

UNIVERSIDAD DE CUENCA
FACULTAD DE INGENIERÍA
ESCUELA DE INFORMÁTICA



**“DISEÑO E IMPLEMENTACIÓN DE UN
REPOSITORIO ECUATORIANO DE DATOS
ENLAZADOS GEOESPACIALES”**

Tesis de Grado previa a la obtención del Título:
Ingeniero de Sistemas

Autores:

John Fernando Baculima Cumbe
Marcelo Xavier Cajamarca Bermeo

Director de Tesis:

Ing. Víctor Hugo Saquicela Galarza

CUENCA - ECUADOR
Junio de 2014



RESUMEN

Este trabajo presenta los procesos y actividades realizadas para la generación, publicación y visualización de Datos Enlazados Geoespaciales (Geo Linked Data) del Ecuador ya que el avance de la Web (tanto en Europa como en Norteamérica) se enfoca en la publicación de este tipo de datos, permitiendo que estén estructurados de tal manera que se puedan interconectar entre y desde diferentes fuentes. Por lo tanto, para generar estos Datos Enlazados Geoespaciales, se procedió a la ejecución de cada uno de los pasos de la metodología de publicación de Datos Enlazados. Sin embargo, existieron problemas que dificultaron esta actividad, ya que es necesario el conocimiento de tecnologías semánticas que permitan llevar a cabo la generación de este tipo de información. Entonces, este trabajo describe una solución que permita la fácil generación de Datos Enlazados Geoespaciales, para lo cual, con la ayuda de buscadores Web, se inició con la recolección manual de datos con su respectiva información Geoespacial, después se procedió a desarrollar herramientas que faciliten la generación de Datos Enlazados Geoespaciales y a modificar otras previamente desarrolladas con el fin de dar compatibilidad al consumo del nuevo tipo de información y, finalmente, con la utilización de estas herramientas dentro de la metodología de publicación, realizar la implementación del prototipo de repositorio Ecuatoriano de Datos Enlazados Geoespaciales.

Palabras Clave: Datos Enlazados (Linked Data), Datos Enlazados Geoespaciales (Geo Linked Data), **RDF**, **SPARQL**, **GEOSPARQL**, Web Semántica, Geoespacial.



ABSTRACT

This work presents the processes and activities done for generation, publication and visualization of Geospatial Linked Data of Ecuador, considering that the Web advance (in both, Europe and North America) is focused on publication of this data types and let it to be structured and interconnected, to and from, different sources. Therefore, to generate Geospatial Linked Data, proceeded with the execution of every step of Linked Data's publication methodology. However, there were some troubles for this activity since semantic technologies knowledge is required to implement the generation process. Then, this work describes one solution to allow an easy Geospatial Linked Data generation, for which with a web search, began a manual recollection of data that contains Geospatial information; after, proceeded with the development of new tools and the modification of the old ones for helping the Geospatial Linked Data generation and adding compatibility of new format of information; and finally, with the use of the developed and modified tools, implement and publish the prototype of the Ecuadorian's Geospatial Linked Data repository.

Keywords: Linked Data, Geospatial Linked Data, **RDF**, **SPARQL**, **GEOSPARQL**, Semantic Web, Geospatial.



Índice general

DEDICATORIA	17
AGRADECIMIENTOS	18
1. INTRODUCCIÓN	19
1.1. IDENTIFICACIÓN DEL PROBLEMA	19
1.2. JUSTIFICACIÓN	20
1.3. ALCANCE	21
1.4. OBJETIVO GENERAL	21
1.5. OBJETIVOS ESPECÍFICOS	22
2. MARCO TEORICO	23
2.1. WEB SEMÁNTICA	23
2.1.1. DEFINICIÓN DE WEB SEMÁNTICA	24
2.1.2. RDF y RDFS	25
2.1.3. ONTOLOGÍA	26
2.1.4. SPARQL y GEOSPARQL	27
2.1.5. DATOS ENLAZADOS Y DATOS ENLAZADOS GEOESPA- CIALES	29
2.2. METODOLOGÍA PARA LA GENERACIÓN DE DATOS ENLAZA- DOS GEOESPACIALES	30
2.2.1. ESPECIFICACIÓN	31
2.2.2. MODELADO	31
2.2.3. GENERACIÓN DE RDF	31
2.2.4. GENERACIÓN DE LINKS	32



2.2.5. PUBLICACIÓN	33
2.2.6. EXPLOTACIÓN	34
2.3. HERRAMIENTAS DE DESARROLLO Y SOPORTE	34
2.3.1. API GEOTOOLS	34
2.3.2. GEOKETTLE	35
2.3.3. JENA	35
2.4. FORMATO DE LAS FUENTES DE INFORMACIÓN	36
2.4.1. SHAPEFILE	36
3. ANÁLISIS Y EXTENSIÓN DE LAS HERRAMIENTAS DE GENERACION Y CONSUMO DE DATOS ENLAZADOS GEOESPACIALES	39
3.1. INTRODUCCIÓN A GEOMETRY2RDF	39
3.2. ANÁLISIS DE FUNCIONALIDAD ACTUAL DE GEOMETRY2RDF	41
3.2.1. Limitaciones	41
3.3. ANÁLISIS DE FUNCIONALIDAD A CORREGIR Y EXTENDER A GEOMETRY2RDF	42
3.3.1. Identificación de errores a corregir	42
3.3.2. Descripción de las nuevas funcionalidades para Geometry2RDF	43
3.4. DESARROLLO Y VALIDACIÓN DE FUNCIONALIDAD A CORREGIR Y EXTENDER DE GEOMETRY2RDF	44
3.4.1. Corrección de errores identificados en Geometry2RDF	44
3.4.2. Validación de errores corregidos	44
3.4.3. Implementación de nuevas funcionalidades requeridas en Geometry2RDF	45
3.4.4. Validación de nuevas funcionalidades implementadas	45
3.5. SELECCIÓN DE LA HERRAMIENTA ETL A USAR PARA LA INTEGRACIÓN	48
3.5.1. Análisis de Kettle	48
3.5.2. Análisis de GeoKettle	49
3.6. DISEÑO DEL PLUGIN	49
3.6.1. Esquema funcional del plugin	50
3.6.2. Esquema de relación entre la herramienta de ETL y el plugin .	50



3.7.	DESARROLLO, IMPLEMENTACIÓN Y PRUEBAS DEL PLUGIN	51
3.7.1.	Integración del Plugin con la herramienta de ETL	51
3.7.2.	Pruebas	51
3.7.3.	Resultados	52
3.8.	INTRODUCCIÓN A MAP4RDF	53
3.9.	ANÁLISIS DE FUNCIONALIDAD PREVIO A CAMBIOS DE MAP4RDF	54
3.9.1.	Limitaciones	56
3.10.	ANÁLISIS DE FUNCIONALIDAD A CORREGIR Y EXTENDER A MAP4RDF	57
3.10.1.	Identificación de errores a corregir	57
3.10.2.	Descripción de la o las nuevas funcionalidades requeridas . . .	57
3.11.	DESARROLLO Y VALIDACIÓN DE FUNCIONALIDAD A CO- RREGIR Y EXTENDER DE MAP4RDF	58
3.11.1.	Corrección de errores identificados en MAP4RDF	58
3.11.2.	Validación de errores corregidos	58
3.11.3.	Implementación de nuevas funcionalidades requeridas en MAP4RDF	58
3.11.4.	Validación de nuevas funcionalidades implementadas	60
4.	IMPLEMENTACIÓN DE LA METODOLOGÍA PARA LA GENE- RACIÓN DE DATOS ENLAZADOS GEOESPACIALES ECUA- TORIANOS	61
4.1.	METODOLOGÍA PARA LA PUBLICACIÓN DE DATOS ENLAZA- DOS	61
4.1.1.	ESPECIFICACIÓN	62
4.1.2.	MODELAMIENTO	62
4.1.3.	GENERACIÓN DE RDF	63
4.1.4.	PUBLICACIÓN	63
4.1.5.	EXPLOTACIÓN	63
4.2.	METODOLOGÍA PARA LA PUBLICACIÓN DE DATOS ENLAZA- DOS GEOESPACIALES DEL ECUADOR	63
4.2.1.	ESPECIFICACIÓN	64
4.2.2.	MODELAMIENTO	66
4.2.3.	GENERACIÓN DE RDF	68



4.2.4. ENLACE DE DATOS	75
4.2.5. PUBLICACION	79
4.2.6. EXPLOTACION	81
4.3. GENERACIÓN DE RELACIONES GEOESPACIALES	91
5. RESULTADOS	94
5.1. ANÁLISIS DE LAS FUENTES DE INFORMACIÓN	94
5.2. GENERACIÓN DE RDF	96
5.3. MAP4RDF	97
5.4. ENLACE DE DATOS	98
5.5. REPOSITORIO WEB	99
6. CONCLUSIONES	102
A. MODIFICACIONES Y CONFIGURACIONES DE MAP4RDF	104
B. INSTALACIÓN DEL SERVIDOR PARLIAMENT	111
C. FUENTES DE INFORMACIÓN	114
D. CONFIGURACIÓN PARA PUBLICACIÓN Y CONSUMO DE DATOS ENLAZADOS GEOESPACIALES	115
E. CREACIÓN E INSTALACIÓN DE UN PLUGIN EN GEOKETTLE	117
E.0.1. Creación de un plugin para GeoKettle	117
E.0.2. Instalación del plugin Geometry2RDF en GeoKettle	124
Glosario	127
Bibliografía	130



Índice de figuras

2.1. Evolución de la Web Semantica. Fuente: [14]	24
2.2. Ejemplo de Tripletas	25
2.3. Ejemplo de consulta SPARQL	28
2.4. Visualización Geográfica de las Provincias del Ecuador en QGIS	38
2.5. Tabla de información de un archivo shapefile.	38
3.6. Generación actual (izquierda) y anterior (derecha) de un polígono	44
3.7. Función actual (izquierda) y función obsoleta (derecha)	45
3.8. Geometry2RDF con formato Maven	46
3.9. GEOSPARQL del ejemplo de la especificación del estándar	47
3.10. GEOSPARQL generado manualmente mediante código Java	47
3.11. Generación de un punto antes (derecha) y ahora (izquierda)	48
3.12. Esquema general del plugin	50
3.13. Esquema para la utilización del plugin	50
3.14. Configuración de la ruta del shapefile	52
3.15. Configuración del Sistema de Coordenadas	52
3.16. Configuración RDF para una prueba	53
3.17. Parte del RDF generado de provincias	54
3.18. Parte del RDF con los metadatos del shapefile de provincias	55
3.19. Arquitectura general de MAP4RDF	56
3.20. Esquema general de la funcionalidad de MAP4RDF	56
3.21. Gráfico de los Aereopuertos del Ecuador aplicando filtro	60
4.22. Proceso de Publicación de Linked Data. Fuente [16]	62
4.23. Metodología extendida para la generación de Datos Enlazados Geoes- paciales	64



4.24. Ilustración de la Información GeoEspacial del Ecuador en los Archivos Shapefile	65
4.25. Ontología GeoSPARQL vista desde Protege	67
4.26. Modelo de ejemplo, aplicando la ontología GEOSPARQL	68
4.27. Ontología usada para los metadatos, es parte del ISO-19115	69
4.28. Configuración de la Transformación para Generar RDF	70
4.29. Especificación de la ruta del archivo Shapefile	70
4.30. Uso y configuración del componente “Shape File Input”	71
4.31. Uso y configuración del componente “Get Variables”	72
4.32. Uso y configuración del componente “SRS Transformation”	72
4.33. Uso y configuración del componente “Geometry2RDF”	73
4.34. Uso y configuración del componente “Filter Rows”	74
4.35. Configuración de archivo de salida rdf de Datos	75
4.36. Configuración de archivo de salida rdf de MetaDatos	75
4.37. Silk. Configuración de Fuentes de Datos	76
4.38. Silk. Configuración de condiciones de enlazado	77
4.39. Silk. Resultados de Enlaces entre fuentes locales	78
4.40. Silk. Resultados de Enlaces con la DBPEDIA	78
4.41. Carga de archivo RDF al servidor Parliament	79
4.42. Carga de archivo RDF al servidor Parliament	80
4.43. Explotación de datos cargados al servidor Parliament	81
4.44. Ejemplo de Consulta SPARQL en el Triple Store Parliament	82
4.45. Tabla de resultado de consulta	83
4.46. Relación Visual entre consulta SPARQL y Resultado	84
4.47. Visualización de Información Geométrica en formato RDF (Cantones del Ecuador)	85
4.48. Poblaciones que están dentro de la provincia del Azuay.	86
4.49. Ríos que intersectan la Provincia del Guayas.	87
4.50. Centros Educativos a 500 metros a la redonda de la ciudad de Cuenca.	88
4.51. Cantones que rodean al cantón Quito	89
4.52. Centrales Eléctricas ubicadas en la región SIERRA	90
4.53. Relaciones Espaciales - Fuente: [12], Página 18.	91



4.54. Funciones GeoSPARQL - Fuente: [12], Página 29.	92
4.55. Consulta GeoSPARQL - Intersección: Ríos y la Provincias del Guayas	93
4.56. Visualización GeoSPARQL - Intersección: Ríos y la Provincias del Guayas	93
5.57. Total Archivos RDF Generados	95
5.58. Total de Archivos RDF Generados y No Generados	95
5.59. Shapefile que sufrieron cambio de sistema de coordenadas	96
5.60. Archivos Shapefile con información similar	98
5.61. Resultados de Generación de Enlaces	98
5.62. Resultados de Generación de Enlaces con la DBPEDIA	99
5.63. Repositorio Geo Linked Data Ecuador	100
5.64. Geo Linked Data Ecuador	101
A.1. Error que impedía filtrar los resultados	104
A.2. Error de filtrado de resultados corregido. Filtrado de las provincias del Ecuador	105
A.3. Limite de resultados máximo de consulta corregido, ahora disponible en archivo de configuración	105
A.4. Modificación de función que construye la cadena de consulta	106
A.5. Modificación de función que recupera la propiedad “Figure” de el re- sultado de consulta	106
A.6. Determinación del tipo de Geometría según el WKT recuperado	107
A.7. Envío de los recursos a buffer para graficarlos según el tipo de Geo- metría del WKT	107
A.8. Gráfico de Polígonos	108
A.9. Gráfico de Puntos	108
A.10. Gráfico de Lineas	109
A.11. Funciones para el gráfico de los recursos en OpenLayers con el formato WKT	109
A.12. Construcción del texto de la consulta GeoSPARQL	110
A.13. Gráfico de la Consulta GeoSPARQL	110
B.1. Archivos y carpetas descomprimidas de Parliament	111



B.2. Comando para ejecución de Parliamet	112
B.3. Crear Indices Espaciales para soporte de consultas GEOSPARQL	112
B.4. Servidor listo para consultas Geoespaciales	113
C.1. Fuentes de Información	114
D.1. Aplicación JBOSS sobre el cual se ejecuta MAP4RDF	116
E.1. Estructura del proyecto en Eclipse	119
E.2. Librerías de GeoKettle que deben ser agregadas a Eclipse	119
E.3. Contenido del archivo plugin.xml	121
E.4. Configuración de depuración en Eclipse	123
E.5. Opción para generar el archivo JAR	123
E.6. Configuración para generar el archivo JAR	124
E.7. Directorio donde se debe copiar el plugin Geometry2RDF	125
E.8. Modelo de transformación usando el plugin Geometry2RDF	126



Índice de cuadros

2.1. Clases del Vocabulario RDF	26
2.2. Propiedades del vocabulario RDF	26
3.3. Funcionalidades aplicadas en MAP4RDF	59
3.4. Validaciones de Nuevas Funcionalidades MAP4RDF	60
4.5. URIS para identificación de recursos	65
4.6. URIS para identificación de los recursos asociados a los datos	66
4.7. URIS para representar la información geométrica	66
D.1. Características de Equipo donde se realizo las consultas de prueba	115



Yo, *JOHN FERNANDO BACULIMA CUMBE*, autor de la tesis “DISEÑO E IMPLEMENTACIÓN DE UN REPOSITORIO ECUATORIANO DE DATOS ENLAZADOS GEOESPACIALES”, reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de INGENIERO DE SISTEMAS. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, 20 de junio del 2014

JOHN FERNANDO BACULIMA CUMBE

C.I: 0105669683



Yo, JOHN FERNANDO BACULIMA CUMBE, autor de la tesis “DISEÑO E IMPLEMENTACIÓN DE UN REPOSITORIO ECUATORIANO DE DATOS ENLAZADOS GEOESPACIALES”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 20 de junio del 2014

JOHN FERNANDO BACULIMA CUMBE

C.I: 0105669683



Yo, *MARCELO XAVIER CAJAMARCA BERMEO*, autor de la tesis "DISEÑO E IMPLEMENTACIÓN DE UN REPOSITORIO ECUATORIANO DE DATOS ENLAZADOS GEOESPACIALES", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de INGENIERO DE SISTEMAS. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, 20 de junio del 2014

MARCELO XAVIER CAJAMARCA BERMEO

C.I: 0104736400



Yo, MARCELO XAVIER CAJAMARCA BERMEO, autor de la tesis "DISEÑO E IMPLEMENTACIÓN DE UN REPOSITORIO ECUATORIANO DE DATOS ENLAZADOS GEOESPACIALES", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 20 de junio del 2014

A handwritten signature in blue ink, appearing to read 'Marcelo', written over a horizontal line.

MARCELO XAVIER CAJAMARCA BERMEO

C.I: 0104736400



DEDICATORIA

Dedico esta tesis a mis Padres, Manuel y Alicia, pues gracias a su apoyo he logrado este objetivo , guiándome día y noche con sus palabras de aliento para que no caiga en el camino incorrecto, para levantarme cada vez que caía en algún problema, para obtener este título.

A mis Hermanos Edyson y Mariela, porque con su ejemplo de valor y constancia me motivaron en toda mi carrera universitaria.

A mis abuelos.

Fernando

La presente tesis la dedico a mis padres Miguel y Rosa, los cuales me han apoyado desde un inicio en mis estudios, animándome y brindándome energías para continuar ante cualquier problema; y a mis hermanos que aunque en ocasiones estemos alejados geográficamente, siempre espero que todos salgamos adelante.

Marcelo.



AGRADECIMIENTOS

A mis Padres, Manuel y Alicia , que con su fuerza, valor y perseverancia me apoyaron incondicionalmente durante toda mi carrera, guiándome por el camino correcto para lograr mis objetivos.

Fernando

A mi familia, quienes nos apoyamos y nos damos ánimos para salir adelante, empezando por mis padres los cuales con su ejemplo, me han dado razones para nunca rendirme.

Marcelo

Agradecemos a la Ingeniera Lucia Lupercio Novillo por el gran apoyo que nos brindó durante el desarrollo del presente trabajo de tesis, y al Ingeniero Víctor Saquicela por dirigir y compartir su conocimiento en este proyecto.



1. INTRODUCCIÓN

1.1. IDENTIFICACIÓN DEL PROBLEMA

Debido a la gran cantidad de información existente en la Web actual, la búsqueda de información se vuelve un problema; ante esta situación, muchas de las instituciones tanto públicas como privadas están generando datos para enlazarlos en la Web denominada Web de datos. Gran parte de esta información tiene un componente espacial que también pueden ser publicadas en la Web de datos. La Web Actual, se enfoca en la visualización de documentos escritos en lenguaje **HTML**, es decir, permite que estos documentos y sus objetos sean únicamente entendibles por el humano. Por el contrario, la Web de datos, permite la descripción de contenidos con tecnologías como **RDF**, **OWL** o **XML**; haciendo que la información sea entendible por máquinas y humanos para que así los gestores interpreten y se realicen procesos de captura, enlace y tratamiento de esta información.

Para lograr publicar información en la Web de datos, es necesario, primero disponer de un repositorio donde almacenarlos y consultarlos; y segundo, la información debe estar en el formato adecuado (**RDF**). Sin embargo, para el primer requerimiento, no existe un repositorio del Ecuador donde externamente se puede enlazar datos, sino actualmente cada institución (empresa, persona) publica su propia información sin relacionarla (aun teniendo información que puede ser complementaria), dicha información tampoco puede ser tratada mediante consultas semánticas, por ejemplo consultas semánticas geoespaciales (funciones **GEOSPARQL**). Para el segundo requerimiento, el inconveniente es la falta de herramientas de fácil uso (preferiblemente con interfaz gráfica) para la generación de **RDF** y que sea compatible con el estándar **GEOSPARQL**.



Entonces, debido a estos inconvenientes, se ve la necesidad de la creación de una herramienta con interfaz gráfica, en un entorno amigable al usuario, que permita generar RDF utilizando información proveniente de los archivos Shapefile, y a partir de éste se pueda publicar (siguiendo la respectiva metodología) en un repositorio, con sus respectivas relaciones con otros repositorios, lo que permitiría que nuestro país, esté en la Web de datos y cualquier repositorio externo lo pueda enlazar.

