

**RIKHUNA: VISOR CARTOGRÁFICO INTELIGENTE DE DIRECCIONES
URBANAS Y SITIOS DE INTERÉS DEL MUNICIPIO DE PASTO BASADO EN
POSTGIS**

**DAVID REINALDO ERASO GALLO
WILMER ALEXANDER ESCOBAR BOTINA
YAMID ESTEBAN ESTRADA BASTIDAS**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
2017**

**RIKHUNA: VISOR CARTOGRÁFICO INTELIGENTE DE DIRECCIONES
URBANAS Y SITIOS DE INTERÉS DEL MUNICIPIO DE PASTO BASADO EN
POSTGIS**

**DAVID REINALDO ERASO GALLO
WILMER ALEXANDER ESCOBAR BOTINA
YAMID ESTEBAN ESTRADA BASTIDAS**

**Trabajo de grado presentado como requisito parcial para optar al título de
Ingeniero de Sistemas**

**DIRECTOR:
SILVIO RICARDO TIMARAN PEREIRA, Ph.D.**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
2017**

NOTA DE RESPONSABILIDAD

“Las ideas y conclusiones aportadas en la tesis de grado, son responsabilidad exclusiva de sus autores”.

Artículo 1º del acuerdo No 324 del 11 de octubre de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”

Artículo 13, Acuerdo N. 005 de 2010, emanado del Honorable Consejo Académico.

Nota de aceptación:

Firma Presidente del Jurado

Firma del Jurado

Firma del Jurado

San Juan de Pasto, 2017

AGRADECIMIENTOS

Expresamos nuestro agradecimiento a la Universidad de Nariño por acogernos y formarnos como profesionales.

Al Sistema de Investigaciones de la Universidad de Nariño por el apoyo financiero otorgado para la ejecución de este trabajo.

Al Asesor del trabajo de grado, profesor Ricardo Timaran PhD, por su constante acompañamiento y por permitirnos formar parte del grupo de investigación GRIAS para iniciar nuestra vida investigativa.

A los profesores que fueron parte de nuestra formación, por todas las enseñanzas y conocimientos compartidos.

A los compañeros Juan José Chamorro y Marlon Pasuy, por su colaboración y por los buenos momentos compartidos en el grupo de investigación.

A todos los compañeros, amigos y demás personas que formaron parte de nuestra formación académica, por el apoyo y colaboración en esta importante etapa.

A Dios por permitirnos culminar satisfactoriamente nuestras metas y superar las etapas difíciles que se pudieron presentar.

DEDICATORIA

Dedico este triunfo a mis padres por el apoyo incondicional en todos los momentos y etapas de mi vida. Por el esfuerzo que han realizado para que pueda estar cumpliendo uno de mis sueños. Por el cariño brindado, los consejos recibidos y las enseñanzas compartidas. Por haberme formado con valores y principios para ser una excelente persona y profesional.

A mi hermana Juliana Sofía, porque es una motivación adicional para poder salir adelante y sobrepasar todos los obstáculos de la vida. Por ser una compañera, amiga y que a pesar de las condiciones adversas, recordarme cada día la importancia de soñar.

A mis amigos y compañeros que en el aula de clases compartieron conmigo parte de su experiencia y conocimiento. Por el apoyo durante los momentos difíciles y por todos los momentos vividos a lo largo de este proceso.

Yamid Esteban Estrada Bastidas

DEDICATORIA

Dedico este trabajo a mis padres que siempre me inculcaron sus valores y principios, gracias a ellos soy una persona de bien, responsable y dedicada en todo lo que hago, por siempre confiar en mí y apoyarme en todo momento para cumplir este logro.

A mis Hermanos, por siempre preocuparse por mí y brindar su apoyo incondicional en cualquier momento que lo necesite.

A mi hermano Luis, por ser un ejemplo de inspiración en mi vida y una motivación más para seguir adelante.

A Eliana Carlosama, que siempre me apoyo en los momentos difíciles y gracias a su compañía estoy cumpliendo un logro más en mi vida junto a ella.

A mi prima Mayra, que aunque hoy ya no esté con nosotros, los mejores recuerdos nos queda de ella. Siempre nos enseñó a vivir con alegría.

David Reinaldo Eraso Gallo

DEDICATORIA

Dedico este trabajo a Dios, por brindarme la vida y la oportunidad de llegar a este punto tan importante de mi formación como profesional.

A mis padres, por su cariño, confianza y apoyo brindado durante toda mi formación y por estar presentes en los momentos más significativos de mi vida, guiándome por el camino correcto y haciendo de mí un buen hijo y una buena persona.

A mis hermanos por estar siempre a mi lado brindándome su apoyo incondicional.

A mi hermana Consuelo, mi cuñado Miguel y mi sobrino Oswaldo por abrirme las puertas de su casa, acogerme en su hogar y acompañarme durante mi proceso de formación, con quienes pasamos buenos momentos y logramos superar las dificultades presentadas como una familia.

A mis amigos y compañeros de la universidad por su calidad humana y porque siempre quedarán en mi los mejores recuerdos. Gracias a todos.

Wilmer Alexander Escobar Botina

RESUMEN

En este documento se presentan los resultados del trabajo de investigación cuyo objetivo fue diseñar y desarrollar un visor cartográfico a nivel de direcciones urbanas y sitios de interés para el municipio de Pasto (Colombia) haciendo uso de PostGIS una extensión del SGBD PostgreSQL para el manejo de base de datos espaciales. El visor cartográfico Rikhuna, como se ha denominado a esta herramienta, permite a los usuarios localizar fácilmente sobre la cartografía del municipio de Pasto una dirección urbana, barrio, vereda, corregimiento o sitios de interés tales como lugares turísticos, centros educativos, centros administrativos, hospitales, centros de atención inmediata, entre otros y además es posible generar diferentes tipos de rutas, con información detallada para llegar hasta ellos. Rikhuna está acoplado con una ontología que permite realizar búsquedas de manera semántica e inteligente y para su desarrollo se utilizaron herramientas de software libre. Con este visor se busca generar impacto en el municipio de Pasto, abriendo sus puertas al mundo y creando una herramienta que puede ser replicable en otras ciudades que busquen objetivos similares.

Palabras claves: Georreferenciación, PostGIS, Ontología, Visor cartográfico, Municipio de Pasto.

ABSTRACT

This document presents the results of the research project that aims to design and develop a cartographic viewer of street addresses and places of interest in the city of Pasto based on PostGIS, an extension of DBMS PostgreSQL for the management of spatial database. The viewer Rikhuna allows users to easily locate on a cartography of the city, an address urban, neighborhood, sidewalk, districts or sites of interest such as tourist sites, educational centers, administrative centers, hospitals, immediate attention and others. Also it's possible to generate different types of routes, with detailed information to reach them. Rikhuna works together with an ontology that allows to perform searches for a semantic and intelligent way and for its development were used free software tools. This viewer seeks to generate impact in the municipality of Pasto, opening its doors to the world and this tool that may be replicable in other cities with similar objectives.

Keywords: Georeferencing, PostGIS, Ontology, Cartographic Viewer, City of Pasto.

CONTENIDO

INTRODUCCIÓN	17
1. ANTECEDENTES	22
2. MARCO TEÓRICO	24
2.1. SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG)	24
2.1.1. Dato e información geográfica.....	24
2.1.2. Representación de los datos geográficos.....	24
2.1.3. Cartografía.....	25
2.1.4. Geocodificación de eventos.....	25
2.2. BASES DE DATOS	26
2.2.1. Base de datos espacial.....	27
2.2.2. PostgreSQL.....	27
2.2.3. PostGIS.....	27
2.3. WEB SEMÁNTICA.....	29
2.3.1. Ontología.....	29
2.3.2. Buscador semántico.....	31
2.4. ESTÁNDAR DE DIRECCIONES URBANAS	31
2.5. SCRUM.....	33
2.6. TECNOLOGÍAS USADAS	35
2.6.1. Java OpenStreetMap.....	35
2.6.2. Protégé.....	35
2.6.3. Jena.....	36
2.6.4. Java Server Faces.....	36
2.6.5. Geoserver.....	37
3. METODOLOGÍA	38
4. ARQUITECTURA.....	41
4.1. ARQUITECTURA DEL VISOR CARTOGRÁFICO.....	41
4.2. ARQUITECTURA DEL PANEL ADMINISTRATIVO.....	42
5. RESULTADOS.....	45
5.1. ETAPA DE APROPIACIÓN DEL CONOCIMIENTO	45
5.2. ETAPA DE RECOLECCIÓN DE INFORMACIÓN GEOGRÁFICA.....	45
5.2.1. Recolección de puntos.....	45
5.2.2. Recolección de geometrías.....	50
5.2.3. Utilización de GPS.....	53
5.3. ETAPA DE CONSTRUCCIÓN DE BASE DE DATOS Y ONTOLOGÍA	54
5.3.1. Construcción de la base de datos del visor cartográfico.....	54
5.3.2. Construcción de la base de datos del panel administrativo.....	60
5.3.3. Creación de la ontología Web.....	61
5.4. ETAPA DE ANÁLISIS DE HERRAMIENTAS.....	62
5.5. CONSTRUCCIÓN DE RIKHUNA	65
5.5.1. Implementación visor cartográfico.....	65

5.5.2.	Construcción visor cartográfico.....	70
5.5.3.	Implementación panel administrativo.....	75
5.5.4.	Construcción panel administrativo.:	82
5.6.	APLICACIÓN Y ANÁLISIS DE PRUEBAS.....	86
5.6.1.	Casos de prueba	87
5.6.2.	Pruebas unitarias.....	90
6.	CONCLUSIONES	92
7.	RECOMENDACIONES	94
	BIBLIOGRAFÍA.....	95

LISTA DE TABLAS

Tabla 1. Estándar de abreviaturas más usadas en el municipio de Pasto.....	33
Tabla 2. Casos para la nomenclatura vial.....	45
Tabla 3. Direcciones urbanas obtenidas en la etapa de recolección	47
Tabla 4. Mapas turísticos del Municipio de Pasto.....	48
Tabla 5. Recolección de sitios de interés por comuna.....	48
Tabla 6. Recolección de sitios de interés por corregimiento.....	49
Tabla 7. Barrios recolectados por comuna	51
Tabla 8. Veredas recolectadas por corregimientos.....	52
Tabla 9. Datos iniciales de fuentes externas	56
Tabla 10. Direcciones estandarizadas de fuentes externas.....	56
Tabla 11. Direcciones faltantes de fuentes externas	56
Tabla 12. Direcciones urbanas obtenidas en la etapa de corrección.....	59
Tabla 13. Direcciones de malla vial recolectadas por comuna	59
Tabla 14. Direcciones de manzana recolectadas por comuna	59
Tabla 15. Comparación de lenguajes de programación con soporte a ontologías	62
Tabla 16. Criterios de evaluación para bibliotecas de búsqueda semántica.....	62
Tabla 17. Comparación funcionalidades bibliotecas de búsqueda semántica	63
Tabla 18. Criterios de evaluación de APIs para la creación de rutas.....	64
Tabla 19. Comparación de APIs para la creación de rutas.....	65
Tabla 20. Paquetes módulo de presentación – panel administrativo	78
Tabla 21. CP-001 Estandarización de direcciones malla vial	87
Tabla 22. CP-002 Estandarización de direcciones manzana.....	87
Tabla 23. CP-003 Búsqueda y localización de sitios de interés.....	88
Tabla 24. CP-004 Ubicación de direcciones con dispositivo GPS	89
Tabla 25. CP-004 Ubicación de sitios de interés con dispositivo GPS	89
Tabla 26. Recopilación de pruebas unitarias	90

LISTA DE FIGURAS

Figura 1. Proceso general de geocodificación	26
Figura 2. Esquema vial según IGAC.....	32
Figura 3. Esquema general de Scrum	34
Figura 4. Metodología de la investigación.....	38
Figura 5. Arquitectura general del visor cartográfico.....	41
Figura 6. Arquitectura por paquetes del visor cartográfico.....	42
Figura 7. Arquitectura general del panel administrativo	43
Figura 8. Arquitectura por paquetes del panel administrativo	44
Figura 9. Comparación ciudad de la Plata (Arg) y ciudad de Pasto (Col)	46
Figura 10. Puntos geográficos de direcciones urbanas recolectadas	47
Figura 11. Mapas turísticos del Municipio de Pasto.....	49
Figura 12. Mapas turísticos por corregimiento del municipio de Pasto	49
Figura 13. Comunas del Municipio de Pasto	50
Figura 14. Barrios del Municipio de Pasto	51
Figura 15. Corregimientos del municipio de Pasto	52
Figura 16. Veredas del municipio de Pasto	53
Figura 17. GPS Garmin GPSMAP 64s y recolección de sitio de interés rural	53
Figura 18. Diagrama Entidad-Relación base de datos del visor cartográfico.....	54
Figura 19. Distancia entre direcciones de malla vial	59
Figura 20. Diagrama Entidad-Relación base de datos del panel administrativo	60
Figura 21. Diseño de la ontología web.....	61
Figura 22. Módulo presentación visor cartográfico.	66
Figura 23. Clases del paquete visor.controlador.....	67
Figura 24. Clases del paquete visor.ontologia	68
Figura 25. Clases del paquete visor.ruta	68
Figura 26. Clases del paquete visor.util	69
Figura 27. Clases del paquete visor.jpj.....	70
Figura 28. Clases del paquete visor.ejb.....	70
Figura 29. Búsqueda de dirección basada en malla vial.....	71
Figura 30. Búsqueda de dirección basada en manzana	71
Figura 31. Búsqueda de sitio específico	72
Figura 32. Búsqueda de categoría.....	72
Figura 33. Búsqueda de sitios cercanos	73
Figura 34. Búsqueda de barrios.....	73
Figura 35. Búsqueda de corregimientos	73
Figura 36. Generación de rutas	74
Figura 37. Ubicación del usuario	75
Figura 38. Vista google Street view	75
Figura 39. Módulo presentación panel administrativo.....	76
Figura 40. Interacción Usuario - Módulo Registro.....	76

Figura 41. Interacción Usuario - Módulo inicio de sesión.....	77
Figura 42. Clases del paquete login.controlador.....	79
Figura 43. Clases del paquete login.correo	80
Figura 44. Clase del paquete login.encriptar	80
Figura 45. Clases del paquete login.jpá.....	81
Figura 46. Clases del paquete login.ejb.....	81
Figura 47. Formulario de registro de usuarios	82
Figura 48. Formulario de inicio de sesión	82
Figura 49. Vista principal panel administrativo.....	83
Figura 50. Formulario para la creación de dirección malla vial	83
Figura 51. Formulario para la eliminación de dirección manzana.....	84
Figura 52. Formulario para la edición de sitio de interés	85
Figura 53. Vista principal usuario administrador	86

LISTA DE ANEXOS

ANEXO A. ANÁLISIS SCRUM.....	103
ANEXO B. MANUAL DE INSTALACIÓN VISOR CARTOGRÁFICO.....	120
ANEXO C. MANUAL DE USUARIO.....	124
ANEXO D. MANUAL DEL PROGRAMADOR.....	145
ANEXO E. DICCIONARIO DE DATOS.....	174
ANEXO F. PRUEBAS UNITARIAS.....	180

INTRODUCCIÓN

La geocodificación como proceso permite asignar coordenadas geográficas a los datos temáticos de eventos mediante el procesamiento de objetos espaciales asociados a esta información. Estas coordenadas pueden ser utilizadas para desplegar, localizar y visualizar con detalle estos eventos en mapas cartográficos digitales. Realizar este proceso con alto grado de aproximación a la realidad geográfica es un objetivo fundamental que conlleva a considerar diferentes aspectos sobre este proceso. La información cartográfica debe tener suficiente detalle sobre el área geográfica de estudio que permita identificar con precisión la ubicación que se está buscando, ESRI [1]. Además, González y Torres [2] aconsejan realizar una sistematización de los datos geográficos existentes en la red y las diversas posibilidades de manejo y visualización de la misma con el objetivo de difundir aún más su conocimiento y utilización, tanto en visores como en sistemas de información geográfica robustos.

En la actualidad los visores cartográficos son una herramienta esencial para la visualización y consulta de datos geográficos de forma rápida y sencilla, permitiendo conocer información tomada del mundo real. Los Sistemas de Información Geográfica (SIG) surgen como herramientas de apoyo en la toma de decisiones que permiten procesar esta información desde su dimensión espacial y suministrar datos para la formulación de soluciones a problemas de orden geográfico, Huisman y By [3].

Dada la importancia que tiene un visor cartográfico para la representación de la información, es necesario que la región cuente con un sistema de este tipo. A pesar de la riqueza que posee el municipio de Pasto, por su ubicación geográfica y el constante intercambio cultural, económico y social que se presenta en la región debido al paso de personas que salen e ingresan al país, se puede afirmar que Pasto no cuenta con un sistema que proporcione información geográfica detallada de su división territorial, direcciones urbanas, sitios de interés y demás lugares emblemáticos que a lo largo de la historia han formado parte de la identidad de los pastusos, de tal manera que facilite la ubicación a propios y visitantes.

GENERALIDADES DEL PROYECTO

MODALIDAD

Este trabajo corresponde a la modalidad de trabajo de investigación desarrollado por estudiantes investigadores del Grupo de Investigación Aplicado en Sistemas (GRIAS) del Departamento de Sistemas de la Universidad de Nariño.

LÍNEA DE INVESTIGACIÓN

Esta investigación corresponde a la línea software y manejo de información.

ALCANCE Y DELIMITACIÓN

La investigación comprende el área geográfica del municipio de Pasto, que corresponde a 12 comunas en el área urbana y 17 corregimientos para el área rural. Se recopilaron las direcciones de malla vial y manzanas de las comunas del municipio según la norma de estandarización de nomenclaturas urbanas propuesta por el Instituto Geográfico Agustín Codazzi de Colombia, así como los sitios de interés clasificados por categorías ubicados geográficamente a nivel de comunas y corregimientos.

El visor cartográfico cuenta con un mapa interactivo donde se muestran los resultados de la búsqueda realizada por el usuario con información detallada. Entre las funcionalidades se encuentran la búsqueda de direcciones por malla vial y por barrio-manzana-predio, sitios de interés, barrios, corregimientos y además de la generación de rutas.

También se cuenta con un módulo administrativo integrado al visor principal con el fin de gestionar (crear, editar y eliminar) la información de direcciones urbanas, sitios de interés y usuarios.

PLANTEAMIENTO DEL PROBLEMA

San Juan de Pasto, la Ciudad Sorpresa de Colombia y capital del departamento de Nariño, se destaca por contar con muchas riquezas naturales y culturales que han formado parte del pueblo pastuso a lo largo de su historia; tras el auge tecnológico y el inminente crecimiento de la ciudad se identifica la carencia de suficientes recursos especializados que permitan darse a conocer al mundo.

Entre los múltiples factores a los que se le puede atribuir este hecho esta la inexistencia de información geográfica libre del municipio, que de alguna manera facilite el trabajo de desarrolladores y demás interesados en la construcción de herramientas tecnológicas para la ciudad, las cuales al ser enfocadas en Sistemas de Información Geográfica (SIG) brinden nuevas y modernas formas de conocer a detalle la distribución espacial del municipio.

Muchos de los llamados servidores de mapas de libre acceso no cuentan con funcionalidades que permitan llevar a cabo la búsqueda de una dirección urbana, un sitio de interés o divisiones territoriales (barrios, veredas, corregimientos y comunas) en el municipio de Pasto, de forma detallada, teniendo en cuenta información real y actualizada. Varios de estos servidores no muestran los resultados con la ubicación que realmente corresponde a la búsqueda realizada y brindan información errónea a los usuarios. Esto se presenta porque la ubicación que se provee es obtenida a partir del uso de algoritmos de reconocimiento de vías en donde se estima la ubicación de la dirección, este proceso no garantiza que el resultado obtenido sea acorde a la realidad.

En este trabajo de investigación se propone la creación de un visor cartográfico inteligente basado en PostGIS, que permita visualizar direcciones urbanas, sitios de interés, divisiones territoriales y la generación de rutas en el municipio de Pasto. En Rikhuna se podrá contar con información detallada de los sitios referenciados ya que se construirá una base de datos espacial de uso libre, lo que permitirá a usuarios en general disponer de la ubicación de sitios turísticos, universidades, centros administrativos, hospitales, clínicas, centros de atención inmediata y demás información geográfica de la ciudad. Además, Rikhuna estará integrado con una ontología que facilite la búsqueda de los sitios mencionados. Este sistema será desarrollado haciendo uso de la herramienta gestora de datos espaciales PostGIS del gestor de bases de datos PostgreSQL, lo que significa que está desarrollado bajo software libre, y será de libre distribución.

FORMULACIÓN DEL PROBLEMA

¿Cómo es posible facilitar la búsqueda y localización de direcciones urbanas y sitios de interés en el municipio de Pasto?

OBJETIVOS

Objetivo general

Facilitar la búsqueda y localización de direcciones urbanas y sitios de interés del municipio de Pasto a través de la construcción de un visor cartográfico inteligente basado en la herramienta de software libre PostGIS.

Objetivos específicos

- ✓ Apropiar de conocimientos acerca de conceptos relacionados con sistemas de información geográfica, proceso de estandarización de direcciones urbanas, ontologías y búsquedas semánticas.
- ✓ Conocer el funcionamiento y la implementación de la herramienta PostGIS.
- ✓ Recolectar información de direcciones urbanas y sitios de interés en el municipio de Pasto.
- ✓ Diseñar y construir una base de datos con direcciones urbanas y sitios de interés del Municipio de Pasto definidas según el estándar de la Institución Geográfica Colombiana Agustín Codazzi.
- ✓ Diseñar y construir una ontología para la búsqueda inteligente de direcciones urbanas y sitios de interés del municipio de Pasto.
- ✓ Definir la herramienta apropiada para la construcción del visor cartográfico.
- ✓ Realizar la documentación del funcionamiento del visor cartográfico.
- ✓ Realizar pruebas, análisis y evaluación de resultados obtenidos.

JUSTIFICACIÓN

En el mundo actual es de gran importancia que la información esté disponible para todos en cualquier momento y lugar. Las nuevas tecnologías se han convertido en una parte fundamental del crecimiento acelerado que han tenido las ciudades debido en parte a la implementación de sistemas de información que facilitan el desarrollo de las actividades cotidianas. Teniendo en cuenta la gran cantidad de información circulando por la red, es pertinente desarrollar sistemas que faciliten su acceso y administración, de manera que sea presentada de forma clara y detallada. En el caso de sistemas que involucran datos geográficos, su implementación permite gestionar y mostrar información de una determinada área o región mediante el uso de mapas interactivos.

Con la ejecución de este trabajo el municipio de Pasto contará con un visor cartográfico inteligente que ubicará y mostrará información de direcciones urbanas, barrios, comunas, veredas, corregimientos y sitios de interés tales como lugares turísticos, centros educativos, centros administrativos, hospitales, centros de atención inmediata, entre otros, permitiendo a los habitantes locales y visitantes conocer información detalladamente sobre direcciones y lugares representativos de la ciudad.

La funcionalidad de generación de rutas facilitará el desplazamiento y ubicación de los usuarios entre los distintos puntos de búsqueda por toda el área del municipio de Pasto, en el cual los sitios de interés cercanos se asociarán al camino generado. Por otro lado, el visor cartográfico cuenta con un módulo administrativo que permitirá gestionar la información de toda la base de datos del sistema, facilitando los procesos de inserción, actualización y eliminación de direcciones y sitios de interés. Adicional a esto, Rikhuna se convertirá en una herramienta impulsora del sector turístico regional que permitirá que las nacientes industrias locales tengan un mayor reconocimiento para propios y visitantes.

Para el trabajo se utilizarán herramientas de software libre lo que permitirá ser reutilizado por cualquier organización en el municipio de Pasto que requiera georreferenciar direcciones urbanas y sitios de interés. Finalmente, los investigadores de GRIAS adquirirán experiencia investigativa en el área de sistemas de información geográfica, web semántica, georreferenciación de direcciones urbanas y sitios de interés para cualquier municipio, departamento o país.

1. ANTECEDENTES

En la búsqueda de antecedentes que beneficien al proceso de contextualización de la investigación se encontraron proyectos realizados en diferentes áreas de aplicación, en su mayoría orientados a la creación de sistemas de información geográfica, georreferenciación de eventos y ontologías. A continuación, se mencionan algunos proyectos destacados.

En el ámbito regional se identifica el trabajo de investigación denominado DETECCIÓN DE PATRONES DELICTIVOS CON TÉCNICAS DE MINERÍA DE DATOS EN EL OBSERVATORIO DEL DELITO DEL MUNICIPIO DE PASTO, financiado mediante convenio 615-2011 por COLCIENCIAS con contrapartida de la Universidad de Nariño y la Alcaldía Municipal de Pasto y ejecutado por los investigadores del Grupo de Investigación Aplicada en Sistemas GRIAS del departamento de Sistemas de la Universidad de Nariño. En la primera etapa de este trabajo se desarrolló el software SIGODEP, un sistema de información georreferenciado del Observatorio del Delito de Pasto bajo software libre [4]. Una de las funcionalidades que soporta SIGODEP es la generación de mapas temáticos, a partir de cruce de variables que describen las lesiones fatales y no fatales que ocurrieron en el municipio de Pasto desde el año 2003 hasta año 2014. A partir de esta funcionalidad se identifica la visualización de estadísticas correspondientes a estas incidencias desplegada en geometrías espaciales a nivel de barrios y comunas como el gránulo más pequeño de georreferenciación.

