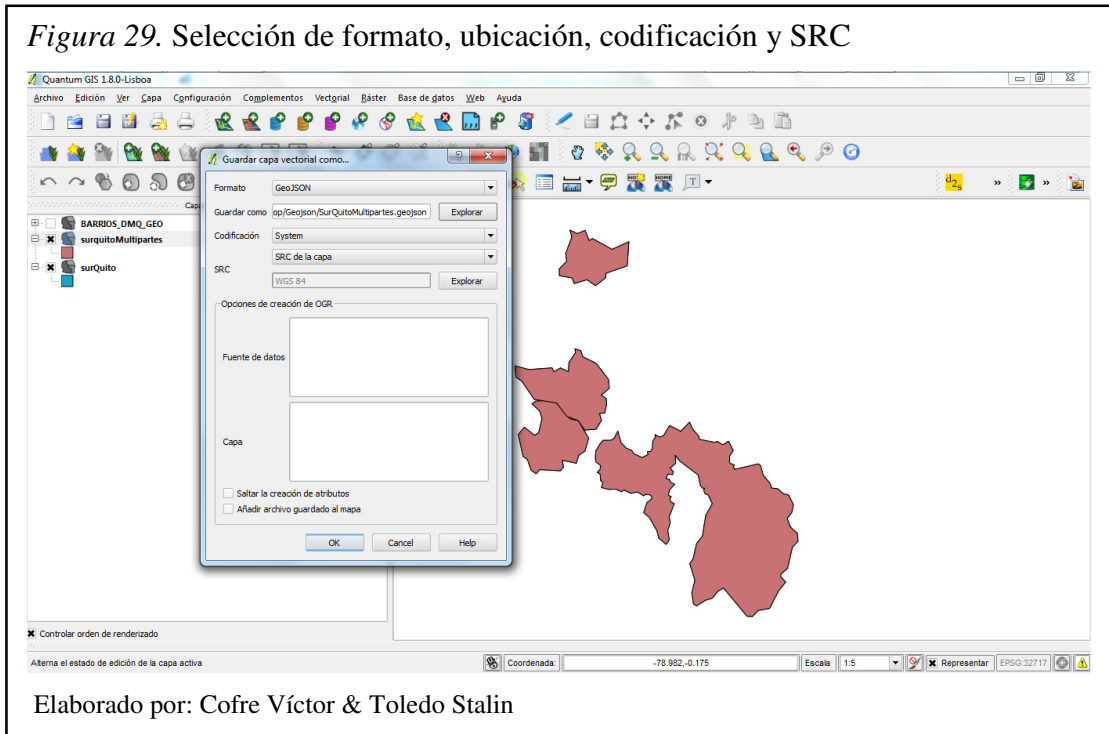
Figura 28. Guardar como GeoJson la capa multipartes



Elaborado por: Cofre Víctor & Toledo Stalin

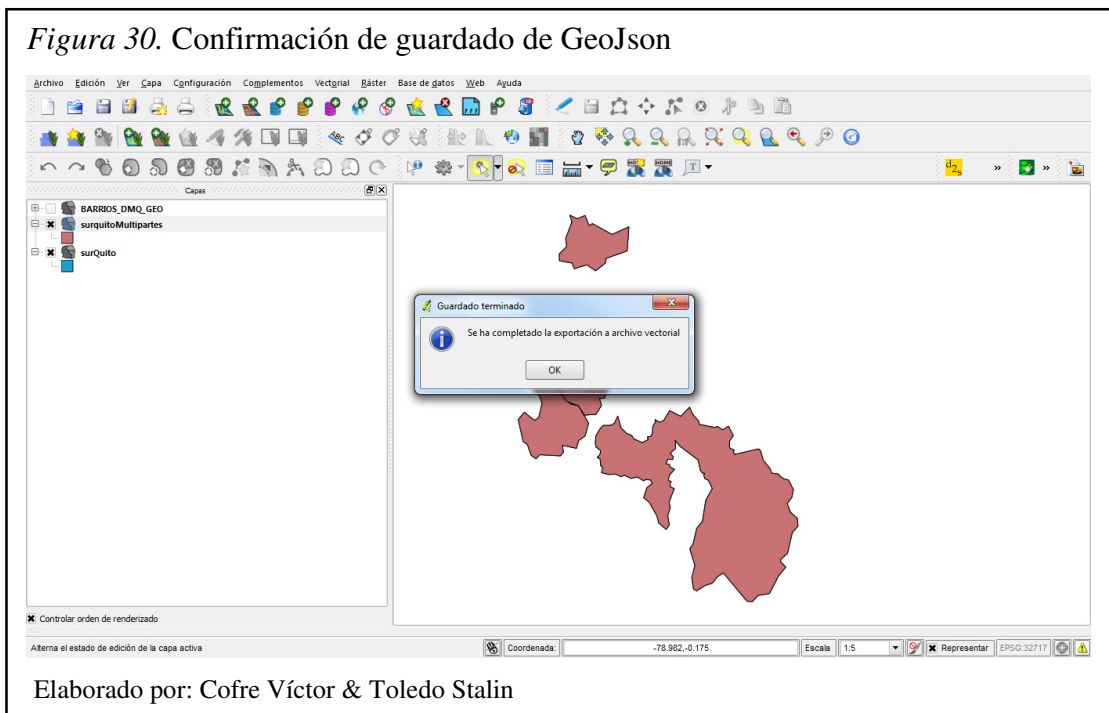
Seleccionar el formato GeoJson, el lugar en donde se guarda el archivo generado e indicar el nombre que se le asignará, verificar que la codificación se mantenga en system y que el campo SRC se encuentre con la selección de WGS-84 y presionar el botón OK.

Figura 29. Selección de formato, ubicación, codificación y SRC



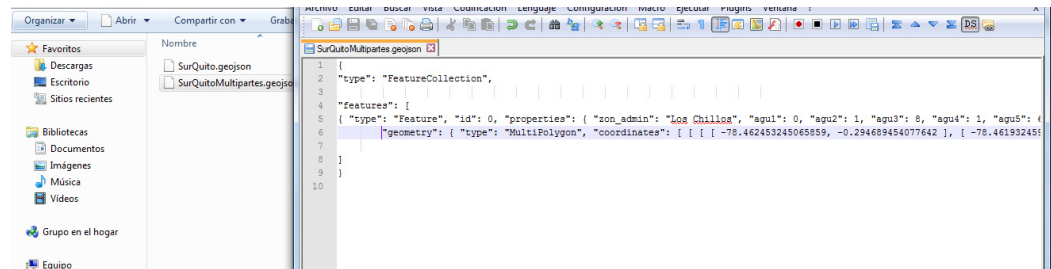
Se muestra la confirmación de la generación del GeoJson archivo.

Figura 30. Confirmación de guardado de GeoJson



El archivo generado se puede abrir con cualquier editor de texto y se debe verificar el tipo de elemento geométrico (MultiPolygon) y que contenga las coordenadas en grados longitud y latitud.

*Figura 31. Contenido de archivo GeoJson con multipolígono*



```
1 {
2   "type": "FeatureCollection",
3
4   "features": [
5     { "type": "Feature", "id": 0, "properties": { "zon_admin": "Los Chillos", "agu1": 0, "agu2": 1, "agu3": 0, "agu4": 1, "agu5": 4
6       "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [-78.462453245066859, -0.294689454077642 ], [ -78.46193245:
```

Elaborado por: Cofre Víctor & Toledo Stalin