1.2. JUSTIFICACIÓN

Con el crecimiento que se está dando últimamente en cuanto a la publicación de información en la gran red de datos enlazados, es importante la presencia de Ecuador en la misma, porque, al tener publicados los datos, se puede también tener enlaces hacia otras fuentes permitiendo consultar dicha información y a la vez ofreciendo la información del Ecuador al exterior. Sin embargo, la no existencia de un repositorio público dedicado a la información geoespacial del Ecuador, causa que la consulta de esa información se realice de una manera muy manual, ya que se debe consultar cada sitio que publica esa información y extraerla. Pero incluso, si se llega a obtener de esta manera la información, la misma no estaría relacionada, y peor aún, visible por otros grandes repositorios de datos como la DBpedia. Por ello, se ve la necesidad de la creación y publicación de un repositorio de Datos Enlazados Geoespaciales, donde se publique información geoespacial relacionada y visible desde el exterior.

Para la publicación de información en la Web de datos se necesita de herramientas que apoyen a esta labor. Sin embargo, las herramientas actuales que son de acceso público, tales como herramientas que permite generación de Datos Enlazados Geoespaciales o que permiten la visualización de los Datos Enlazados; tienen deficiencias, tales como: la incompatibilidad con lenguajes de consulta geo, el uso de un estándar para la generación de Datos Enlazados, el uso de archivos de configuración complejos, y en unos casos, el no ser una herramienta visual. Por tal razón, es necesario el desarrollo de componentes que permitan la integración de las herramientas existentes, que a partir del formato actual de la información disponible se obtenga la publicación final de Datos Enlazados Geoespaciales; y así la información generada se podrá integrar



a repositorios más grandes, como por ejemplo la [DBpedia](#) , o en nuestro caso, un repositorio Ecuatoriano.

Tomando en cuenta que los Datos Enlazados Geoespaciales, es un área poco explotada a nivel mundial y mucho menos en Ecuador, es importante que esta tesis tome la iniciativa, para que su implementación en el país vaya aumentando y brindando aportes de importancia a la comunidad. Para lograr este objetivo, se seguirá la siguiente metodología de publicación de Datos Enlazados, utilizando herramientas libre acceso: Modelamiento, elaboración de un componente visual para generación de Datos Enlazados Geoespaciales, Enlace de Datos, Publicación y Explotación.

1.3. ALCANCE

Se trabajará con información geoespacial, concretamente con archivos en formato Shapefile. La mejora de las herramientas se enfocará únicamente a las destinadas a la generación (Geometry2RDF que permitirá llevar la información de los shapefile hacia documentos en formato RDF) y consumo (MAP4RDF que permite la visualización de archivos en formato RDF), esta mejora radica principalmente en que, deben trabajar con un formato estándar, tanto como modelo de relaciones de datos (ontología), y como lenguaje de consulta geoespacial, este estándar es GEOSPARQL. La mejora en el caso de Geometry2RDF además, será la implementación de una interfaz gráfica y el evitar el uso de archivos de configuración externos, como se lo hacía antes.

Se deberá integrar estas mejoras, en un entorno gráfico donde se permita, trabajar desde múltiples fuentes de datos, darles un nuevo formato, limpiarlos y obtenerlos en algún archivo resultante, es decir, un entorno **ETL** (Extract, Transform and Load). Finalmente, todos los datos generados mediante las herramientas mejoradas, se publicarán en un repositorio disponible, el cual dará lugar al: repositorio de Datos Enlazados Ecuatoriano el cual debe permitir su acceso al público en general.

1.4. OBJETIVO GENERAL

Creación de un prototipo de Repositorio de Datos Enlazados Geoespaciales del Ecuador, utilizando las herramientas mejoradas para generación de RDF.



1.5. OBJETIVOS ESPECÍFICOS

1. Analizar y definir las fuentes de información (Archivos Shapefile).
2. Analizar el modelo ontológico a utilizar (GEOSPARQL).
3. Analizar las herramientas existentes para la generación de Datos Enlazados Geoespaciales.
4. Crear un Plugin para la generación de RDF dentro de GeoKettle.
5. Analizar y extender las herramientas para el consumo de Datos Enlazados Geoespaciales (MAP4RDF)
6. Aplicar el proceso de publicación de Datos Enlazados Geoespaciales.



2. MARCO TEORICO

En este capítulo se tratarán los términos clave a los cuales se hará referencia a lo largo de la presente tesis. Algunos términos pueden llegar a ser muy generales y usarse en diferentes campos, por esta razón se define a cada termino, siendo estos tecnologías o herramientas, acorde al tema de la presente tesis.

2.1. WEB SEMÁNTICA

En la actualidad, la Web esta compuesta por varios documentos distribuidos en diferentes lugares del planeta, durante muchos años ha sido una poderosa herramienta para publicar, buscar y compartir información convirtiéndose en una gran base de datos, siendo imposible explotar adecuadamente esa información.

Desde sus inicios, la **WWW**, ha dado solución para la publicación de información de cualquier tipo; pero, el problema surge al consultar dicha información. Además, la veracidad de gran parte del contenido de la WWW no es completamente garantizada, pues, cualquier persona que tenga acceso a Internet puede publicar información de cualquier tipo, llegando así a formar parte de la gran base de datos compuesta por un número elevado de paginas Web, convirtiéndose en una Web Sintáctica cargada de documentos HTML siendo comprensible unicamente por el humano.

Durante algunos años ciertos grupos de investigación han puesto sus esfuerzos para lograr solventar este problema de consulta de información, y la falta de métodos para encontrar información relacionada, obteniendo como resultado el surgimiento de la Web de Datos o Web Semántica.

En la figura 2.1 se muestra gráficamente la evolución de la Web desde sus inicios, mostrándonos en que punto se encuentra actualmente el desarrollo de la Web

Semántica o Web 3.0.

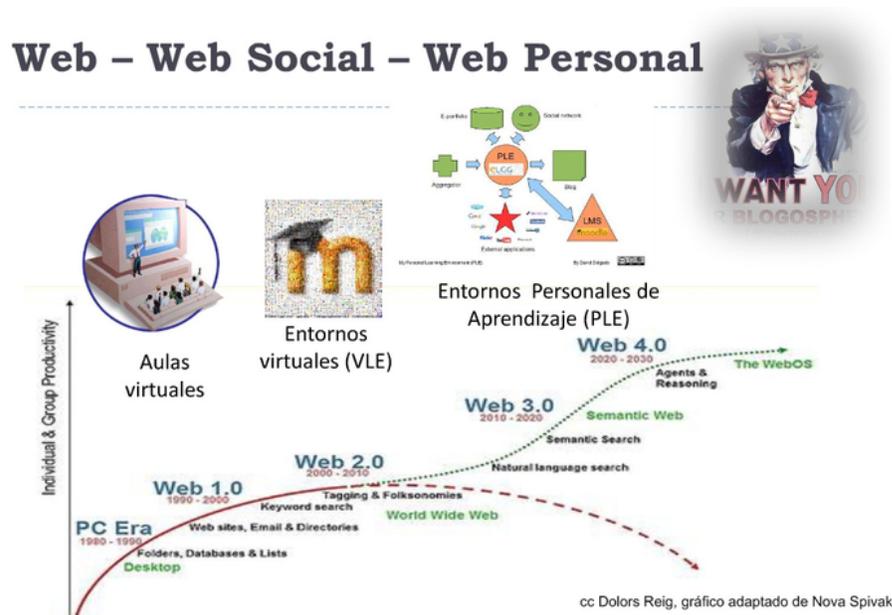


Figura 2.1: Evolución de la Web Semántica. Fuente: [14]

2.1.1. DEFINICIÓN DE WEB SEMÁNTICA

La Web Semántica es 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. Al dotar a la Web de más significado se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de modelos comunes, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante [18].

A continuación se describen los términos, tecnologías y herramientas, de carácter importante que se utilizaron durante el desarrollo de la presente tesis.

2.1.2. RDF y RDFS

Es un modelo estándar para el intercambio de datos en la Web semántica, brindando facilidades en cuanto a las relaciones entre los datos provenientes de diferentes fuentes. Además es muy adaptable y un modelo ya definido puede evolucionar sin necesidad de cambiar todos los datos [5].

La base sintáctica de un RDF son las triplas, las cuales consisten en sujeto, predicado y objeto formando un grafo RDF, el cual se lo puede procesar y manipular utilizando diferentes herramientas. En la figura 2.2 se muestra un ejemplo de triplas que indican la relación entre un recurso, nombre del recurso y geometría, en este caso el sujeto es el “Recurso” que tiene la propiedad “Nombre” cuyo valor es “Quito”, además “Recurso” tiene la propiedad “Geometría” cuyo valor es “Punto” y finalmente se indica que “Punto”, siendo el sujeto, tiene la propiedad “WKT” ,la cual indica que es el formato de la geometría, cuyo valor es “POINT(-78.52495 -0.22985)”

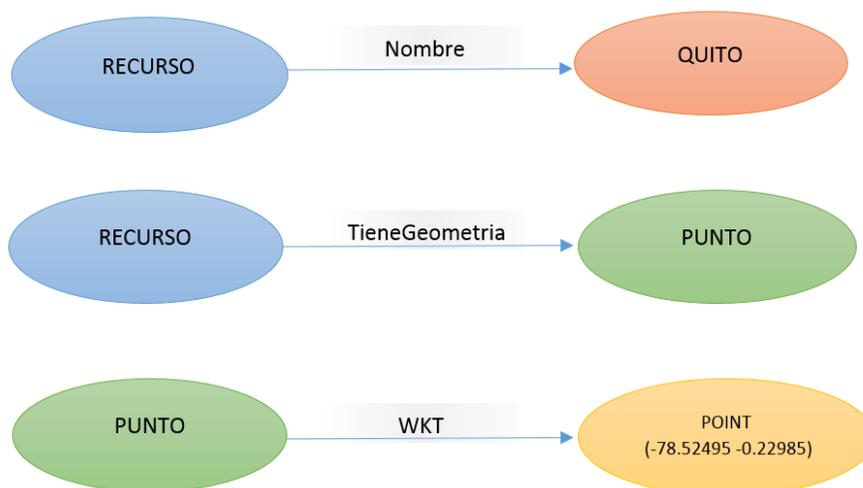


Figura 2.2: Ejemplo de Tripletas

Básicamente RDF es un conjunto de declaraciones que definen clases y propiedades de los recursos. A continuación en la tabla 2.1 y en la tabla 2.2 se muestra el vocabulario empleado por RDF y RDFS.

CLASE	DESCRIPCIÓN
rdfs:Resource	La clase de recurso, cada uno.
rdfs:Literal	La clase del valor literal, por ejemplo, cadenas de texto y números enteros.
rdf:XMLLiteral	La clase de los valores literales de los valores literales XML.
rdfs:Class	La clase de las clases.
rdf:Property	La clase de las propiedades RDF.
rdfs:Datatype	La clase de los tipos de datos RDF.
rdf:Statement	La clase de las declaraciones RDF.
rdf:Bag	La clase de los contenedores desordenados.
rdf:Seq	La clase de los contenedores ordenados.
rdf:Alt	La clase de los contenedores de alternativas.
rdfs:Container	La clase de los contenedores RDF.
rdf:List	La clase de las listas RDF.

Cuadro 2.1: Clases del Vocabulario RDF

PROPIEDAD	DESCRIPCIÓN
rdf:type	El sujeto es una instancia de una clase.
rdfs:subClassOf	El sujeto es una subclase de una clase.
rdfs:subPropertyOf	El sujeto es una subpropiedad de una propiedad.
rdfs:domain	Un dominio de la propiedad del sujeto.
rdfs:range	Un rango de la propiedad del sujeto.
rdfs:label	Un nombre para el sujeto legible por seres humanos.
rdfs:comment	Una descripción del recurso sujeto.
rdfs:member	Un miembro del recurso sujeto.
rdf:first	El primer ítem en la lista RDF del sujeto.
rdf:rest	El resto de la lista RDF del sujeto después del primer ítem.
rdfs:seeAlso	Más información sobre el recurso sujeto.
rdfs:isDefinedBy	La definición del recurso sujeto.
rdf:value	Propiedad idiomática usada para valores estructurados .
rdf:subject	El sujeto de la declaración RDF del sujeto.
rdf:predicate	El predicado de la declaración RDF del sujeto.
rdf:object	El objeto de la declaración RDF del sujeto.

Cuadro 2.2: Propiedades del vocabulario RDF

2.1.3. ONTOLOGÍA

Se trata de un esquema conceptual que define los términos a utilizar para describir y representar un área de conocimiento dado, con el objetivo de facilitar la comunicación entre diferentes entidades. Es decir, codifican el conocimiento de un dominio y también el conocimiento que extiende los dominios y de esta manera hacen el conocimiento reutilizable [7].



2.1.4. SPARQL y GEOSPARQL

A continuación se describen los términos SPARQL y GEOSPARQL, utilizados a menudo en el desarrollo del presente trabajo de Tesis.

SPARQL

SPARQL es una especificación que ofrece un lenguaje y protocolos para consultar y manipular RDF. Se trata de un lenguaje estandarizado por la W3C clave para el desarrollo de la Web semántica, en principio SPARQL trata únicamente funciones para consulta de datos RDF pero actualmente incluye funciones para inserción, modificación y eliminación de datos [6].

Entre las características de SPARQL, resaltan:

- Lenguaje de consulta.
- Formato XML para resultados en una consulta.
- Actualización de grafos RDF.
- Protocolo para RDF.
- Descubrir información acerca de la información almacenada (dataset).
- Manejo de grafos mediante el protocolo HTTP.

En la figura 2.3 se muestra un ejemplo de consulta SPARQL, haciendo referencia a las tripletas de la figura 2.2, se observa la estructura de la consulta para obtener los datos deseados como el Nombre del recurso, tipo de geometría y la geometría en formato WKT. Lo que se encuentra en el recuadro de color verde permite identificar el recurso, en este caso “Quito”, posteriormente en el recuadro de color rojo se observa la parte de la consulta que permite obtener el tipo de geometría del recurso “Quito” y finalmente en el recuadro de color azul se encuentra la parte de la consulta que permite obtener la geometría en formato WKT del recurso “Quito”, también se muestra el resultado generado por la consulta SPARQL.

```
SELECT DISTINCT
?Nombre ?figura ?wkt
WHERE {
  ?r a <http://geo.linkeddata.ec/ontology/ciudades_principales>.
  ?r rdfs:label "Quito"@es.
  ?r geo:hasGeometry ?geo.
  ?geo a ?figura.
  ?geo geo:asWKT ?wkt
  OPTIONAL{?r rdfs:label ?Nombre}
}
```

RESULTADO

Nombre	figura	wkt
"Quito" @es	http://www.opengis.net/ont/sf#Point	POINT (-78.50549649860159 -0.2234814224194014)

Figura 2.3: Ejemplo de consulta SPARQL

GEOSPARQL

Es un estándar establecido por Open Geospatial Consortium (OGC) para la representación de datos geospaciales en la Web Semántica. GEOSPARQL define un vocabulario para representar datos con información Geoespacial en RDF y también define una extensión al lenguaje de consulta SPARQL para la consulta y procesamiento de datos geospaciales [11].

Entre las características fundamentales de GEOSPARQL se mencionan las siguientes:

- Un vocabulario RDF/OWL para la representación de información geoespacial
- Un conjunto de funciones extensión a SPARQL para cálculos espaciales
- Un conjunto de reglas RIF para la transformación de Consultas
- GEOSPARQL no define un vocabulario amplio para la representación de información geoespacial. en su lugar define un conjunto básico de clases, propiedades



y tipo de datos que son utilizados para construir patrones de consulta.

2.1.5. DATOS ENLAZADOS Y DATOS ENLAZADOS GEO-ESPACIALES

A continuación se describen los términos DATOS ENLAZADOS Y DATOS ENLAZADOS GEOESPACIALES, utilizados a menudo en el desarrollo del presente trabajo de Tesis.

DATOS ENLAZADOS

Los Datos Enlazados (Linked Data), hacen referencia a la vinculación de los datos de la Web semántica que se encuentra distribuida en la Web de forma que el usuario pueda obtener información relacionada proveniente de diferentes fuentes. La Web Semántica, igual que la Web del hipertexto, también se constituye mediante documentos publicados en la Web . Sin embargo, en la Web del hipertexto, los enlaces relacionan documentos HTML, mientras que, en la Web de Datos se enlazan propiedades descritas en los documentos RDF .

Los Datos Enlazados se basan en la aplicación de cuatro principios básicos que permitirán un crecimiento de la Web de Datos. Estos principios deben ser respetados para conseguir los datos interconectados y así reutilizar la información publicada [17].

A continuación se escriben los Principios básicos de Datos Enlazados:

- Usar URIs para identificar las cosas: Al usar URIs para nombrar a las cosas se emplea una abstracción del lenguaje natural y se ofrece una forma estándar y unívoca de referirse a cualquier recurso.
- Usar URIs HTTP: Este principio permite que cualquier recurso pueda ser buscado y accedido desde la Web.
- Ofrecer información sobre los recursos usando RDF: Una vez que se busca un recurso y ha sido identificado según los dos pasos anteriores, se debe obtener información útil representada mediante descripciones estándares RDF permitiendo que esa información sea fácilmente procesable.



- Incluir enlaces a otros URIs: Los datos que se tiene deben ser interconectados hacia otros enlaces para que esta información no quede aislada y así también pueda ser accedida desde otros enlaces.

DATOS ENLAZADO GEOESPACIALES

Los Datos Enlazados Geoespaciales (Geo Linked Data) permiten el enriquecimiento de la Web de datos con información geoespacial, aplicando los principios de linked data, es decir, se puede publicar información georreferenciada en la Web de datos para poder realizar análisis geográfico de los recursos [15].

A continuación se indican algunos ejemplos de análisis que se pueden realizar según la información geográfica.

- Mostrar los elementos georreferenciados a través de un Mapa.
- Cuantos o cuales elementos se encuentran a una distancia X del elemento A
- Cuales elementos se intersectan con el elemento B
- Cuales elementos tocan al elemento C
- Cuales elementos se encuentran dentro del elemento D
- etc.

2.2. METODOLOGÍA PARA LA GENERACIÓN DE DATOS ENLAZADOS GEOESPACIALES

Para la generación y consumo de datos enlazados (linkeddata) es necesario seguir un conjunto de actividades, desde el análisis de las fuentes de información hasta la explotación de la información ya publicada en formato RDF, también se indicara las respectivas herramientas y tecnologías que se utilizaron. [9]. Estas actividades, que



describen la metodología para la generación y consumo de Datos Enlazados, son las siguientes: (1) Especificación,(2) Modelado,(3) Generación de RDF,(4) Generación de Links,(5) Publicación y (6) Explotación.

2.2.1. ESPECIFICACIÓN

Esta actividad esta compuesta por dos pasos.

- Identificación y Análisis de las fuentes de datos: En este paso se definen claramente cuales serán las fuentes y el formato que se encuentra alojada la información con la que se va a trabajar. También se realiza el respectivo análisis para verificar la validez de las fuentes y formas de alojamiento de la información.
- Diseño de URIs: Uno de los principios para la generación de Datos Enlazados es usar URIs para identificar las cosas, estas URIs deben ser significativas para facilitar la identificación de los elementos y el enlace hacia otras fuentes, por lo tanto, el diseño de estas, debe ser parte de un adecuado y moderado análisis para la posterior generación de RDF.

2.2.2. MODELADO

En esta actividad, según los datos de las fuentes de información, se realiza la selección de la respectiva ontología para la generación de RDF, en caso de no existir la ontología adecuada se debe modificar o crear según las necesidades requeridas.

2.2.3. GENERACIÓN DE RDF

Luego de analizar la información y con el respectivo modelo ontológico se procede a la generación de los datos a formato RDF. Para ésta actividad se trabajó con la herramienta Geometry2RDF.

Geometry2RDF es un programa que toma como entrada archivos de formato shapefile para , mediante un archivo de configuración, generar RDF de todas las geometrías. Originalmente este programa es independiente y para poder utilizarlo se debe tener conocimiento de datos geoespaciales, programación y principalmente de tecnología semántica.



Para la generación de las geometrías dentro del RDF, utiliza el formato **WKT** y **GML**; usa el sistema de coordenadas especificado en el archivo de configuración.

Las características principales de Geometry2RDF son:

- Soporta archivos SHP
- Soporta salida WKT y GML.
- Usa un archivo de configuración por cada archivo a convertir.
- Requiere Java para ejecutarse.
- Es un programa independiente, puede funcionar sin necesidad de un programa padre que lo contenga.

2.2.4. GENERACIÓN DE LINKS

Una vez generada la información en RDF es importante descubrir enlaces entre diferentes fuentes internas o enlaces externos para encontrar información relacionada y compartirla.

Para solucionar este problema se utiliza Silk Workbench, el cual es una aplicación Web que a través de un proceso permite al usuario crear una especificación o condición de enlazado para encontrar recursos relacionados entre diferentes fuentes de datos SPARQL [13].