Se destaca también el proyecto denominado CONSTRUCCIÓN DE UNA ONTOLOGÍA DE APLICACIÓN QUE SOPORTE LA BÚSQUEDA INTELIGENTE SOBRE LOS TRABAJOS DE GRADO DE LA UNIVERSIDAD DE NARIÑO, UTILIZANDO LA HERRAMIENTA DE SOFTWARE LIBRE PROTÉGÉ [5] desarrollado por investigadores del Grupo de Investigación Aplicada en Sistemas GRIAS del departamento de Sistemas de la Universidad de Nariño, mediante el cual se construye una ontología para la búsqueda inteligente de trabajos de grado de los estudiantes de pregrado de la Universidad de Nariño. La ontología denominada Sawa contempla diversas etapas donde se diseña y define los componentes, implementa la ontología y finalmente se aplica las pruebas correspondientes. El estudio realizado concluye que el dominio de la ontología puede ampliarse a la totalidad del material disponible en la biblioteca de la universidad.

En el ámbito nacional se debe mencionar el geoportal del Instituto Geográfico Agustín Codazzi – IGAC [6], que es la entidad encargada de producir el mapa oficial y la cartografía en Colombia. Dentro de esta plataforma se destaca principalmente el Mapa del Sistema Nacional Catastral que permite la consulta de información sobre los predios a lo largo del territorio regional y nacional, por medio de cédula

catastral, matrícula inmobiliaria o dirección. Con la ayuda de un mapa interactivo es posible visualizar la ubicación de los predios del área urbana del municipio de Pasto con el fin de facilitar el proceso de recolección de información geográfica de direcciones urbanas.

En el ámbito internacional destaca el proyecto denominado ANÁLISIS DE LA TECNOLOGÍA JAVASERVER FACES, COMO FRAMEWORK DE DESARROLLO DE APLICACIONES WEB, CASO PRÁCTICO: PROCESOS DE COMERCIALIZACIÓN DE COMBUSTIBLES DE EP-PETROECUADOR [7]. Se plantea un estudio sobre el framework JSF como alternativa para el desarrollo de aplicaciones web y se hace un análisis comparativo con otro framework Java Web llamado Struts. También se desarrolla una aplicación web para procesos de comercialización de combustibles usando JSF. Se mencionan las ventajas y desventajas de las dos tecnologías teniendo en cuenta diversos criterios de evaluación, como pueden ser lógica de navegación, líneas de código generadas, validación y conversión de datos, soporte AJAX, entre otros. Como resultado de este trabajo se menciona que JSF mejora el rendimiento de procesos en línea, tiempos de espera para el proceso de comercialización de combustibles.

Finalmente, se menciona el proyecto MODELO PARA LA REPRESENTACIÓN DE CONOCIMIENTO BASADO EN ONTOLOGÍAS WEB Y AGENTES INTELIGENTES DE BÚSQUEDA, CASO DE ESTUDIO: ARTÍCULOS CIENTÍFICOS WSIA [8], que plantea la creación de una herramienta para la búsqueda de información de artículos científicos por medio de agentes inteligentes y ontologías web. Este trabajo muestra la arquitectura, implementación y un análisis comparativo con el desarrollo basado en tecnologías tradicionales usando PHP y MySQL. El estudio permitió establecer que con el uso de los agentes inteligentes y ontologías web fue posible mejorar tiempos de respuesta y concordancia en las búsquedas. Es posible ampliar el alcance del proyecto llevándolo a empresas del sector público.

2. MARCO TEÓRICO

2.1.SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG)

Un sistema de información geográfica (SIG o GIS, en su acrónimo inglés) es “un sistema informático capaz de realizar las tareas para manejar datos georreferenciados: entrada, almacenamiento, recuperación, manipulación, análisis y representación” como lo menciona Aranoff [9], posteriormente Burrough y Mcdonell [10] definen a un SIG como “Un conjunto de herramientas para reunir, introducir, almacenar, recuperar, transformar y cartografiar datos espaciales sobre el mundo real con el fin de satisfacer múltiples propósitos”.

En términos prácticos, un SIG está integrado por un componente grafico que permite representar, visualizar y analizar la información geográfica y una base de datos espacial que permite almacenar, manipular y consultar dicha información.

2.1.1.Dato e información geográfica. La información geográfica es un conjunto de datos geográficos procesados y organizados, en donde los datos geográficos o espaciales son características geográficas de los objetos descritos (ubicación, dimensión, forma), respecto a una posición en un sistema de coordenadas terrestres.

Este tipo de información es realmente útil en búsquedas de soluciones a problemas de orden geográfico. Goodchild, Longley, Rhind y Maguire [11] identifican algunos de estos posibles problemas:

- ✓ Ubicación y georreferenciación de predios, entidades de servicios públicos, entidades gubernamentales, etc.
- ✓ Identificación de rutas para construcción de nuevas vías.
- ✓ Determinación de rutas optimas de recorrido.
- ✓ Ubicación estratégica de negocios o sucursales de ventas.
- ✓ Gestión y preservación de recursos naturales.
- ✓ Gestión y promoción de sitios turísticos y de interés.

2.1.2.Representación de los datos geográficos. Según Bernabé y López [12] los datos geográficos se representan en dos tipos de modelos, estos son:

- ✓ **Modelo Vectorial:** en este modelo los elementos espaciales son representados por coordenadas geográficas, las geometrías más utilizadas son:
 - **Elementos puntuales:** se representan a través de un par de coordenadas geográficas (latitud, longitud) que definen su posición en la superficie de la tierra.

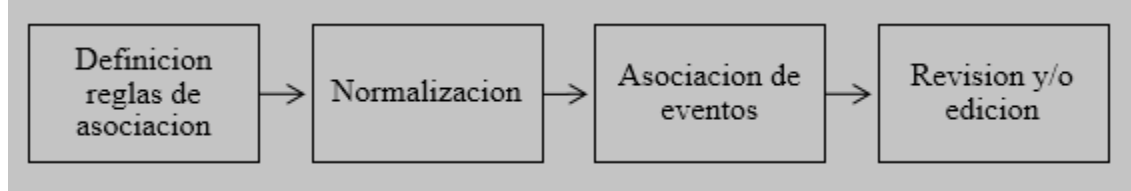
- **Elementos lineales:** se representan a través de dos o más coordenadas geográficas que forman una línea en donde el primero y último punto deben ser diferentes.
 - **Elementos superficiales:** se representan a través de dos o más coordenadas geográficas que forman un polígono de recinto cerrado. Para formar esta geometría se necesitan al menos tres puntos diferentes no alineados en donde el primero y el último punto deben ser iguales.
- ✓ **Modelo Raster:** en este modelo los elementos espaciales son representados por pixeles que forman parte de una imagen, fotografía o modelo digital. Para conseguir esto se realiza una superposición de una rejilla de unidades poligonales de igual forma y tamaño, en donde cada pixel se le asigna un valor al área que representa de la imagen original, a este proceso se le llama rasterización.

2.1.3. Cartografía. La cartografía se encarga del estudio y recolección de datos de regiones de la tierra para de esta manera representarlos de forma visual y clara a una escala menor de la real. Generalmente, se hace uso de mapas para poder llevar a cabo el estudio de la superficie terrestre, relieves, perspectivas, cartogramas, etc. La herramienta principal en el estudio de la cartografía es el mapa, con el cual se agregan rótulos para la identificación de los detalles más importantes de la superficie terrestre esto según Raisz [13].

En términos generales, la Asociación Cartográfica Internacional (ICA por sus siglas en inglés) define la cartografía como un conjunto de estudios y técnicas científicas y artísticas, las cuales parten de la base de resultados de las observaciones directas o el análisis de documentación para de esta manera elaborar mapas y demás formas de representación de objetos, elementos geográficos, entre otros.

2.1.4. Geocodificación de eventos. Se puede definir geocodificación o geolocalización como la acción de atribuir coordenadas geográficas a una descripción de una ubicación como una dirección, lugar o cualquier tipo de evento esto según ArcGIS [14]. Estas coordenadas pueden ser utilizadas para desplegar, localizar y visualizar con detalle la información relacionada a estos eventos en mapas cartográficos digitales. En la Figura 1, se muestra el proceso general de geocodificación de eventos expuesto por ESRI [15].

Figura 1. Proceso general de geocodificación



Fuente: ArcGIS 9, Geocoding Rule Base Developer Guide. ESRI, 2003.

Las reglas de asociación de eventos definen aspectos del método de geocodificación a utilizar como la estructura de la base de datos, atributos sobre los cuales se realizará la geocodificación, atributos que deben indexarse para optimización de búsquedas, manejo de errores, criterios de evaluación de resultados, entre otros. La normalización hace referencia a describir los atributos georreferenciados de los datos descriptivos y los datos de referencia ajustándose a normas o formatos de abreviaturas o escritura. En la asociación de eventos el software compara cada registro descriptivo con cada registro de referencia mediante el atributo seleccionado generando puntajes que determinan el grado de similitud de estos dentro de un rango de aceptación establecido. De acuerdo con estas revisiones se ajustan las reglas definidas para obtener resultados acordes a los requerimientos de información establecidos según el caso de estudio.

2.2.BASES DE DATOS

Una base de datos según Oppel y Sheldon [16] se define como una colección de datos organizada en un formato estructurado que es definido como metadatos que describe esa estructura. Se puede pensar en los metadatos como información sobre los datos almacenados, que define cómo se almacenan éstos en una base de datos.

En los últimos años se ha implementado varios modelos de base de datos para almacenar y administrar la información. Los más comunes que mencionan Oppel y Sheldon, son los siguientes:

- ✓ **Jerárquico:** modelo que tiene una organización similar a un árbol invertido, esto es lo que forma la jerarquía. Los datos se organizan en nodos, el equivalente lógico de las tablas en el modelo de base de datos relacional. El nodo principal puede tener bastantes nodos secundarios, pero un nodo secundario solo puede tener un nodo principal.
- ✓ **Red:** al igual que el modelo jerárquico, el modelo de red usa la estructura de un árbol invertido, pero los tipos de registro se organizan en una estructura que relaciona pares de tipos de registro. Cualquier tipo de registro puede participar en cualquier conjunto con otros tipos de registro en la base de datos, que apoya a las consultas y relaciones más complejas de lo que es posible en el modelo jerárquico.

- ✓ **Relacional:** este modelo aborda algunas de las limitaciones de los modelos jerárquicos y de red. Una base de datos relacional es independiente de la aplicación; puede hacer modificaciones no destructivas a la estructura sin afectar la aplicación. Además, la estructura de la base de datos relacional se basa en la relación o tabla, junto con la capacidad de definir relaciones complejas entre ellas. Se puede acceder directamente a cada relación sin la lentitud de los modelos jerárquicos que requiere de una navegación a través de una estructura compleja de datos. El modelo relacional es el más ampliamente aplicado en los sistemas de negocios modernos y es el modelo relacional el que proporciona la base para SQL. El lenguaje estructurado de consultas SQL (Structured Query Language) apoya la creación y mantenimiento de la base de datos relacional y la gestión de los datos dentro de la misma.

2.2.1.Base de datos espacial. Es una colección lógica de datos sobre características que se refieren a ubicaciones en la superficie terrestre interrelacionada que se maneja y almacena como una unidad. Una base de datos de SIG contiene información sobre la ubicación de las características del mundo real y sobre sus particularidades como lo menciona las Naciones Unidas [17]. Es importante indicar que, para el almacenamiento de la información geográfica, se consideran las bases de datos espaciales, en las que se almacenan de manera estructurada los objetos cartográficos (posición, tamaño y forma) y sus características no geométricas (atributos alfanuméricos) respectivamente esto según Galán y Alegría [18]. En algunos sistemas las dos bases de datos están separadas, mientras que en otros se integran en una sola.

2.2.2.PostgreSQL. La página oficial [19] menciona que PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo una licencia similar a la BSD (Berkeley Software Distribution) y con su código fuente disponible libremente. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y corrección. Se ejecuta en todos los sistemas operativos, incluyendo Linux, UNIX, Mac OS X, Solaris y Windows.

PostgreSQL utiliza un modelo cliente/servidor y es compatible con el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos, o de vídeo. El desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada.

2.2.3.PostGIS. La página oficial [20] menciona que PostGIS es una extensión del gestor de base de datos objeto-relacional PostgreSQL que permite almacenar, consultar y manipular datos espaciales, convirtiéndola en una base de datos espacial para su utilización en sistemas de información geográfica.

PostGIS fue desarrollado por Refraction Research Inc, como un proyecto de investigación de código abierto en tecnología de bases de datos espaciales y está liberado bajo la licencia GNU General Public License (GPLv2 o posterior). Además, está financiado constantemente por muchos desarrolladores, así como por compañías de todo el mundo que hacen beneficios gracias a su funcionalidad y versatilidad.

PostGIS soporta cuatro tipos de datos mencionados por Obe y Hsu [21], los cuales son:

- ✓ **Geometry:** tipo plano. Este fue el primer modelo y sigue siendo el tipo más popular que PostGIS soporta, además es la base de los otros tipos. Se denominó así porque su base es la geometría analítica (geometría cartesiana). Por ejemplo, el camino más corto entre dos puntos en un plano es una línea recta. Esto significa que los cálculos en geometrías (áreas, distancias, longitudes, intersecciones, etc.) pueden calcularse utilizando matemáticas cartesianas y vectores lineales.
- ✓ **Geography:** tipo geodésico esferoidal. Proporciona soporte nativo para objetos espaciales representados por coordenadas geográficas, (a veces llamadas coordenadas geodésicas) expresados en unidades angulares (grados). La base del tipo geography es la superficie curvada de la tierra. Por ejemplo, el camino más corto entre dos puntos en la esfera es el arco de circunferencia, esto significa que los cálculos geográficos (áreas, distancias, longitudes, intersecciones, etc.) deben calcularse en la esfera, utilizando matemáticas más complejas, debido a esto los cálculos en él son más lentos de lo que son en geometry.
- ✓ **Raster:** tipo de celda multibanda. Modela el espacio como una cuadrícula de celdas rectangulares, cada una contiene una matriz numérica de valores. Este tipo de dato organiza la información mediante píxeles.
- ✓ **Topology:** tipo de modelo relacional. La topología modela el mundo como una red de nodos conectados, aristas y caras. Los objetos se componen de éstos y pueden compartirlos con otros objetos.

PostGIS ofrece muchas funciones y operadores para cada uno de sus tipos de datos, a continuación, se muestran las más utilizadas, teniendo en cuenta que existe un mayor número para geometry y otras funciones solo están disponibles para geography por ser más complejo en su manejo.

- ✓ **Funciones de setters y getters:** estas funciones trabajan con un único objeto espacial y devuelven o establecen atributos del objeto.

- ✓ **Funciones de medición:** estas funciones devuelven mediciones escalares de un objeto espacial.
- ✓ **Funciones de descomposición y composición:** estas funciones extraen y agrupan objetos espaciales de un objeto espacial de entrada.
- ✓ **Funciones de simplificación:** estas funciones devuelven representaciones simplificadas mediante la eliminación de puntos o cadenas de líneas o redondeando las coordenadas. Los objetos espaciales resultantes todavía tienen la apariencia básica de los originales, pero contienen menos puntos o elementos de menor precisión.

2.3.WEB SEMÁNTICA

La web semántica es adoptada por W3C [22] quien la define como “una web extendida, dotada de mayor significado en la que cualquier usuario en internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida”.

La web semántica permite solucionar problemas habituales en la búsqueda pues ayuda a resolver problemas de comunicación, transacciones económicas y el acceso a recursos web independientemente de la ubicación geográfica o idioma del cliente. También permite que los productos software sean capaces de llevar a cabo un mejor procesamiento y razonamiento de contenidos, además de realizar deducciones lógicas que resuelven problemas automáticamente.

2.3.1.Ontología. Horridge [23] menciona que las ontologías se usan para capturar conocimiento sobre algún dominio de interés. Una ontología describe los conceptos en el dominio y también las relaciones que mantienen entre esos conceptos. Studer, Benjamins y Fensel [24] en su definición de ontología mencionan que “es una especificación formal y explícita de una conceptualización compartida”, donde “Conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo a través de la identificación de los conceptos relevantes de dicho fenómeno. Explícita significa que el tipo de conceptos y restricciones usados se definen explícitamente. Formal representa el hecho de que la ontología debería ser entendible por las máquinas. Compartida refleja la noción de que una ontología captura conocimiento consensual, esto es, que no es de un individuo, sino que es aceptado por un grupo”.

Las ontologías tienen componentes que servirán para representar el conocimiento de algún dominio según Horridge, estos son:

- ✓ **Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- ✓ **Relaciones:** representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio.
- ✓ **Funciones:** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.
- ✓ **Instancias:** se utilizan para representar objetos determinados de un concepto.
- ✓ **Axiomas:** son teoremas que se declaran sobre las relaciones que deben cumplir los elementos de la ontología. Permiten junto al mecanismo de la herencia de conceptos, inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

Según Heijst y Wielinga [25] las ontologías pueden clasificarse en:

- ✓ **Clasificación por el conocimiento que contienen:** se compone de ontologías terminológicas que representan un conocimiento determinado, ontologías de información que representan los esquemas de una base de datos y ontologías para modelar conocimiento donde tienen una estructura más completa que las primeras.
- ✓ **Clasificación por motivación:** se componen de ontologías que representan el conocimiento, ontologías genéricas que pueden ser reutilizables, ontologías de algún dominio en específico y ontologías de aplicación.

Según las W3C [26] para definir una ontología se utilizan varios tipos de lenguajes que son:

- ✓ **XML:** proporciona una sintaxis superficial para documentos estructurados, pero no impone restricciones semánticas en el significado de estos documentos.
- ✓ **XML Schema (XMLS):** es un lenguaje que se utiliza para restringir la estructura de los documentos XML, además para ampliar XML con tipos de datos.
- ✓ **RDF:** es un modelo de datos para objetos (“recursos”) y relaciones entre ellos, proporcionando una semántica simple para estos. Este tipo de modelo de datos puede ser representado en una sintaxis XML.

- ✓ **RDF Schema (RDFS):** es un vocabulario utilizado para describir propiedades y clases de recursos RDF con una semántica para la generalización y jerarquización tanto de propiedades como de clases.
- ✓ **OWL:** añade más vocabulario para describir propiedades, clases, relaciones (por ejemplo, desunión), cardinalidad (por ejemplo, “uno exacto”), igualdad, más tipos de propiedades, características de propiedades (por ejemplo, simetría), y clases enumeradas.

Actualmente, el lenguaje OWL (Ontology Web Language) es el más usado, debido a que es una evolución de los anteriores. También, es importante mencionar el lenguaje de consultas de ontologías SPARQL (Simple Protocol and RDF Query Language). La W3C [27] menciona que SPARQL se puede utilizar para expresar consultas a través de diversas fuentes de datos, si los datos se almacenan nativamente como RDF o se ven como RDF. SPARQL contiene capacidades para consultar los patrones de gráficos requeridos y opcionales junto con sus conjunciones y disyunciones. SPARQL también soporta agregación, subconsultas, negación, creación de valores por expresiones, pruebas de valores extensibles y restricción de consultas. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o gráficos RDF. Varios estándares del W3C, incluyendo RDF y OWL, proporcionan interpretaciones semánticas para los gráficos RDF que permiten inferir declaraciones RDF adicionales a partir de aseveraciones explícitamente dadas.

2.3.2. Buscador semántico. La definición de lo que es un buscador semántico desde un punto de vista técnico, se dice que es una aplicación que comprende las búsquedas de los usuarios y los textos de los documentos de la web, todo mediante el uso de algoritmos que simulan comprensión de palabras y así establecer una relación entre ellas, esto según Morales [28]. A partir de estas premisas proporciona resultados que son más acordes a lo que el usuario espera encontrar, donde coinciden con las palabras claves de la búsqueda y demás documentos que se relacionan semánticamente con las palabras claves usadas.

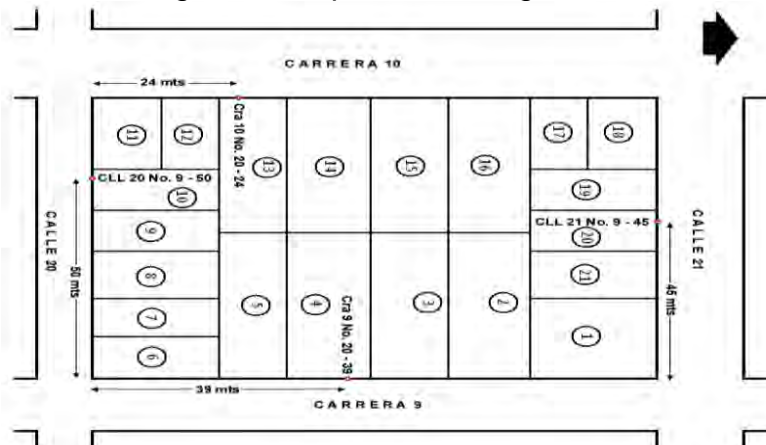
2.4. ESTÁNDAR DE DIRECCIONES URBANAS

Debido a que una dirección urbana puede ser escrita de múltiples maneras es fácil encontrar diversas formas para referirse a un mismo elemento (Calle, Cal, Cl, etc.) y consecuentemente se presentan diversas formas de organizar los mismos elementos. Según lo consignado en IGAC [29] se argumenta que el estándar de direcciones urbanas sirve para optimizar resultados al momento de emplear una geocodificación y que por lo tanto es necesario para establecer los parámetros para la obtención y respectivo almacenamiento de direcciones. En términos generales definen al estándar como un conjunto de reglas que proporcionan una forma de estructurar datos y facilitar su desempeño automático e interoperable.

El Instituto Geográfico Agustín Codazzi, identifica los elementos presentes en áreas urbanas que requieren o no asignación de nomenclaturas:

- ✓ **Calle:** vía pública con orientación predominante definida de acuerdo al modelo de una ciudad.
- ✓ **Carrera:** vía pública generalmente perpendicular a la calle. Genera nomenclatura predial.
- ✓ **Diagonal:** vía pública que generalmente tiene el mismo sentido que la calle sin ser paralela a esta. Puede generar o no una nomenclatura predial.
- ✓ **Transversal:** vía pública que generalmente tiene el mismo sentido que la carrera sin ser paralela a esta. Puede o no generar nomenclatura predial.

Figura 2. Esquema vial según IGAC



Fuente: Instituto Geográfico Agustín Codazzi, 2009.

Para direcciones basadas en manzana se destacan en gran medida los siguientes conceptos:

- ✓ **Manzana:** es el espacio geográfico donde se ubica un conjunto de predios urbanos edificados o sin edificar delimitado generalmente por vías públicas.
- ✓ **Barrio:** es el espacio geográfico en que se divide el área urbana, y donde se ubican un conjunto de manzanas.
- ✓ **Casa:** este campo hace referencia a la información complementaria, que puede definir la condición y/o posición del predio dentro de un determinado lote.

La definición del estándar para los elementos está definida en la tabla 1.

Tabla 1. Estándar de abreviaturas más usadas en el municipio de Pasto

Elemento	Abreviatura
Calle	CL
Carrera	KR
Diagonal	DG
Transversal	TV
Barrio	BR
Manzana	MZ
Casa	CS

Fuente: Instituto Geográfico Agustín Codazzi, 2009.

2.5.SCRUM

Para referirse a una metodología ágil para el desarrollo de software o cualquier otro tipo de desarrollo es válido referirse al concepto ofrecido por Menzinsky, López y Palacio [30] donde mencionan que “La agilidad es un comportamiento persistente o habilidad, de entidad sensible, que presenta flexibilidad para adaptarse a los cambios esperados o inesperados rápidamente; persigue la duración más corta en tiempo, usa instrumentos económicos; y utiliza los conocimientos y experiencias previas para aprender tanto del entorno interno como del externo”.

Según Rubin [31], se define a Scrum, como una colección de procesos para la gestión de proyectos, que permite centrarse en la entrega de valor para el cliente y la potenciación del equipo para lograr su máxima eficiencia, dentro de un esquema de mejora continua.

Scrum es un marco de trabajo, framework iterativo e incremental para el desarrollo de proyectos. Su estructura se basa en ciclos de trabajo llamados Sprints. Para entender los ciclos de trabajo de Scrum es importante destacar las 5 fases que definen el ciclo de desarrollo ágil:

- ✓ **Concepto:** se define de forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.
- ✓ **Especulación:** en esta fase se hacen disposiciones con la información obtenida y se establecen los límites que marcarán el desarrollo del producto, tales como costes y agendas. Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno.
- ✓ **Exploración:** se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.
- ✓ **Revisión:** el equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.

- ✓ **Cierre:** se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios, denominados “mantenimiento”, que hará que el producto final se acerque al producto final deseado.

En Scrum se pueden identificar dos tipos de roles que son:

- ✓ **Comprometidos:** participante principal del proyecto y del proceso de Scrum.
- ✓ **Implicados:** no son parte del proceso del proyecto, pero intervienen directa o indirectamente en él. Ejemplos: otros interesados, Dirección General, Dirección Comercial, Marketing, Usuarios.

Figura 3. Esquema general de Scrum



Fuente: Essential Scrum, 2013.

El desarrollo en Scrum puede comprender varios equipos de trabajo, donde es importante como mínimo contar con personas que desempeñen roles (comprometidos), tales como product owner, Scrum master y development team, cuyas funciones se describen a continuación:

- ✓ **Product owner:** es la persona que toma las decisiones y es la que realmente conoce el negocio del cliente y su visión del producto.
- ✓ **Scrum master:** es el encargado de comprobar que el modelo y la metodología funciona. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores.
- ✓ **Development team:** suele ser un equipo pequeño de unas 5-9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo.