Silk en su entorno de trabajo ofrece los siguientes componentes:

- Workspace Browser: Permite al usuario navegar a través del proyecto y especificar las fuentes de datos y en la sección Linking Task, permite especificar las tareas para enlazado.
- Linkage Rule Editor: Un editor gráfico que permite al usuario fácilmente crear y editar especificaciones de enlazado.
- Evaluation: El usuario ejecuta la búsqueda de enlaces, una vez devuelto los resultados debe seleccionar si son o no son correctos dichos enlaces.



2.2.5. PUBLICACIÓN

Los datos RDF deben ser publicados para que otras personas puedan acceder mediante consultas a esta información. Para ello, la información debe ser cargada a un servidor de datos RDF o servidor de Tripletas, que permite gestionar la información de los archivos RDF. A continuación se menciona algunos de estos servidores.

- VIRTUOSO
- PARLIAMET
- FUSEKI

El servidor seleccionado es Parliamet. Se trata de una aplicación que permite el almacenamiento de tripletas diseñado para la Web Semántica, fue lanzado como proyecto de código abierto en junio del 2009. Compatible con RDF, RDFS, OWL, SPARQL y las normas GEOSPARQL, Parliament emplea un innovador sistema de almacenamiento de datos que entrelaza los datos con un índice único permitiendo responder a consultas de manera eficiente debido a su re-ordenación de la consulta para que las partes mas restrictivas se ejecuten primero. Como resultado a esta optimización, Parliament soporta la inserción rápida y la consulta rápida, en comparación a otros servidores de tripletas que favorecen unicamente a la consulta [10].

Entre las características principales se tiene las siguientes:

- Parliament tiene un índice temporal, de manera que pueda responder eficiente las consultas como “encontrar todos los eventos que se produjeron entre los tiempos X y Y”.
- Soporta GEOSPARQL, norma aprobada por la OGC para datos semánticos geoespaciales. Usando su índice geoespacial, Parliament puede responder eficientemente consultas como “encontrar todos los artículos que se encuentran dentro de la región X”.
- Es una aplicación de fácil instalación y de fácil uso.
- Ofrece un SPARQL Endpoint



2.2.6. EXPLOTACIÓN

Finalmente, cuando los datos han sido publicados, el usuario puede consumir esta información. A través de consultas sobre un SPARQL Endpoint. Un SPARQL Endpoint es un servicio que ofrece el servidor de datos RDF, al cual el usuario o aplicación puede conectarse para consultar información.

Al tratarse de información geoespacial esta información puede ser visualizada a través de mapas, para ello se trabajó con la aplicación MAP4RDF. MAP4RDF es un herramienta para explorar y visualizar conjuntos de datos RDF enriquecidos con información geométrica, es un software open source que necesita ser configurado para usar cualquier SPARQL Endpoint y que proporciona a los usuarios una visualización de datos georeferenciados en RDF en un mapa [2].

2.3. HERRAMIENTAS DE DESARROLLO Y SOPORTE

A continuación se describen algunas herramientas que han sido utilizadas para el desarrollo y prueba del plugin de Geometry2RDF, así como para la mejora de MAP4RDF.

2.3.1. API GEOTOOLS

Es una librería que permite la obtención y manipulación de información geoespacial, por ejemplo leer el sistema de coordenadas, los datos dentro de un shapefile, etc.

Esta API es usada en la versión original de Geometry2RDF donde principalmente lee los datos y realiza las conversiones a un sistema de coordenadas estándar .

Sus funciones principales son las siguientes [3]:

- Acceso a datos GIS en varios formatos y bases de datos.
- Trabaja con varias proyecciones.
- Filtra y analiza datos en términos en atributos espaciales y no espaciales.



- Muestra mapas con algún estilizado complejo.
- etc.

2.3.2. GEOKETTLE

Es un ambiente integrador orientado a datos geospaciales, donde gráficamente se colocan diferentes componentes que permiten la obtención y manipulación de los datos. Dentro de este ambiente funciona el Plugin creado (uno de los objetivos específicos de ésta tesis) para generar RDF.

Características principales de Geokettle [4]:

- Extracción de datos de PostGIS, Oracle spatial, Mysql; archivos con formato SHP, GML, KML, OGR; servicios web OGC.
- Cálculos de área, longitud, distancia, intersecciones, uniones.
- Procesamiento de líneas, polígonos, división de polilíneas.
- Cargado de datos en bases de datos espaciales, data warehouse espacial, formato GML, KML, OGR, SHP, servicios web OGC.

2.3.3. JENA

Es una librería destinada a la creación, manipulación de ontologías y la generación de su respectiva salida en RDF, además de permitir expandir los vocabularios existentes de RDF; para lograr esto, Jena ofrece funciones que retornan información de todas las tripletas contenidas en un modelo ontológico, el cual puede ser creado desde cero o cargado desde un archivo independiente. Un modelo ontológico cargado en Jena, contendrá nodos que deberán estar relacionados en una sola dirección, por ejemplo:

ejemplo:obj foaf:name Juan



Se indica que el recurso ejemplo:obj tiene la propiedad name con el valor Juan, pero la relación inversa no es cierta (que el recurso Juan tenga la propiedad name con el valor ejemplo:obj).

Para la realización de consultas, Jena dispone de un motor llamado ARQ, el cual soporta SPARQL y permite buscar dentro de un modelo ontológico cargado.

Entre sus características principales, están [8]:

- Generación y lectura de RDF.
- Consultas en SPARQL.
- Navegación en un modelo ontológico.
- Uso de OWL.
- etc.

2.4. FORMATO DE LAS FUENTES DE INFORMACIÓN

A continuación se describe el formato más usado para almacenar información geoespacial, el Shapefile, el cual será el punto de partida para la lectura de toda la información disponible del Ecuador.

2.4.1. SHAPEFILE

Se trata de un formato de archivos que almacena información no topológica con características espaciales de elementos geográficos.

Los archivos shapefile tienen grandes ventajas sobre otras fuentes de datos, como mayor velocidad de dibujo y edición. Además, éstos archivos suelen requerir poco espacio de almacenamiento en disco y se pueden leer y escribir con facilidad. Estos shapefile soportan a geometrías como puntos, líneas y polígonos, los atributos son almacenados en un archivo de formato dBase.



Descripción técnica de los shapefiles

Un archivo shape, originario de Environmental Systems Research Institute (ESRI), consiste de un archivo principal, un archivo de índice y una tabla dBase.

- El archivo principal (*.shp) es un archivo de longitud variable en el que cada registro describe una forma con sus respectivos vértices.
- El archivo de índice (*.shx) acompaña al archivo principal (*.shp) que almacena la posición de los Id. de entidades individuales en el archivo .shp
- La tabla dBase (*.dbf) almacena la información de atributos de las entidades.

Los diferentes archivos que componen el formato shapefile deben tener el mismo nombre cada uno con su respectiva extensión, por ejemplo:

- Archivo Principal: provinciasdeecuador.shp
- Archivo de Índice: provinciasdeecuador.shx
- Tabla de Atributos: provinciasdeecuador.dbf

Visualización de los Archivos Shapefile

Los archivos Shapefile pueden ser visualizados a través de cualquier software GIS como por ejemplo: Arc GIS, Quantum GIS, etc. La visualización permitirá saber que son Shapefile válidos. En la figura 2.4 se visualiza un archivo Shapefile válido utilizando el software de análisis geoespacial QGIS.

En la figura 2.5 se visualiza una tabla de registros de la información contenida en el archivo shapefile que ha sido cargada a QGIS.

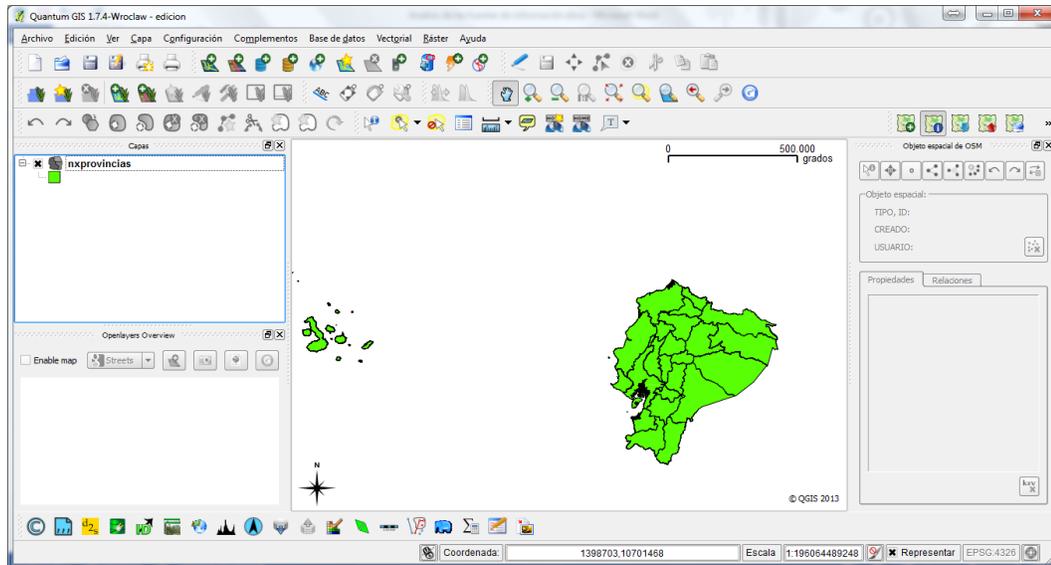


Figura 2.4: Visualización Geográfica de las Provincias del Ecuador en QGIS

Attribute table - nxprovincias :: 0 / 25 feature(s) selected

	DPA_PROVIN	DPA_DESPRO	DPA_VALOR	DPA_ANIO	REL_CODIGO	REN_CODIGO	PEE_CODIGO
0	01	AZUAY	0	2011	05	01	593
1	02	BOLIVAR	0	2011	02	01	593
2	03	CAÑAR	0	2011	05	01	593
3	04	CARCHI	0	2011	04	01	593
4	05	COTOPAXI	0	2011	02	01	593
5	06	CHIMBORAZO	0	2011	02	01	593
6	07	EL ORO	0	2011	03	02	593
7	08	ESMERALDAS	0	2011	04	02	593
8	09	GUAYAS	0	2011	03	02	593
9	10	IMBABURA	0	2011	04	01	593
10	11	LOJA	0	2011	05	01	593
11	12	LOS RIOS	0	2011	03	02	593
12	13	MANABI	0	2011	03	02	593
13	14	MORONA SAN...	0	2011	05	03	593
14	15	NAPO	0	2011	04	03	593
15	16	PASTAZA	0	2011	02	03	593
16	17	PICHINCHA	0	2011	04	01	593
17	18	TUNGURAHUA	0	2011	02	01	593
18	19	ZAMORA CHIN...	0	2011	05	03	593
19	20	GALAPAGOS	0	2011	03	04	593
20	21	SUCUMBIOS	0	2011	04	03	593
21	22	ORELLANA	0	2011	04	03	593
22	23	SANTO DOMIN...	0	2011	04	01	593
23	24	SANTA ELENA	0	2011	03	02	593
24	90	ZONA NO DELL...	0	2011	90	05	593

Figura 2.5: Tabla de información de un archivo shapefile.



3. ANÁLISIS Y EXTENSIÓN DE LAS HERRAMIENTAS DE GENERACION Y CONSUMO DE DATOS ENLAZADOS GEOESPACIALES

En este capítulo se tratará el análisis realizado de las herramientas Geometry2RDF y MAP4RDF para su posterior mejora, empezando desde la especificación de sus funcionalidades originales, hasta la prueba de las nuevas funcionalidades. El objetivo es que las herramientas estén preparadas para la generación y visualización de RDF utilizando tecnologías estándar.

3.1. INTRODUCCIÓN A GEOMETRY2RDF

Es una librería *Open Source*, utilizada para generar ficheros RDF a partir de información geográfica, la cual puede estar en formato GML, WKT, o proveniente de alguna base de datos geoespacial de Oracle o MySQL.

Esta herramienta permite mapear información entre la fuente de información geoespacial y la ontología, para ello, se debe llenar los parámetros del archivo `options.properties` de acuerdo al tipo de información que se disponga.

Esta librería utiliza GeoTools para la lectura de archivos shp, y Jena para la generación de las respectivas tripletas. Además se necesitar de Java 1.5 o posterior para



poder ejecutarse.

Entre los parámetros más importantes que se deben configurar para que funcione correctamente, se tiene:

- outputFile: Es la dirección completa donde se almacenará el archivo.
- dbType: Puede ser 0 para MSAccess, 1 para MySQL y 2 para Oracle.
- dbName: Nombre de la base de datos.
- dbUserName: Usuario de la base de datos.
- dbHost: Dirección del host de la base de datos.
- dbPort: Puerto utilizado para conectarse a la base de datos.
- resourceName: Nombre del recurso a leer.
- tableName: Nombre de la tabla.
- condition: Va la condición para que se tome o no una fila leída, debe estar en lenguaje java.
- labelColumnName: Nombre de la columna que contiene un nombre que identifique al recurso.
- nsPrefix: Prefijo dentro del RDF para los recursos generados.
- nsURI: Dirección que será usada de base para la generación de URIs de cada uno de los recursos.
- ontologyNSPrefix: Prefijo de la nueva ontología que utiliza el RDF.
- ontologyNS: Namespace de la ontología utilizada por el RDF.
- sourceRS: Sistema de coordenadas EPSG que se leerá.
- targetRS: Sistema de coordenadas EPSG que resultará en el RDF generado.
- pointType: URI del tipo de dato punto.



- `linestringType`: URI del tipo de dato línea.
- `polygonType`: URI del tipo de dato polígono.
- `formBy`: Nombre de la propiedad que indicará la relación que tiene una geometría con sus puntos que la conforman.

3.2. ANÁLISIS DE FUNCIONALIDAD ACTUAL DE GEOMETRY2RDF

Las funcionalidades que provee esta librería son las siguientes:

- Leer la información que se desea transformar, desde bases de datos geoespaciales
- Trabajar con formatos GML y WKT en la generación de RDF.
- Utilizar un fichero de configuración para realizar el mapeo entre el archivo geoespacial y la ontología.
- Ejecución independiente (no requiere de un programa externo para funcionar).
- Compatibilidad con los gestores de bases de datos MySQL, Oracle y archivos SHP.
- Compatibilidad con formatos GML y WKT en la generación de RDF.
- Utilización de un fichero de configuración `options.properties` por cada archivo a transformar.

3.2.1. Limitaciones

Las limitaciones actuales de Geometry2RDF son las siguientes:

- No dispone de una interfaz gráfica.
- La configuración inicial es compleja.



- No dispone de una configuración predeterminada base.
- El RDF resultante no es compatible con GEOSPARQL.
- La generación de geometrías complejas como Polygon, Multipolygon, Multiline no sigue un estándar y en algunos casos, la geometría es transformada primero a un enorme conjunto de puntos contiguos.
- No existe nuevas actualizaciones de la librería, desde el punto de vista tecnológico.

Además, se encuentran limitaciones en cuanto a la estructura del proyecto utilizado:

- El proyecto de desarrollo (hecho en NetBeans) que contiene a Geometry2RDF, no es maven (que permite indicar todas las librerías que son necesarias en el proyecto, y que así pueda ser construido en cualquier entorno de desarrollo que lo soporte).
- La versión de la librería Jena es muy antigua y no es compatible por completo con la creación de clases en RDF.

3.3. ANÁLISIS DE FUNCIONALIDAD A CORRIGIR Y EXTENDER A GEOMETRY2RDF

Debido a la estandarización de vocabularios dentro del área de la Web semántica para datos geográficos, la W3C ha definido el estándar GEOSPARQL, razón por la cual en esta sección se analizará todo lo necesario para añadir la su compatibilidad dentro de Geometry2RDF, y que a posterior se pueda realizar consultas geográficas usando sus funciones integradas.

3.3.1. Identificación de errores a corregir

Se necesita corregir los siguientes detalles técnicos:



- Se generan demasiadas tripletas para el caso de geometrías complejas (Polygon, Multiline, etc) y muchas de estas tripletas, contienen información repetidas (un polígono es representado tanto en WKT de toda la figura, como en tripletas individuales de cada uno de sus puntos).
- Se están usando funciones obsoletas, por lo que existe gran probabilidad que hayan sido eliminadas en versiones más recientes de Jena.
- Se utiliza un directorio temporal para el manejo una ontología, lo cual aumenta el tiempo para la creación de RDF.

3.3.2. Descripción de las nuevas funcionalidades para Geometry2RDF

- Proyecto en Maven: El proyecto debe estar en maven para un mejor manejo de las versiones de las librerías y además que el tamaño del proyecto sea mucho menor.
- RDF estándar de GEOSPARQL: La generación de RDF estándar de GEOSPARQL ayudará a poder realizar consultas sobre el RDF generado e incluso buscar posibles enlaces hacia RDFs externos.
- Generación de WKT estándar: La generación de WKT dentro del RDF debe ser estándar, y no repetitivo, utilizando clases existentes dentro del estándar RDF y siguiendo el formato WKT compatible con GEOSPARQL.
- Detección de coordenadas: Detectar automáticamente el sistemas de coordenadas actual del archivo SHP, eliminando así configuraciones adicionales.

3.4. DESARROLLO Y VALIDACIÓN DE FUNCIONALIDAD A CORREGIR Y EXTENDER DE GEOMETRY2RDF

3.4.1. Corrección de errores identificados en Geometry2RDF

Para la corrección del error de repetición de tripletas, se eliminó el código que generaba las tripletas sobrantes y se trabajó solo con el WKT de la geometría. Esto permite que Parliament reconozca la tripleta como compatible con GEOSPARQL y además permite ahorrar espacio en disco.

En el problema del uso de funciones obsoletas, primeramente se actualizaron las librerías de Jena a la 2.10 y GeoTools a la 2.7.3, y posteriormente el IDE NetBeans identificó automáticamente las funciones obsoletas, las cuales fueron reemplazadas. Para el problema del uso de un directorio temporal dentro de la creación de la ontología, se lo solucionó utilizando la clase OntModel de Jena, que permite trabajar con ontologías completamente en memoria, sin necesidad de crear directorios temporales.

3.4.2. Validación de errores corregidos

La Figura 3.6 muestra la representación de un polígono en RDF, tanto en la versión actual como en la anterior, apreciándose que el RDF generado actualmente es mucho menor y que el anterior genera un recurso en WKT con cada uno de sus puntos, duplicando la información.

```
<geontology:provincia rdf:about=
"http://geo.linkeddata.ec/resource/provincia/AZUAYA9e55c768
609fef24ec3e6ca5c0af0c365dda696">
  <geontology:hasExactGeometry>
    <sf:Multipolygon rdf:about=
"http://geo.linkeddata.ec/resource/provincia/a9e55c768609fe
f24ec3e6ca5c0af0c365dda696">
      <geo:asWKT rdf:datatype=
"http://www.opengis.net/ont/sf#wktLiteral"
        >&lt;http://www.opengis.net/def/crs/ogc/1.3/CRS84
&gt; MULTIPOLYGON (((-78.56954431188377
-2.5586524111890663, -78.56919905766202
...
-78.56954431188377 -2.5586524111890663)))/>
    </sf:Multipolygon>
  </geontology:hasExactGeometry>
  <rdfs:label xml:lang="es">AZUAY</rdfs:label>
</geontology:provincia>
```

```
<rdf:Description rdf:about=
"http://geo.linkeddata.ec/resource/22ece26c30ad1a82
e86dddbcbad65bf3221e3a5f">
  <rdf:type rdf:resource=
"http://geo.linkeddata.ec/ontology/Po1%3ADgono"/>
  <geontology:gm1>POLYGON ((-1.521623770849759
-78.35799742261177, -1.5216258187413705
...
-78.35799747670171, ))/>
  <geontology:formadoPor rdf:resource=
"http://geo.linkeddata.ec/resource/wgs84/-78.569544
31188377_-2.5586524111890663"/>
  <geontology:formadoPor rdf:resource=
"http://geo.linkeddata.ec/resource/wgs84/-78.569199
05766202_-2.5582844655097183"/>
  ...
</rdf:Description>
```

Figura 3.6: Generación actual (izquierda) y anterior (derecha) de un polígono



Para el reemplazo de funciones obsoletas, se investigó en la Web por sus reemplazantes, la figura 3.7 muestra un ejemplo de reemplazo. Ambos códigos tiene la misma funcionalidad, con la diferencia que el actual es compatible con versiones posteriores de GeoTools.

```
File file = new File(inputFileName);           File file = new File(fileString);
Map map = new HashMap();                       Map map = new HashMap();
map.put("url", file.toURI().toURL());          map.put("url", file.toURI());
```

Figura 3.7: Función actual (izquierda) y función obsoleta (derecha)

Finalmente, para el problema de la creación de un directorio temporal durante la creación de la ontología, este se vio solucionado cuando se revisó el directorio del proyecto y se notó que no existía un directorio adicional, como sucedía antes.

3.4.3. Implementación de nuevas funcionalidades requeridas en Geometry2RDF

El proyecto base de Geometry2RDF se convirtió a maven, actualizando también todas las librerías que se usaban anteriormente; si se desea distribuir el proyecto, se lo hará solo con el código fuente, ya que maven se encargará de la descarga de las librerías faltantes en el computador que se desee ejecutar el proyecto.

Al solucionar el error de información duplicada, se incluyó además la compatibilidad con GEOSPARQL y además usando el WKT que ya generaba pero que no estaba en el lugar indicado para que sea compatible con el estándar.

3.4.4. Validación de nuevas funcionalidades implementadas

La Figura 3.8 muestra la estructura actual del proyecto transformado, que ahora es compatible con Maven y que además, sigue una estructura de paquetes hasta dar con las clases principales.

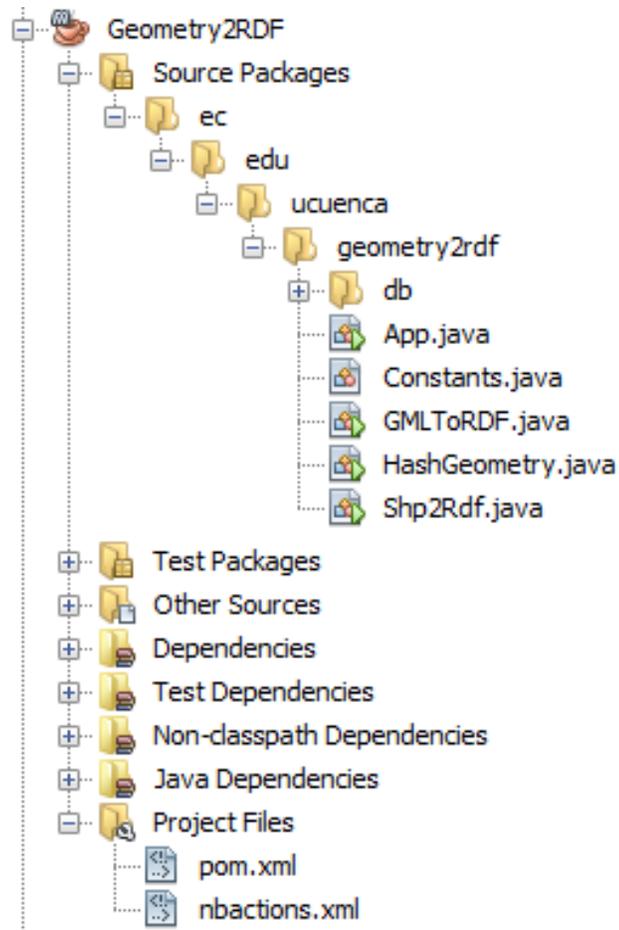


Figura 3.8: Geometry2RDF con formato Maven

Ahora, para comprobar que se pueda generar RDF compatible con GEOSPARQL, primeramente se revisó el ejemplo proporcionado por el documento del estándar GEOSPARQL, el cual se muestra en la Figura 3.9. Con el ejemplo citado, se procede a generarlo mediante código, para lo cual se modificó parte del proyecto de Geometry2RDF, teniendo como resultado el RDF de la Figura 3.10 donde se puede notar que es muy similar al original de GEOSPARQL y demostrando así la posibilidad de que ese formato sea implementado.

```
<rdfs:Class rdf:about="http://example.org/Applicationschema#PlaceOfInterest">
  <rdfs:subClassof rdf:resource="http://www.opengis.net/ont/geosparql#Feature"/>
</rdfs:Class>
<rdf:Property rdf:about="http://example.org/Applicationschema#hasExactGeometry">
  <rdfs:subPropertyof rdf:resource="http://www.opengis.net/ont/geosparql#hasGeometry"/>
  <rdfs:subPropertyof rdf:resource="http://www.opengis.net/ont/geosparql#hasDefaultGeometry"/>
</rdf:Property>
<rdf:Property rdf:about="http://example.org/Applicationschema#hasPointGeometry">
  <rdfs:subPropertyof rdf:resource="http://www.opengis.net/ont/geosparql#hasGeometry"/>
</rdf:Property>
<my:PlaceOfInterest rdf:about="http://example.org/Applicationschema#A">
  <my:hasExactGeometry rdf:resource="http://example.org/Applicationschema#AExactGeom"/>
  <my:hasPointGeometry rdf:resource="http://example.org/Applicationschema#APointGeom"/>
</my:PlaceOfInterest>
<sf:Polygon rdf:about="http://example.org/Applicationschema#AExactGeom">
  <geo:asWKT rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">
    <![CDATA[
      <http://www.opengis.net/def/crs/OGC/1.3/CRS84>Polygon((-83.6 34.1, -83.2 34.1,
        -83.2 34.5, -83.6 34.5, -83.6 34.1)]]>
    </geo:asWKT>
  </sf:Polygon>
<sf:Point rdf:about="http://example.org/Applicationschema#APointGeom">
  <geo:asWKT rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">
    <![CDATA[<http://www.opengis.net/def/crs/OGC/1.3/CRS84>Point(-83.4 34.3)]]>
  </geo:asWKT>
</sf:Point>
```

Figura 3.9: GEOSPARQL del ejemplo de la especificación del estándar

```
<rdfs:Class rdf:about="http://example.com/Applicationschema#PlaceOfInterest">
  <rdfs:subClassof rdf:resource="http://www.opengis.net/ont/geosparql#Feature"/>
</rdfs:Class>
<rdf:Property rdf:about="http://example.com/Applicationschema#hasExactGeometry">
  <rdfs:subPropertyof rdf:resource="http://www.opengis.net/ont/geosparql#hasDefaultGeometry"/>
  <rdfs:subPropertyof rdf:resource="http://www.opengis.net/ont/geosparql#hasGeometry"/>
</rdf:Property>
<rdf:Property rdf:about="http://www.opengis.net/ont/sf#Polygon"/>
<rdf:Property rdf:about="http://www.opengis.net/ont/sf#Point"/>
<rdf:Property rdf:about="http://www.opengis.net/ont/geosparql#asWKT"/>
<rdf:Property rdf:about="http://example.com/Applicationschema#hasPointGeometry">
  <rdfs:subPropertyof rdf:resource="http://www.opengis.net/ont/geosparql#hasGeometry"/>
</rdf:Property>
<my:PlaceOfInterest rdf:about="http://example.com/ApplicationResource/A">
  <my:hasPointGeometry>
    <sf:Point rdf:about="http://example.com/ApplicationResource/APointGeom">
      <geo:asWKT rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">
        >&lt;http://www.opengis.net/def/crs/OGC/1.3/CRS84&gt;POINT(0.0001 0.0002)</geo:asWKT>
      </sf:Point>
    </my:hasPointGeometry>
  <my:hasExactGeometry>
    <sf:Polygon rdf:about="http://example.com/ApplicationResource/AExactGeom">
      <geo:asWKT rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">
        >&lt;http://www.opengis.net/def/crs/OGC/1.3/CRS84&gt;POLYGON(0.0001 0.0002,
          0.003 0.004, 0.005 0.006)</geo:asWKT>
      </sf:Polygon>
    </my:hasExactGeometry>
  </my:PlaceOfInterest>
</rdf:RDF>
```

Figura 3.10: GEOSPARQL generado manualmente mediante código Java

Al generar RDF compatible con GEOSPARQL, las tripletas que contengan la información de las figuras debe tener un formato estándar, al realizar los cambios respectivos, se realizó una comparación con el formato que se utilizaba anteriormente, esta comparación está representada en la Figura 3.11.

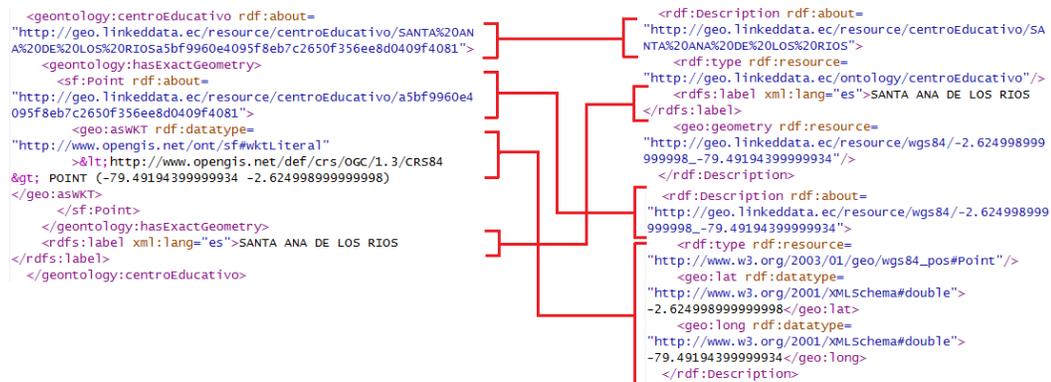


Figura 3.11: Generación de un punto antes (derecha) y ahora (izquierda)

3.5. SELECCIÓN DE LA HERRAMIENTA ETL A USAR PARA LA INTEGRACIÓN

Para la elaboración de un plugin, primero se realizó un análisis de las herramientas actuales y si éstas herramientas pueden ayudar a la elaboración de partes del plugin o si todo el proceso debería ser integrado dentro del plugin.

Las herramientas a analizar serán Kettle y GeoKettle, las cuales son libres y gratuitas. El análisis se lo hará tomando en cuenta las funciones geoespaciales que posean, de allí saldrá la herramienta que sea más apropiada.

3.5.1. Análisis de Kettle

Kettle es una herramienta ETL, lo cual significa que, como toda herramienta ETL, recibe datos de entrada en diferentes formatos, los transforma y genera una o varias salida en los formatos deseados; teniendo además la particularidad, que no se requiere de programación, sino de utilización y configuración de componentes ya programados.

Kttle está destinado para un uso general, siendo los plugins más útiles los que leen desde bases de datos. Sin embargo, no se encontró un plugin útil para lectura de un archivo con información geoespacial, el único disponible es ESRI Shapefile Input pero genera muchas filas con información duplicada.



Tampoco disponía de algún plugin destinado a generar WKT, algún conversor de sistemas de coordenadas o un visualizador de figuras. Por lo tanto, si se decide realizar el plugin en este entorno, la transformación de coordenadas, generación de WKT y la lectura de archivos SHP debería estar incluida en la funcionalidad del plugin.

3.5.2. Análisis de GeoKettle

Debido a las limitaciones de Kettle con respecto al manejo de geometrías, se procedió a analizar GeoKettle, el cual es una extensión de Kettle, pero destinado a realizar operaciones con figuras. Los plugins que dispone, cumplen funciones que en Kettle sería necesario desarrollarlas, como por ejemplo:

- Plugin para lectura de archivos SHP sin generar duplicidad de datos.
- Plugin para la transformación de coordenadas.
- Se podía obtener WKT automáticamente en el plugin que lee SHP.
- Se puede previsualizar la figura que se leyó.

Entonces, debido a que una buena parte de la funcionalidad que se necesita, está ya desarrollada, se optó por utilizar esta opción y así enfocarse totalmente en la generación del plugin de RDF.

3.6. DISEÑO DEL PLUGIN

Para la realización del plugin, se tomó en cuenta los datos de entrada, los cuales son cada una de las entidades geométricas, y como resultado del plugin, se obtendrá el RDF, mismo que será compatible con GEOSPARQL. Entonces, se realizará primero un esquema de entrada-salida del plugin, para posteriormente integrarlo en otro esquema donde se separe más el proceso de generación de RDF, ya que algunos pasos son complementados mediante el uso de plugins ya existentes.

Dentro de la generación de RDF, además de la generación de tripletas de figuras, se encuentra la obtención de metadatos del archivo SHP, el cual ayudara a dar información adicional al que consulte el RDF. Sin embargo, el RDF que se genere

con los metadatos, debe ser generado aparte del RDF de figuras, para dejar al usuario que use el plugin, la decisión de usar dicha información o no. Finalmente, para la generación de los metadatos, se lo hará en un formato estándar, el ISO 19115.

3.6.1. Esquema funcional del plugin

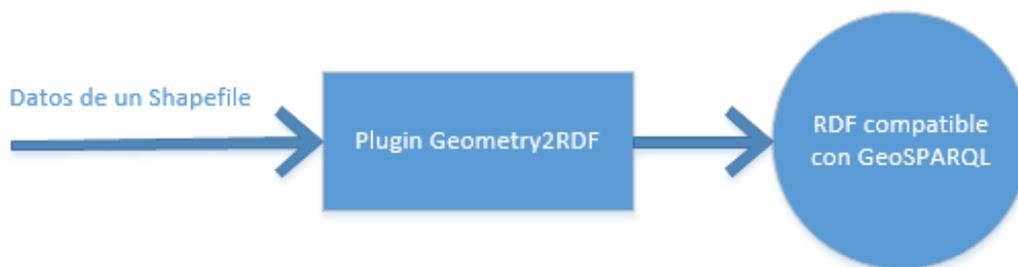


Figura 3.12: Esquema general del plugin

La figura 3.12 muestra la arquitectura del plugin. Los datos de ingreso deben estar en las coordenadas deseadas y además en formato WKT. Para la generación de estos datos de ingreso, se pueden utilizar otros plugins, pero ello requiere dividir más el proceso de generación, como se indicará a continuación.

3.6.2. Esquema de relación entre la herramienta de ETL y el plugin



Figura 3.13: Esquema para la utilización del plugin

Como se muestra en la figura 3.13 primero se realiza la lectura de archivo SHP, que da como resultado el WKT de entidad geográfica; posteriormente se realiza la transformación de coordenadas donde el usuario indica con que sistema de coordenadas se trabajará, y finalmente el proceso que genera RDF, el cual recibirá el WKT en un determinado sistema de coordenadas, y generará los RDF correspondientes tanto



de las figuras como de los metadatos (siguiendo el estándar ISO 19115). Como parte opcional, se puede dar un tratamiento a los RDFs generados, como por ejemplo, que se guarden cada uno en archivos diferentes.

3.7. DESARROLLO, IMPLEMENTACIÓN Y PRUEBAS DEL PLUGIN

En el desarrollo del plugin, se basó en la aplicación mejorada de Geometry2RDF, donde se tomó fracciones de código, para posteriormente adaptarlas dentro del plugin; posteriormente se realiza una aplicación del esquema del plugin, colocando en cada fase del proceso el plugin respectivo; y finalmente se realiza las pruebas respectivas para obtener un RDF.

3.7.1. Integración del Plugin con la herramienta de ETL

En el Anexo E se describe el proceso para la creación del plugin y su instalación.

3.7.2. Pruebas

Para la realización de pruebas, se usó un archivo SHP de provincias. A continuación se describirá el proceso de configuración para generar el RDF del shapefile.

Primero, se estableció una variable (clic derecho sobre GeoKettle y se elige la opción *Transformation Settings*) donde irá la ruta del archivo, como muestra la Figura 3.14.

Posteriormente, en la configuración de las coordenadas, se estableció el sistema de coordenadas a EPSG:4326, seleccionándolo de la lista de la derecha tal y como muestra la Figura 3.15.

Para la obtención del RDF, en el plugin Geometry2RDF se estableció la configuración de la Figura 3.16, en la cual se indica que el campo descriptivo (en este caso, los nombres de cada provincia) será *DPA_DESPRO*, que el nombre de la clase sea *provincia*, que tenga como URIs de ontología y recursos a *geo.linkeddata.ec*, que tenga como prefijo de la URI de la ontología *geontology*, con la URI indicada que

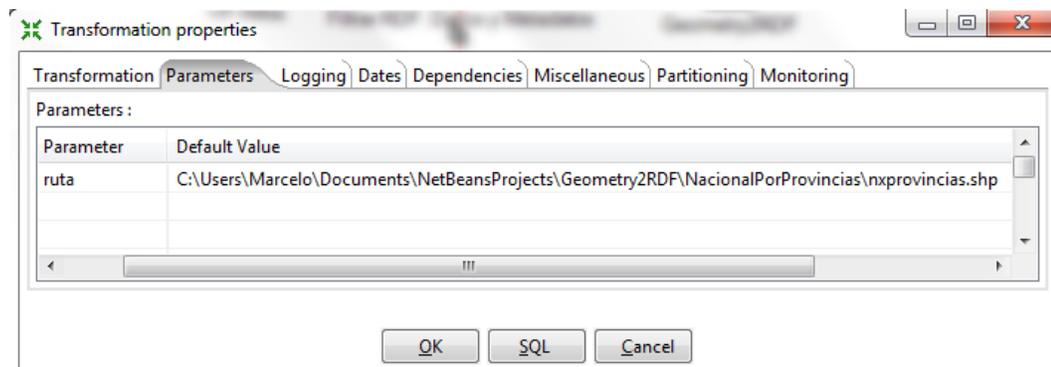


Figura 3.14: Configuración de la ruta del shapefile

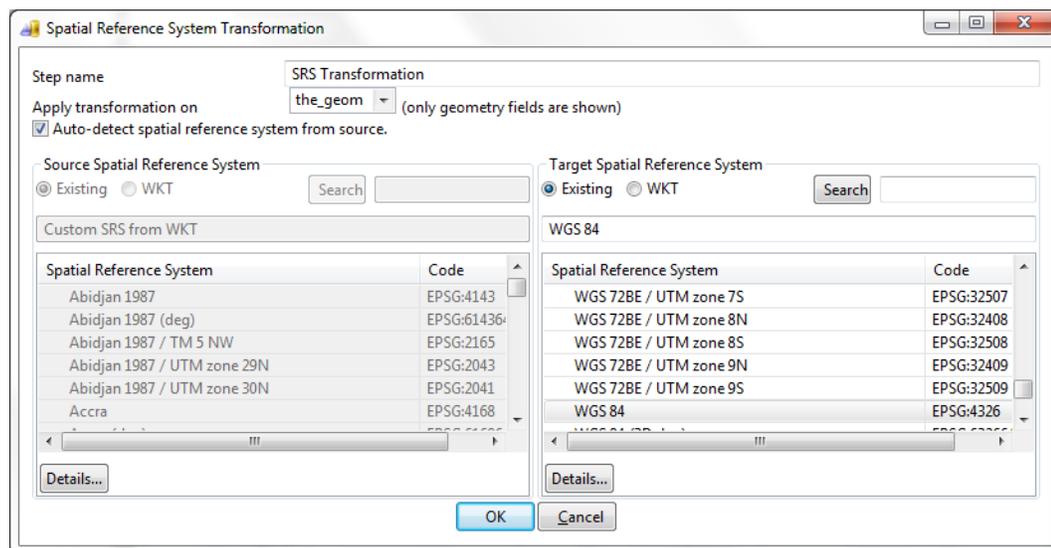


Figura 3.15: Configuración del Sistema de Coordenadas

es estándar para el sistema de coordenadas, que estará en idioma español (indicado de manera corta: es) y que el campo donde se encuentra la ruta del archivo SHP es *ruta*.

3.7.3. Resultados

Al realizar el proceso de generación, se obtuvieron dos archivos:

1. *datos.rdf*: Contiene el RDF de las geometrías del archivo SHP. El contenido de este archivo se muestra en la Figura 3.17, el cual se puede notar parte

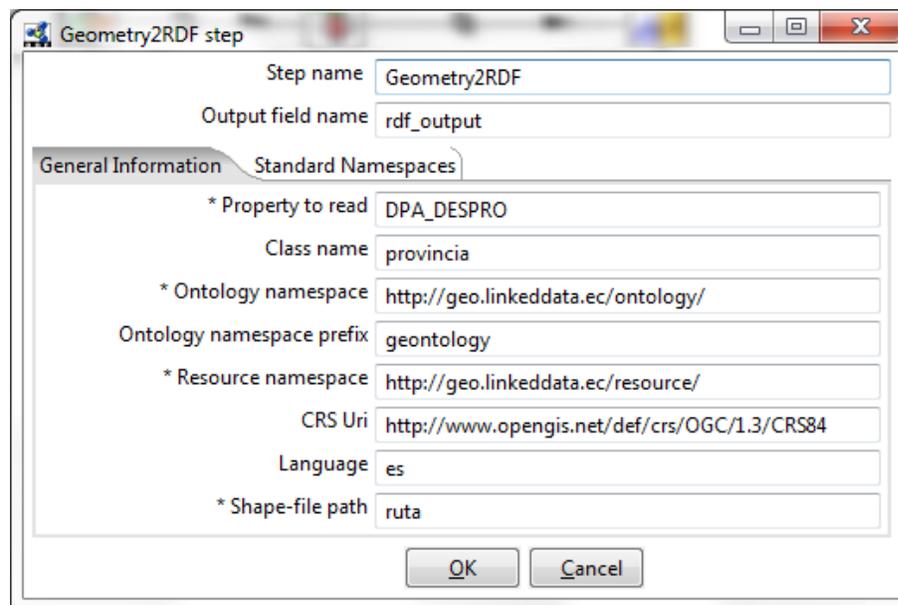


Figura 3.16: Configuración RDF para una prueba

de la representación de la provincia de Manabí mediante un Multi-polígono en formato WKT. Se puede observar además, dónde van ubicados cada dato ingresado en la configuración del plugin, como la URI para la ontología, la URI de los recursos y la URI del sistema de coordenadas.