2.6.TECNOLOGÍAS USADAS

2.6.1.Java OpenStreetMap. La página oficial [32] menciona que OpenStreetMap (OSM) es un proyecto colaborativo para crear mapas libres y editables. Los mapas se crean utilizando imágenes aéreas, dispositivos GPS, mapas y otras fuentes de datos libres. Los datos que se encuentra en OSM son abiertos, lo que significa que pueden ser usados libremente para cualquier propósito, siempre y cuando se de crédito a OSM y a sus colaboradores. Todos los datos vectoriales almacenados en su base de datos, se distribuye bajo licencia abierta ODbL (Open Data Commons Open Database License). La diferencia de OSM con otros mapas es que cualquier persona puede ser parte de esta comunidad, añadiendo, corrigiendo o actualizando información.

JOSM (Java OpenStreetMap Editor) es un editor para OpenStreetMap (OSM) escrito en Java. Fue desarrollado por Immanuel Scholz y actualmente está mantenido por Dirk Stöcker. La versión actual soporta trazas GPX únicas, datos de trazas GPX almacenadas en la base de datos OSM y nodos existentes, segmentos de líneas y etiquetas de metadatos de la base de datos OSM esto según Ferrer, Sanchez y Tramoyeres [33]. A pesar de que tiene una curva de aprendizaje relativamente pronunciada, JOSM es muy popular entre los editores experimentados por su estabilidad y por las posibilidades que ofrece gracias en gran parte a sus componentes externos, los cuales extienden en gran medida sus capacidades de edición esto según el proyecto OSGeo-Live [34].

2.6.2.Protégé. Protégé es un editor de ontologías libre y de código abierto para la construcción de sistemas inteligentes, cuenta con el apoyo de una fuerte comunidad de usuarios académicos, gubernamentales y corporativos que la utilizan para construir soluciones basadas en el conocimiento en áreas tan diversas como la biomedicina, el comercio electrónico y el modelado organizacional esto según la Universidad de Stanford [35].

Existen dos versiones de este editor:

- ✓ **WebProtégé:** es un entorno de desarrollo de ontologías para la web que facilita la creación, carga, modificación y compartición de ontologías para la visualización y edición en colaboración. Tiene un soporte completo a las últimas especificaciones de la W3C.
- ✓ **Protégé Desktop:** es un entorno de edición de ontologías basado en Java, rico en funciones con soporte completo a las últimas especificaciones de la W3C. Admite la creación y edición de una o más ontologías en un solo espacio de trabajo a través de una interfaz de usuario completamente personalizable. Las herramientas de visualización permiten la navegación interactiva de las relaciones de ontología.

2.6.3.Jena. La página oficial [36] menciona que Apache Jena es un framework de Java de código abierto y libre para la construcción de aplicaciones web semánticas y datos vinculados. Licenciado bajo la Licencia Apache Versión 2.0 y desarrollado por la Apache Software Foundation (ASF). Proporciona una API para extraer y escribir datos de ontologías en formato RDF, RDFS y OWL, además de soportar consultas en el lenguaje SPARQL siguiendo las recomendaciones de la W3C. Jena incluye un motor de inferencia basado en reglas para realizar razonamiento basado en los diferentes formatos de ontologías.

2.6.4.Java Server Faces. Java Server Faces (JSF) es un estándar de Java hacia la construcción de interfaces de usuario para aplicaciones web del lado del cliente simplificando el desarrollo, está basado en la tecnología Java Enterprise Edition (Java EE). Una página JSF utiliza la extensión *.xhtml, es decir, una combinación de XML con HTML y puede incluir componentes como CSS, JavaScript, entre otros. JSF está diseñado para aliviar significativamente la carga de escribir y mantener aplicaciones que se ejecutan en un servidor de aplicaciones Java y convertir sus GUI de nuevo a un cliente de destino esto según Burns y Schalk [37].

Burns y Kitain [38], mencionan que JSF proporciona facilidad de uso de las siguientes maneras:

- ✓ Facilita la construcción de una interfaz de usuario a partir de un conjunto de componentes de interfaz de usuario reutilizables.
- ✓ Simplifica la migración de los datos de la aplicación hacia y desde la interfaz de usuario.
- ✓ Ayuda a administrar el estado de la interfaz de usuario en las solicitudes de servidor.
- ✓ Proporciona un modelo sencillo para el cableado de eventos generados por el cliente al código de la aplicación del servidor.
- ✓ Permite que los componentes personalizados de la interfaz de usuario se puedan construir y volver a utilizar fácilmente.

La facilidad de uso es el objetivo principal. La arquitectura Java Server Faces define claramente una separación entre la lógica de la aplicación y la presentación, al mismo tiempo que facilita su conexión. Este diseño permite a cada miembro de un equipo de desarrollo centrarse en su parte del proceso, y también proporciona un modelo de programación simple para vincular sus módulos.

La página oficial [39] menciona que JSF tiene dos funciones principales, la primera es generar una interfaz de usuario, normalmente una respuesta HTML que se sirve a un navegador y se ve como una página web. Esta interfaz de usuario está representada en el servidor por un árbol de componentes. La segunda función de JSF es responder a los eventos generados por el usuario en la página, invocando a los oyentes del lado del servidor, seguido por la generación de otra interfaz de usuario (por ejemplo, página web) o una actualización de la interfaz de usuario ya

mostrada (posiblemente a través de Ajax). En este sentido, se dice que JSF es un framework web basado en eventos.

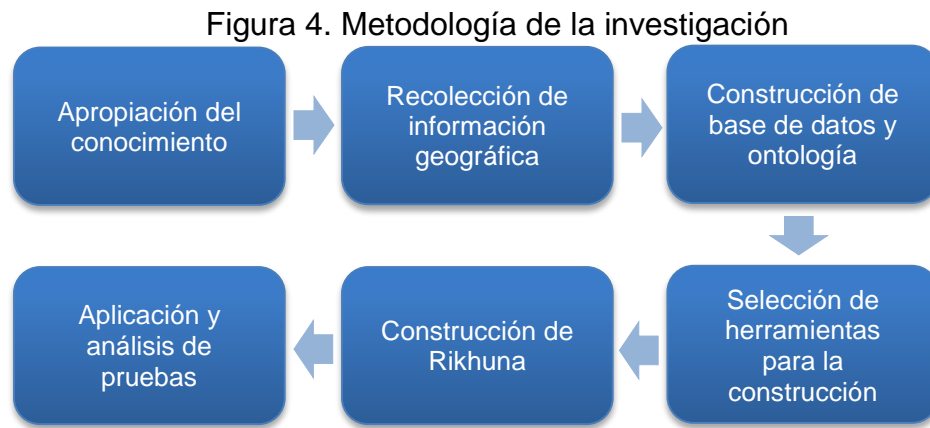
2.6.5.Geoserver. GeoServer es un servidor de mapas de código abierto escrito en lenguaje Java que permite a los usuarios compartir y editar información geoespacial usando estándares abiertos. En efecto, implementa varios estándares definidos por el Open Geospatial Consortium (OGC), como por ejemplo el ampliamente utilizado Web Map Service (WMS) [40]. Además, permite el despliegue de la aplicación sobre cualquier servidor de aplicaciones conforme a la especificación J2EE, tanto libres (como Tomcat, Geronimo de Apache o JBoss de RedHat) como propietarios (WebLogic de BEA o WebSphere de IBM, entre otros) esto según García, Fonts y González [41].

Szczerban [42] menciona que entre las características principales se destacan:

- ✓ Multiplataforma, es de fácil utilización y administración a través de una herramienta de administración web. No es necesario tocar archivos de configuración.
- ✓ Soporte para edición de datos a través del protocolo WFS transaccional (WFS-T).
- ✓ Al ser programado en Java está basado en servlets (JEE), puede funcionar en cualquier contenedor de servlets.
- ✓ Es compatible para ser usado con extensiones y soporte de estilos CSS.
- ✓ Soporte para multitud de orígenes de datos (PostGIS, Oracle Spatial, ArcSDE, DB2, MySQL, entre otros).
- ✓ Incluye integrado un cliente OpenLayers que permite visualizar las capas de datos en múltiples formatos (KML, GML, Shapefile, GeoRSS, PDF, GeoJSON, JPEG, GIF, SVG, PNG).

3. METODOLOGÍA

La metodología empleada en el desarrollo de este trabajo comprende las etapas: apropiación de conocimiento, recolección de información geográfica, construcción de la base de datos y ontología, selección de herramientas, construcción del visor cartográfico Rikhuna, y finalmente aplicación y análisis de pruebas, como se muestra en la figura 4.



✓ **Apropiación del conocimiento.**

El objetivo de esta etapa fue apropiarse del conocimiento correspondiente con la temática, entendiendo la problemática que se va a solucionar, el alcance propuesto y las metas que se pretende alcanzar, de manera que se adquieran unas bases teóricas sólidas para la construcción del visor cartográfico. Se planteó asimilar conceptos sobre sistemas de información geográfica, estándares y proceso de geocodificación de direcciones urbanas, búsquedas semánticas, un acercamiento a PostGIS, conocer el funcionamiento de diversas librerías y herramientas para el almacenamiento, tratamiento, visualización de información geográfica, así como el mantenimiento y gestión de ontologías.

✓ **Recolección de información geográfica.**

En esta etapa se recolectó toda la información geográfica correspondiente a direcciones urbanas, sitios de interés, comunas, barrios, corregimientos y veredas del municipio de Pasto. Este proceso se realizó teniendo en cuenta los lineamientos y estándares que presenta el Instituto Geográfico Agustín Codazzi. La información obtenida proviene de distintas fuentes como trabajos de investigación, mapas turísticos, sitios web gubernamentales, junto con el uso de métodos y técnicas de interpolación de puntos.

Entre los sitios de interés se encuentran sitios turísticos, templos, museos, restaurantes, esculturas, monumentos, hitos arquitectónicos, centros educativos, centros administrativos, hospitales, centros de atención inmediata y demás zonas de interés de la ciudad.

✓ **Construcción de base de datos y ontología.**

El objetivo de esta etapa fue diseñar y construir la base de datos con el diagrama entidad relación implementada en el Sistema Gestor de Base de Datos PostgreSQL. Se hizo la migración de datos geográficos en formato OSM a la base de datos espacial utilizando scripts en el lenguaje de programación Python. Se validó y corrigió los datos recolectados usando funciones de PostGIS con el fin de obtener información limpia y completa. En cuanto a la ontología, se utiliza la herramienta de software libre Protégé en formato OWL para su creación y mantenimiento. Para definir cada una de las clases se realizó una abstracción del esquema de la base de datos, y se definió los atributos necesarios para que las búsquedas puedan relacionar diversos conceptos sobre el dominio establecido.

✓ **Selección de herramientas para construcción.**

El objetivo de esta etapa fue establecer las herramientas adecuadas para la construcción del visor cartográfico, para lo cual se realizó un estudio de las tecnologías actuales teniendo en cuenta los requerimientos de soporte para ontologías en formato OWL (un tipo de lenguaje de ontologías web) y soporte para realizar consultas en SPARQL (lenguaje para las consultas en las ontologías). Se analizaron lenguajes de programación, librerías, su funcionamiento y algunas características como velocidad de respuesta, escalabilidad, documentación, entre otras.

✓ **Construcción de Rikhuna.**

La construcción del visor cartográfico fue realizada bajo el sistema operativo Linux Ubuntu en su versión 16.04. El lenguaje de programación utilizado fue Java con el Framework JSF y Primefaces, con el entorno de desarrollo (IDE) Netbeans versión 8.2 y el servidor de aplicaciones Glassfish en su versión 4.1, además se utilizó Geoserver 2.11, Jena 3.1.1, y OpenLayers en su versión 3. La arquitectura de Rikhuna se basa en el patrón Modelo Vista Controlador (MVC). En esta etapa se hizo una adaptación de la metodología de desarrollo ágil SCRUM, la cual permite que el desarrollo se lleve a cabo por medio de iteraciones o sprints. La arquitectura del visor cartográfico se presenta detalladamente en el capítulo denominado ARQUITECTURA.

✓ **Aplicación y análisis de pruebas.**

El objetivo de esta etapa fue diseñar, aplicar y analizar resultados de pruebas realizadas al visor cartográfico con el fin de verificar aspectos de funcionamiento y calidad en las búsquedas realizadas.

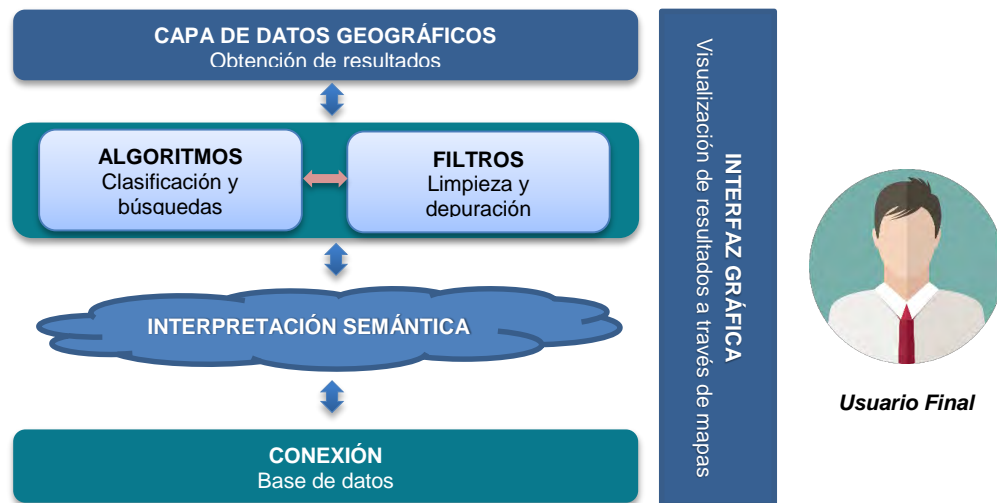
4. ARQUITECTURA

A continuación, se describe la arquitectura para el visor cartográfico Rikhuna y para el panel administrativo:

4.1.ARQUITECTURA DEL VISOR CARTOGRÁFICO

Para el desarrollo del visor cartográfico se utilizó el patrón de diseño Modelo Vista Controlador (MVC), donde la arquitectura se evidencia en la figura 5.

Figura 5. Arquitectura general del visor cartográfico



Teniendo en cuenta el patrón utilizado en el desarrollo del visor cartográfico, se ha definido un diseño de una arquitectura complementaria en la cual se muestran los paquetes que componen el núcleo de la herramienta, el módulo de conexión y las principales tecnologías empleadas para la presentación de las búsquedas al usuario, como se muestra en la figura 6.

✓ Módulo de presentación

Este módulo permite la interacción con el usuario final, en donde se presenta la vista con el menú de opciones para los distintos tipos de búsqueda disponibles, la información ingresada es procesada por el módulo núcleo y los resultados se hacen visible a través de un mapa donde el usuario puede interactuar y visualizar la información correspondiente a las solicitudes realizadas.

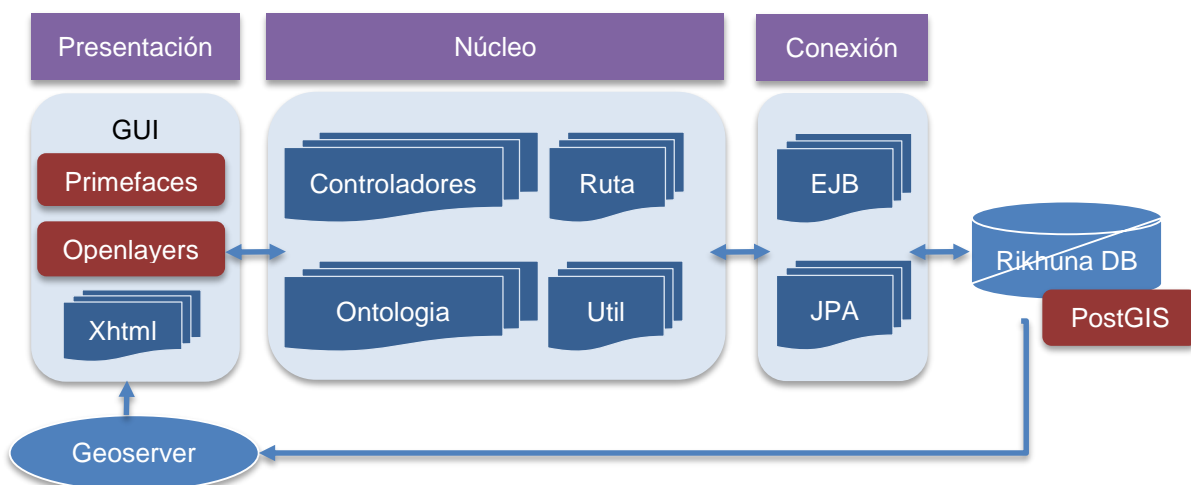
✓ **Módulo núcleo**

Módulo que contiene la lógica de la aplicación. Se encarga de validar los datos ingresados por el usuario, limpiarlos y llevar a cabo el procesamiento de las búsquedas ingresadas, realizando consultas a la ontología a través de una interpretación semántica y a la base de datos a través del módulo de conexión. Con la implementación de este módulo se pretende dar respuestas eficientes y eficaces a las distintas búsquedas realizadas.

✓ **Módulo de conexión**

Este módulo es el encargado de gestionar la conexión con la base de datos Rikhuna en su esquema público a través del mapeo de sus tablas y el procesamiento de consultas personalizadas sobre los datos recibidos por el módulo núcleo.

Figura 6. Arquitectura por paquetes del visor cartográfico

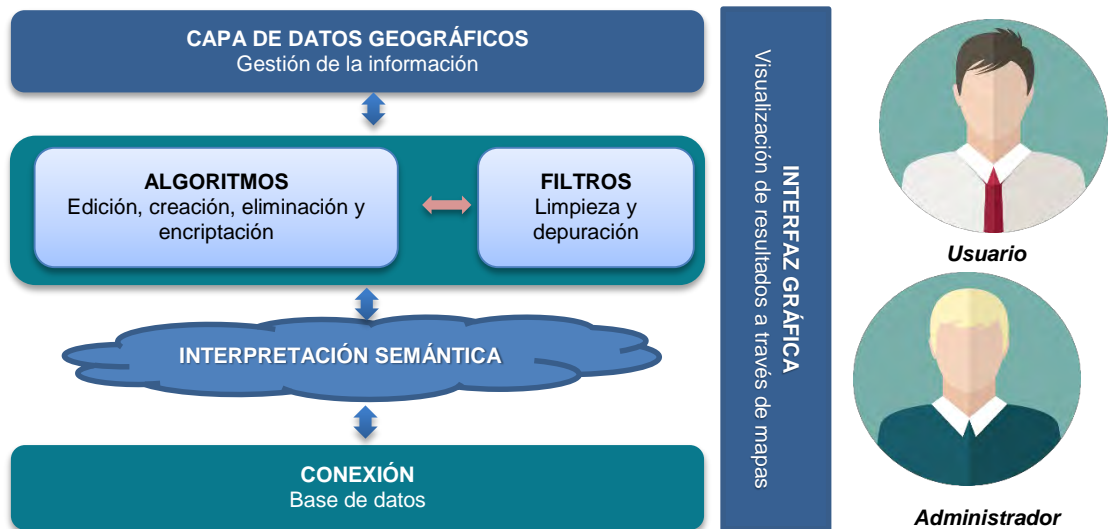


En la sección de Implementación del Visor Cartográfico, dentro del capítulo de RESULTADOS se explican detalladamente cada uno de los paquetes y demás archivos que conforman cada uno de los módulos mencionados anteriormente.

4.2.ARQUITECTURA DEL PANEL ADMINISTRATIVO

El panel administrativo fue pensado para llevar a cabo tareas que permitan a largo plazo mantener los datos espaciales de direcciones urbanas y sitios de interés actualizados. Para cumplir este objetivo se definieron funcionalidades que permiten llevar a cabo tareas de actualización, modificación y eliminación sobre los datos almacenados, esto posterior a un registro y autenticación de usuarios. La arquitectura se basó en el patrón de diseño Modelo Vista Controlador (MVC), donde la arquitectura se evidencia en la figura 7.

Figura 7. Arquitectura general del panel administrativo



De manera similar a la arquitectura de visor cartográfico se ha definido un diseño de una arquitectura complementaria en la cual se muestran los paquetes que componen el núcleo del panel administrativo, el módulo de conexión y las principales tecnologías empleadas para la gestión de las tareas de los usuarios, como se muestra en la figura 8.

✓ Módulo de presentación

Este módulo permite la interacción con el usuario final, donde se presenta las vistas que componen el panel administrativo, tanto para la gestión de los usuarios como para las distintas opciones de registro de nueva información de direcciones y sitios de interés. Las vistas asociadas son:

- **Módulo registro:** hace parte del módulo de presentación, y es el encargado de llevar a cabo el registro de los nuevos usuarios en la base de datos, solicitando la información mínima para identificar al usuario que pretende hacer uso del panel administrativo.
- **Módulo de inicio de sesión:** una vez se cuenta con un registro de usuario, este podrá iniciar sesión y acceder a la vista principal del panel administrativo. Si las credenciales brindadas por el usuario son válidas se procede a mostrar la interfaz principal, caso contrario muestra un mensaje de error donde se notifica que existen inconsistencias en la información suministrada. En este módulo se tiene en cuenta que existe un único usuario administrador que se encargará de validar la información suministrada por los usuarios que se han registrado.

- **Módulo principal:** módulo que presenta la vista principal del panel administrativo, donde en la cartográfica del municipio de Pasto se cuenta con opciones para llevar a cabo las tareas esenciales y mantener la información actualizada. Sobre este módulo los usuarios pueden llevar a cabo tareas tales como agregar, actualizar y eliminar datos de direcciones urbanas y sitios de interés.
- **Módulo principal administrador:** módulo que presenta la vista principal del panel administrativo para el administrador del sistema, donde se despliega toda la información proporcionada por los usuarios y donde el administrador verifica si esta información es válida para almacenar en el esquema principal de la base de datos Rikhuna.

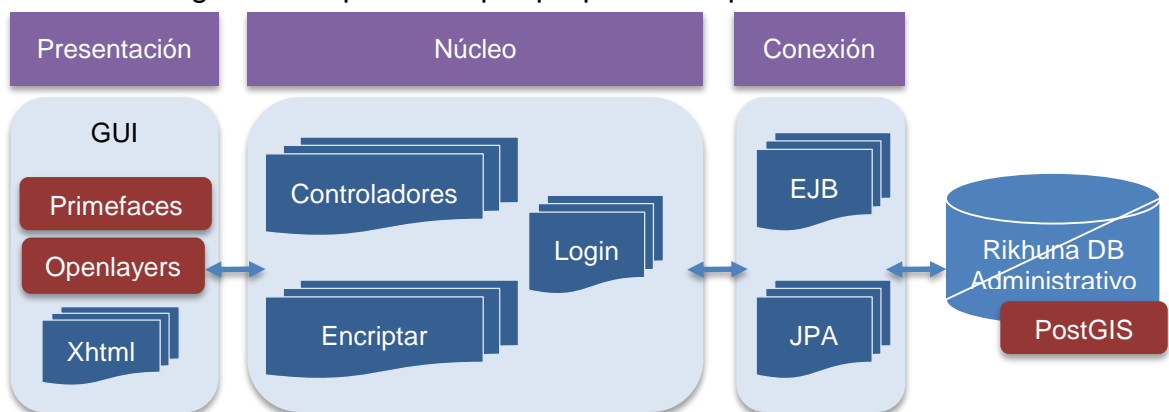
✓ **Módulo núcleo**

Módulo que contiene la lógica de la aplicación. Se encarga de llevar a cabo el procesamiento y validación de las tareas que llevan a cabo los usuarios desde el módulo de presentación. También se encarga de llevar un control sobre la sesión de usuarios.

✓ **Módulo de conexión**

Permite llevar a cabo la conexión con la base de datos Rikhuna en su esquema administrativo a través del mapeo de sus tablas, para realizar la gestión de los datos ingresados por el usuario para sus diferentes tareas.

Figura 8. Arquitectura por paquetes del panel administrativo



En la sección de Implementación del panel administrativo, dentro del capítulo de RESULTADOS se explican detalladamente cada uno de los paquetes y demás archivos que conforman cada uno de los módulos mencionados anteriormente.

5. RESULTADOS

A continuación, se presentan los resultados obtenidos en cada etapa de la metodología aplicada:

5.1.ETAPA DE APROPIACIÓN DEL CONOCIMIENTO

En esta etapa se apropió del conocimiento que permitió abordar las temáticas necesarias para completar cada etapa de la metodología. El principal resultado de esta etapa se puede evidenciar en los conceptos adquiridos, que se encuentran documentados en el marco teórico y en la realización exitosa de cada etapa, las cuales son detalladas en los numerales siguientes.

5.2.ETAPA DE RECOLECCIÓN DE INFORMACIÓN GEOGRÁFICA

En esta etapa se realizó la recolección de toda la información geográfica necesaria para el funcionamiento del visor cartográfico, la fuente inicial de esta información fue tomada de diferentes trabajos de investigación realizados en el grupo de investigación GRIAS, fuentes externas como la Alcaldía de Pasto y la información restante se recolectó desde cero. Para llevar a cabo este proceso se utilizó la herramienta de software libre para la edición de datos cartográficos Java OpenStreetMap Editor [32] (en adelante JOSM).

5.2.1.Recolección de puntos. En esta etapa se recolectó información geográfica para direcciones urbanas y sitios de interés de la siguiente manera:

✓ **Recolección de direcciones urbanas**

A partir del documento del Instituto Geográfico Agustín Codazzi (en adelante IGAC) para la estandarización de direcciones [29], menciona que existen diferentes tipos de casos para la Nomenclatura Predial disponibles en la tabla 2.

Tabla 2. Casos para la nomenclatura vial

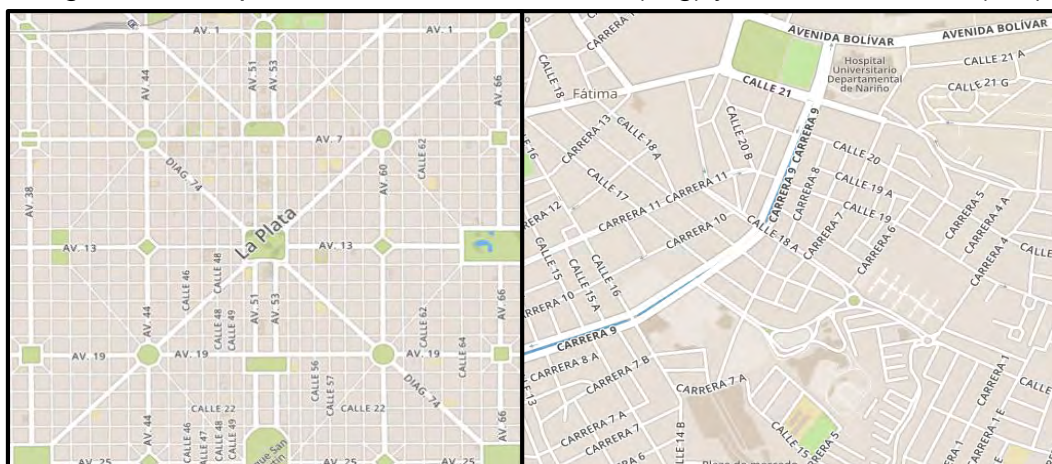
Casos para la nomenclatura Predial
Nomenclatura basada en la estructura de malla vial
Nomenclatura basada en la estructura Barrio-Manzana-Predio
Nomenclatura Barrio-Manzana-Predio como complemento de la nomenclatura basada en la malla vial.
Barrio como complemento a la nomenclatura basada en la estructura de malla vial

Fuente: Instituto Geográfico Agustín Codazzi, 2009

En la ciudad de Pasto predominan los dos primeros, nomenclatura basada en la estructura de malla vial y Barrio-Manzana-Predio (en adelante manzana), estos son los únicos casos que se tienen en cuenta para esta investigación. Ya delimitado el área de estudio se continuó con la recolección de las direcciones, la primera opción para esto es obtenerlas de alguna fuente externa, pero debido a que esta información no es de libre acceso por el IGAC u otra institución, se vio la necesidad de recolectar estas direcciones desde cero. Esta se realizó de forma manual y no de manera automática o usando algún algoritmo debido a que la ciudad presenta varios problemas en asignación de direcciones como lo son:

- La zona urbana de Pasto no es homogénea en tamaño y posición de las cuadras, como se muestra en la figura 9.
- Muchas casas no presentan dirección o no tienen placa de identificación predial.
- Algunas cuadras presentan direcciones mal asignadas que no tienen en cuenta el estándar del IGAC.
- El número de placa de algunas direcciones no corresponde con la distancia en metros como lo menciona el estándar del IGAC.
- Algunas casas presentan doble dirección, basadas en malla vial y manzana.
- Algunas direcciones no respetan la asignación del número de la placa que menciona que si el predio esta al costado derecho de las vías deben tener número par y al costado izquierdo número impar.
- Dado el crecimiento de la ciudad con la creación de nuevas cuadras y vías, es complejo llevar un orden correcto de asignación de direcciones urbanas.

Figura 9. Comparación ciudad de la Plata (Arg) y ciudad de Pasto (Col)



Lo anterior justifica la recolección manual de esta información, pero debido a que el municipio de Pasto es muy amplio para que solo una investigación realice este proceso, se realizó esta recolección en conjunto con los miembros pertenecientes al grupo de investigación GRIAS, siendo esta información útil para varios proyectos

de investigación de trabajos de grado de la Universidad de Nariño como SIGODEP-SIG [43] y SITAPP [44].

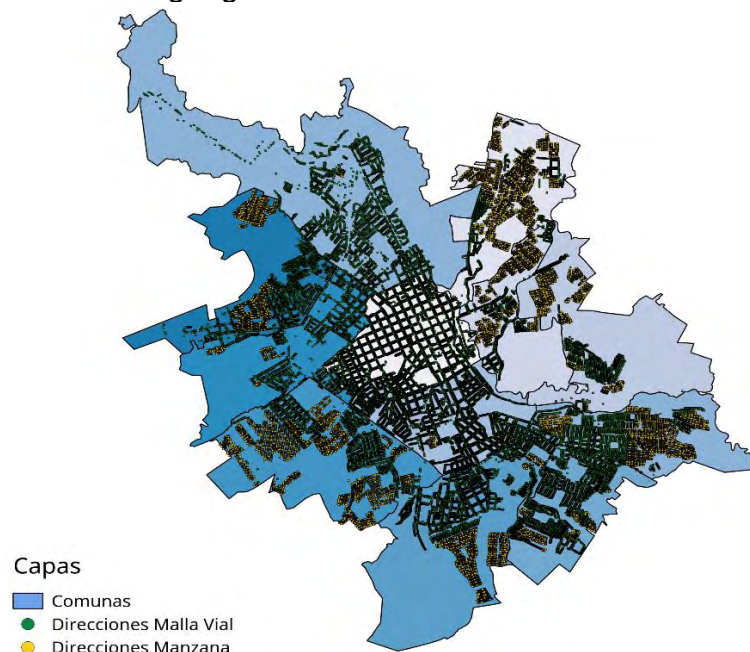
Para facilitar este proceso se dividió la recolección por comunas creando un archivo en formato OSM para cada una de ellas, se utilizó herramientas como Google Street View [45], el geoportal de visualización de datos geográficos y catastrales del IGAC [6] y la herramienta JOSM con el plugin Address Interpolation [46].

En la tabla 3, se consigna el total de direcciones que se recolectó en esta etapa:

Tabla 3. Direcciones urbanas obtenidas en la etapa de recolección

Tipo de dirección	Número
Malla vial	82.828
Manzana	21.437

Figura 10. Puntos geográficos de direcciones urbanas recolectadas



✓ **Recolección de sitios de Interés**

Para la recolección de sitios de interés se tomó como base mapas turísticos brindados por el IGAC y la Subsecretaria de Turismo de la Alcaldía de Pasto los cuales están en la tabla 4.

Tabla 4. Mapas turísticos del Municipio de Pasto

Mapa	Institución	Descripción
Mapa Turístico de Pasto y sus alrededores	IGAC	Guía que muestra información general del municipio de Pasto con alrededor de 500 sitios de interés agrupados en 49 categorías.
Agenda Turística 2016-2017	Subsecretaría de Turismo	Guía que muestra los sitios más emblemáticos de la ciudad y sus alrededores, con información detallada, además de las fiestas más representativas del municipio. Presenta alrededor de 135 sitios de interés.
PASTO ¡Déjate sorprender!	Subsecretaría de Turismo	Guía que muestra información general de los corregimientos y sitios de interés más representativos de la ciudad, agrupados en las categorías: hoteles, restaurantes, templos, museos, sitios de interés y arte y artesanía. Presenta alrededor de 73 sitios de interés.
Ruta Gastronómica	Subsecretaría de Turismo	Guía que muestra los principales platos típicos de Nariño y restaurantes a donde ir. Presenta alrededor de 55 sitios de interés.
Ruta Artesanal	Subsecretaría de Turismo	Guía que muestra las principales artesanías del Municipio y los lugares donde se realizan. Presenta alrededor de 34 sitios de interés.
Guías Turísticas de los Corregimientos de Pasto	Subsecretaría de Turismo	Conjunto de guías por cada uno de los 17 corregimientos pertenecientes al municipio de Pasto, cada guía describe lo más representativo de cada corregimiento, como sus fiestas, atractivos turísticos y platos típicos.

A partir de estos mapas se recolectaron 779 sitios de interés en la zona urbana, gran parte de los sitios se encontraban repetidos en los mapas mencionados, lo que redujo el número total de estos, además, se crearon 121 categorías para poder agruparlos con sus respectivos iconos tomados del proyecto The Map Icons Collection [47]. En la tabla 5, se agrupan los sitios de interés recolectados.

Tabla 5. Recolección de sitios de interés por comuna

Comuna	Sitios de Interés	Comuna	Sitios de Interés	Comuna	Sitios de Interés
Comuna 1	169	Comuna 5	87	Comuna 9	126
Comuna 2	62	Comuna 6	87	Comuna 10	16
Comuna 3	43	Comuna 7	57	Comuna 11	17
Comuna 4	77	Comuna 8	21	Comuna 12	17

Figura 11. Mapas turísticos del Municipio de Pasto



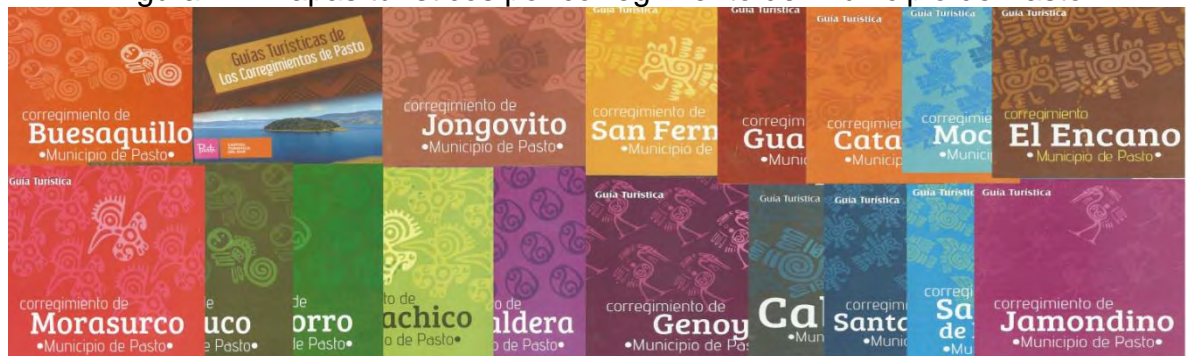
Fuente: IGAC y Subsecretaria de Turismo, 2016.

En la parte rural se recolectaron algunos de los sitios más relevantes de los corregimientos, en total 43, no se pudo obtener la mayoría de los sitios mostrados en los mapas por la complejidad de la ubicación de algunos puntos.

Tabla 6. Recolección de sitios de interés por corregimiento

Corregimiento	Sitios de interés	Corregimiento	Sitios de interés	Corregimiento	Sitios de interés
Buesaquillo	1	Gualmatan	1	Mocondino	1
Cabrera	3	Jamondino	5	Morasurco	6
Catambuco	1	Jongovito	3	Obonuco	8
El Encano	2	La Caldera	1	San Fernando	3
El Socorro	1	La Laguna	2	Santa Bárbara	1
Genoy	3	Mapachico	1		

Figura 12. Mapas turísticos por corregimiento del municipio de Pasto



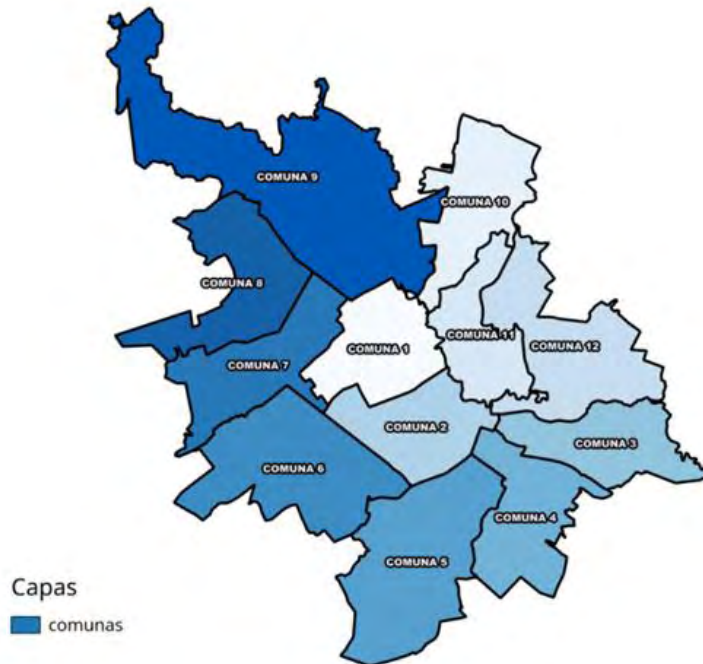
Fuente: Subsecretaria de Turismo, 2016

5.2.2.Recolección de geometrías. En esta etapa se recolectó información geográfica para comunas, barrios, corregimientos y veredas de la siguiente manera:

✓ **Recolección de comunas**

Para la recolección de comunas se tomó como base información de las comunas perteneciente al proyecto SIGODEP [4] desarrollado también en el grupo de investigación GRIAS, a partir de este archivo se corrigió el rango de sus geometrías, contemplando sus alrededores como corregimientos e internamente que su área contemple todos los barrios. Se recolectaron 12 comunas como se muestra en la figura 13.

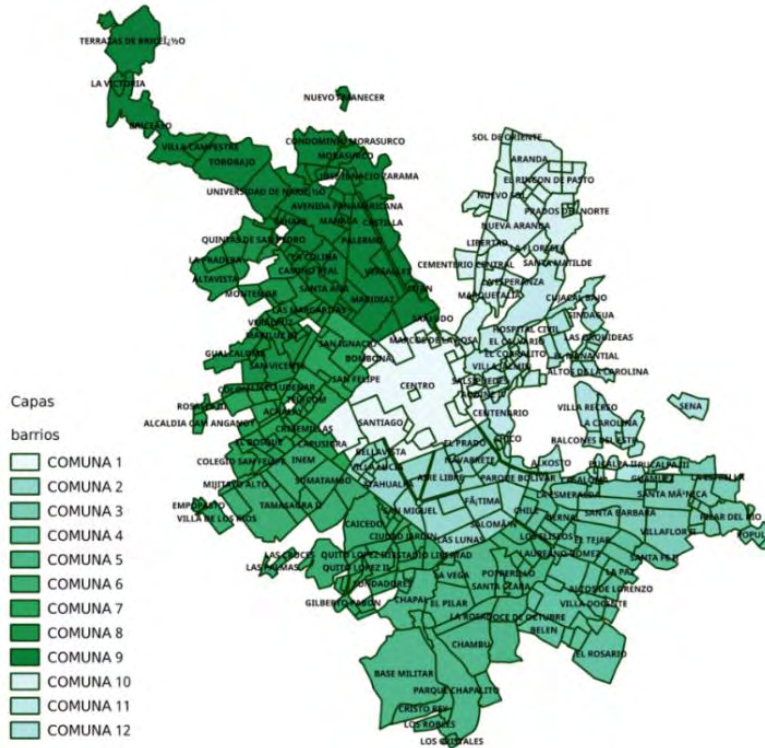
Figura 13. Comunas del Municipio de Pasto



✓ **Recolección de barrios**

Para la recolección de barrios se tomó como base información de los barrios pertenecientes al proyecto SIGODEP [4], a partir de este archivo se corrigió la geometría de algunos de ellos, se eliminó barrios sin nombre y se adicionó nuevos, que debido al crecimiento de la ciudad antes no estaban presentes, esta actualización se la realizó con ayuda de mapas de cada comuna pertenecientes a la Alcaldía de Pasto, estos mapas en formato PDF se los agregó a la herramienta JOSM utilizando el plugin PDF Import [48].

Figura 14. Barrios del Municipio de Pasto



Se recolectaron 392 barrios como se muestra en la tabla 7:

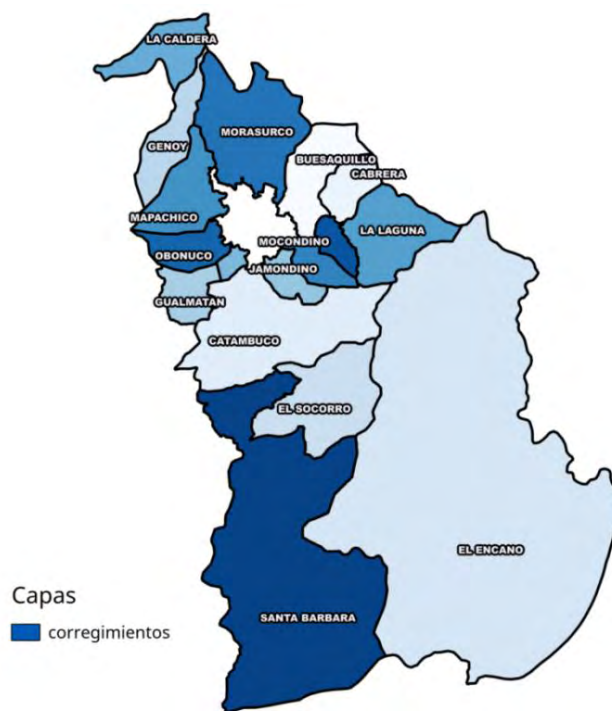
Tabla 7. Barrios recolectados por comuna

Comuna	Barrios	Comuna	Barrios	Comuna	Barrios
Comuna 1	20	Comuna 5	29	Comuna 9	52
Comuna 2	32	Comuna 6	42	Comuna 10	31
Comuna 3	27	Comuna 7	41	Comuna 11	23
Comuna 4	30	Comuna 8	33	Comuna 12	28

✓ **Recolección de corregimientos**

Para la recolección de corregimientos se tomó como base únicamente los mapas de la Alcaldía de Pasto, dado que no existía un archivo en anteriores investigaciones, se tomó como guía el mapa general del municipio de Pasto dividido en corregimientos, y el mapa perteneciente a cada uno de ellos, estos mapas se transformaron en imagen en un formato PNG y se los agregó a la herramienta JOSM utilizando el plugin Import Image Plugin [49], en total se agregaron 17 corregimientos como se muestra en la figura 15.

Figura 15. Corregimientos del municipio de Pasto



✓ **Recolección de veredas**

Para la recolección de veredas se tomó como base únicamente los mapas de la Alcaldía de Pasto dividido en corregimientos, al igual que en el anterior caso, los mapas se transformaron en imagen en un formato PNG y se los agregó a la herramienta JOSM utilizando el plugin Import Image Plugin [49]. En la tabla 8, se muestran las veredas recolectadas.

Tabla 8. Veredas recolectadas por corregimientos

Corregimiento	Veredas	Corregimiento	Veredas	Corregimiento	Veredas
Buesaquillo	6	Gualmatan	6	Mocondino	4
Cabrera	6	Jamondino	3	Morasurco	10
Catambuco	22	Jongovito	7	Obonuco	7
El Encano	17	La Caldera	6	San Fernando	7
El Socorro	4	La Laguna	4	Santa Bárbara	8
Genoy	9	Mapachico	12		

Figura 16. Veredas del municipio de Pasto



5.2.3.Utilización de GPS. Para obtener una mayor precisión en la información recolectada, se utilizó un GPS Garmin GPSMAP 64s como se muestra en la figura 17, principalmente se usó en la recolección de sitios de interés rurales, debido a la complejidad para ubicarlos en el mapa, además se utilizó en la etapa de pruebas descrito en el capítulo APLICACIÓN Y ANÁLISIS DE PRUEBAS.

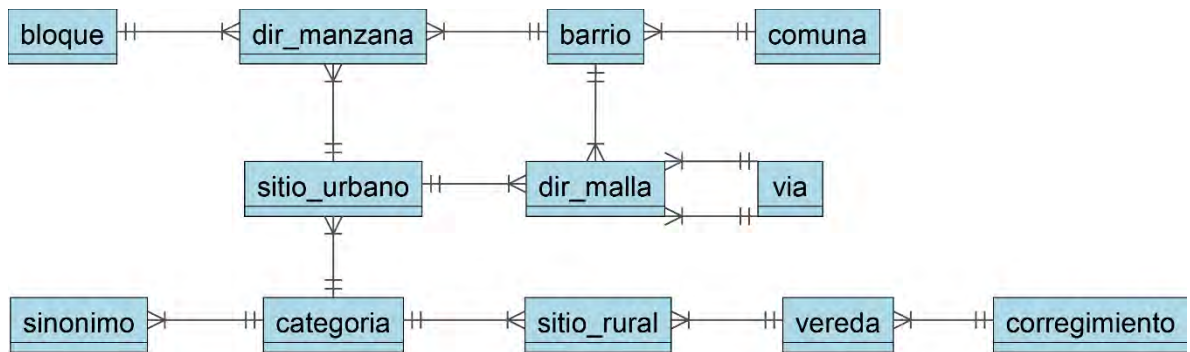
Figura 17. GPS Garmin GPSMAP 64s y recolección de sitio de interés rural



5.3. ETAPA DE CONSTRUCCIÓN DE BASE DE DATOS Y ONTOLOGÍA

5.3.1. Construcción de la base de datos del visor cartográfico. Una vez recolectada la información geográfica, se continuó con la etapa de diseñar la base de datos con el diagrama entidad relación y su posterior implementación en el Sistema Gestor de Base de Datos (SGBD) PostgreSQL, para el diseño se crearon las tablas que se describen a continuación con sus respectivas relaciones. La figura 18 muestra las tablas del esquema principal de la base de datos, el diagrama completo y el diccionario de datos se encuentra disponible en el ANEXO E.

Figura 18. Diagrama Entidad-Relación base de datos del visor cartográfico



- **Comuna:** tabla que representa las comunas pertenecientes al municipio de Pasto, tiene los atributos id, nombre y un campo que representa su geometría.
- **Corregimiento:** tabla que representa los corregimientos pertenecientes al municipio de Pasto, tiene los atributos id, nombre y geometría.
- **Barrio:** tabla que representa los barrios pertenecientes al municipio de Pasto, tiene los atributos id, nombre, geometría y la relación:
 - ❖ Un barrio está asociado a una sola comuna.
- **Vereda:** tabla que representa las veredas pertenecientes al municipio de Pasto, tiene los atributos id, nombre, geometría y la relación:
 - ❖ Una vereda está asociada a un solo corregimiento.
- **Vía:** tabla que representa las vías (calle, carrera, transversal, diagonal) presentes en las direcciones de malla vial, tiene los atributos id y nombre.
- **Bloque:** tabla que representa los bloques (manzanas) presente en las direcciones de tipo manzana, tiene los atributos id y nombre.
- **Categoría:** tabla que representa las agrupaciones a las que puede pertenecer

algún sitio de interés ya sea urbano o rural, tiene los atributos id y nombre.

- **Sitio de interés urbano:** tabla que representa los sitios de interés que se encuentran en las comunas del municipio de Pasto, tiene los atributos id, nombre, longitud, latitud, teléfono, sitio web, prioridad y la relación:
 - ❖ Un sitio urbano está asociado a una sola categoría.
 - **Sitio de interés rural:** tabla que representa los sitios de interés que se encuentran en los corregimientos del municipio de Pasto, tiene los atributos id, nombre, longitud, latitud, teléfono, sitio web, dirección, prioridad y las relaciones:
 - ❖ Un sitio rural está asociado a una sola categoría.
 - ❖ Un sitio rural está asociado a una sola vereda.
 - **Dirección de malla vial:** tabla que presenta las direcciones de malla vial pertenecientes al municipio de Pasto, tiene los atributos id, distancia, longitud, latitud y las relaciones:
 - ❖ Una dirección está asociada a un solo barrio.
 - ❖ Una dirección está asociada a una sola vía principal.
 - ❖ Una dirección está asociada a una sola vía generadora.
 - ❖ Una dirección está asociada a ninguno o a un solo sitio de interés urbano.
 - **Dirección manzana:** tabla que representa las direcciones de tipo manzana pertenecientes al municipio de Pasto, tiene los atributos id, casa, longitud, latitud y las relaciones:
 - ❖ Una dirección está asociada a un solo barrio.
 - ❖ Una dirección está asociada a un solo bloque.
 - ❖ Una dirección está asociada a ninguno a un solo sitio de interés urbano.
 - **Sinónimo:** tabla que representa sinónimos posibles para describir una misma categoría, tiene los atributos id, nombre y la relación:
 - ❖ Un sinónimo está asociado a una sola categoría.
- ✓ **Migración de datos geográficos a PostgreSQL**

Para realizar la migración de datos geográficos recolectados en formato OSM se utilizó como base scripts escritos en lenguaje Python del proyecto SIGODEP-SIG [43], los cuales al ser software libre (con licencia GNU General Public License v3) se los modificó para que funcionen en este trabajo, agregando nuevas funciones como agregar los datos a sus respectivas tablas normalizadas y evitar la duplicación

de datos al volver a ejecutar el script, esta es una forma óptima para migrar este tipo de datos.