2. *metadata.rdf*: Contiene el RDF de los metadatos, siguiendo el formato estándar ISO 19115. el contenido del archivo se indica en la Figura 3.18, donde se puede observar casi al final, que el idioma de los datos del RDF está en español (escrito de manera corta: es), que el nombre del archivo leído tiene de nombre *nprovincias.shp* y que además tiene dentro de la descripción del archivo, el sistema de coordenadas en el que está.

3.8. INTRODUCCIÓN A MAP4RDF

MAP4RDF es un software open source que necesita ser configurado para usar cualquier SPARQL endpoint y que proporciona a los usuarios una visualización de datos RDF georeferenciados en un mapa, a través de una aplicación web, la cual

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://geo.linkeddata.ec/resource/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:sf="http://www.opengis.net/ont/sf#"
  xmlns:geontology="http://geo.linkeddata.ec/ontology#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:geo="http://www.opengis.net/ont/geosparql#">
  <rdfs:Class rdf:about="http://geo.linkeddata.ec/ontology#provincia">
    <rdfs:subClassof rdf:resource="http://geo.linkeddata.ec/ontology#Feature"/>
  </rdfs:Class>
  <rdf:Property rdf:about="http://geo.linkeddata.ec/ontology#hasExactGeometry">
    <rdfs:subPropertyof rdf:resource=
      "http://geo.linkeddata.ec/ontology#hasDefaultGeometry"/>
    <rdfs:subPropertyof rdf:resource=
      "http://geo.linkeddata.ec/ontology#hasGeometry"/>
  </rdf:Property>
  <rdf:Property rdf:about="http://www.opengis.net/ont/geosparql#aswkt"/>
  <rdf:Property rdf:about="http://www.opengis.net/ont/sf#Multipolygon"/>
  <geontology:provincia rdf:about=
    "http://geo.linkeddata.ec/resource/provincia/MANABI15e666e7e54a14fcab7e27459a31
    be7595585467">
    <geontology:hasExactGeometry>
      <sf:Multipolygon rdf:about=
        "http://geo.linkeddata.ec/resource/provincia/15e666e7e54a14fcab7e27459a31be7595
        585467">
        <geo:aswkt rdf:datatype="http://www.opengis.net/ont/sf#wktLiteral"
          >&lt;http://www.opengis.net/def/crs/OGC/1.3/CRS84&gt; MULTIPOLYGON
          (((-80.83116820036211 -1.683295345370231, -80.83149545043048
```

Figura 3.17: Parte del RDF generado de provincias

que permite al usuario seleccionar un tipo de recurso para visualizarlos, además la aplicación realiza la búsqueda a través de consultas SPARQL al servidor que previamente debe haber sido cargado con información semántica (RDF).

En la figura 3.19 se presenta la arquitectura general de MAP4RDF, donde se indica las tecnologías aplicadas en la herramienta para lograr el consumo y visualización de datos RDF.

3.9. ANÁLISIS DE FUNCIONALIDAD PREVIO A CAMBIOS DE MAP4RDF

En esta sección se realiza el respectivo análisis de MAP4RDF, donde se indica cada una de las características de la aplicación así como los errores y las limitaciones encontradas. En la figura 3.20 se presenta el esquema general de trabajo de MAP4RDF. Para la visualización de los recursos la aplicación consulta a través de SPARQL o GEOSPARQL, posteriormente se extrae la información geométrica de

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:md="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  <owl:Ontology rdf:about="http://geo.linkeddata.ec/resource/metadata/d6f63b6db11fa44effa
  <owl:imports rdf:resource="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata
  </owl:Ontology>
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#D
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#M
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#i
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#B
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#g
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#f
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#f
  <rdf:Property rdf:about="http://def.seegrid.csiro.au/isotc211/iso19115/2003/metadata#l
  <md:Metadata rdf:about="http://geo.linkeddata.ec/resource/metadata/d6f63b6db11fa44effa
  <md:language rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >es</md:language>
  <md:identificationInfo>
  <md:DataIdentification rdf:about="http://geo.linkeddata.ec/resource/metadata/d6f63b
  <md:graphicoverview>
  <md:BrowseGraphic rdf:about="http://geo.linkeddata.ec/resource/metadata/d6f63b
  <md:fileName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >nxprovincias.shp</md:fileName>
  <md:fileDescription rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WGS_1984_UTM_Zone_17S
```

Figura 3.18: Parte del RDF con los metadatos del shapefile de provincias

los recursos que se encuentran en formato WKT para finalmente se grafican a través de *OPENLAYERS*.

A continuación se indica las funcionalidades que ofrece MAP4RDF:

- Visualización Geométrica usando *Google Maps*, *Open Street Maps* y *Open Layers*
- Filtrado de Consultas
- Conexión a un SPARQL Endpoint para la consulta de información.
- Fácil configuración para el usuario usando el archivo de configuración.
- Visualización Textual de Resultados

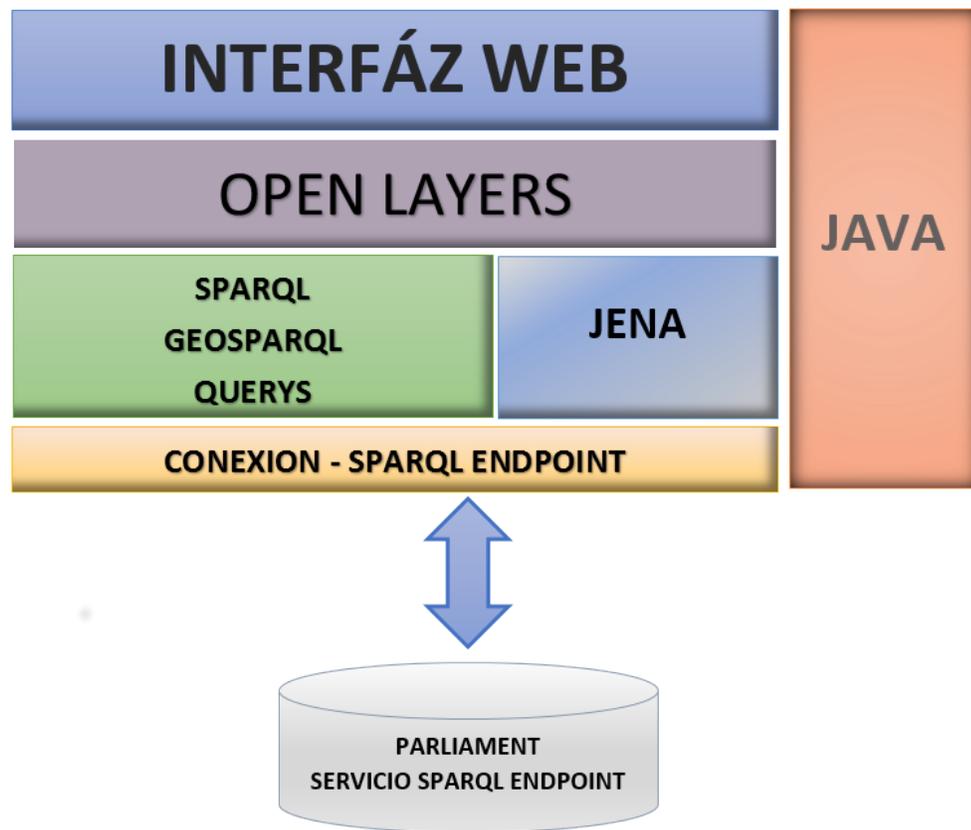


Figura 3.19: Arquitectura general de MAP4RDF



Figura 3.20: Esquema general de la funcionalidad de MAP4RDF

3.9.1. Limitaciones

MAP4RDF presenta las siguientes Limitaciones:

- No soporta gráfico de información geométrica (Puntos, Lineas o Polígonos) en formato WKT.
- No soporta consultas GEOSPARQL.



- Limite máximo de resultados de consulta no es configurable por el usuario
- Limite máximo de resultados de consulta es 1000.

3.10. ANÁLISIS DE FUNCIONALIDAD A CORRREGIR Y EXTENDER A MAP4RDF

En esta sección se realiza el análisis para la corrección de errores y limitaciones identificadas, además se describe las nuevas funcionalidades que se implementaron como parte del proceso de desarrollo de la presente tesis.

3.10.1. Identificación de errores a corregir

Al realizar el respectivo análisis de la versión sin cambios de la herramienta se encontró los siguientes errores.

- Desactivada la función para filtrado de resultados
- Limite máximo de resultados de consulta no es controlable por el usuario

3.10.2. Descripción de la o las nuevas funcionalidades requeridas

- Consultas SPARQL de los recursos con información geográfica en formato WKT
- Soporte de gráfico de Puntos Lineas y Polígonos del formato WKT
- Soporte de consultas GEOSPARQL.
- Filtrado de Información aplicando GEOSPARQL



3.11. DESARROLLO Y VALIDACIÓN DE FUNCIONALIDAD A CORREGIR Y EXTENDER DE MAP4RDF

Es esta sección se describe las implementaciones de las funcionalidades corregidas y nuevas funcionalidades de MAP4RDF.

3.11.1. Corrección de errores identificados en MAP4RDF

Los errores identificados en la sección “Identificación de errores a corregir” han sido corregidos previo a la modificación e inserción de nuevas funcionalidades requeridas. El error “Desactivada la función para filtrado de información” ha sido corregido con éxito, pues, la línea de código que llamaba a esta función estaba inhabilitada como se muestra en la figura A.1 de la sección ANEXOS.

El error “Limite máximo de resultados de consulta no es controlable por el usuario” ha sido corregido, se considero un error porque esta versión del MAP4RDF si soporta la consulta y visualización de una mayor cantidad de resultados. En la figura A.3 se muestra que el valor de limite máximo de consulta ahora se encuentra en el archivo de configuración “configuration.properties” para que el usuario pueda modificarlo según sus necesidades.

3.11.2. Validación de errores corregidos

En la figura A.2 de la sección ANEXOS se muestra la validación del error “Desactivada la función para filtrado de resultados ” ya corregido, la aplicación nuevamente permite filtrar los resultados según el recuadro dibujado por el usuario.

3.11.3. Implementación de nuevas funcionalidades requeridas en MAP4RDF

Se ha implementado cada una de las nuevas funcionalidades como se muestra en la tabla 3.3.



Funcionalidad Requerida	Método de implementación
Consultas SPARQL de los recursos con información geográfica en formato WKT	Para esta funcionalidad se modificó las funciones en las que se construye la cadena de texto de la consulta SPARQL que se envía al Endpoint. Modificaciones necesarias en varias funciones en la capa de acceso a datos para recuperar la información del EndPoint, una de las principales modificaciones es la recuperación de la propiedad "Figure" que representa la geometría del recurso en el formato WKT
Soporte de gráfico de Puntos Lineas y Polígonos con formato WKT	Se modifíco la función "GetGeometry" y "GetPoint" y se agrego funciones como "GetMultiPolygon", "GetMultiLinestring" para recuperar el tipo de geometría del WKT devuelto por el servidor y posteriormente graficar los recursos. Se modifíco la función "draw" y se agrego nuevas funciones como "drawMultipolygon" "drawMultilinestring", para que desde el formato WKT se grafique el recurso a través de <i>OpenLayers</i> .
Soporte de consultas GEOSPARQL	Se estableció 5 consultas GEOSPARQL predefinidas para que el usuario escoja. Se agrego funciones como "createGetGeoSparqlResourceQuery" que construyen la cadena de texto para cada una de las consultas GEOSPARQL. Cuando se obtiene los resultados de la consulta construida y enviada al servidor, se continua con las mismas funciones de obtención de geometría y de graficado mencionadas anteriormente.
Filtrado de información aplicando GEOSPARQL	Se modifíco una función que permite filtrar información de los resultados consultados. La condición de filtrado es de acuerdo a un recuadro que el usuario gráfica en el mapa. Debido a que los datos que se están consultando soportan el estándar GEOSPARQL, para el filtrado de los datos ahora se aplica una función GEOSPARQL y con pocas lineas de código se obtiene el mismo resultado mediante el análisis de las geometrías.

Cuadro 3.3: Funcionalidades aplicadas en MAP4RDF

3.11.4. Validación de nuevas funcionalidades implementadas

En la tabla 3.4 se indican las respectivas figuras que muestran la validez de cada funcionalidad implementada.

Funcionalidad	Figura
Consultas SPARQL de los recursos con información geográfica en formato WKT	En la figura A.4 y A.5 de A de la sección ANEXOS se observa secciones de código que se agregaron para esta funcionalidad.
Soporte de gráfico de Puntos Lineas y Polígonos del formato WKT	Recuperación de los tipos de Geometrías: Figuras A.6, A.7 y A.8 de A de la sección ANEXOS. Gráfico de Recursos: Figuras A.9, A.10 y A.11 de A de la sección ANEXOS
Soporte de consultas GEOSPARQL	Figura A.12 y A.13 de A de la sección ANEXOS
Filtrado de Información aplicando GEOSPARQL	Se ha modificado parte del código de filtrado para esta funcionalidad, la figura A.2 de A muestra el resultado de el filtrado, filtrando polígonos en este caso.

Cuadro 3.4: Validaciones de Nuevas Funcionalidades MAP4RDF

En la figura 3.21 se muestra el gráfico de los aeropuertos del Ecuador, en la cual se ha aplicado un filtro para mostrar a los aeropuertos que se encuentran adentro del recuadro dibujado, de esta manera se muestran las funcionalidades de MAP4RDF, graficando recursos con información geoespacial en formato WKT aplicando consultas GEOSPARQL.

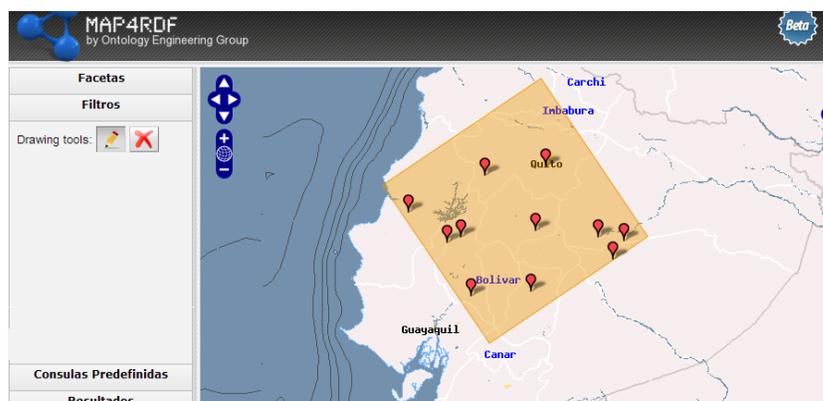


Figura 3.21: Gráfico de los Aeropuertos del Ecuador aplicando filtro



4. IMPLEMENTACIÓN DE LA METODOLOGÍA PARA LA GENERACIÓN DE DATOS ENLAZADOS GEOESPACIALES ECUATORIANOS

En este capítulo se describe el respectivo proceso metodológico para la publicación de Datos Enlazados Geospaciales. Además se detalla cada actividad que permite la generación Datos Enlazados Geospaciales, herramientas y tecnologías aplicadas para el cumplimiento del objetivo principal de la presente tesis.

4.1. METODOLOGÍA PARA LA PUBLICACIÓN DE DATOS ENLAZADOS

En esta sección se describe la metodología para la publicación de Datos Enlazados, la cual es el punto de partida para la generación de Datos Enlazados Geospaciales; además, aplicando los principios de Datos Enlazados descritos en la sección 2.1.5, permite la creación del prototipo de repositorio de Datos Enlazados Geospaciales del Ecuador.

La Metodología para la generación de Datos Enlazados consta de cinco actividades como se muestra en la figura 4.22 y, a continuación se describe cada una de ellas.

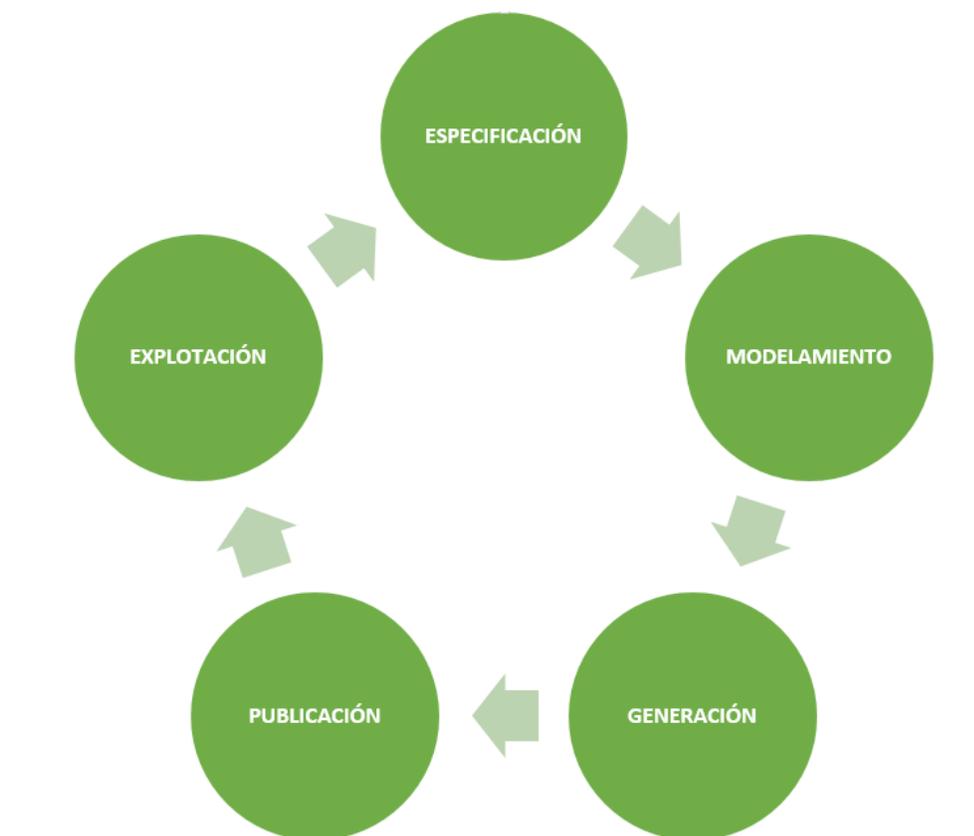


Figura 4.22: Proceso de Publicación de Linked Data. Fuente [16]

4.1.1. ESPECIFICACIÓN

Esta actividad consiste en la determinación, obtención y análisis de las fuentes de información. En esta actividad también se deben definir las URI's cuyo objetivo principal es identificar los recursos y cada una de sus propiedades.

4.1.2. MODELAMIENTO

En esta actividad se define la ontología con la que se va a trabajar, en este caso se trabajó con el estándar GEOSPARQL que define un modelo de datos con información geoespacial, donde también se estableció utilizar el formato WKT para definir la geometría de cada recurso proveniente de las fuentes de información.



4.1.3. GENERACIÓN DE RDF

El objetivo principal de esta actividad es generar los archivos RDF una vez establecidas las fuentes de información, URIs y el respectivo modelo ontológico. Luego de haber generado los archivos RDF, se procede a la Generación de Enlaces o Interlinking cuyo propósito es descubrir información relacionada en otras fuentes de datos.

4.1.4. PUBLICACIÓN

El fin de esta actividad es publicar la información de los archivos RDF en un repositorio semántico (Triple Store) brindando acceso a estos datos a través de un Endpoint mediante consultas SPARQL.

4.1.5. EXPLOTACIÓN

Finalmente, la actividad de Explotación de datos describe la manera de como el usuario puede acceder a la información publicada en el Triple Store, ya sea manualmente donde el usuario escribe la consulta o mediante visualización gráfica.

4.2. METODOLOGÍA PARA LA PUBLICACIÓN DE DATOS ENLAZADOS GEOESPACIALES DEL ECUADOR

En esta sección se aplica la metodología para la generación de Datos Enlazados. Siguiendo el proceso descrito en la sección anterior y adaptando una serie de actividades se logra la generación, publicación y consumo de RDF con información geoespacial. En la figura 4.23 se muestra la metodología extendida para la generación de Datos Enlazados Geoespaciales, en la cual se observan las sub-actividades que se han adaptado dentro de la actividad de Generación de RDF (GeoKettle RDF - Plugin, Metadata Generation), también el proceso de Enlace de Datos o Interlinkig se ha desarrollado como una actividad independiente debido a que la actividad de Generación de RDF describe únicamente la creación de los archivos RDF desde el

Plugin y la actividad Interlinking describe el proceso de relacionar datos desde los archivos RDF generados anteriormente.

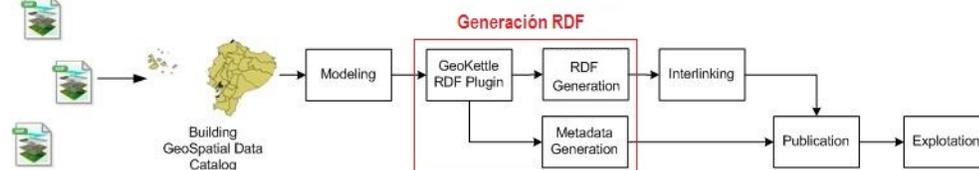


Figura 4.23: Metodología extendida para la generación de Datos Enlazados Geoespaciales

A continuación se describe a detalle cada una de las actividades desarrolladas para la generación de Datos Enlazados Geoespaciales.