✓ **Fuentes externas de direcciones**

Para completar la base de datos con la mayoría de direcciones existentes, se tomaron fuentes externas de direcciones de malla vial y manzana disponibles en la web, con el fin de incrementar el número total de datos recolectados. Estas fuentes fueron: el directorio telefónico de Pasto del año 2014, la curaduría urbana segunda de Pasto, el servicio público de alumbrado de Pasto (Sepal) y Empopasto. Todos estos nuevos datos al no encontrarse en una base de datos, se migraron de formatos EXCEL, PDF o HTML y se realizó una posterior limpieza:

Tabla 9. Datos iniciales de fuentes externas

Fuentes	Datos iniciales	Datos limpios
Directorio de pasto	24908	19919
Curaduría urbana segunda	27776	9461
Sepal	14041	11938
Empopasto	92203	58343

Luego se verificó si estos datos limpios tienen un formato válido de dirección urbana, para esto se utilizó una función escrita en lenguaje SQL del proyecto SIGEODEP-SIG [43] y se determinó cuales datos pasan a la siguiente etapa:

Tabla 10. Direcciones estandarizadas de fuentes externas

Fuentes	Direcciones estandarizadas	Sin formato
Directorio de pasto	13407	6512
Curaduría urbana segunda	5972	3489
Sepal	5103	6835
Empopasto	35679	22664

Posteriormente, para los datos que tienen un formato completo y válido de dirección urbana, se verificó si se encuentran almacenados en la base de datos Rikhuna y cuales no lo están y así determinar el número de direcciones faltantes.

Tabla 11. Direcciones faltantes de fuentes externas

Fuentes	Direcciones ya existentes	Direcciones faltantes
Directorio de pasto	11595	1812
Curaduría urbana segunda	4918	1054
Sepal	4236	867
Empopasto	29579	6100
TOTAL		9833

Del total de direcciones faltantes se pudo agregar a la base de datos un 44,4%, debido a que el porcentaje restante de datos no se podían ubicar en la cartografía de la ciudad o porque no eran direcciones coherentes, también estos datos se utilizaron para corregir algunas direcciones mal ubicadas, aumentando así el número total de direcciones.

✓ **Corrección de datos geográficos**

Para validar y corregir las direcciones urbanas, sitios de interés y las diferentes geometrías obtenidas, se utilizaron funciones de la extensión PostGIS [20] con el fin de obtener datos limpios y completos, para esto se determinaron varios casos de prueba y según el resultado se realizó su respectiva revisión y/o corrección.

- **Área sin delimitar Barrio:** en este caso se desea evaluar cuál es la cantidad de área dentro de las geometrías de las comunas que no tienen barrio, para esto se hace unión de las geometrías de las comunas, una unión de las geometrías de los barrios y luego se hace una diferencia, la geometría resultante debe ser nula o tener un área muy pequeña, de lo contrario se amplía la geometría de los barrios para completar la misma área que presentan las comunas.
- **Barrios que no tienen comuna:** se verificó que todos los barrios se encuentren dentro de alguna comuna, de manera similar se verifico para veredas dentro de los corregimientos, para los registros que no cumplen con esto, significa que hay problemas en las geometrías.
- **Barrios sin ninguna dirección:** se verificó si existen barrios que en su interior no se encuentre alguna dirección, ya sea de malla vial o manzana, para los registros que cumplen con esto, puede deberse a que su geometría de barrio es muy pequeña, o por la complejidad del lugar puede que no contenga ninguna dirección.
- **Barrios con pocas direcciones:** se analizó los barrios que contienen pocas direcciones de malla vial o de manzana, debido a que puede existir un error en la geometría del barrio o por la complejidad del lugar no se pudo georreferenciar más direcciones, para satisfacer este caso se tomó un valor menor a 20 direcciones.
- **Direcciones repetidas en malla vial:** según el estándar para la asignación de direcciones de malla vial no deben existir direcciones repetidas con el mismo nombre de vía principal, vía generadora y distancia, esto puede pasar por error al momento de la georreferenciación o en casos muy extraños la dirección está mal asignada en el predio.

- **Direcciones repetidas en manzana:** según el estándar para la asignación de direcciones de manzana no deben existir direcciones repetidas con el mismo nombre de barrio, bloque y casa, el error puede ser igual que en el anterior caso o porque la geometría del barrio se encuentra mal ubicada.
- **Direcciones sin barrio:** similar al caso del barrio que debe pertenecer a alguna comuna, las direcciones basadas en malla vial o de manzana también deben pertenecer a algún barrio, si esto no es así significa que el punto está fuera del área de las geometrías y debe ser reubicado o en caso contrario agrandar la geometría del barrio.
- **Direcciones con el mismo tipo de vía en malla vial:** según el estándar para la asignación de direcciones basadas en malla vial existen dos tipos de vías que la componen, una principal y otra generadora, estas dos vías no deben ser del mismo tipo, por ejemplo, no debe existir una dirección con dos calles o dos carreras, esto se da principalmente por error humano al momento de la georreferenciación.
- **Direcciones con nomenclatura diferente al estándar:** todas las direcciones recolectadas basadas en malla vial o de manzana deben cumplir una nomenclatura para nombrar a sus componentes, si esto no ocurre significa que fue un error humano al momento de realizar la georreferenciación.
- **Distancia entre direcciones de malla vial del mismo sector con un valor alto:** al momento de asignar una dirección de malla vial se define una vía principal, secundaria y una distancia, en el ejemplo de la figura 19, se muestra direcciones de vía principal 'KR 36' y vía generadora 'CL 17', una distancia mínima y máxima de 14 y 97 metros respectivamente, en este ejemplo se puede asumir que la distancia real máxima entre las diferentes direcciones no puede superar los 97 metros por ser el mayor valor, si este valor es superado puede significar que existe algún error al momento de georreferenciar los puntos o en algunos casos los valores son correctos pero el error se encuentra en la asignación de direcciones en la ciudad. Para este caso se tomó un valor mayor o igual a 100 metros.
- **Distancia entre direcciones de manzana del mismo sector con un valor alto:** igual que en el caso anterior se realiza el mismo análisis, pero ahora para direcciones basadas en manzana, la principal diferencia es que estas direcciones no tienen un atributo de distancia, por lo cual solo se toma como valor la distancia real entre los puntos, para este caso se tomó un valor mayor o igual a 100 metros.

Figura 19. Distancia entre direcciones de malla vial



Todos los casos mencionados para la corrección de datos geográficos fueron muy importantes para obtener información confiable y viable en esta investigación, principalmente para las direcciones urbanas y con ayuda de las fuentes externas aumentaron su número total, como se muestran en la tabla 12:

Tabla 12. Direcciones urbanas obtenidas en la etapa de corrección

Tipo de dirección	Numero
Malla vial	87171
Manzana	21459

En la tabla 13, se consigna el total de direcciones de malla vial que se recolectó al finalizar las etapas anteriores.

Tabla 13. Direcciones de malla vial recolectadas por comuna

Comuna	Direcciones	Comuna	Direcciones	Comuna	Direcciones
Comuna 1	19613	Comuna 5	8915	Comuna 9	5642
Comuna 2	18635	Comuna 6	4533	Comuna 10	3005
Comuna 3	3322	Comuna 7	2855	Comuna 11	2889
Comuna 4	10482	Comuna 8	4163	Comuna 12	3117

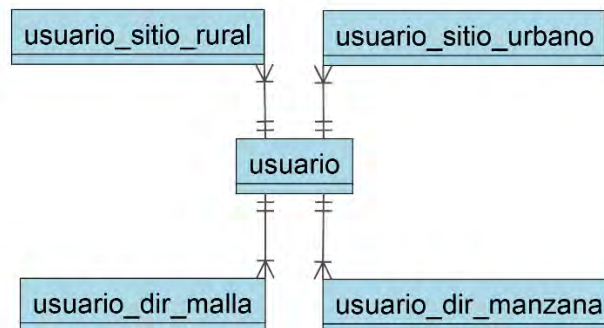
En la tabla 14, se consigna el total de direcciones de manzana que se recolectó al finalizar las etapas anteriores.

Tabla 14. Direcciones de manzana recolectadas por comuna

Comuna	Direcciones	Comuna	Direcciones	Comuna	Direcciones
Comuna 1	0	Comuna 5	2198	Comuna 9	64
Comuna 2	97	Comuna 6	4513	Comuna 10	4123
Comuna 3	3673	Comuna 7	265	Comuna 11	1736
Comuna 4	910	Comuna 8	2282	Comuna 12	1598

5.3.2. Construcción de la base de datos del panel administrativo. Teniendo en cuenta de que es necesario mantener actualizada la información de las direcciones urbanas y de los sitios de interés se planteó el desarrollo de un panel administrativo. Fue necesario implementar un nuevo esquema administrativo en la base de datos, en el cual se definió una serie de entidades únicamente dedicadas a la administración y almacenamiento provisional de datos. De esta manera se garantiza una validación antes de que los nuevos registros sean almacenados en el esquema principal de Rikhuna. Para ello se diseñó el diagrama entidad relación respectivo para su posterior implementación en el Sistema Gestor de Base de Datos (SGBD) PostgreSQL, para el diseño se crearon las tablas que se describen a continuación con sus respectivas relaciones. La figura 20 muestra las tablas del esquema administrativo de la base de datos, el diagrama completo y el diccionario de datos se encuentra disponible en el ANEXO E.

Figura 20. Diagrama Entidad-Relación base de datos del panel administrativo



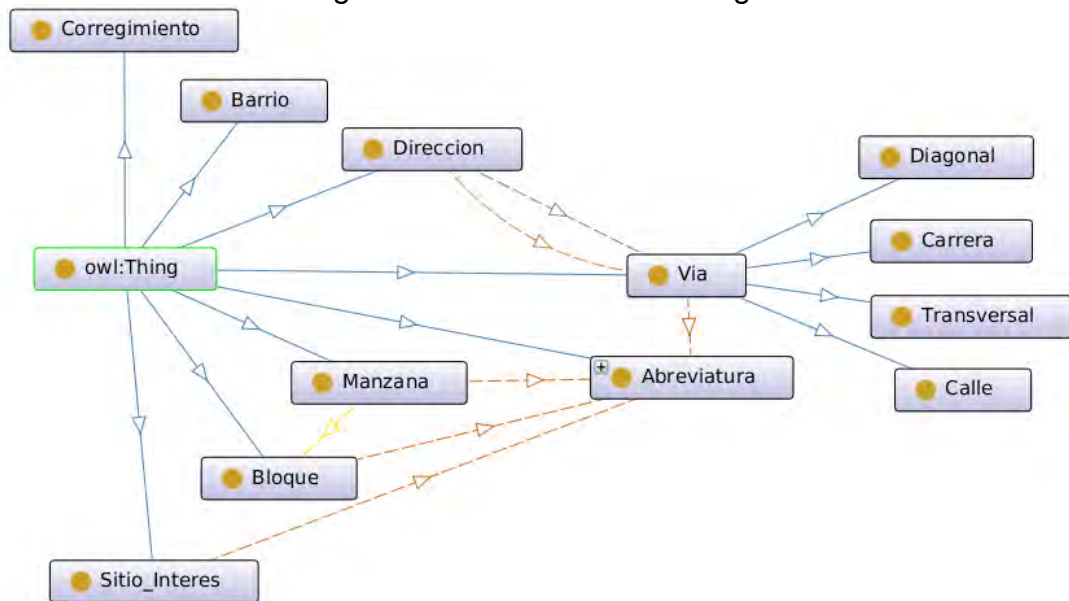
- ✓ **Usuario:** tabla que representa a los usuarios que pueden llevar a cabo las acciones de agregar, editar y eliminar direcciones urbanas y sitios de interés. Tiene los atributos id, login, password, correo, dirección, teléfono, fecha de nacimiento, usuario validado y el tipo de usuario.
- ✓ **Usuario sitio rural:** tabla que representa los sitios rurales que son añadidos por los usuarios registrados. Tiene los atributos id, id de categoría, nombre, longitud, latitud, teléfono, sitio web, dirección, id de vereda, prioridad, acción, comentarios, id de sitio rural de origen y la relación:
 - Un sitio rural tiene asociado un usuario.
- ✓ **Usuario sitio urbano:** tabla que representa los sitios urbanos que son añadidos por los usuarios registrados. Tiene los atributos id, id de categoría, nombre, longitud, latitud, teléfono, sitio web, dirección, prioridad, acción, comentarios, id de sitio urbano de origen y la relación:
 - Un sitio urbano tiene asociado un usuario.

- ✓ **Usuario malla vial:** tabla que representa las direcciones basadas en malla vial que son añadidas por los usuarios registrados. Tiene los atributos id, vía principal, vía generadora, distancia, longitud, latitud, id del barrio, acción, comentarios, id de malla vial de origen y la relación:
 - Un sitio urbano tiene asociado un usuario.

- ✓ **Usuario manzana:** tabla que representa direcciones basadas en manzana que son añadidas por los usuarios registrados. Tiene los atributos id, id del bloque, casa, longitud, latitud, id del barrio, acción, comentarios, id de manzana de origen y la relación:
 - Un sitio urbano tiene asociado un usuario.

5.3.3.Creación de la ontología Web. Posteriormente, se realizó la creación de la ontología para sitios de interés y direcciones urbanas, para esto se utilizó la herramienta de software libre Protégé [35] en formato OWL. Para definir cada una de las clases se realizó una abstracción del esquema público de la base de datos y se definió los atributos necesarios para que las búsquedas puedan relacionar diversos conceptos sobre el dominio establecido.

Figura 21. Diseño de la ontología web



La mayoría de tablas de la base de datos pasaron a ser conceptos o clases, las relaciones entre tablas pasan a ser relaciones entre clases, los atributos de cada tabla pasan a ser atributos de la clase, en donde solo se tomó los necesarios para realizar las búsquedas, como ejemplo se deja los nombres y atributos como coordenadas o geometrías son ignorados, finalmente cada registro en las tablas se

convierte en una instancia de cada clase. La migración de datos de la base de datos a la ontología se realizó desde código Java y se especifica en la Implementación del visor cartográfico.

5.4.ETAPA DE ANÁLISIS DE HERRAMIENTAS

Para definir el lenguaje de programación a utilizar para crear el visor cartográfico se tuvo en cuenta los lenguajes más populares que tienen soporte para ontologías en formato OWL y soporte para realizar consultas en SPARQL como se muestra en la tabla 15. Actualmente los lenguajes no soportan estas consultas de forma nativa, sino que todos ellos utilizan bibliotecas de clases para este uso.

Tabla 15. Comparación de lenguajes de programación con soporte a ontologías

Lenguaje de Programación	Soporte Librerías			
	RDF	RDFS	OWL	SPARQL
Php	x	x	x	x
Java (JSF)	x	x	x	x
Python (Django)	x	x		x
JavaScript (NodeJS)	x	x		x

La intención de este análisis fue encontrar una biblioteca de clases que soporte OWL y SPARQL al mismo tiempo. En algunos casos el soporte era en librerías diferentes, lo cual no era la mejor opción. A partir de lo anterior, se realizó pruebas con Php y Java, para determinar cuál es mejor para este trabajo. En la tabla 16, se muestra los criterios de evaluación para analizar estas bibliotecas y en tabla 17, se muestra el resultado obtenido.

Tabla 16. Criterios de evaluación para bibliotecas de búsqueda semántica

Prueba de aceptación	Descripción	Evaluación
Velocidad de respuesta	Velocidad de respuesta a las consultas realizadas en lenguaje SPARQL.	Alta: menor a 1 segundo. Media: entre 1 a 2 segundos. Baja: mayor a 2 segundos.
Soporte a consultas complejas	Soporte a consultas en lenguaje SPARQL de una complejidad avanzada y una petición de gran cantidad de datos.	Alta: realiza la consulta correctamente con un tiempo menor a 3 segundos. Media: realiza la consulta correctamente con un tiempo entre 3 o 5 segundos. Baja: no realiza la consulta o realiza la consulta en un tiempo mayor a 5 segundos.

Documentación	Calidad de la documentación y la comunidad que presenta.	Alta: documentación amplia con una comunidad creada. Media: documentación amplia sin una comunidad creada. Baja: documentación básica sin una comunidad creada.
Escalabilidad	Adaptación de nuevos cambios en la ontología por parte de la biblioteca.	Alta: no presenta problemas en rendimiento a los cambios. Media: el rendimiento baja a la mitad debido a los cambios. Baja: el rendimiento baja más de la mitad debido a los cambios.
Usabilidad	Facilidad de uso e integración de la biblioteca en un proyecto.	Alta: complejidad baja para la instalación, configuración y utilización. Media: complejidad media para la instalación, configuración y utilización. Baja: complejidad alta para la instalación, configuración y utilización.

Tabla 17. Comparación funcionalidades bibliotecas de búsqueda semántica

Prueba de aceptación	ARC2 (Php)	Jena (Java)
Velocidad de respuesta	Alta	Alta
Soporte a consultas complejas	Baja	Alta
Documentación	Baja	Media
Escalabilidad	Baja	Alta
Usabilidad	Media	Alta

A partir de lo anterior, para el desarrollo de este trabajo se definió a Java como el lenguaje de programación a utilizar con el framework Jena. En cuanto a la tecnología del lado del servidor se decidió desarrollar en el lenguaje de programación Java EE con el framework JSF debido a las facilidades para integrar la biblioteca elegida anteriormente.

Aparte de seleccionar con que lenguaje de programación trabajar, también se analizó diferentes APIs (Application Programming Interface) web para la creación de rutas, que sirven como base para satisfacer una funcionalidad del visor que se detalla en la etapa de Implementación del visor cartográfico; En la tabla 18 se muestra los criterios de evaluación para las cinco APIs escogidas, en la tabla 19 se presentan el resultado obtenido, en donde se escogió el API Mapzen [50] por presentar mayores ventajas que las otras.

Tabla 18. Criterios de evaluación de APIs para la creación de rutas

Pruebas de aceptación	Descripción	Evaluación
Soporte a idioma español	Traducción de la descripción de la ruta de idioma inglés a idioma español.	<p>Alta: traducción detallada de la descripción de la ruta.</p> <p>Media: traducción básica de la descripción de la ruta.</p> <p>Baja: no realiza la traducción de la descripción de la ruta.</p>
Compatibilidad con PostGIS	Formato de la geometría tipo línea de la ruta y su compatibilidad con PostGIS.	<p>Alta: no se deben realizar modificaciones a la geometría para utilizar con PostGIS.</p> <p>Media: se deben realizar modificaciones a la geometría para utilizar con PostGIS.</p> <p>Baja: no genera una geometría compatible con PostGIS.</p>
Usabilidad	Facilidad de uso e integración de la API en un proyecto.	<p>Alta: complejidad baja para la configuración y utilización.</p> <p>Media: complejidad media para la configuración y utilización.</p> <p>Baja: complejidad alta para la configuración y utilización.</p>
Velocidad de respuesta	Velocidad de respuesta a las consultas realizadas.	<p>Alta: menor a 0.5 segundos.</p> <p>Media: menor a 1 segundo.</p> <p>Baja: mayor a 1 segundo.</p>
Documentación	Calidad de la documentación en el uso y respuesta de la API.	<p>Alta: documentación completa sobre peticiones, parámetros y errores.</p> <p>Media: documentación parcial sobre peticiones, parámetros y errores.</p> <p>Baja: documentación básica sobre peticiones, parámetros y errores.</p>
Consultas personalizadas	Consulta a la API con atributos personalizables de la ruta para obtener más información de ella.	<p>Alta: gran cantidad de atributos personalizables y opcionales.</p> <p>Media: cantidad regular de atributos personalizables y opcionales.</p> <p>Baja: cantidad mínima o nula de atributos personalizables y opcionales.</p>
Descripción de la ruta	Calidad y precisión de la descripción de la ruta.	<p>Alta: descripción completa del recorrido de la ruta.</p> <p>Media: descripción parcial del recorrido de la ruta.</p> <p>Baja: descripción básica del recorrido de la ruta.</p>

Soporte a formato JSON	Calidad del formato JSON e información adicional de la ruta.	Alta: formato JSON con información adicional de la ruta. Media: formato JSON con información básica de la ruta. Baja: No soporta formato JSON.
------------------------	--	---

Tabla 19. Comparación de APIs para la creación de rutas

Pruebas de aceptación	Mapbox	MapQuest	Open Source Routing Machine	Open Route Service	Mapzen
Soporte a idioma español	Baja	Alta	Baja	Alta	Alta
Compatibilidad con PostGIS	Alta	Alta	Alta	Alta	Media
Usabilidad	Alta	Alta	Alta	Alta	Alta
Velocidad de respuesta	Media	Alta	Alta	Baja	Alta
Documentación	Alta	Alta	Media	Media	Alta
Consultas personalizadas	Media	Alta	Baja	Media	Alta
Descripción de la ruta	Alta	Media	Baja	Alta	Alta
Soporte a formato JSON	Alta	Alta	Alta	Baja	Alta

5.5.CONSTRUCCIÓN DE RIKHUNA

En este capítulo se presentan los resultados del proceso de construcción de Rikhuna, tanto del visor cartográfico como del panel administrativo. En la implementación se hace referencia a la forma de cómo está estructurada la aplicación y los paquetes de código teniendo en cuenta las funcionalidades; en la parte de los resultados de la construcción se muestran las interfaces graficas de cada una de las partes que componen el visor cartográfico y el panel administrativo.

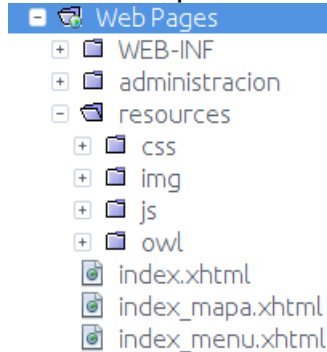
5.5.1.Implementación visor cartográfico. En este capítulo se describe la implementación realizada para la construcción del visor cartográfico, explicando cada uno de los paquetes mencionados en la arquitectura del sistema.

✓ Implementación módulo presentación

Este módulo permite al usuario final la visualización de resultados de las búsquedas por medio de una interfaz gráfica, compuesta por un menú de búsqueda y un mapa. Estas páginas en formato xhtml de JSF se encargan de interactuar con el usuario y el intercambio de información entre los controladores y la base de datos. Además, este módulo comprende distintos recursos como archivos CSS, imágenes, un

archivo JavaScript y archivos OWL para la representación de la ontología. En la figura 22, se muestra la estructura de este módulo.

Figura 22. Módulo presentación visor cartográfico.



A continuación, se describen detalladamente los archivos xhtml que componen el módulo de presentación.

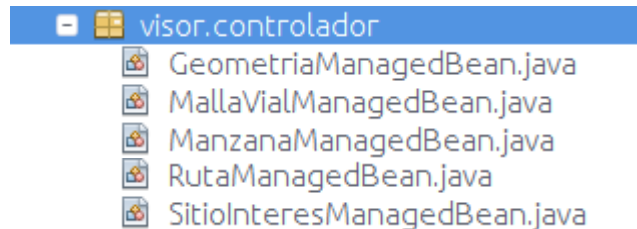
- **index.xhtml:** es la vista principal del visor cartográfico y agrupa a las vistas del menú y mapa.
- **index_mapa.xhtml:** es la vista que presenta el mapa del visor cartográfico. Inicialmente se presenta el mapa de Pasto con los principales sitios de interés de la ciudad. En esta sección se presentan los resultados visuales de las búsquedas, se presenta un mensaje cuando no se generan resultados, se indica la dirección o el sitio de interés ingresado con el correspondiente icono y se cargan los sitios de interés cercanos al lugar de la búsqueda. Se presenta además la opción de generar rutas desde la ubicación de la dirección, sitio de interés o haciendo clic en el mapa.
- **index_menu.xhtml:** es la vista que presenta el menú de búsquedas para el visor cartográfico. Para cada una de las búsquedas, se presenta un cuadro de texto en el cual el usuario ingresa la dirección urbana o sitio de interés. Posteriormente, se despliega información relacionada con la búsqueda realizada. Seguido al menú se presenta una vista de Google Street View para proporcionar una ayuda visual al usuario final obteniendo una imagen bastante aproximada del lugar ingresado.

✓ Implementación módulo núcleo

Este módulo contiene los paquetes encargados de procesar las búsquedas ingresadas por el usuario. A continuación, se describen los paquetes que componen el núcleo del sistema.

- **visor.controlador:** este paquete es el encargado de manipular información relacionada con geometrías, direcciones, sitios de interés y rutas. A continuación, se detallan en la figura 23, cada una de las clases que lo componen.

Figura 23. Clases del paquete visor.controlador

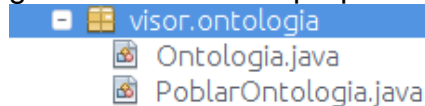


- ❖ **MallaVialManagedBean.java:** es la clase encargada de procesar las búsquedas de malla vial ingresadas por el usuario. Envía estas búsquedas a una clase que gestiona las consultas en la ontología, a partir de este resultado se decide si se obtiene la información completa de la dirección en la base de datos o se hace una búsqueda por la dirección más cercana al valor ingresado o el mensaje que no encontró dirección y retorna el resultado al módulo de presentación.
- ❖ **ManzanaManagedBean.java:** es la clase encargada de procesar las búsquedas de manzana ingresadas por el usuario. En esta búsqueda se piden dos valores, la dirección y el barrio, cada una de ellas es enviada a una clase que gestiona las consultas en la ontología, según el resultado de ambos valores se decide si se obtiene la información completa de la dirección en la base de datos o se hace una búsqueda con nombre similares del barrio o el mensaje que no encontró dirección y retorna el resultado al módulo de presentación.
- ❖ **SitiInteresManagedBean.java:** es la clase encargada de procesar las búsquedas de sitios de interés ingresadas por el usuario. En esta clase existen tres tipos de búsquedas, sitio de interés específico, categoría y sitios de interés cercanos, para cada una de ellas la clase decide que métodos ejecutar y enviar estas búsquedas a una clase que gestiona las consultas en la ontología, a partir de este resultado se decide si se obtiene la información completa del sitio en la base de datos o se hace una búsqueda por palabras similares o el mensaje que no encontró sitio de interés y retorna el resultado al módulo de presentación.
- ❖ **GeometriaManagedBean.java:** es la clase encargada de procesar las búsquedas de geometrías ingresadas por el usuario. En esta búsqueda se pueden buscar dos opciones, barrios y corregimientos, cada una de ellas es enviada a la clase para las consultas en la ontología, según el

resultado de ambos valores se decide si se obtiene la información completa de la dirección en la base de datos o se hace una búsqueda con nombre similares o el mensaje que no encontró la geometría y retorna el resultado al módulo de presentación.

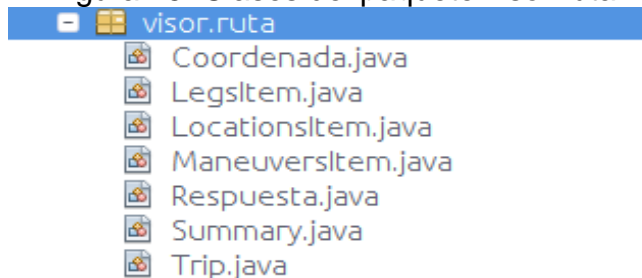
- ❖ **RutaManagedBean.java:** es la clase encargada de procesar las rutas solicitadas por el usuario. Para generar una ruta son necesarios dos puntos, un punto inicial y otro final, estos valores el usuario los puede seleccionar al momento de realizar una búsqueda de direcciones o sitios de interés o al hacer clic en el mapa, luego estos valores son enviados a la Api web para la creación de rutas Mapzen, los resultados son procesados y se retorna el resultado al módulo presentación.
- **visor.ontologia:** este paquete contiene las clases encargadas de la conexión, gestión y mantenimiento de la ontología. En la figura 24, se detallan las clases que componen este paquete.

Figura 24. Clases del paquete visor.ontologia



- ❖ **Ontologia.java:** es la clase encargada de la conexión con la ontología para las direcciones urbanas y sitios de interés, así como para definir los distintos tipos de consultas en lenguaje SPARQL.
- ❖ **PoblarOntologia.java:** es la clase que permite la actualización de la ontología a partir de la base de datos Rikhuna. Cada registro de las tablas de la base de datos se insertan a la ontología con ayuda del lenguaje SPARQL.
- **visor.ruta:** en este paquete se almacenan las clases necesarias para el funcionamiento de la creación de rutas en el visor cartográfico. En la figura 25, se detallan las clases correspondientes a este paquete.

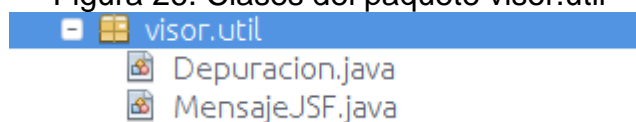
Figura 25. Clases del paquete visor.ruta



Estas clases funcionan en conjunto con el archivo RutaManagedBean.java, se componen simplemente de atributos con sus métodos getters, setters y permiten el parseo del archivo JSON que llega desde el API de rutas Mapzen a clases Java mediante la biblioteca de clases Retrofit y Gson.

- **visor.util:** este paquete contiene algunas clases utilitarias necesarias para el correcto funcionamiento del visor cartográfico. En la figura 26, se detallan las clases que componen este paquete.

Figura 26. Clases del paquete visor.util



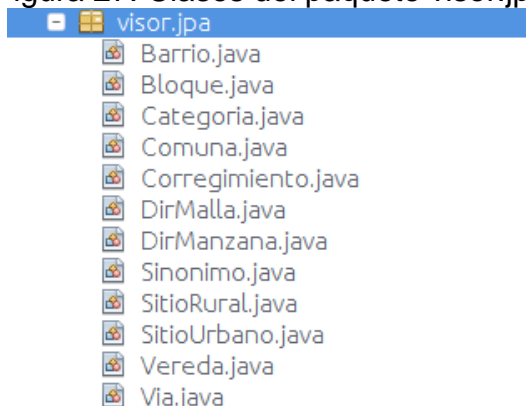
- ❖ **Depuracion.java:** es la clase encargada de limpiar la cadena ingresada por el usuario, quitando espacios, caracteres especiales, haciendo una separación entre las letras y números para que las búsquedas, en el caso de direcciones urbanas se hagan de acuerdo al estándar. Se eliminan palabras repetidas y algunas palabras frecuentes que son innecesarias para realizar la búsqueda.
- ❖ **MensajeJSF.java:** es la clase encargada de definir los distintos tipos de mensaje que se mostraran como resultado de las búsquedas realizadas. Se presentan mensajes para información, advertencia o error según corresponda.

✓ Implementación módulo conexión

En este módulo se reúnen los paquetes encargados de la conexión y gestión con la base de datos Rikhuna en el esquema público. Para esta conexión se utilizaron recursos como la API Java Data Base Conector (por sus siglas JDBC), Java Persistence Application (JPA) y Enterprise JavaBeans (EJB). A continuación, se describen los paquetes que componen este módulo.

- **visor.jpa:** este paquete contiene las clases mapeadas a partir de las tablas de la base de datos, con sus atributos y relaciones. Cada clase es creada a partir de cada tabla de la base de datos Rikhuna en su esquema público, además se encuentran definidos métodos getter, setter para cada atributo y algunas consultas predefinidas en lenguaje EJB-QL (Enterprise JavaBeans Query Language). En la figura 27, se detallan las clases que componen este paquete.

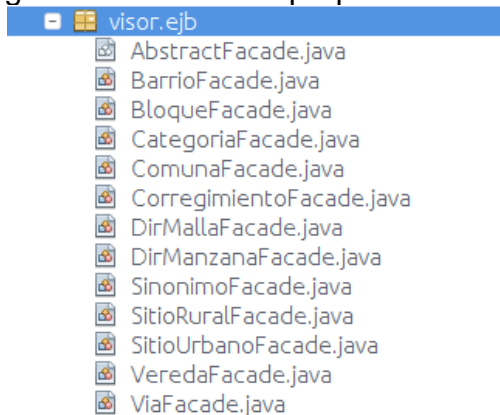
Figura 27. Clases del paquete visor.jpj



- **visor.ejb:** este paquete contiene las clases que se encargan de realizar la gestión de la información contenida en la base de datos Rikhuna. En la figura 28 se detallan las clases que componen este paquete.
 - ❖ **AbstractFacade.java:** es la clase abstracta que realiza a través de sus métodos las tareas de listar, buscar, contar, actualizar, borrar y/o insertar cualquier tipo de dato en la respectiva tabla.

El resto de clases heredan de la clase AbstractFacade.java para realizar cualquiera de las tareas mencionadas en la respectiva tabla, además algunas clases tienen definidas consultas personalizadas para el uso del algoritmo jarowinkler de la extensión de PostgreSQL pgsimilarity y otras funciones propias de PostGIS.

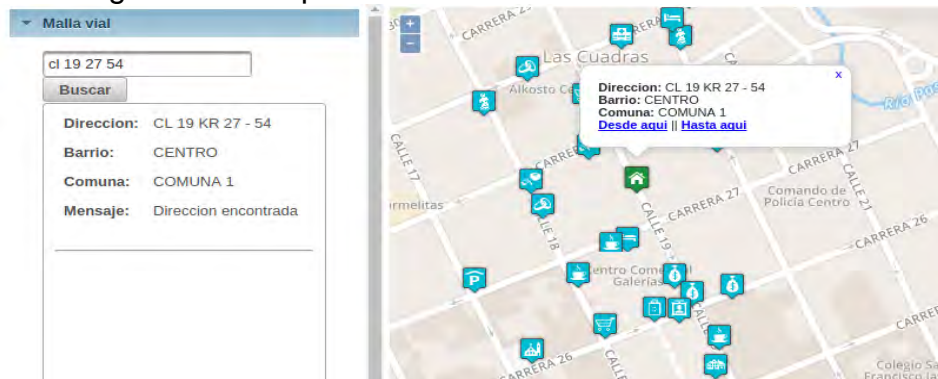
Figura 28. Clases del paquete visor.ejb



5.5.2.Construcción visor cartográfico. El visor cartográfico permite realizar cuatro tipos de búsquedas principales, en todas ellas hace una interpretación semántica para obtener un resultado, estas son:

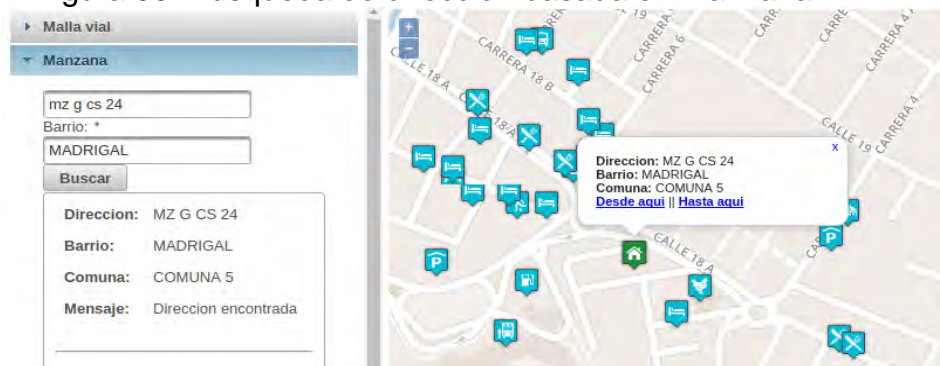
- ✓ **Búsqueda de dirección basada en malla vial:** como se muestra en la figura 29, en esta búsqueda el usuario ingresa una dirección de malla vial y se muestra información relacionada con la dirección ingresada, los resultados se presentan en el mapa con la ubicación geográfica y sitios de interés cercanos para que el usuario se pueda ubicar fácilmente en la cartografía de la ciudad.

Figura 29. Búsqueda de dirección basada en malla vial



- ✓ **Búsqueda de dirección basada en manzana:** como se muestra en la figura 30, en esta búsqueda el usuario ingresa una dirección correspondiente a manzana y el nombre del barrio, el sistema muestra información relacionada con la dirección ingresada, además de ayudar al usuario autocompletando el nombre del barrio, los resultados se presentan en el mapa con la ubicación geográfica y sitios de interés cercanos.

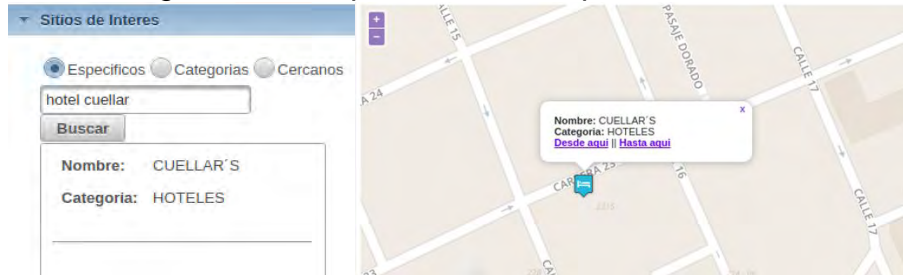
Figura 30. Búsqueda de dirección basada en manzana



- ✓ **Búsqueda de sitios de interés:** este tipo de búsqueda se divide en 3 casos:
 - ❖ **Búsqueda sitio específico:** como se muestra en la figura 31, en esta búsqueda el usuario ingresa el nombre del sitio de interés específico, el visor realiza una búsqueda de sinónimos si se ingresa para una categoría y se muestran los resultados relacionados junto con información detallada

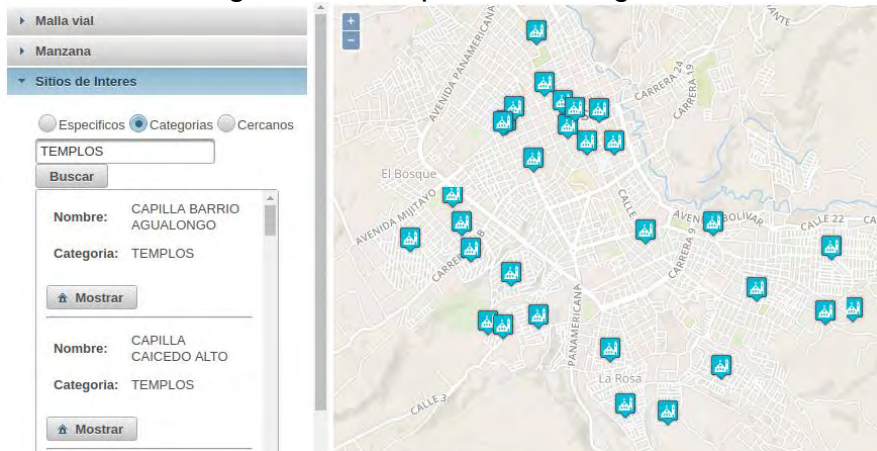
de cada uno de éstos. En el mapa se presenta la ubicación geográfica del sitio.

Figura 31. Búsqueda de sitio específico



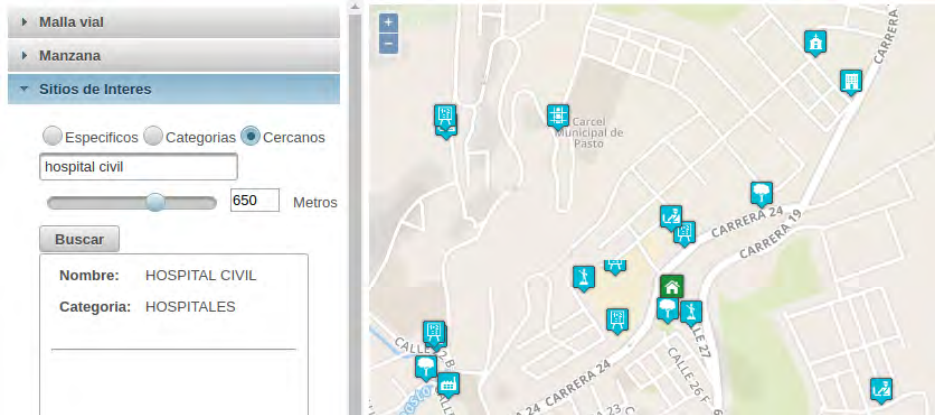
- ❖ **Búsqueda categorías:** como se muestra en la figura 32, en esta búsqueda el usuario ingresa el nombre de la categoría de los sitios de interés, el visor realiza una búsqueda de sinónimos además de autocompletar el nombre de la categoría, el sistema muestra los sitios que pertenecen a la búsqueda ingresada y proporciona información detallada de cada uno de éstos. En el mapa se presenta la ubicación geográfica de todos los sitios de la categoría.

Figura 32. Búsqueda de categoría



- ❖ **Búsqueda sitios cercanos:** como se muestra en la figura 33, en esta búsqueda el usuario ingresa el nombre del sitio de interés junto con el rango de metros para mostrar los sitios cercanos. En el mapa se presenta la ubicación geográfica y sitios de interés cercanos de acuerdo con la distancia indicada. El valor máximo es de 1000 metros.

Figura 33. Búsqueda de sitios cercanos



- ✓ **Búsqueda de barrios y corregimientos:** como se muestra en la figura 34 y 35, en esta búsqueda el usuario ingresa el nombre del barrio o corregimiento que pertenece al municipio, el visor ayuda al usuario autocompletando dicho nombre y muestra información relacionada sobre el resultado obtenido. En el mapa se muestra la ubicación geográfica de la geometría correspondiente a la búsqueda, las geometrías de veredas en el caso de los corregimientos y los sitios de interés que pertenecen a ese barrio o corregimiento.

Figura 34. Búsqueda de barrios

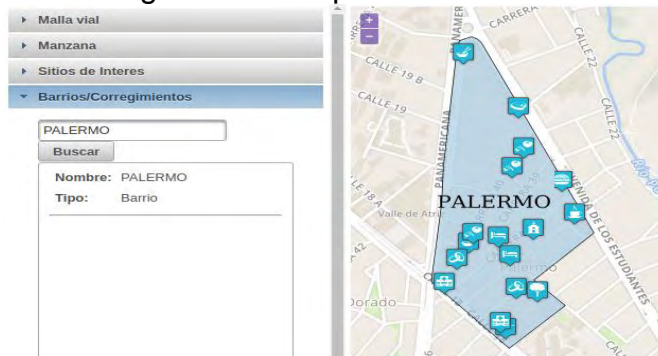
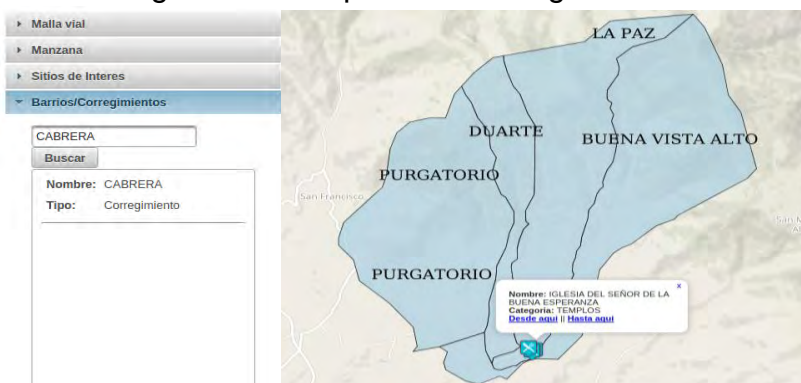


Figura 35. Búsqueda de corregimientos



Junto con las opciones de búsqueda, se han desarrollado funcionalidades adicionales que se detallan a continuación:

- ✓ **Generación de rutas:** como se muestra en la figura 36, el sistema permite generar una ruta entre 2 puntos definidos por el usuario. Para el inicio y final de la ruta es posible elegir un punto cualquiera de la cartografía de la ciudad, una búsqueda por malla vial, manzana o un sitio de interés. La longitud de la ruta debe ser menor a 10 km. Es posible elegir entre diversos medios de transporte para la generación de la ruta: a pie, bicicleta o vehículo. Se muestran las indicaciones para llegar al destino, el tiempo y la distancia de la ruta. En el mapa se presenta la línea de la ruta junto con los sitios de interés cercanos a la misma en un rango de 90 metros.
- ✓ **Ubicación del usuario:** como se muestra en la figura 37, el usuario puede elegir si desea que el sistema acceda a su ubicación actual para mostrar los sitios de interés cercanos a dicha posición. El botón está en la parte superior derecha de la ventana del visor cartográfico. En el mapa se muestra la ubicación actual del usuario y los sitios de interés cercanos.
- ✓ **Vista Google Street View:** como se muestra en la figura 38, se incorporó al visor cartográfico Rikhuna una ventana de Google Street View y funciona para cualquier búsqueda dentro del área urbana del municipio de Pasto. Esta vista estará disponible en las búsquedas de direcciones de malla vial, manzana y al momento de visualizar cualquier sitio de interés. Teniendo en cuenta que es una herramienta externa a Rikhuna, las imágenes proporcionadas pueden ser relativamente antiguas e incompletas.

Figura 36. Generación de rutas

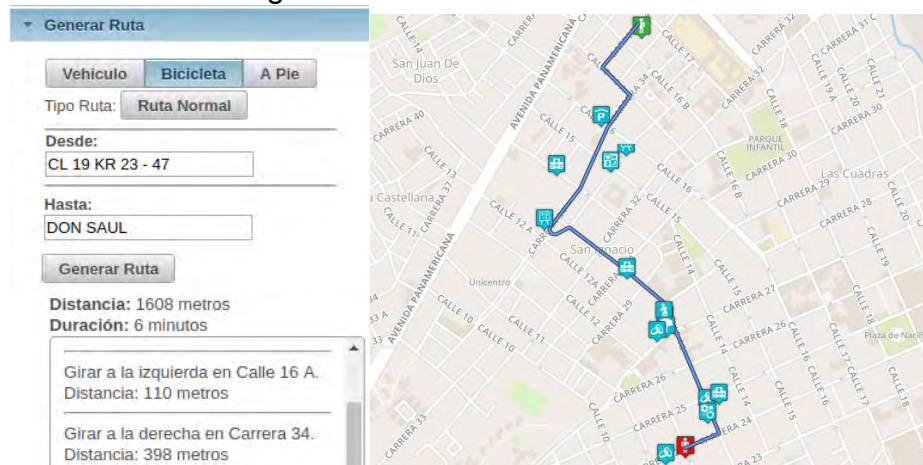


Figura 37. Ubicación del usuario

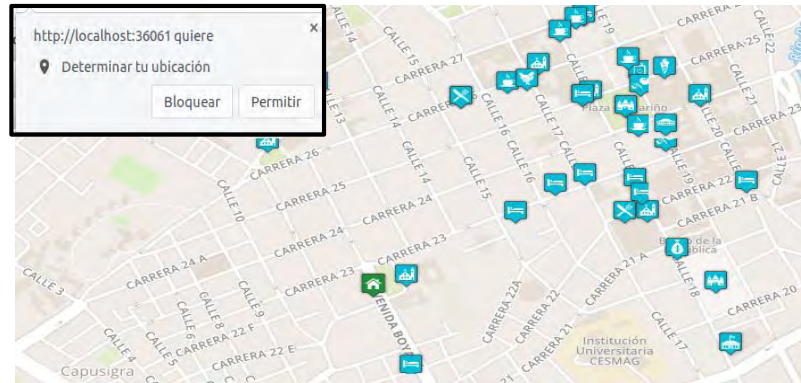
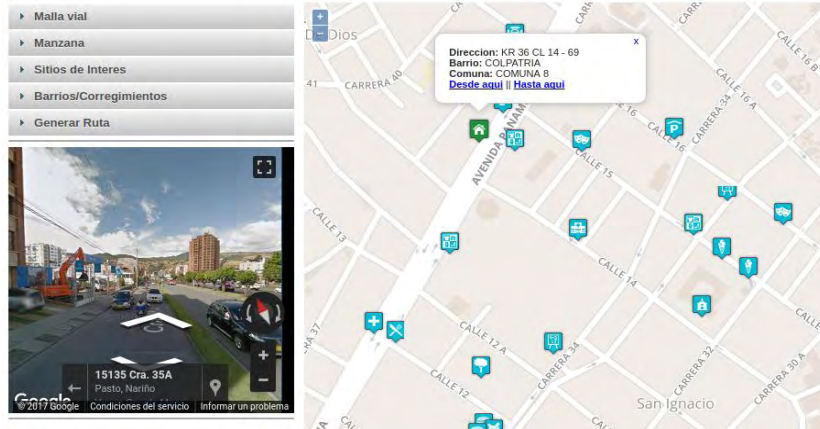


Figura 38. Vista google Street view

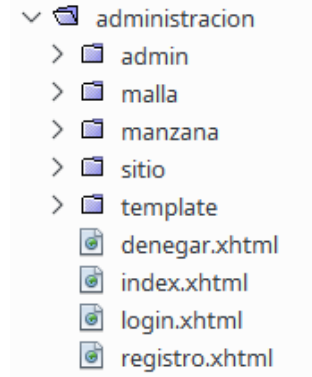


5.5.3. Implementación panel administrativo. En este capítulo se describe la implementación realizada en el panel administrativo, explicando los paquetes que compone a cada módulo de la arquitectura del sistema.

✓ **Implementación módulo de presentación**

Este módulo permite el ingreso al panel administrativo tanto para el usuario normal como para el usuario administrador, así como gestionar sus sesiones. Está compuesto por páginas en formato xhtml de JSF, las cuales se encargan de interactuar con el usuario y el intercambio de información entre los controladores y la base de datos. En la figura 39, se muestra la estructura de este módulo.

Figura 39. Módulo presentación panel administrativo



- **Implementación módulo de registro**

A continuación, se describe el archivo.xhtmll que compone este módulo:

- ❖ **registro.xhtmll:** se compone de la interfaz para el registro usuarios. Cuenta con un formulario donde se solicita la información mínima para almacenar un nuevo registro. Este módulo hace uso de funciones de la biblioteca JavaMail con la cual se facilita el envío y recepción de e-mail desde código Java. Gracias a esta biblioteca de clases se puede realizar una confirmación de registro exitoso al correo que el usuario registra en el formulario, cerciorándose así de que la información brindada es válida. Cabe mencionar que el registro de contraseñas pasa por un algoritmo de encriptación MD5 para un mayor nivel de seguridad. En la figura 40, se muestra la interacción entre el usuario y este módulo.