4.2.1. ESPECIFICACIÓN

Análisis de las Fuentes de Información

Este proceso permite tener un conocimiento claro de las fuentes de información, es el punto de partida para todo el proceso de generación de Datos Enlazados Geoespaciales. Para ellos se ha realizado la búsqueda manual en internet, usando buscadores Web se logro encontrar archivos de formato shapefile alojados en paginas Web de instituciones publicas e instituciones privadas del Ecuador mismas que se mencionan en la sección C, obteniendo así una gran cantidad de archivos que fueron analizados para el respectivo proceso de publicación de Datos Enlazados Geoespaciales. Además, se utilizó este formato debido a la gran cantidad de información digital geográfica existente en este formato. Como se explicó en la sección 2.4 los archivos shapefile almacenan tanto información de entidades espaciales como puntos, líneas y polígonos y atributos o información temática sobre provincias, ciudades, entidades educativas, centrales eléctricas y otros tipos de información geoespacial del Ecuador como se ilustra en la figura 4.24. La geometría que se encuentra en estos archivos en formato WKT es la que se utiliza en el proceso para el cumplimiento de toda la metodología antes mencionada.

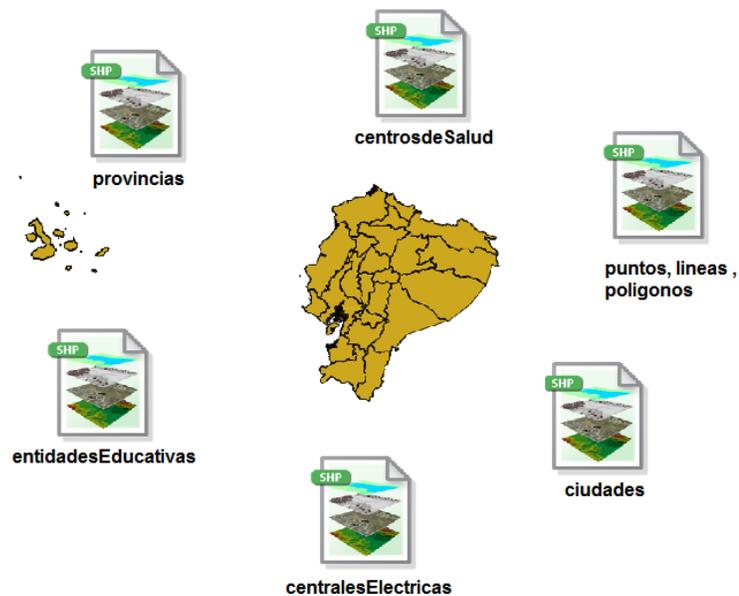


Figura 4.24: Ilustración de la Información GeoEspacial del Ecuador en los Archivos Shapefile

Diseño de URIS

La definición del formato de los Identificadores de las instancias (URI) es de mucha importancia previo a la generación de los archivos RDF, puesto que estas permiten identificar inequívocamente los elementos o recursos que han sido generados y publicados en formato RDF. Para el diseño de las URIS de este proyecto se adoptan las recomendaciones de [1] que serán descritas a continuación:

- La raíz de las URI será: *http://geo.linkeddata.ec*
- La URI para la identificación de los recursos modelados en la ontología se presenta en la tabla 4.5.

Patron	http://geo.linkeddata.ec/ontology/class
Ejemplo	http://geo.linkeddata.ec/ontology/provincia

Cuadro 4.5: URIS para identificación de recursos



- En la tabla 4.6 se muestran las URIs para identificar los recursos asociados a los datos.

Patron	http://geo.linkeddata.ec/resource/tipo/nombre del recurso+GeometryHash
Ejemplo	http://geo.linkeddata.ec/resource/Centro Educativo/ANTONIO %20BORRERO4052ca02e881aef868ee56d0b3f76708ac6378fa

Cuadro 4.6: URIS para identificación de los recursos asociados a los datos

- Las URIS para representar la información geométrica conforme a GeoSparql implementada en el formato WKT y en el sistema de referencia WGS84, se presentan en la tabla 4.7.

Patron	http://geo.linkeddata.ec/resource/GeometryHash
Ejemplo	http://geo.linkeddata.ec/resource/4052ca02e881aef868ee56d0b3f76708ac6378fa

Cuadro 4.7: URIS para representar la información geométrica

4.2.2. MODELAMIENTO

La ontología seleccionada para trabajar con información geoespacial, es GEOSPARQL, el cual es un estándar para el almacenamiento de información geoespacial en formato RDF y que permite además la realización de consultas sobre esos datos, integrando además ciertas funciones especializadas en geometrías.

La figura 4.25 muestra un extracto de la ontología de GEOSPARQL, donde se observa en primera instancia la herencia de clases. Las clases usadas más importantes son: la clase *Point* (representación de puntos) y la clase *Polygon* (representación de polígonos). Además, las líneas circulares que posee la clase *SpatialObject* son propiedades que se aplican a sí mismo, por ejemplo la propiedad *intersects* que permite conocer si dos geometrías se llegan a cruzar. Al ser las clases *Point* y *Polygon*, hijas de *SpatialObject*, también heredan sus propiedades, así que cualquier polígono o punto, también tienen su propiedad *intersect*.

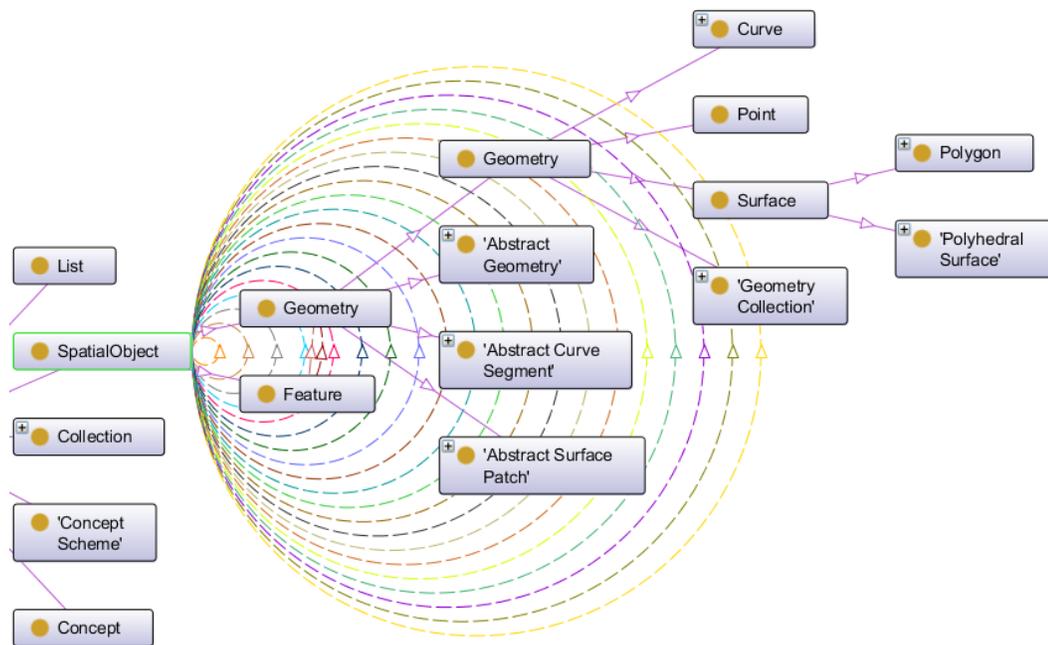


Figura 4.25: Ontología GeoSPARQL vista desde Protege

No todas las clases de la ontología GEOSPARQL son usadas ya que no han sido necesario. Para tener una mejor idea de las clases utilizadas, está la Figura 4.26 donde se muestra un ejemplo de una ontología que tiene un punto y un polígono, y que además, muestra los atributos que se usan entre las diferentes clases. Se dispuso un gráfico con colores diferentes para resaltar lo siguiente:

- Verde: clases o atributos propios del estándar GEOSPARQL.
- Naranja: clases del estándar de figuras simples. GEOSPARQL los usa para la representación de geometrías.
- Azul: clases o datos definidos por el usuario. En el caso de la clase principal y sus instancias, tienen un formato estándar (ya que heredan de la clase Feature); pero en el caso de las etiquetas, es un dato adicional no estándar (ya que GEOSPARQL no lo menciona) pero que es de mucha utilidad ya que dota de significado a la geometría.

Para el caso práctico, las etiquetas (ver Figura 4.26) serán llenadas con la información que contiene una determinada columna, de preferencia donde cada dato de la columna

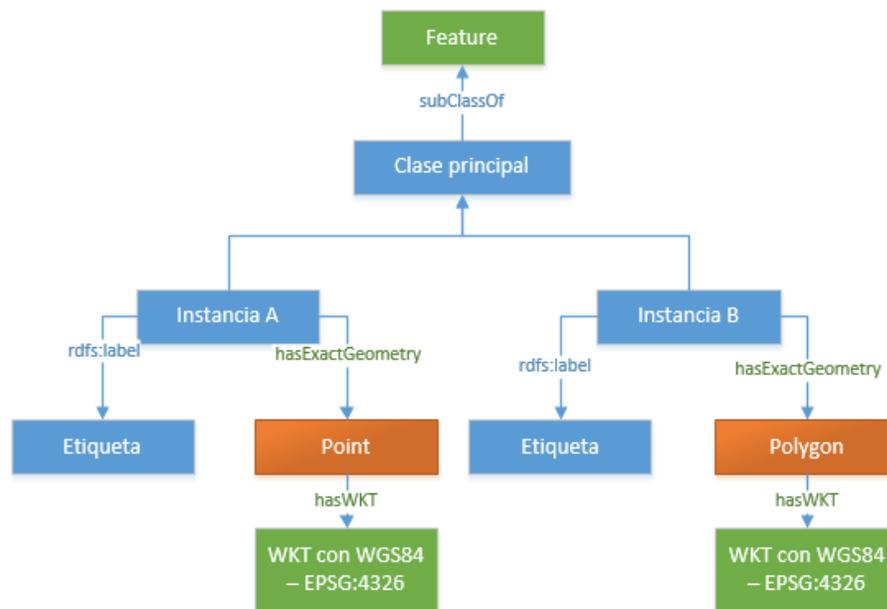


Figura 4.26: Modelo de ejemplo, aplicando la ontología GEOPARQL

sea único. Por ejemplo, si se trabaja con shapefiles de provincias, la columna clave sería el nombre de las provincias, puesto que son únicos. Se podría igualmente elegir alguna columna que no aporte mucho significado, como por ejemplo, un ID, pero eso no ayudaría a conocer a simple vista con qué dato se está trabajando.

Para el caso de los metadatos, se utilizó el ISO-19115 (Geographic Information - Metadata) el cual define una gran cantidad de información la cual en gran parte la llenaría el usuario, por esta razón y para evitar el crear en el plugin demasiados campos, se utilizó solo algunos, que pueden ser llenados automáticamente. La ontología usada se la puede ver en la Figura 4.27. El único dato que pide es el idioma, que debe estar en formato corto (por ejemplo: *es* para Español, *en* para Inglés, *it* para Italiano, etc). El significado de los colores son los mismos que los utilizados para mostrar la ontología del RDF.

4.2.3. GENERACIÓN DE RDF

Esta sección especifica la generación de los archivos RDF a través de un proceso ETL. Para la transformación del shapefile a archivo *.rdf, se utiliza la herramienta

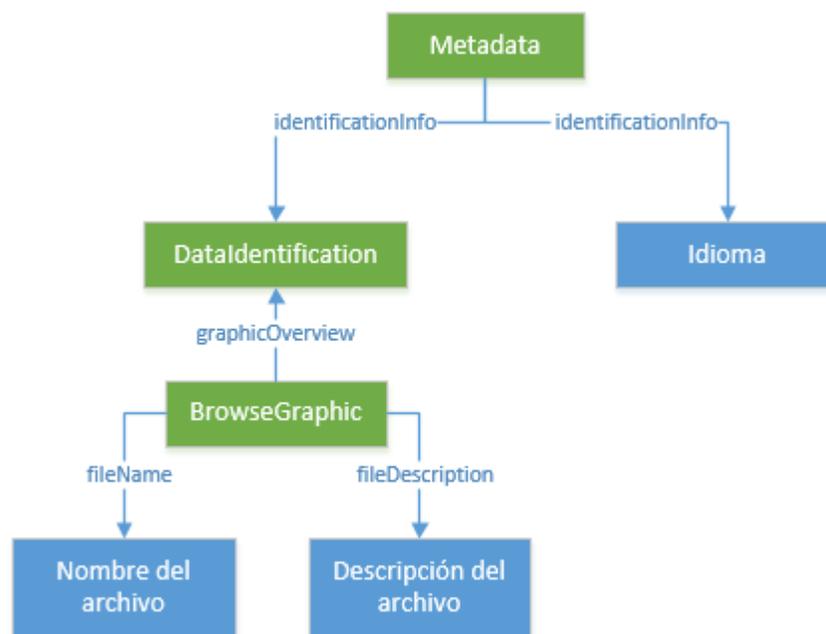


Figura 4.27: Ontología usada para los metadatos, es parte del ISO-19115

GeoKettle, para ello, se configura una transformación para que la generación de los archivos rdf se realice de una manera simple utilizando algunos plugins existentes en la herramienta y también el plugin Geometry2RDF.

Este plugin fue creado previamente y configurado para que sea utilizable en GeoKettle, este proceso está descrito en el Anexo E.

En la Figura 4.28 se representa gráficamente el estado final de la transformación para la generación del RDF, esta transformación ha sido configurada de forma que recibe como entrada el archivo shapefile, posteriormente se configura el sistema de coordenadas con el cual se trabaja, luego se genera los datos en formato RDF y finalmente estos datos se envían a la salida que es un archivo con extensión RDF.

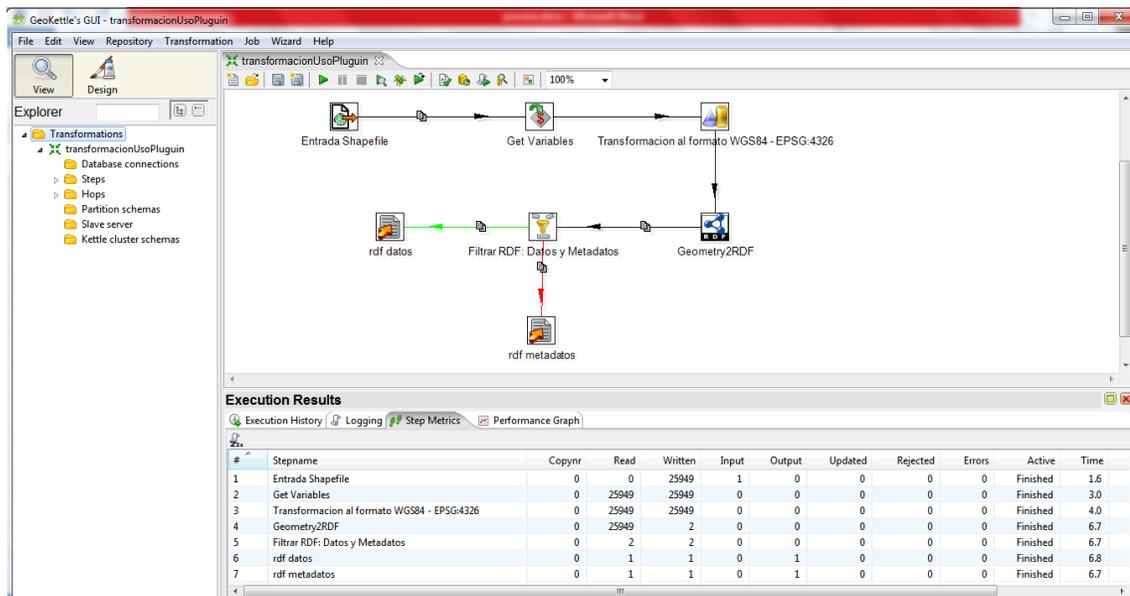
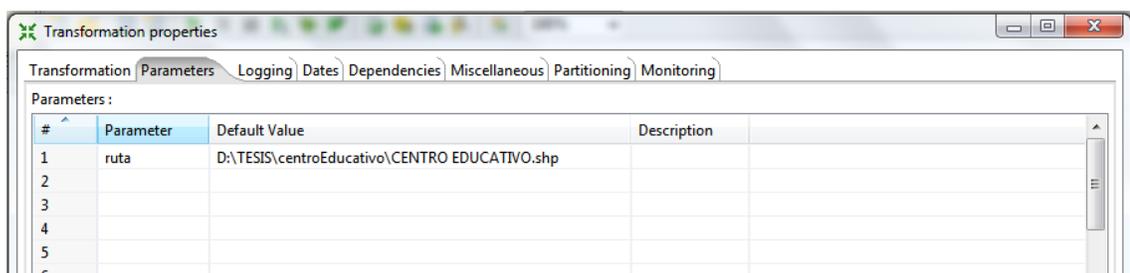


Figura 4.28: Configuración de la Transformación para Generar RDF

A continuación se describe la configuración de cada uno de los componentes utilizados.

Especificar la ruta el archivo de entrada Shapefile

En las propiedades de la transformación en la sección de parámetros se escribe el nombre de una variable y como valor se especifica la ruta en donde se encuentra el archivo shapefile como se muestra en la Figura 4.29.



#	Parameter	Default Value	Description
1	ruta	D:\TESIS\centroEducativo\CENTRO EDUCATIVO.shp	
2			
3			
4			
5			
6			

Figura 4.29: Especificación de la ruta del archivo Shapefile

Uso y configuración del componente “Shape File Input”

Para la configuración del componente “Shape File Input” se presiona doble click sobre el mismo y se observa una ventana de propiedades en donde se debe especificar la variable que contiene la ruta del archivo en el campo “File name” como se observa en la figura 4.30. Este componente permite recuperar la información del archivo shapefile

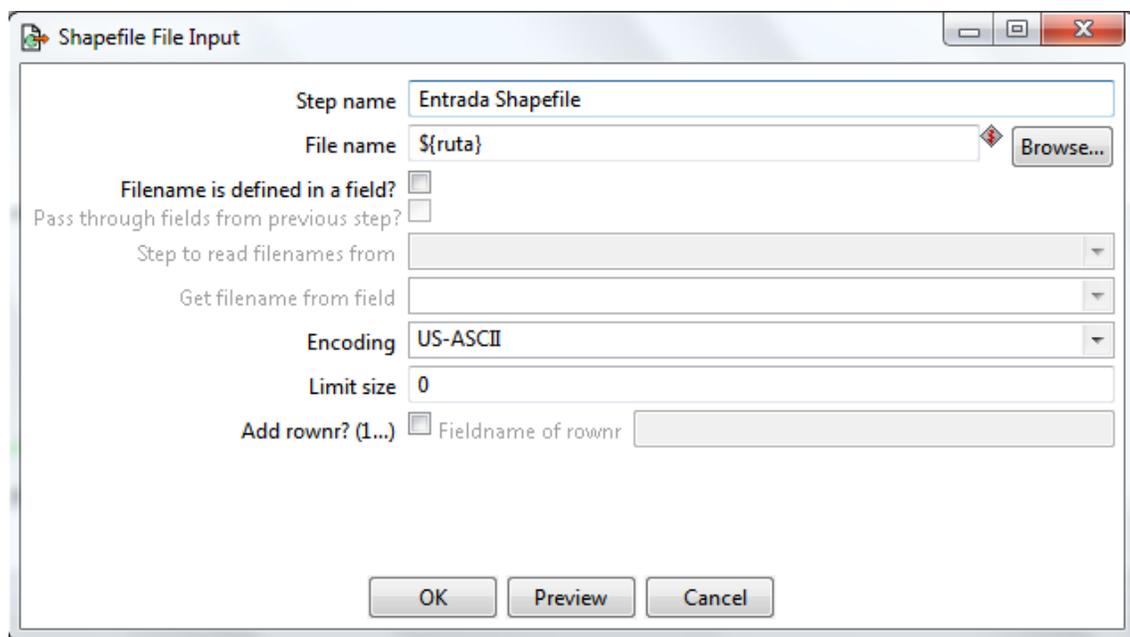


Figura 4.30: Uso y configuración del componente “Shape File Input”

Uso y configuración del componente “Get Variables”

Este componente permite recuperar el path o la ruta del Shapefile que fue almacenada en una variable llamada “ruta”, se presiona doble click sobre el componente y se especifica las propiedades del componente como se observa en la figura 4.31. El path recuperado del Shapefile es necesario para utilizarla en el componente “Geometry2RDF” que será explicado posteriormente.

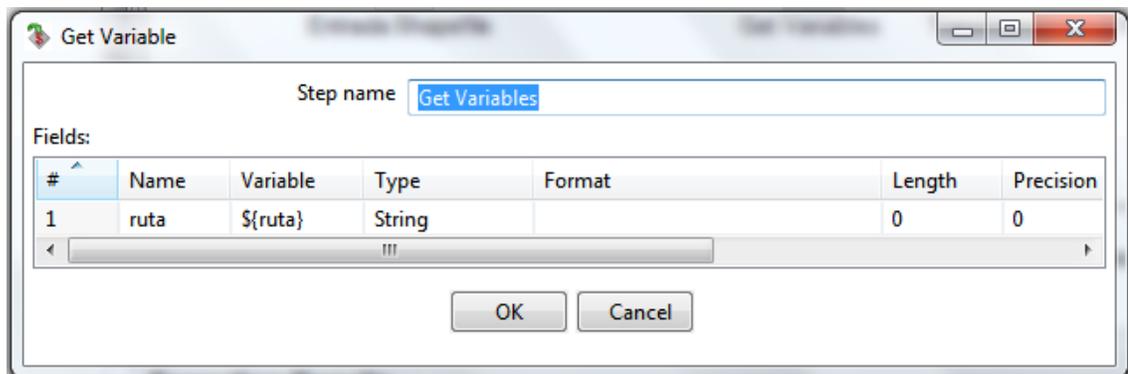


Figura 4.31: Uso y configuración del componente “Get Variables”

Uso y configuración del componente “SRS Transformation”

Este componente permite seleccionar el sistema de coordenadas al que se quiere transformar la geometría leída desde el archivo shapefile. Para esto, al presionar doble click sobre este componente se observa una ventana de propiedades en donde se debe seleccionar el campo “the_geom”, el cual contiene la geometría de las figuras y luego se debe seleccionar el sistema de coordenadas “WGS84 EPSG:4326” como se indica en la figura 4.32.

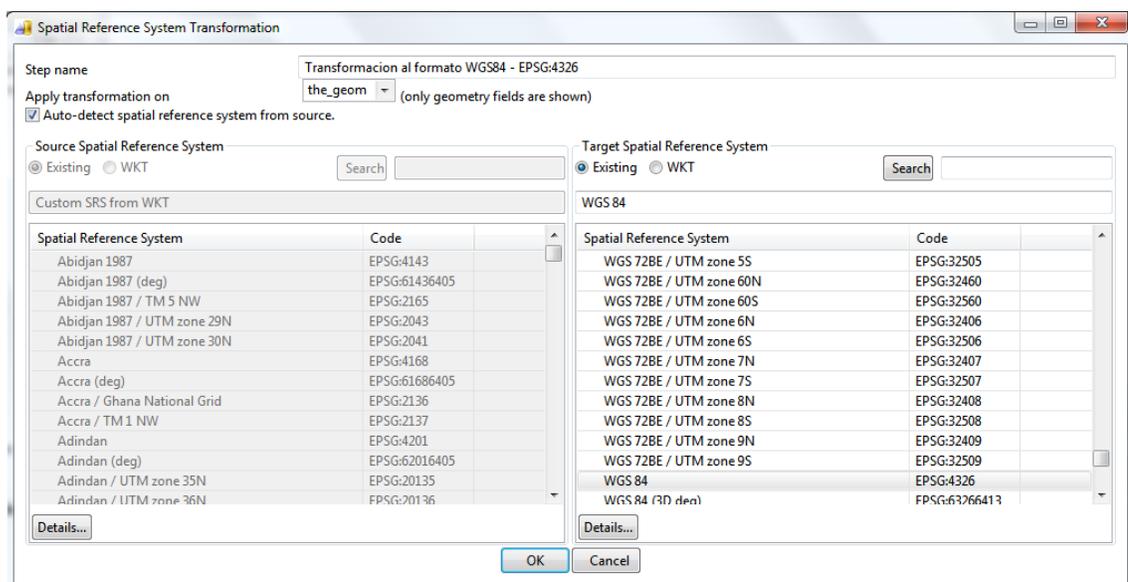


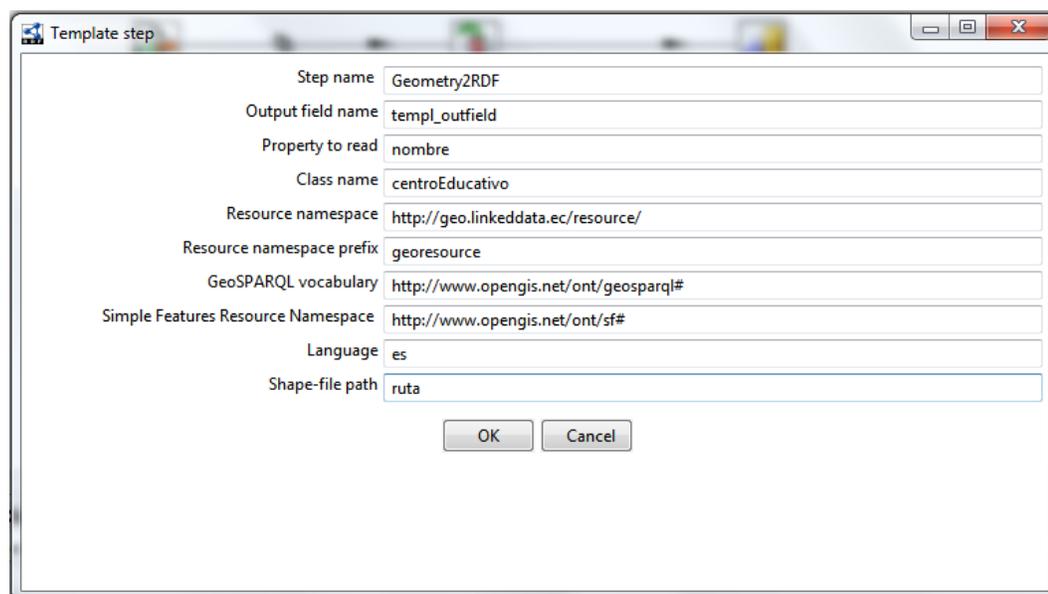
Figura 4.32: Uso y configuración del componente “SRS Transformation”

Uso y configuración del componente “Geometry2RDF”

En un capítulo anterior se habló sobre este componente, ahora se especifica su uso como parte de la metodología de generación de Datos Enlazados Geoespaciales.

Este componente se encarga de la generación del archivo RDF con la información obtenida desde el archivo shapefile, a continuación se describe cada uno de los campos que deben ser editados al momento de configurar este componente, previamente, al presionar doble click sobre el componente aparece una ventana con los campos como se observa en la figura 4.33.

- Campo “Property to read” : permite especificar el nombre del campo proveniente del shapefile que determinará la etiqueta (label) de cada recurso generado en el archivo RDF, en este caso “nombre”.
- Campo “Class name” : se especifica el nombre de la clase de entidades que serán generadas en el archivo RDF, en este caso “centroEducativo”.
- Campo “Shape-file-path”: se debe especificar el nombre de la columna que contiene el path del shapefile.



Step name	Geometry2RDF
Output field name	templ_outfield
Property to read	nombre
Class name	centroEducativo
Resource namespace	http://geo.linkeddata.ec/resource/
Resource namespace prefix	georesource
GeoSPARQL vocabulary	http://www.opengis.net/ont/geosparql#
Simple Features Resource Namespace	http://www.opengis.net/ont/sf#
Language	es
Shape-file path	ruta

Figura 4.33: Uso y configuración del componente “Geometry2RDF”

Uso y configuración del componente “Filter Rows”

Puesto que el componente “Geometry2RDF” genera dos filas de información, donde cada fila contiene la información de archivos .rdf de datos y de metadatos, por esta razón se necesita filtrar para finalmente obtener los dos archivos .rdf por separado. Como se observa en la figura 4.34 se verifica que en una de las filas no contenga el texto “iso9115” puesto que solo un archivo .rdf de los dos generados contiene ese texto.

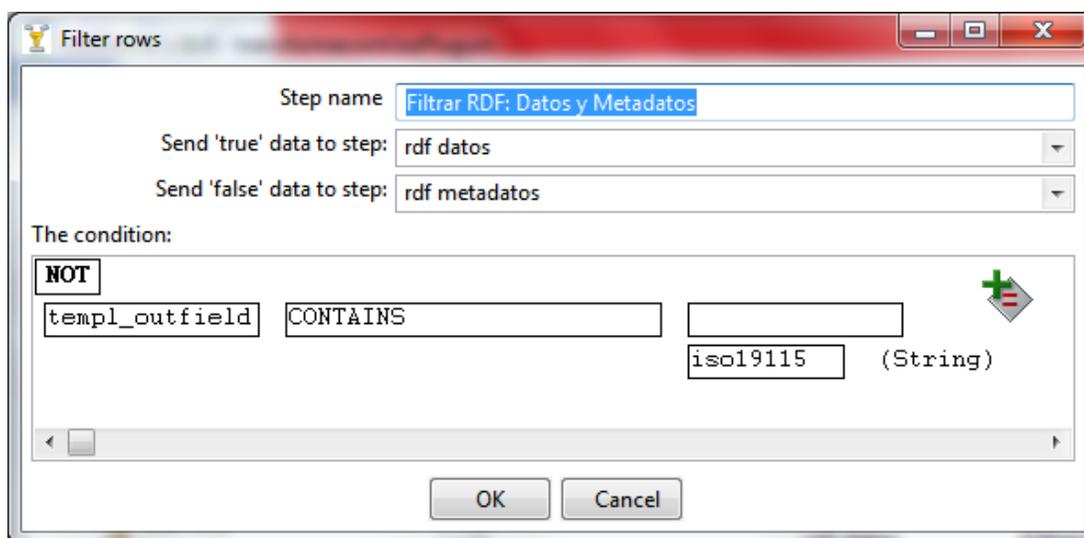


Figura 4.34: Uso y configuración del componente “Filter Rows”

Uso y configuración del componente “Text File Output”

En este componente se especifica el nombre y la ruta del archivo de salida en el campo “Filename” y la extensión del archivo de salida que será en formato *.rdf en el campo “Extension”

En las figuras 4.35 y 4.36 se muestran las configuraciones para el archivo de salida RDF con información de los datos y metadatos respectivamente provenientes del shapefile.

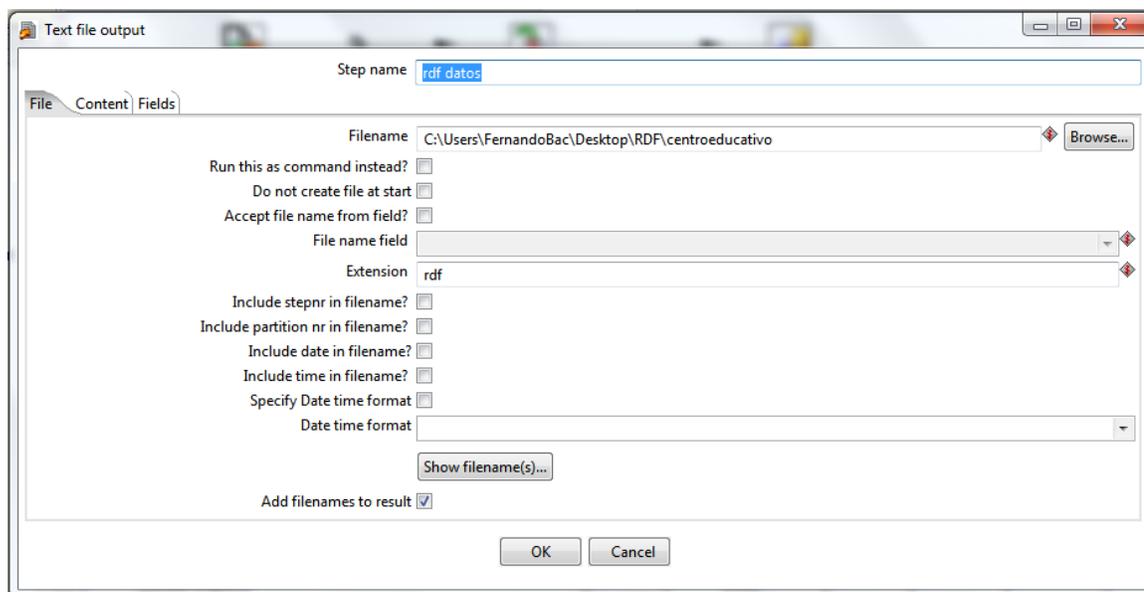


Figura 4.35: Configuración de archivo de salida rdf de Datos

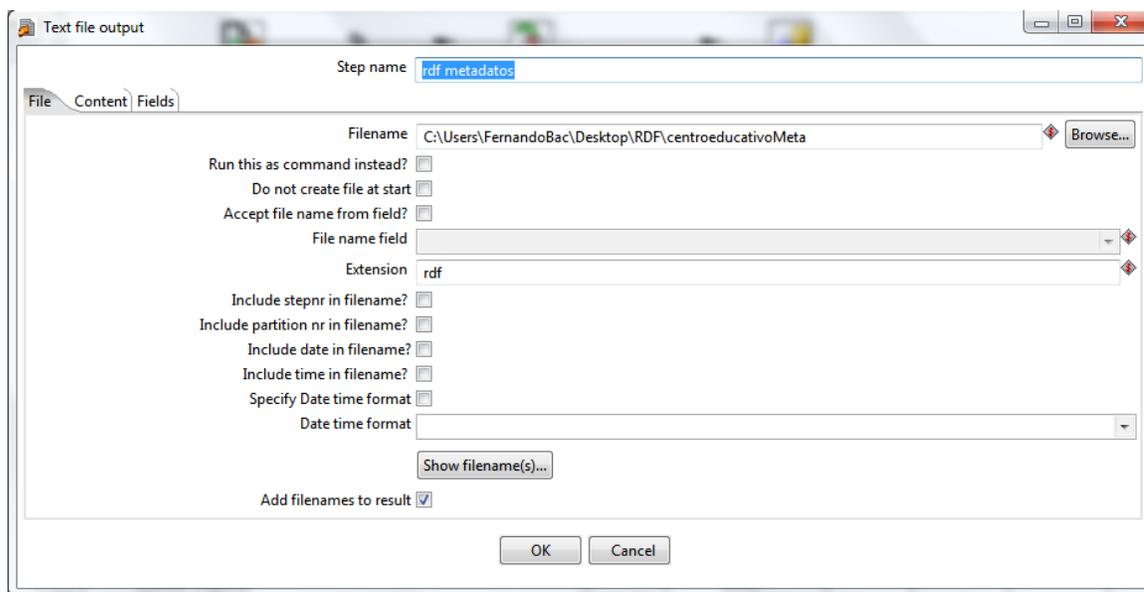


Figura 4.36: Configuración de archivo de salida rdf de MetaDatos

4.2.4. ENLACE DE DATOS

En esta sección, se muestra el enlazado de datos. Este proceso permite encontrar definiciones del mismo recurso en diferentes fuentes. Para ello se trabajó con la

herramienta SILK Workbench y a continuación se muestra el respectivo proceso. Inicialmente como se muestra en la figura 4.37 se debe especificar las fuentes de datos, un archivo RDF o un SPARQL Endpoint, en este caso las fuentes son “dbpedia” y “nxprovincias”, posteriormente se especifica la tarea, en este caso “ProvinciasTask”, para obtener las propiedades de cada fuente de información mediante condiciones, además se especifica que las coincidencias serán indicadas bajo la propiedad “sameAs”.

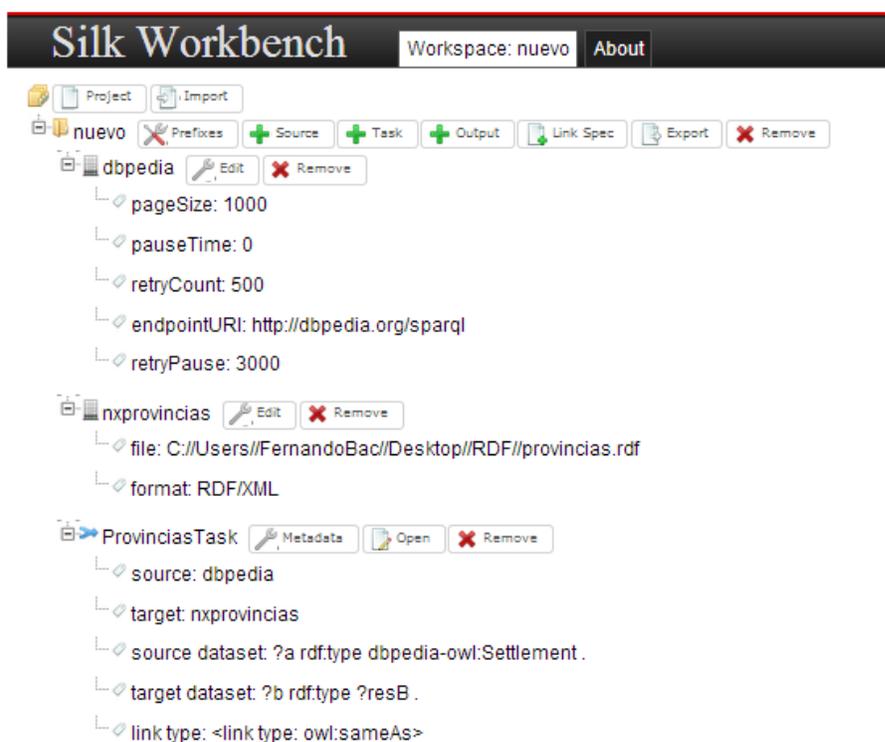


Figura 4.37: Silk. Configuración de Fuentes de Datos

Luego de la configuración inicial se especifica los parámetros bajo los cuales se desea buscar las coincidencias. Como se observa en la figura 4.38, del lado izquierdo se tiene un menú donde están varios componentes que permiten configurar la comparación. Inicialmente la sección “source” y la sección “Target” muestran las clases y propiedades leídas de las dos fuentes de información, se selecciona la propiedad que sera

comparada, en este caso “Label”; también de la sección “Transformations” se utiliza el componente “Lower Case” que permite la transformación a minúsculas de la palabra que será comparada, posteriormente de la sección “Comparators” se selecciona el algoritmo de comparación con el cual la aplicación buscara las coincidencias, en este caso el algoritmo seleccionado es “Levenshtein distance” que permite una rápida comparación con la modalidad de inserción o reemplazo de caracteres y finalmente se muestran los resultados.

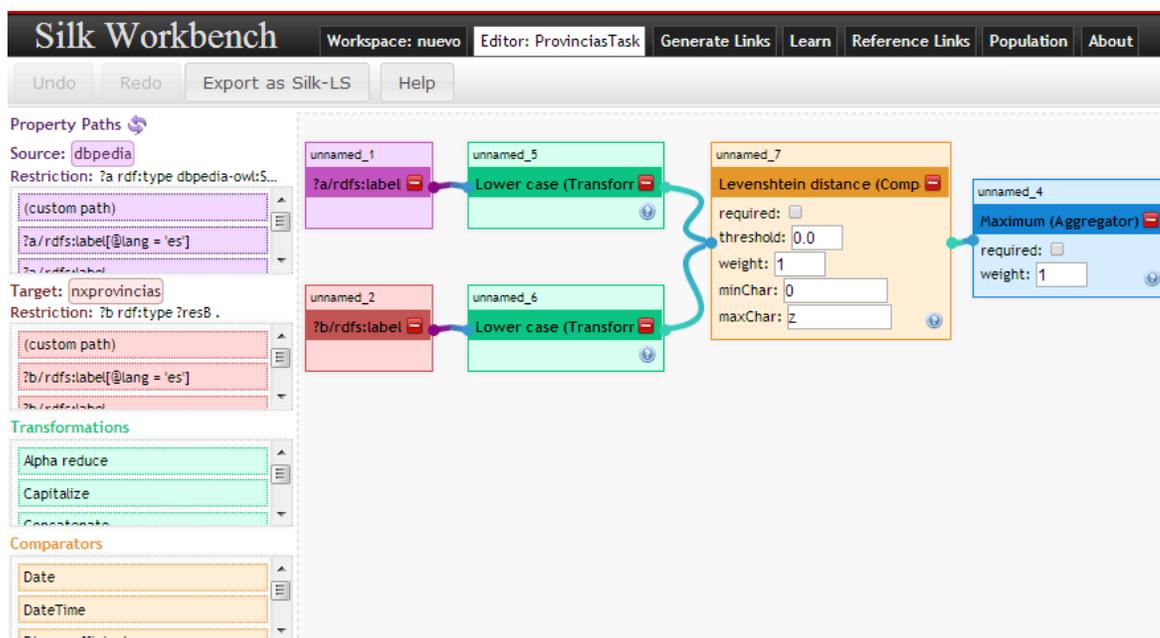


Figura 4.38: Silk. Configuración de condiciones de enlazado

Finalmente en las figuras 4.39 y 4.40 se observa los resultados de enlaces obtenidos de las diferentes fuentes de datos. En donde se debe seleccionar cuales enlaces son correctos y cuales son incorrectos. Esta validación se hace manualmente para cada enlace generado, el usuario lee y verifica cada enlace, si el usuario duda o no tiene conocimiento sobre el tema analizado se puede basar en apoyo externo para su correcta validación. Se validó los enlaces entre fuentes locales con información de Ecuador como se observa en la figura 4.39 , también se validó correctamente enlaces que se generaron con la DBpedia como se observa en la figura 4.40.