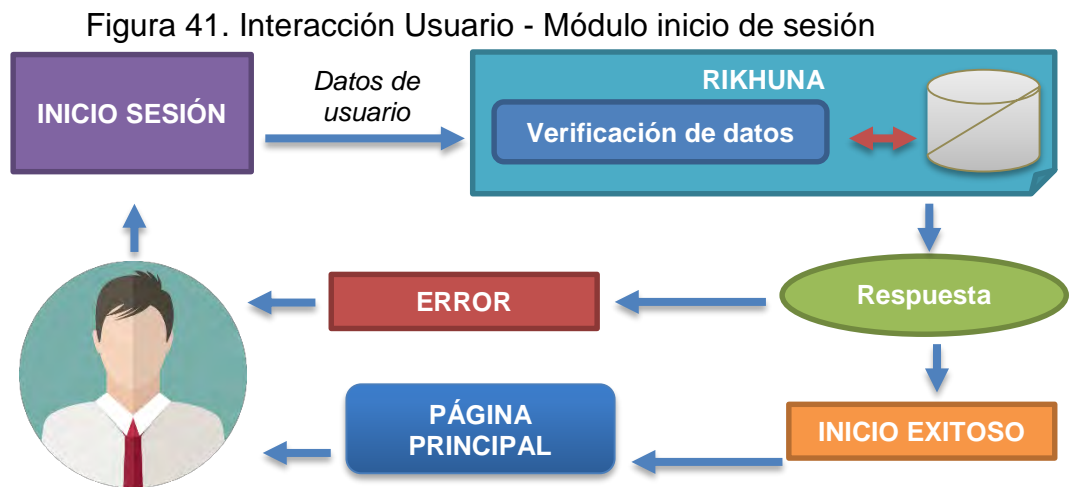
Figura 40. Interacción Usuario - Módulo Registro



- **Implementación módulo de inicio de sesión**

Una vez se lleve a cabo el registro en el esquema administrativo de la base de datos, un usuario puede iniciar sesión y acceder a la vista principal del panel administrativo. En la figura 41 se muestra la interacción entre el usuario y este módulo. A continuación, se describe los archivos xhtml que componen este módulo:

- ❖ **login.xhtml:** esta vista presenta dos campos donde se solicita el login y password del usuario, los cuales posterior al ingreso son verificados en la tabla de usuarios del esquema administrativo, de presentarse algún tipo de inconsistencia se muestra el respectivo mensaje de error en la vista.
- ❖ **denegar.xhtml:** esta vista se encarga de mostrar un mensaje de error cuando el usuario no ha iniciado sesión o cuando no tiene los permisos suficientes para realizar una tarea.



- **Implementación módulo principal**

Sobre este módulo los usuarios pueden llevar a cabo tareas como agregar, actualizar y eliminar información relacionada con las direcciones o sitios de interés. Para esto se dispone de una serie de formularios, en los cuales se solicita el ingreso de datos mínimos que puedan ser validados y de esta manera llevar a cabo el respectivo registro. A continuación, se describe los archivos xhtml que componen este módulo:

- ❖ **index.xhtml:** en este módulo se trabaja sobre una plantilla .xhtml, en la cual se incrusta una serie vistas también en formato .xhtml dependiendo de las acciones que el usuario quiera realizar.

Cada uno de los paquetes (malla, manzana, sitio) contiene los archivos .xhtml que serán la presentación final al usuario. En cada paquete se cuenta con los respectivos archivos como se muestra en la tabla 20.

Tabla 20. Paquetes módulo de presentación – panel administrativo

Malla vial	Manzana	Sitio de interés	Administrador
crear.xhtml	crear.xhtml	crear.xhtml	index.xhtml
editar.xhtml	editar.xhtml	editar.xhtml	
eliminar.xhtml	eliminar.xhtml	eliminar.xhtml	

La función que estos archivos deben desempeñar es similar pero cada uno en su respectivo dominio. La información solicitada está basada en los parámetros mínimos necesarios definidos por el IGAC para establecer una dirección urbana basada en malla vial o manzana, y de parámetros mínimos que se definieron como necesarios para sitios de interés tanto urbanos como rurales. A continuación, se describe de forma general que hace cada archivo .xhtml:

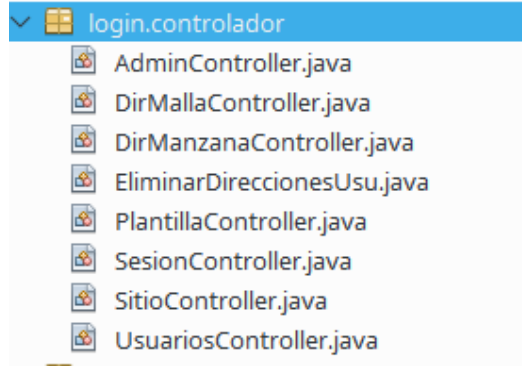
- ❖ **crear.xhtml:** brinda al usuario la posibilidad de agregar un nuevo registro en la base de datos, para ello despliega un formulario y solicita la información correspondiente. Esta información pasa a ser procesada y si es válida se lleva a cabo el respectivo registro de la operación realizada.
- ❖ **editar.xhtml:** brinda al usuario la posibilidad de corregir un registro que cuente con inconsistencias tales como una mala geolocalización, posibles datos erróneos en sitios de interés, vías principales/generadoras o en las placas de una dirección, entre otros. Los datos que se pueden editar son los mismos que se solicitan en la vista de crear.xhtml.
- ❖ **eliminar.xhtml:** permite eliminar un registro realizado en las opciones de editar y crear mencionadas anteriormente, puesto que la actualización final de la información suministrada se llevará a cabo solamente hasta cuando un administrador del sistema valide que la información suministrada es correcta y real.
- ❖ **index.xhtml:** permite al administrador del sistema ver la lista de direcciones y sitios de interés que los usuarios desean agregar, editar o eliminar. Se le despliega la información para verificar si es válida o no y las opciones para proceder al registro o descarte respectivo.

✓ **Implementación módulo núcleo**

Este módulo contiene los paquetes necesarios para gestionar las diferentes tareas que se realiza en el panel administrativo. A continuación, se describen los paquetes que componen el núcleo del sistema.

- **login.controlador:** este paquete se encarga del procesamiento y validación de la información registrada por los usuarios respecto a direcciones o sitios de interés, controla sus sesiones y realiza la encriptación de contraseñas. En la figura 42, se detallan las clases que componen este paquete.

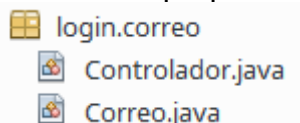
Figura 42. Clases del paquete login.controlador



- ❖ **AdminController.java:** es la clase encargada de procesar las diferentes solicitudes que provienen del administrador (gestión de usuarios y registros) y validar la información de direcciones urbanas y sitios de interés que provienen de los usuarios registrados en el sistema, además se encarga de la gestión de la ontología.
- ❖ **DirManzanaController.java:** es la clase encargada de validar, actualizar y almacenar las direcciones basadas en manzana registradas por los usuarios.
- ❖ **EliminarDireccionesUsu.java:** es la clase encargada de eliminar las direcciones basadas tanto en malla vial como en manzana que se encuentran en la base de datos Rikhuna en su esquema administrativo.
- ❖ **DirMallaController.java:** es la clase que se encarga de validar, actualizar y almacenar direcciones basadas en malla vial registradas por los usuarios.
- ❖ **PlantillaController.java:** es la clase que verifica que los usuarios estén autenticados para poder acceder a la plantilla base que se encuentra en el módulo principal.
- ❖ **SesionController.java:** es la clase que lleva a cabo las funcionalidades de iniciar y cerrar sesión por parte de los usuarios registrados.
- ❖ **SitioController.java:** es la clase que se encarga de adicionar, eliminar y actualizar los sitios de interés.

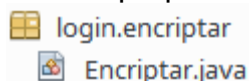
- ❖ **UsuariosController.java:** es la clase encargada de registrar nuevos usuarios en la base de datos Rikhuna en su esquema administrativo.
- **login.correo:** este paquete contiene las clases encargadas de enviar el correo de confirmación cuando un usuario lleva a cabo el registro en el panel administrativo. En la figura 43, se detallan las clases que componen este paquete.

Figura 43. Clases del paquete login.correo



- ❖ **Controlador.java:** es la clase que se encarga de definir las propiedades necesarias para enviar la confirmación al correo que el usuario ha suministrado. Implementa el protocolo SMTP para el intercambio de correos electrónicos.
- ❖ **Correo.java:** es la clase que representa un objeto de tipo correo, el cual permite la definición y acceso a los atributos mínimos que un correo electrónico necesita para poder ser enviado (destinatario, asunto, contenido, etc.).
- **login.encriptar:** este paquete almacena la clase Encriptar.java como se muestra en la figura 44, se encarga de llevar a cabo la encriptación de las claves de usuarios basado en el algoritmo MD5. Cumple la función de dar mayor seguridad a la información almacenada por el usuario al momento del registro y posterior inicio de sesión. Esta clase recibe como parámetro una cadena de texto la cual pasa a ser procesada por el algoritmo y posteriormente retorna la encriptación respectiva.

Figura 44. Clase del paquete login.encriptar

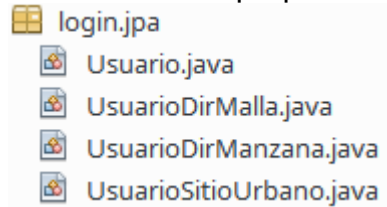


✓ Implementación Módulo de conexión

En este módulo se reúnen los paquetes encargados de la conexión y gestión con la base de datos Rikhuna en su esquema administrativo. Para esta conexión se utilizaron recursos como la API Java Data Base Conector (JDBC), Java Persistence Application (JPA) y Enterprise JavaBeans (EJB). A continuación, se describen los paquetes que componen este módulo.

- **login.jpj:** este paquete contiene las clases mapeadas a partir de las tablas de la base de datos, con sus atributos y relaciones. Cada clase es creada a partir de cada tabla de la base de datos, además se encuentran definidos métodos getter, setter para cada atributo y algunas consultas predefinidas en lenguaje EJB-QL (Enterprise JavaBeans Query Language). En la figura 45, se detallan las clases que componen este paquete.

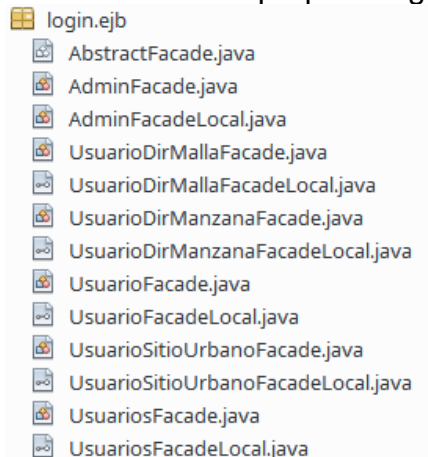
Figura 45. Clases del paquete login.jpj



- **login.ejb:** este paquete contiene las clases que se encargan de realizar la gestión de la base de datos Rikhuna en su esquema administrativo. En la figura 46, se detallan las clases que componen este paquete.
 - ❖ **AbstractFacade.java:** es la clase abstracta que realiza a través de sus métodos las tareas de listar, buscar, contar, actualizar, borrar y/o insertar cualquier tipo de dato en la respectiva tabla.

Las clases terminadas en *Facade.java heredan de la clase AbstractFacade.java para realizar cualquiera de las tareas mencionadas en la respectiva tabla, además algunas clases tienen definidas consultas personalizadas para la gestión de información almacenados por los usuarios y otras con el uso de PostGIS. Las clases terminadas en *FacadeLocal.java se implementaron por buena práctica ya que dan una mayor seguridad para el panel administrativo, estas hacen uso de las rutinas que cada clase *Facade.java utiliza.

Figura 46. Clases del paquete login.ejb



5.5.4.Construcción panel administrativo. A continuación, se presentan las principales funcionalidades que se pueden realizar en el panel administrativo, estas son:

- ✓ **Registro de usuarios:** como se muestra en la figura 47, se presenta un formulario al momento que un usuario decida registrarse y contribuir con información a esta investigación. Se presenta los datos que el usuario debe suministrar para acceder al panel principal. Los campos que se solicitan son: nombres, apellidos, correo electrónico, login y password.

Figura 47. Formulario de registro de usuarios



Registro

Nombre(s) *	<input type="text"/>
Apellido(s) *	<input type="text"/>
Correo *	<input type="text"/>
Login *	<input type="text"/>
Password *	<input type="password"/>
Repetir Password *	<input type="password"/>
	<input type="checkbox"/> Registrar

(*) Campos requeridos

- ✓ **Inicio de sesión:** como se muestra en la figura 48, se presenta un formulario con los campos de login y password que el usuario debe ingresar para acceder al módulo principal del panel administrativo. En caso de presentarse inconsistencias en el inicio de sesión el usuario será notificado con el mensaje de error respectivo.

Figura 48. Formulario de inicio de sesión



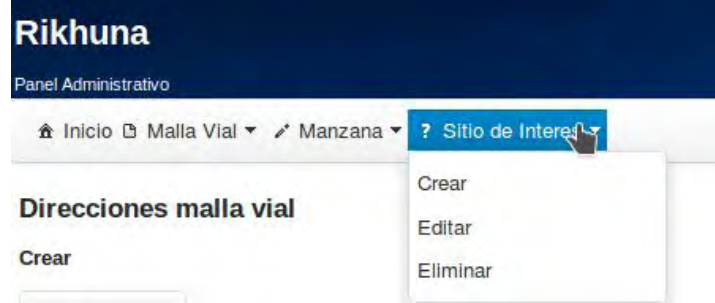
Inicio de sesión

Login *	<input type="text"/>
Password *	<input type="password"/>
	<input type="button" value="Ingresar"/>

(*) Campos requeridos

- ✓ **Vista principal:** como se muestra en la figura 49, en esta vista se presenta las funcionalidades de crear, editar y eliminar implementadas para direcciones de malla vial, manzanas y sitios de interés.

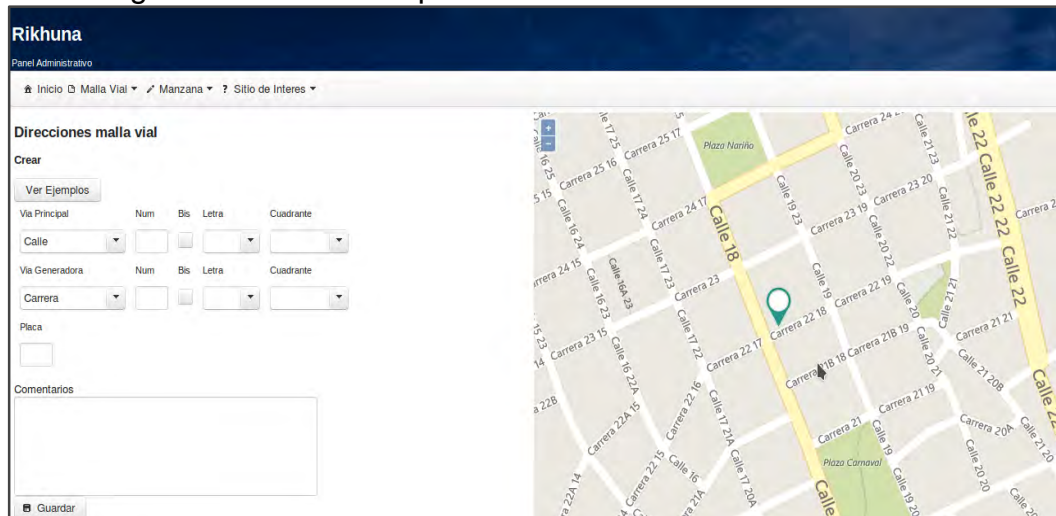
Figura 49. Vista principal panel administrativo



A continuación, se describe las vistas para cada caso:

- **Crear malla vial:** como se muestra en la figura 50, se compone de dos secciones, en la primera se muestra el formulario de ingreso de datos y en la segunda se muestra la cartografía del municipio de Pasto. En el formulario obligatoriamente el usuario debe ingresar la vía principal, vía generadora y la placa para definir una nueva dirección, opcionalmente puede definir la letra, los sufijos y/o cuadrantes asociados a la dirección y de ser necesario dejar un comentario. En la cartografía del municipio se encuentra un marcador el cual puede ser desplazado hasta ubicarlo en la posición que el usuario desee.

Figura 50. Formulario para la creación de dirección malla vial

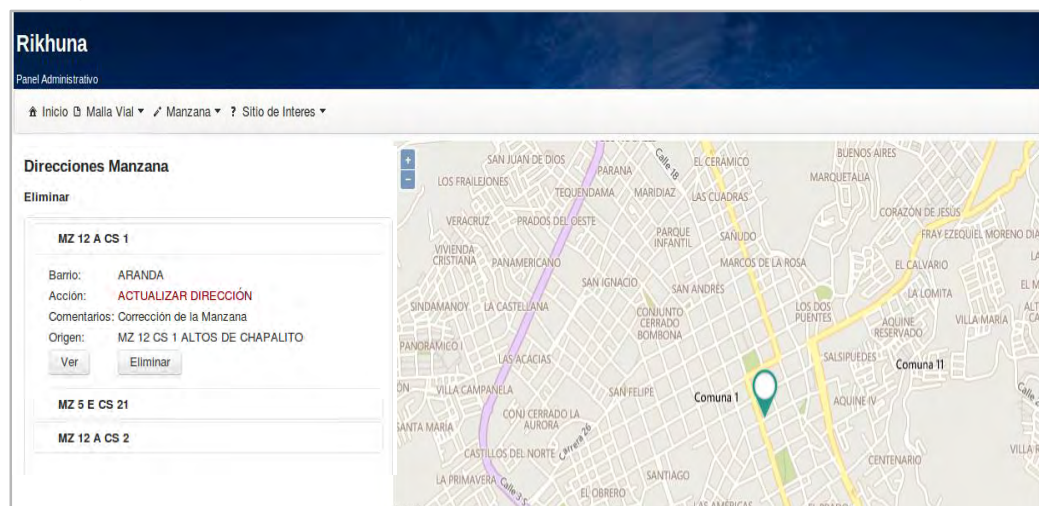


- **Editar malla vial:** se compone de dos secciones, una con un campo de texto y otra con la cartografía del municipio. En la primera sección se buscará la dirección que el usuario desea editar, en caso de no encontrar registros en la base de datos se procede a informar al usuario de este hecho, caso contrario se despliega la información básica como barrio y comuna a la cual

pertenece, así como también se le mostrará los campos que el usuario puede editar para redefinir la dirección.

- **Eliminar malla vial:** se compone de dos secciones, una con una lista de direcciones registradas por el usuario y en la otra la cartografía del municipio. En la primera sección cada dirección está acompañada del barrio al cual pertenece, la acción que se pretende hacer con ella (crear o editar), los comentarios y la dirección de origen en caso que la acción sea editar.
- **Crear manzana:** se compone de dos secciones, en la primera se muestra el formulario de ingreso de datos y en la segunda se muestra la cartografía del municipio. En el formulario obligatoriamente el usuario debe ingresar el nombre de la manzana y el número de la casa, opcionalmente y de ser necesario puede definir un comentario. En la cartografía del municipio se encuentra un marcador el cual puede ser desplazado hasta ubicarlo en la posición que el usuario desee.
- **Editar manzana:** se compone de dos secciones, una con un campo de texto y otra con la cartografía del municipio. En la primera sección se buscará la dirección que el usuario desea editar, para lo cual se solicita los campos de manzana y barrio, en caso de no encontrar resultados se procede a informar al usuario, caso contrario se despliega la información básica de la dirección encontrada, así como también se le mostrará los campos que el usuario puede editar para redefinir la dirección.

Figura 51. Formulario para la eliminación de dirección manzana



- **Eliminar manzana:** como se muestra en la figura 51, se compone de dos secciones, una con una lista de direcciones de manzana registradas por el usuario y en la otra la cartografía del municipio. En la primera sección cada

dirección basada en manzana está acompañada del barrio al cual pertenece, la acción que se pretende hacer con ella (crear o editar), los comentarios y la dirección de origen en caso que la acción sea editar.

- **Crear sitio de interés:** se compone de dos secciones, la primera de ellas muestra opciones de selección para sitio de interés urbano y sitio de interés rural, dependiendo de la opción se despliega el formulario de ingreso de datos y en la segunda sección se muestra la cartografía del municipio. En el formulario obligatoriamente el usuario debe ingresar el nombre del sitio de interés y la categoría, opcionalmente se puede definir una dirección, teléfono, sitio web y de ser necesario un comentario. En la cartografía del municipio se encuentra un marcador el cual puede ser desplazado hasta ubicarlo en la ubicación que el usuario desee.
- **Editar sitio de interés:** como se muestra en la figura 52, se compone de dos secciones, una con un campo de texto y otra con la cartografía del municipio. En la primera sección se buscará el sitio de interés se desea editar, en caso de no encontrar resultados se procede a informar al usuario, caso contrario se despliega la información básica del sitio encontrado y los campos que el usuario puede editar para redefinir el sitio.

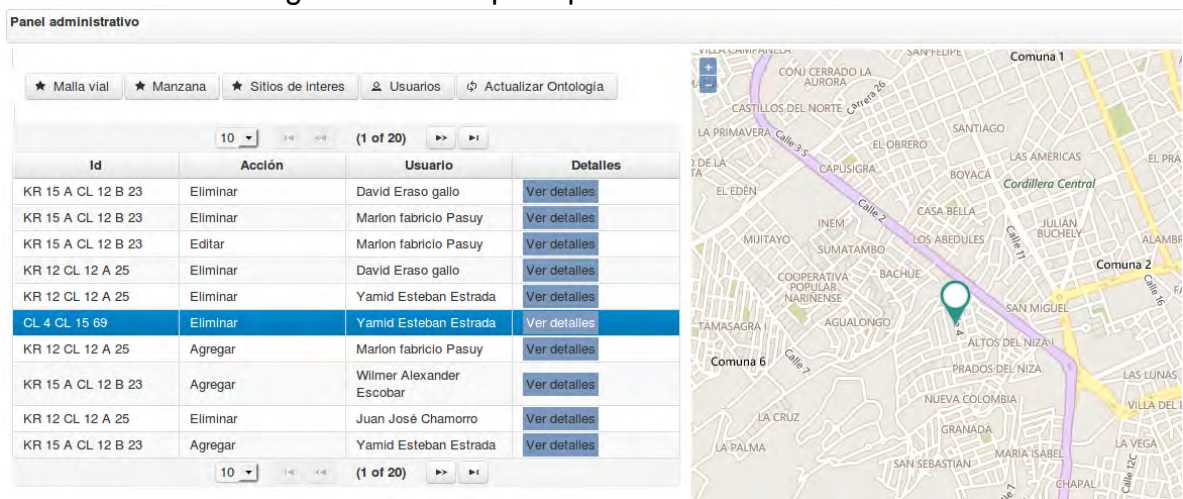
Figura 52. Formulario para la edición de sitio de interés

The image shows a screenshot of the Rikhuna administrative interface. At the top, there is a navigation bar with the text 'Rikhuna' and 'Panel Administrativo'. Below this, there is a breadcrumb trail: 'Inicio > Malla Vial > Manzana > ? Sitio de Interes'. The main content area is divided into two sections. On the left, there is a search form with the text 'universidad de nariño' in the search box. Below the search box, there are three search results, each with a 'Mostrar' button and an 'Editar' button. The first result is 'OBSERVATORIO ASTRONOMICOMICO UNIVERSIDAD DE NARIÑO' with the category 'UNIVERSIDADES'. The second result is 'UNIVERSIDAD ANTONIO NARIÑO' with the category 'UNIVERSIDADES'. The third result is 'UNIVERSIDAD AUTONOMA DE NARIÑO' with the category 'UNIVERSIDADES'. On the right, there is a map of the municipality of Nariño, showing various comunas and streets. A green location pin is placed on the map, indicating the location of the selected site of interest in Comuna 1. The map also shows the 'Cordillera Central' and various streets like 'Calle 2' and 'Calle 22'.

- **Eliminar sitio de interés:** se compone de dos secciones, una con una lista de sitios de interés registrados por el usuario y en la otra la cartografía del municipio. En la primera sección cada sitio está acompañado de información relevante, la acción que se pretende hacer con ella (crear o editar), los comentarios y el sitio de origen en caso que la acción sea editar.

- ✓ **Vista del administrador del sistema:** como se muestra en la figura 53, esta vista es única para el usuario administrador del sistema y permite gestionar las acciones que desean hacer los demás usuarios. Tanto para malla vial, direcciones urbanas y sitios de interés podrá visualizar dos secciones, en la primera de ellas aparece la lista con la información que los usuarios registrados han proporcionado, mostrando la dirección o sitio de interés, la acción que se pretende realizar (crear, editar o eliminar), el usuario que registro la información, una opción donde se muestra mayor detalle los registros almacenados y las acciones que el administrador desee realizar. En la segunda sección se muestra la cartografía del municipio de Pasto donde se muestra la ubicación del registro que se seleccione.

Figura 53. Vista principal usuario administrador



También se destacan las opciones para la gestión de usuarios y de la ontología que se describen a continuación.

- **Usuarios:** permite al administrador obtener una lista con todos los usuarios registrados en el sistema y realizar gestión sobre estos. De los usuarios se carga información básica como nombres, login, correo, dirección, teléfono, fecha de nacimiento y si este usuario se encuentra validado o no.
- **Actualizar ontología:** permite al administrador tener un control sobre la ontología del sistema y realizar cambios sobre esta cada vez que se lleve a cabo un nuevo registro en el esquema principal de la base de datos.

5.6. APLICACIÓN Y ANÁLISIS DE PRUEBAS

En esta etapa se describen las pruebas aplicadas al visor cartográfico Rikhuna con el objetivo de medir el grado de eficiencia en las búsquedas y la calidad de los datos recolectados. Se definieron 4 tipos de casos de prueba con 10 iteraciones cada uno,

los cuales son calificados como Correcto o Incorrecto, según el resultado obtenido en el proceso. Además de pruebas unitarias para comprobar el correcto funcionamiento del visor cartográfico y el panel administrativo.

5.6.1.Casos de prueba

- ✓ **Caso de prueba 1 - Estandarización de direcciones malla vial:** el objetivo de esta prueba es verificar el funcionamiento del proceso de estandarización para direcciones basadas en malla vial haciendo uso de la ontología. Los resultados se muestran en la tabla 21.

Tabla 21. CP-001 Estandarización de direcciones malla vial

No	Dirección ingresada	Resultado	Calificación
1	Carrera 25 # 17-62	KR 25 CL 17 - 62	Correcto
2	cl 19 27 54	CL 19 KR 27 - 54	Correcto
3	cr 3 N 21B-102 B/ STA BARBARA	KR 3 CL 21 B - 102	Correcto
4	centro calle 19 23-47	CL 19 KR 23 - 47	Correcto
5	4 15-69 caicedo	CL 4 KR 15 - 69	Correcto
6	kll 16 18 15	CL 16 KR 18 - 15	Correcto
7	Cra 40 A numero 18A 61 AV PANA	KR 40 A CL 18 A - 61	Correcto
8	Cra. 32 21A-30 – Los Estudiantes	KR 32 CL 21 A - 30	Correcto
9	dg 16d kr 7 este 4	DG 16 D KR 7 ESTE - 4	Correcto
10	transversal 39 b # 7 -10	TV 39 B CL 7 - 10	Correcto

- ✓ **Caso de prueba 2 - Estandarización de direcciones manzana:** El objetivo de esta prueba es verificar los resultados del proceso de estandarización de direcciones basadas en manzana haciendo uso de la ontología. Los resultados pueden observarse en la tabla 22.

Tabla 22. CP-002 Estandarización de direcciones manzana

No	Dirección ingresada	Barrio ingresado	Resultado	Calificación
1	manz g cs 24	MADRIGAL	MZ G CS 24 MADRIGAL	Correcto
2	mz b casa 5	JUAN PABLO II	MZ B CS 5 JUAN PABLO II	Correcto
3	MANZANA D C 29	RINCON DE PASTO	MZ D CS 29 EL RINCÓN DE PASTO	Correcto
4	man 2 c 12	AGUALONGO	MZ 2 CS 12 AGUALONGO	Correcto
5	m d casa 8	FUTURO	MZ D CS 8 EL FUTURO	Correcto
6	manz 11b casa 6	CORAZON DE JESUS	MZ 11 B CS 6 CORAZÓN DE JESUS	Correcto

7	mza 1 a cas 1	BELEN	MZ 1 A CS 1 BELÉN	Correcto
8	maz 35 cs 12 a	TAMASAGRA I	MZ 35 CS 12 A TAMASAGRA I	Correcto
9	mz a casa 13b	VILLAFLOR I	MZ A CS 13 B VILLAFLOR I	Correcto
10	manzana c2 casa 3a	BELEN	MZ C 2 CS 3 A BELEN	Correcto

- ✓ **Caso de prueba 3 - Búsqueda y localización de sitios de interés:** el objetivo de esta prueba es verificar los resultados del buscador semántico implementado en el visor cartográfico con ayuda de la ontología. Se ingresan distintas búsquedas con errores ortográficos y sintácticos para evaluar la calidad de las búsquedas y el correcto funcionamiento del Rikhuna. Los resultados se presentan en la tabla 23.

Tabla 23. CP-003 Búsqueda y localización de sitios de interés

No.	Sitio ingresado	Resultado	Calificación
1	iglesia santiago	SANTIAGO APOSTOL, TEMPLOS	Correcto
2	hotel cahmbu	HOTEL CHAMBÚ, HOTELES	Correcto
3	futbol tejar	CANCHA DE FUTBOL BARRIO EL TEJAR, FÚTBOL	Correcto
4	iglesia miraflores	PARROQUIA BARRIO MIRAFLOREZ, TEMPLOS	Correcto
5	centro com. unico	CENTRO COMERCIAL UNICO, CENTROS COMERCIALES	Correcto
6	antonio ruena	COLISEO SERGIO ANTONIO RUANO, COLISEOS	Correcto
7	restaurante chipichapee	CHIPICHAPE, COMIDAS REGIONALES	Correcto
8	ospital san pedro	HOSPITAL SAN PEDRO, HOSPITALES	Correcto
9	colegio libertad	I.E.M LIBERTAD, INSTITUCIONES EDUCATIVAS	Correcto
10	eladeria santiago	HELADOS DE PAILA SANTIAGO, HELADERÍAS	Correcto

- ✓ **Caso de prueba 4 - Ubicación de direcciones y sitios de interés con dispositivo GPS:** el objetivo de esta prueba fue comprobar la ubicación de direcciones y sitios de interés del visor cartográfico Rikhuna. Para esto se decidió hacer uso de un dispositivo GPS Garmin GPSMAP 64s y almacenar la ubicación de diversas direcciones, tanto de malla vial y sitios de interés a lo largo del área urbana del municipio de Pasto, con el fin de contrastar los datos obtenidos con aquellos almacenados en la base de datos. Se considera como Correcto cuando la diferencia entre los valores es menor a 10 metros, de lo contrario será

Incorrecto. El cálculo de estas distancias se realiza usando funciones de PostGIS. En las tablas 24 y 25, se muestran los resultados.

Tabla 24. CP-004 Ubicación de direcciones con dispositivo GPS

No	Dirección	Ubicación Base de datos	Ubicación GPS	Diferencia (metros)	Calificación
1	CL 20 # 40 - 80	1.2308113, -77.2841937	1.2307679, -77.2841719	5,4	Correcto
2	KR 31 C # 19 A - 17	1.2217120, -77.2801853	1.2217229, -77.2802170	3,7	Correcto
3	KR 29 # 18 - 09	1.2184142, -77.2802707	1.2183909, - 77.2803290	7,0	Correcto
4	CL 17 # 26 - 42	1.2156998, -77.2801966	1.2156679, -77.28021	3,8	Correcto
5	CL 21 B # 19 - 57	1.2124439, -77.276293	1.212400, -77.2763155	5,5	Correcto
6	CL 22 A # 20 - 24	1.2113973, -77.2737432	1.2113469, -77.273812	9,5	Correcto
7	KR 16 # 20 - 15	1.2071424, -77.2737788	1.2071069, -77.273767	4,2	Correcto
8	KR 6 A # 21 - 21	1.2039294, -77.2669081	1.2039179, -77.266922	2,0	Correcto
9	CL 21 # 3 B - 05	1.2017449, -77.2641582	1.2017979, -77.2641910	6,9	Correcto
10	KR 2 A # 21 B - 07	1.2024585, -77.2624072	1.2024390, -77.2624020	2,3	Correcto

Tabla 25. CP-004 Ubicación de sitios de interés con dispositivo GPS

No	Sitio de interés	Ubicación Base de datos	Ubicación GPS	Diferencia (metros)	Calificación
1	Hotel Morasurco	1.2313722, -77.2843223	1.2313919, -77.284332	2,4	Correcto
2	Parque Infantil	1.2188998, -77.2816178	1.2188319, -77.281592	8,1	Correcto
3	Iglesia San Andrés	1.2171350, -77.2814914	1.2171379, -77.281478	1,5	Correcto
4	Catedral de Pasto	1.2156279, -77.2799089	1.2156199, -77.279926	2,1	Correcto
5	Hotel Agualongo	1.2148586, -77.2790335	1.2148289, -77.279113	9,4	Correcto
6	Iglesia Cristo Rey	1.2149602, -77.2768498	1.2149369, -77.2768199	4,2	Correcto
7	Gobernación de Nariño	1.2142338, -77.2775559	1.2142519, -77.277561	2,1	Correcto

8	Udenar Centro	1.2127244, -77.2770503	1.2127259, -77.27709	4,4	Correcto
9	Mercado los dos puentes	1.2155234, -77.2752812	1.2155790, -77.27531	7,0	Correcto
10	Colegio San Juan Bosco	1.2060063, -77.2755690	1.2059529, -77.275564	6,0	Correcto

5.6.2.Pruebas unitarias. Las pruebas unitarias se plantearon para comprobar el correcto desempeño de cada funcionalidad presente en el visor cartográfico y el panel administrativo. Cada prueba unitaria se compone de varias pruebas de aceptación que se definieron en las historias de usuario en el análisis de Scrum (Ver ANEXO A), por cada una de estas pruebas se especificaron preguntas de respuesta Si o No sobre funciones específicas que se puede realizar en el visor. En la tabla 26, se muestra la recopilación de las 16 pruebas unitarias creadas, las pruebas completas están disponibles en el ANEXO F.

Tabla 26. Recopilación de pruebas unitarias

ID PRUEBA	DESCRIPCIÓN	ESTADO
PU-001 Búsqueda de dirección de malla vial	Comprueba que el ingreso, procesamiento y visualización de datos de malla vial se realice correctamente.	APROBADA
PU-002 Búsqueda de dirección de manzana	Comprueba que el ingreso, procesamiento y visualización de datos de manzanas se realice correctamente.	APROBADA
PU-003 Búsqueda de sitios de interés	Comprueba que el ingreso, procesamiento y visualización de datos sitios de interés se realice correctamente.	APROBADA
PU-004 Búsqueda de geometrías	Comprueba que el ingreso, procesamiento y visualización de datos de barrios y corregimientos se realice correctamente.	APROBADA
PU-005 Generación de rutas	Comprueba que se generen y visualicen rutas correctamente.	APROBADA
PU-006 Ubicación del usuario	Comprueba la visualización de la posición del usuario en la cartografía de Pasto.	APROBADA
PU-007 Registro de usuarios	Comprueba datos del usuario y hace seguimiento al proceso de registro.	APROBADA
PU-008 Inicio de sesión	Hace un seguimiento en el proceso de inicio de sesión.	APROBADA
PU-009 Cerrar sesión	Cierra todos los procesos y verifica permisos.	APROBADA
PU-010 Direcciones malla vial Usuario	Comprueba operaciones de crear, editar y eliminar direcciones de malla.	APROBADA

PU-011 Direcciones manzana Usuario	Comprueba operaciones de crear, editar y eliminar direcciones de manzana.	APROBADA
PU-012 Sitios de interés Usuario	Comprueba operaciones de crear, editar y eliminar sitios de interés.	APROBADA
PU-013 Usuario administrador	Comprueba la gestión sobre usuarios y ontología del sistema.	APROBADA
PU-014 Direcciones malla vial Administrador	Comprueba la administración de direcciones de malla almacenadas por usuarios registrados.	APROBADA
PU-015 Direcciones manzana Administrador	Comprueba la administración de direcciones de manzana almacenadas por usuarios registrados.	APROBADA
PU-016 Sitios de interés Administrador	Comprueba la administración de sitios de interés almacenados por usuarios registrados.	APROBADA

6. CONCLUSIONES

- ✓ Con la culminación de este trabajo de investigación se obtiene Rikhuna: un visor cartográfico inteligente de direcciones urbanas y sitios de interés del Municipio de Pasto basado en PostGIS. Mediante la construcción de esta herramienta se cumplen los objetivos planteados y la obtención de resultados satisfactorios. Esta herramienta facilita la búsqueda y localización de direcciones urbanas, sitios de interés y divisiones territoriales en el municipio de Pasto (Colombia). El visor cartográfico cuenta además con funcionalidades complementarias como la generación de rutas entre puntos de búsqueda y la gestión de la base de datos por medio de un panel administrativo.
- ✓ En la construcción del visor cartográfico fue necesario hacer un tratamiento y gestión de la información geográfica almacenados en la base de datos para responder a las solicitudes de búsquedas realizadas por los usuarios del sistema. Por este motivo, fue de vital importancia el uso de PostGIS y las funciones para datos espaciales. Gracias al uso de esta extensión, fue posible construir la base de datos, hacer consultas sobre los datos de direcciones y sitios de interés, cálculo de distancias entre puntos, áreas de las geometrías de barrios y otras funcionalidades.
- ✓ La etapa de recolección de información geográfica fue clave en el desarrollo de la investigación. Con el uso de herramientas de software libre como Java OpenStreetMap y usando diversas fuentes de datos propias y otras públicas permitieron la construcción de una base de datos geográfica del municipio de Pasto, con información completa y actualizada de direcciones urbanas, barrios, comunas, corregimientos, veredas y sitios de interés. Todos los datos almacenados están georreferenciados de acuerdo con los estándares del Instituto Geográfico Agustín Codazzi y disponibles para el uso por parte de las distintas entidades interesadas.
- ✓ La definición de la ontología teniendo en cuenta los dominios necesarios permite que haya búsquedas inteligentes y resultados con mayor proximidad a las necesidades del usuario. La integración de una ontología al visor cartográfico permitió obtener resultados acordes a las búsquedas de direcciones urbanas y sitios de interés del usuario.

- ✓ Este trabajo permitió aplicar los conocimientos adquiridos durante toda la formación académica junto con el proceso investigativo desarrollado en el grupo GRIAS. Es importante también reconocer la importancia que tiene para los estudiantes el hecho de pertenecer a un grupo de investigación debido a la experiencia adquirida y la constante asesoría por parte de los docentes e investigadores del grupo.
- ✓ El visor cartográfico Rikhuna cuenta actualmente con las características propias de un software de calidad. En miras de realizar mejoras al sistema se plantea como trabajo futuro el desarrollo de una aplicación nativa para dispositivos móviles, de manera que se facilite la navegación del usuario por la aplicación, se aprovechen las características hardware de los equipos y el gran mercado actual de estos dispositivos.
- ✓ Durante el proceso de construcción del visor cartográfico, en el desarrollo de la etapa de recolección de información geográfica, se presentaron dificultades relacionadas con la complejidad de la división territorial del municipio y la inexistencia de normas o estándares para la georreferenciación de nuevas direcciones. Por este motivo, se dificultó la obtención y procesamiento de la totalidad de información de direcciones.
- ✓ Los resultados de este proyecto investigativo se han presentado por medio de la publicación de posters y presentación de ponencias en diversos eventos a nivel nacional e internacional, los cuales se mencionan a continuación:
 - V Congreso Internacional de Gestión Tecnológica y de la Innovación en la ciudad de Bucaramanga, Colombia.
 - IV Congreso Científico Internacional UNIANDES en la ciudad de Ambato, Ecuador.
 - III Congreso Internacional de Ingenierías en la ciudad de Tulcán, Ecuador.
 - III Congreso Andino de Computación, Informática y Educación en la ciudad de Pasto, Colombia.
 - I Encuentro Internacional de Innovación Aplicada a la Empresa en la ciudad de Pasto, Colombia.
 - Ciclo de conferencias realizadas durante la semana de ingeniería en la Universidad de Nariño, Pasto, Colombia.
 - ¡Todos podemos programar! Muestra de proyectos investigativos en la Universidad del Cauca en la ciudad de Popayán, Colombia.
 - Encuentro nacional de estudiantes de ingeniería informática y afines en la Institución Universitaria Colegio Mayor del Cauca en la ciudad de Popayán, Colombia.

7. RECOMENDACIONES

- ✓ Definir un formato único para el registro de nuevas direcciones urbanas, de manera que facilite su posterior análisis y tratamiento, que permita reducir las dificultades presentadas en el desarrollo de este trabajo, principalmente en la etapa de recolección de información geográfica, donde no fue posible recolectar y procesar la totalidad de direcciones debido a la falta de un estándar para su asignación y también por la complejidad de la división territorial de la ciudad.
- ✓ Establecer nuevos requerimientos para el visor cartográfico Rikhuna con el fin de obtener una herramienta más completa, con nuevas funcionalidades y características. Además, para contribuir al desarrollo de nuevos sistemas de este tipo, se recomienda utilizar el visor cartográfico, panel administrativo y demás productos generados por esta investigación como base para las nuevas herramientas. También es recomendable usar el modelo implementado en Rikhuna para la construcción de sistemas similares en distintos lugares.
- ✓ Aportar con el crecimiento de la base de datos de direcciones urbanas y sitios de interés, mediante la inserción de nuevos datos o actualización de los existentes, por medio de un panel de administración en el cual los usuarios pueden gestionar la base de datos de direcciones y sitios de interés, esto con el fin de que este trabajo siga creciendo y sea conocido por muchas personas e instituciones en la comunidad investigativa y demás interesados.
- ✓ Comunicar y compartir los datos recolectados y demás productos generados en esta investigación con el objetivo de contribuir con la filosofía de software libre adoptada en el grupo de investigación, de manera que haya una participación colaborativa de estudiantes, investigadores y demás usuarios interesados, buscando que este intercambio sea el paso inicial para la generación de nuevas investigaciones orientadas a solucionar problemas relacionados con el uso y análisis de datos georreferenciados.
- ✓ Fomentar la investigación desde las aulas de clases, para que los estudiantes despierten el espíritu investigativo y la capacidad de profundizar sus conocimientos y aplicar lo aprendido en el planteamiento de soluciones a los distintos problemas que se presentan en la región y el mundo, ya que en este trabajo la investigación permitió que los estudiantes asociados apropiaran y generaran nuevo conocimiento sobre un área específica de estudio.

BIBLIOGRAFÍA

- [1] ESRI, «ArcGIS 9, Geocoding Rule Base Developer Guide,» 2003. [En línea]. Available: https://socsci.flinders.edu.au/geog/sisl/ESRI_Library/Developing_with_ArcObjects/Engine_Developers_Guide.pdf.
- [2] J. González y M. L. d. L. Torres, «La geoinformación y su importancia para las tecnologías de la información geográfica,» Revista electrónica de recursos en internet sobre geografía y ciencias sociales, N° 148, 1 junio 2011. [En línea]. Available: <http://www.ub.edu/geocrit/aracne/aracne-148.htm>.
- [3] O. Huisman y R. A. d. By, Principles of Geographic Information Systems: An Introductory Textbook, International Institute for Geo-Information Science and Earth Observation (ITC), 2009.
- [4] R. T. Pereira, A. Barón, G. Hernández, A. Hidalgo y C. Betancourth, «SIGODEP: Un primer paso para la Detección de Patrones Delictivos con Técnicas de Minería de Datos,» Memorias de IX Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC 2012), Pontificia Universidad Católica del Perú, Lima, Perú, 2012.
- [5] O. Cabrera, J. Guerrero, M. Benavides y R. T. Pereira, «Un acercamiento a la construcción de ontologías con la herramienta libre Protégé,» Revista Ventana Informática, Universidad de Manizales, 2014. [En línea]. Available: <http://revistasum.umanizales.edu.co/ojs/index.php/ventanainformatica/article/viewFile/278/418>.
- [6] IGAC, «Geoportal,» [En línea]. Available: <http://geoportal.igac.gov.co/ssigl2.0/visor/galeria.req?mapald=23&title=Catastro%20Nacional>.
- [7] M. Á. B. Ordóñez, «Análisis de la tecnología javaserver faces, como framework de desarrollo de aplicaciones web, caso práctico: procesos de comercialización de combustibles de EP-Petroecuador,» 2011. [En línea]. Available: <http://dspace.epoch.edu.ec/bitstream/123456789/1919/1/18T00488.pdf>.
- [8] P. A. Colpas, M. P. Melo y W. N. Bernal, «Buscador Ontológico Web basado en Agentes Inteligentes,» 9th Latin American and Caribbean Conference for Engineering and Technology, 2011. [En línea]. Available:

http://www.laccei.org/LACCEI2011-Medellin/RefereedPapers/IT313_Ariza.pdf.

- [9] S. Aranoff, *Geographic Information Systems: A Management Perspective*, WDL Publications, Ottawa, 1989.
- [10] P. A. Burrough y R. A. Mcdonell, *Principles of Geographical Information Systems*, Oxford University Press, Oxford, 1998.
- [11] M. F. Goodchild, P. Longley, D. W. Rhind y D. J. Maguire, *Geographic Information Systems and Science*, John Wiley & Sons, 2005.
- [12] M. A. Bernabé y C. M. López, *Fundamentos de las infraestructuras de datos espaciales (IDE)*, UPM Press, 2012.
- [13] E. Raisz, *Cartografía general*, 7 ed. OMEGA, 2005.
- [14] ArcGIS, «¿Qué es la geocodificación?,» 2016. [En línea]. Available: <http://desktop.arcgis.com/es/arcmap/10.3/guide-books/geocoding/what-is-geocoding.htm>.
- [15] ESRI, *ArcGIS 9, Geocoding Rule Base Developer Guide*, 2003.
- [16] A. Opiel y R. Sheldon, *Fundamentos de SQL. Tercera Edición.*, McGraw Hill, México, 2010.
- [17] O. d. I. N. Unidas, *Manual de infraestructura geoespacial en apoyo de actividades censales*, Departamento de Asuntos Económicos y Sociales de las Naciones Unidas, Nueva York, 2010.
- [18] C. O. Galán y R. M. Alegría, *Sistemas de Información Geográfica, aplicaciones prácticas con Idrisi32 al análisis de riesgos naturales y problemáticas medioambientales*, RA-MA Editorial, España, 2003.
- [19] PostgreSQL, «About,» [En línea]. Available: <https://www.postgresql.org/about/>.
- [20] PostGIS, «What is PostGIS?,» [En línea]. Available: <http://postgis.refractions.net/>.
- [21] R. O. Obe y L. S. Hsu, *PostGIS in Action - Second Edition*, Manning Publications, 2015.

- [22] W3C, «Guía breve de web semántica,» [En línea]. Available: <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>.
- [23] M. Horridge, «A Practical Guide To Building OWL Ontologies,» The University Of Manchester, 2011. [En línea]. Available: http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf.
- [24] R. Studer, R. Benjamins y D. Fensel, «Knowledge engineering: Principles and methods,» [En línea]. Available: <http://www.sciencedirect.com/science/article/pii/S0169023X97000566>.
- [25] G. v. Heijst, A. T. Schreiber y B. J. Wielinga, «Using explicit ontologies in KBS development,» International Journal of Human-Computer Studies, 1997. [En línea]. Available: <http://www.cs.vu.nl/~guus/papers/Heijst97a.pdf>.
- [26] W3C, «Lenguaje de Ontologías Web (OWL) Vista General,» 2004. [En línea]. Available: <https://www.w3.org/2007/09/OWL-Overview-es.html>.
- [27] W3C, «SPARQL 1.1 Query Language,» 2013. [En línea]. Available: <https://www.w3.org/TR/sparql11-query/>.
- [28] M. S. Morales, Manual de Desarrollo Web basado en ejercicios y supuestos prácticos, Lulu Enterprises Inc., 2012.
- [29] IGAC, «Manual de reconocimiento predial,» 1997. [En línea]. Available: http://www2.igac.gov.co/igac_web/UserFiles/File/Catastro/manualreconocimiento.pdf.
- [30] A. Menzinsky, G. López y J. Palacio, Scrum Manager v 2.6, Llubaris Info 4 Media SL, 2016.
- [31] K. S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison-Wesley, 2013.
- [32] JOSM, «Java OpenStreetMap Editor,» [En línea]. Available: <http://wiki.openstreetmap.org/wiki/ES:JOSM>.
- [33] P. J. Ferrer, I. Sanchez y S. Tramoyeres, «Qué son OSM y JOSM,» [En línea]. Available: http://taller-de-josm-imposm-tilemill.readthedocs.io/es/latest/osm/osm_intro.html.

- [34] OSGeo-Live, «Guía de inicio rápido de OpenStreetMap,» [En línea]. Available: https://live.osgeo.org/es/quickstart/osm_quickstart.html.
- [35] S. University, «Protégé,» Center for Biomedical Informatics Research at the Stanford University School of Medicine, [En línea]. Available: <https://protege.stanford.edu/>.
- [36] A. S. Foundation, «What is Jena?,» [En línea]. Available: https://jena.apache.org/about_jena/about.html.
- [37] E. Burns y C. Schalk, JavaServer Faces 2.0, MC Graw Hill, 2010.
- [38] E. Burns y R. Kitain, JavaServer Faces Specification Version 2.0, Sun Microsystems, Inc, 2009.
- [39] J. C. Process, «JSF in a nutshell,» [En línea]. Available: <http://www.java-serverfaces.org>.
- [40] MADEJA, «Mapserver y Geoserver,» 2009. [En línea]. Available: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/5-0>.
- [41] M. García, O. Fonts y V. González, «Geoserver, más allá de un servidor WMS,» 2013. [En línea]. Available: http://www.sigte.udg.edu/jornadassiglibre2013/uploads/articulos_13/a27.pdf.
- [42] W. Szczerban, «GeoServer,» 2017. [En línea]. Available: https://bolollo.github.io/SDIS_mtig2017/geoserver/geoserver.html.
- [43] R. T. Pereira, G. Hernández y N. Q. Taimbud, «Geocodificador de eventos delictivos georreferenciados a nivel de direcciones urbanas en el municipio de pasto,» 2016. [En línea]. Available: [http://cogestec.ingenio.com.co/db/separated/2016%20\(44\).pdf](http://cogestec.ingenio.com.co/db/separated/2016%20(44).pdf).
- [44] M. F. P. Espinal, J. J. Chamorro y R. T. Pereira, «Una aplicación inteligente para dispositivos móviles del sistema estratégico de transporte público del municipio de pasto,» 2016. [En línea]. Available: [http://cogestec.ingenio.com.co/db/separated/2016%20\(54\).pdf](http://cogestec.ingenio.com.co/db/separated/2016%20(54).pdf).
- [45] Google, «Google Street View,» [En línea]. Available: <https://www.google.com.co/maps>.
- [46] JOSM, «JOSM/Complementos/AddrInterpolation,» [En línea]. Available: <http://wiki.openstreetmap.org/wiki/ES:JOSM/Complementos/AddrInterpolation>

- [47] N. Mollet, «Map Icons Collection,» [En línea]. Available: <https://mapicons.mapsmarker.com/>.
- [48] JOSM, «Plugins for JOSM,» [En línea]. Available: <https://josm.openstreetmap.de/wiki/Plugins>.
- [49] JOSM, «JOSM/Plugins/ImportImagePlugin,» [En línea]. Available: <https://wiki.openstreetmap.org/wiki/JOSM/Plugins/ImportImagePlugin>.
- [50] Mapzen, «Mapzen Turn-by-Turn routing service,» [En línea]. Available: <https://mapzen.com/documentation/mobility/turn-by-turn/api-reference/>.