Silk Workbench Workspace: PROVINCIAS Editor: linkProvincias Generate Links Learn Reference Links About

Positive Negative Import Export Download Help

Expand All Collapse All Prev 1 Next Filter:

Source	Target	Score	Status
source/provincias/LOS%20RIOSec16945798314f19723833d10	source/provincias/LOS%20RIOSec16945798314f19723833d10	100,0%	correct
Aggregation: max (unnamed_6) 100,0% Comparison: levenshteinDistance (unnamed_5) 100,0% Transform: lowerCase (unnamed_3) los rios Input: ?a<http://www.w3.org/2000/01/rdf-schema#label> (unnamed_1) LOS RIOS Transform: lowerCase (unnamed_4) los rios Input: ?b<http://www.w3.org/2000/01/rdf-schema#label> (unnamed_2) LOS RIOS			
source/provincias/CA%20CA%91ARe55e0881827e5240ebe5fddb77e	source/provincias/CA%20CA%91ARe55e0881827e5240ebe5fddb77e	100,0%	correct
source/provincias/LOJAe48e9c0c1effb056454672def	source/provincias/LOJAe48e9c0c1effb056454672def	100,0%	correct
source/provincias/PICHINCHAd068767f92c6518cebb17dc552	source/provincias/PICHINCHAd068767f92c6518cebb17dc552	100,0%	correct
source/provincias/EL%20ORO7f17c80e0ddecf0d1d692da528f	source/provincias/EL%20ORO7f17c80e0ddecf0d1d692da528f	100,0%	correct
source/provincias/ORELLANA15ab1e676720ae778b4a399d32	source/provincias/ORELLANA15ab1e676720ae778b4a399d32	100,0%	correct
source/provincias/AZUAY1d381dc1668fe66a925efd5f321	source/provincias/AZUAY1d381dc1668fe66a925efd5f321	100,0%	correct

Figura 4.39: Silk. Resultados de Enlaces entre fuentes locales

Silk Workbench Workspace: PROVINCIAS Editor: linkProvincias Generate Links Learn Reference Links About

Positive Negative Import Export Download Help

Expand All Collapse All Prev 1 Next Filter:

Source	Target	Score	Status
dbr:Morona-Santiago_Province	_inec/MORONA%20SANTIAGOc031365809c299b0dd058b14	100,0%	correct
dbr:Azuary_Province	resource/provincias_inec/AZUAY1d381dc1668fe66a925efd5f321	100,0%	correct
Aggregation: max (unnamed_6) 100,0% Comparison: levenshteinDistance (unnamed_5) 100,0% Transform: lowerCase (unnamed_3) provincia de azuay, provincia azuay, provinz azuay, 阿苏艾省, azuay, provincia di azuay, アスアイ県, azuay province Input: ?a<http://www.w3.org/2000/01/rdf-schema#label> (unnamed_10) Azuay, 阿苏艾省, Provinz Azuay, Azuay Province, Prowincja Azuay, Transform: lowerCase (unnamed_4) azuay Input: ?b<http://www.w3.org/2000/01/rdf-schema#label> (unnamed_9) AZUAY			
dbr:El_Oro_Province	ce/provincias_inec/EL%20ORO7f17c80e0ddecf0d1d692da5	100,0%	correct
dbr:Orellana_Province	ce/provincias_inec/ORELLANA15ab1e676720ae778b4a399d32	100,0%	correct
dbr:Carchi_Province	source/provincias_inec/CARCHI2794f6e67c6d517adb30ccaf	100,0%	correct
dbr:Imbabura_Province	source/provincias_inec/IMBABURAf67a234e1551025f42019b:	100,0%	correct
dbr:Guayas_Province	source/provincias_inec/GUAYAS87a19ed44bac8ff787cd3461	100,0%	correct

Figura 4.40: Silk. Resultados de Enlaces con la DBPEDIA

4.2.5. PUBLICACION

Una vez generado los RDF se procede a cargar estos archivos a un Triple Store, como se ilustra en la figura 4.41, el mismo que debe cumplir la condición de soporte del estándar GEOSPARQL y también debe ofrecer el servicio SPARQL Endpoint para que el usuario pueda realizar las respectivas consultas.

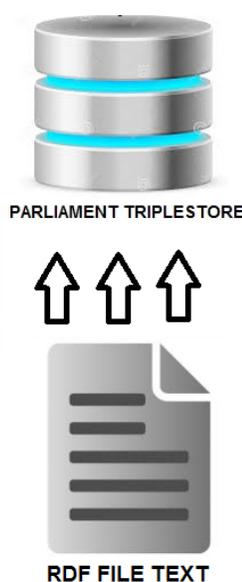


Figura 4.41: Carga de archivo RDF al servidor Parliament

Selección del Triple Store para RDF

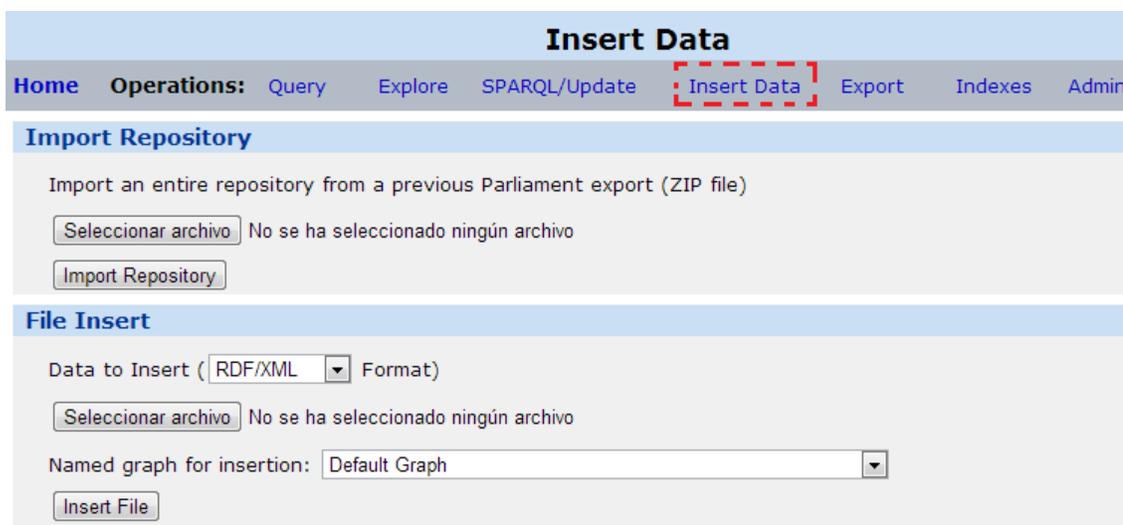
Debido a que en el presente trabajo para la generación de Datos Enlazados Geoespaciales se optó por el estándar GEOSPARQL, el servidor de tripletas debe cumplir con la principal función de soporte de este modelo de datos semántico. Luego de un análisis de cada uno de los servidores posibles a utilizar, entre ellos Virtuoso, Parliament, Fuseki entre otros, se determinó que Parliament es el único servidor que actualmente soporta el mencionado estándar para la publicación y aplicación de funciones GEOSPARQL para la consulta de información.

Instalación y Configuración del Servidor de Tripletas

Parliament es un Triple Store que permite almacenar RDF y de igual manera permite la consulta de esa información a través de su SPARQL EndPoint. Soporta el estándar GEOSPARQL y a través de un manejo de índices espaciales devuelve resultados de consultas de información geoespacial en tiempo aceptables. Se trabajó con la última versión 2.7.4 que está disponible para su descarga en [10], en B de la sección ANEXOS se muestra toda la configuración necesaria de Parliament.

Carga de los archivos RDF al servidor de Tripletas

Una vez instalado y configurado el servidor Parliament es posible la carga de los archivos en formato RDF al servidor. Para ello, al ejecutar Parliament, como se observa en la figura 4.42, en el menú principal se selecciona la opción “Insert Data”, luego en la sección “File Insert” en “Seleccionar Archivo” se agrega el archivo RDF para cargarlo al servidor. En la sección “4.2.6 EXPLOTACIÓN” en la figura 4.44 se muestra un ejemplo para la consulta de información aplicando el lenguaje de consulta SPARQL.



The screenshot shows the Parliament web interface. At the top, there is a navigation bar with the title "Insert Data" and several menu items: Home, Operations, Query, Explore, SPARQL/Update, Insert Data (highlighted with a red dashed box), Export, Indexes, and Admin. Below the navigation bar, there are two main sections: "Import Repository" and "File Insert". The "Import Repository" section has a text input field with a "Seleccionar archivo" button and the text "No se ha seleccionado ningún archivo", and an "Import Repository" button. The "File Insert" section has a "Data to Insert" dropdown menu set to "RDF/XML" with a "Format" label, another "Seleccionar archivo" button with the text "No se ha seleccionado ningún archivo", a "Named graph for insertion:" dropdown menu set to "Default Graph", and an "Insert File" button.

Figura 4.42: Carga de archivo RDF al servidor Parliament

4.2.6. EXPLOTACION

Siguiendo con la metodología del proceso de publicación de Datos Enlazados Geoespaciales, la explotación de los datos se da una vez que éstos se encuentren en el formato RDF y hayan sido cargados al servidor o Triple Store, pues, dicho servidor ofrece un SPARQL Endpoint al que un usuario se puede conectar desde el exterior y así poder consultar la información, este proceso se ilustra en la figura 4.43, el usuario obtiene la información a través de consultas SPARQL o GEOSPARQL.

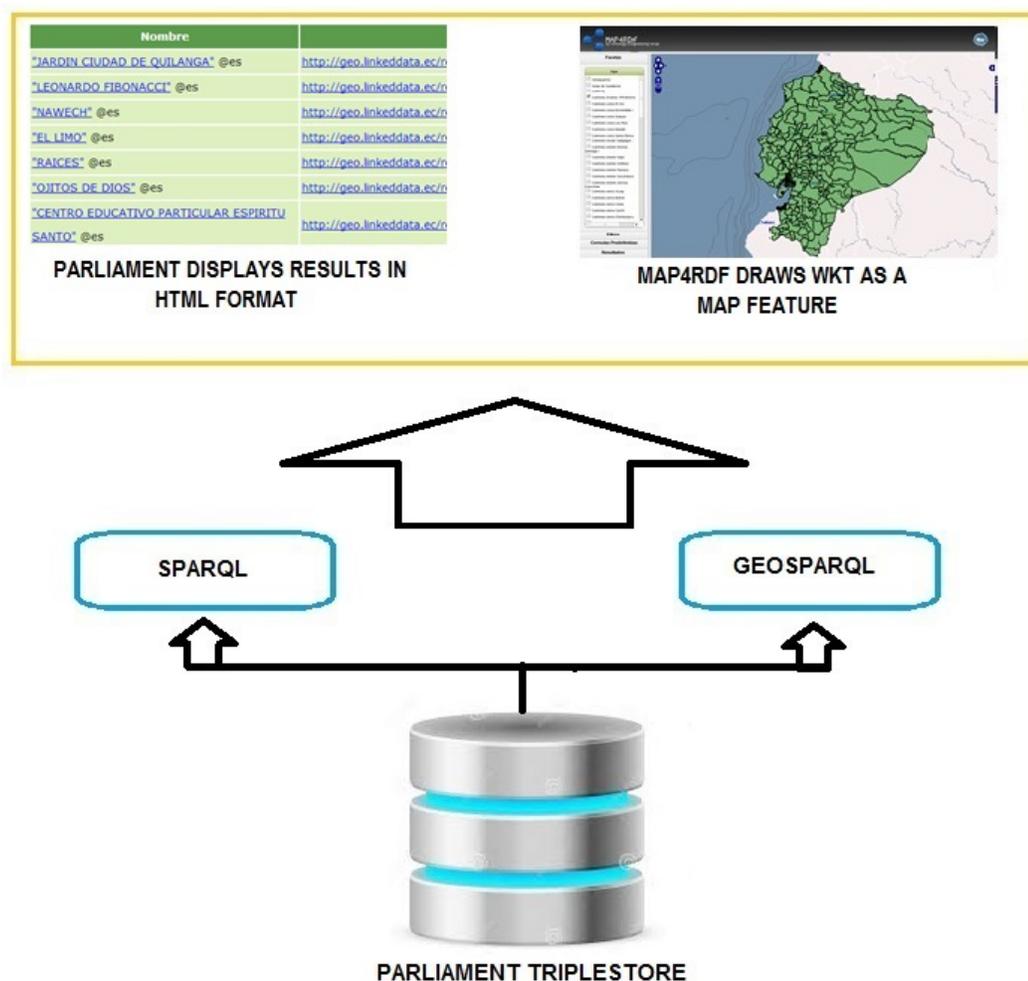
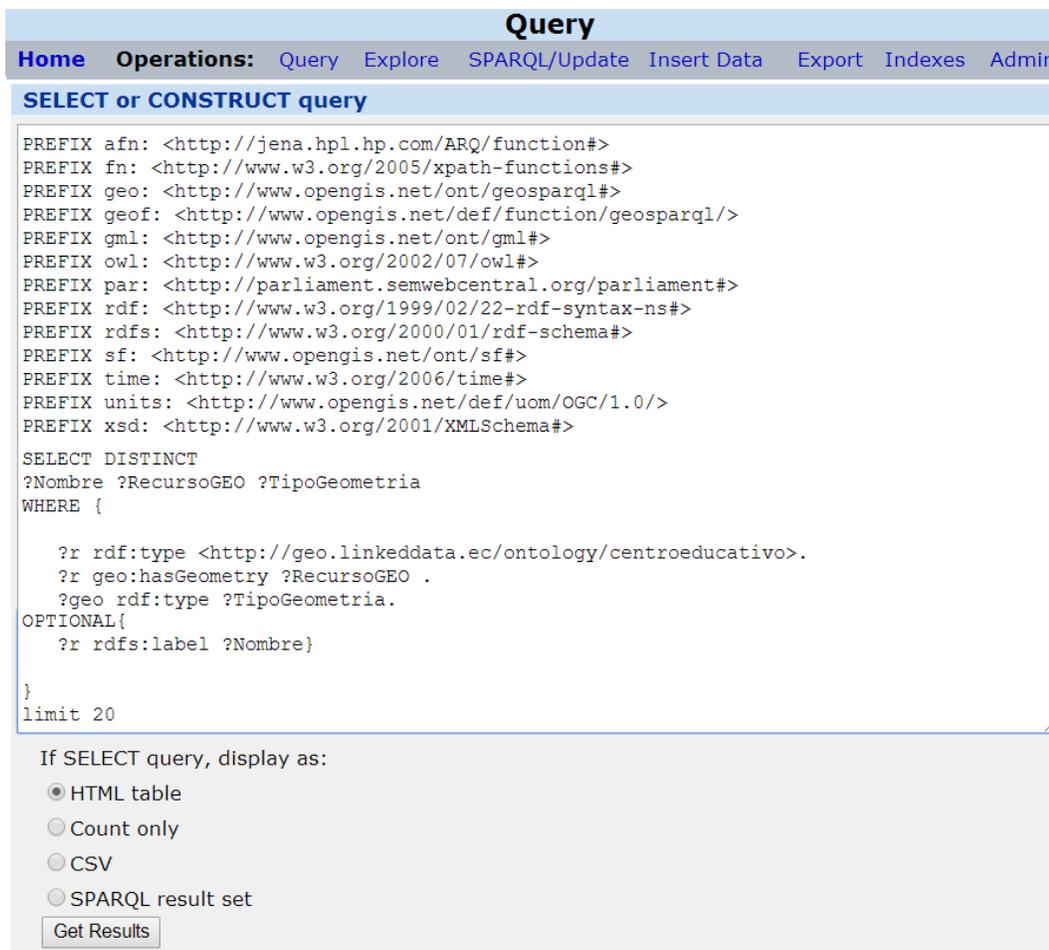


Figura 4.43: Explotación de datos cargados al servidor Parliament

Explotación de información a través del servidor de tripletas

Parliament permite consultar la información que se ha subido previamente. Si se explora el repositorio de Parliament se puede ver toda la información subida desde los archivos *.rdf, dicha información es visualizada mediante tablas, lo que permite al usuario observar la información con mayor facilidad. También se puede hacer consultas sobre la información, utilizando el lenguaje de consulta SPARQL y GEOSPARQL. En la figura 4.44 se observa un ejemplo de consulta SPARQL en el Triple Store Parliament, la consulta permite buscar recursos de tipo “centros educativos”, además se indica que se quiere visualizar el nombre de los recursos “?Nombre”, la geometría “?RecursoGEO” y finalmente el tipo de geometría “?TipoGeometria”.



The screenshot shows the Parliament SPARQL query interface. At the top, there is a navigation bar with links: Home, Operations, Query, Explore, SPARQL/Update, Insert Data, Export, Indexes, and Admin. Below this is a section titled "SELECT or CONSTRUCT query" containing a SPARQL query. The query is as follows:

```
PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX fn: <http://www.w3.org/2005/xpath-functions#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX gml: <http://www.opengis.net/ont/gml#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX par: <http://parliament.semwebcentral.org/parliament#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sf: <http://www.opengis.net/ont/sf#>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT
?Nombre ?RecursoGEO ?TipoGeometria
WHERE {

    ?r rdf:type <http://geo.linkeddata.ec/ontology/centroeducativo>.
    ?r geo:hasGeometry ?RecursoGEO .
    ?geo rdf:type ?TipoGeometria.
OPTIONAL{
    ?r rdfs:label ?Nombre}
}
limit 20
```

Below the query, there is a section titled "If SELECT query, display as:" with four radio button options: HTML table (selected), Count only, CSV, and SPARQL result set. A "Get Results" button is located at the bottom of this section.

Figura 4.44: Ejemplo de Consulta SPARQL en el Triple Store Parliament

Al haber seleccionado la opción “HTML Table” para la visualización de resultados, el usuario obtiene los datos en una tabla HTML como se observa en la figura 4.45, en este caso se muestra el resultado de la consulta indicada anteriormente en la figura 4.44; la información que originalmente contiene el archivo RDF y que ha sido publicado en el Triple Store resulta visualmente amigable para el usuario.

SPARQLer Query Results		
Home	Operations:	Query Explore SPARQL/Update Insert Data Export Indexes Admin
Count: 20		
Nombre	RecursoGEO	TipoGeometria
"JARDIN CIUDAD DE QUILANGA" @es	http://geo.linkeddata.ec/resource/centroeducativo/e489530da796daaf15ca8ffcf	http://www.opengis.net/ont/sf#Point
"LEONARDO FIBONACCI" @es	http://geo.linkeddata.ec/resource/centroeducativo/55d5882548e1e2b6b97c27f	http://www.opengis.net/ont/sf#Point
"NAWECH" @es	http://geo.linkeddata.ec/resource/centroeducativo/402dff6f92bc33714958985a	http://www.opengis.net/ont/sf#Point
"EL LIMON" @es	http://geo.linkeddata.ec/resource/centroeducativo/64fce3abd3f259dfd558404	http://www.opengis.net/ont/sf#Point
"RAICES" @es	http://geo.linkeddata.ec/resource/centroeducativo/6be9dac3e791255e60477bb	http://www.opengis.net/ont/sf#Point
"OJITOS DE DIOS" @es	http://geo.linkeddata.ec/resource/centroeducativo/73ebc9c7a62c41968ce0294	http://www.opengis.net/ont/sf#Point
"CENTRO EDUCATIVO PARTICULAR E	http://geo.linkeddata.ec/resource/centroeducativo/755e8e516d8e0c43c52964d	http://www.opengis.net/ont/sf#Point
"CERRO HERMOSO" @es	http://geo.linkeddata.ec/resource/centroeducativo/354571167cae68f34e33002	http://www.opengis.net/ont/sf#Point
"JOSE OROZCO" @es	http://geo.linkeddata.ec/resource/centroeducativo/0c840d29a026cd0147d5619	http://www.opengis.net/ont/sf#Point
"PATRIA LIBRE" @es	http://geo.linkeddata.ec/resource/centroeducativo/063b94cc3df1e0c9853e5feb	http://www.opengis.net/ont/sf#Point
"BERMEO VASQUEZ" @es	http://geo.linkeddata.ec/resource/centroeducativo/ea30b2cbf038f9729b849cb1	http://www.opengis.net/ont/sf#Point
"TEODORO ANTON VELEZ" @es	http://geo.linkeddata.ec/resource/centroeducativo/62ce2dc41615465503fff0b5	http://www.opengis.net/ont/sf#Point
"INTI PAKARI" @es	http://geo.linkeddata.ec/resource/centroeducativo/78804389fa6b6586872853f	http://www.opengis.net/ont/sf#Point
"ELIAS YUNEZ SIMON" @es	http://geo.linkeddata.ec/resource/centroeducativo/1d1f29b070f5ee89ca37b6ee	http://www.opengis.net/ont/sf#Point
"4 DE NOVIEMBRE" @es	http://geo.linkeddata.ec/resource/centroeducativo/bd46c9149924fe5fb455147	http://www.opengis.net/ont/sf#Point
"ESCUELA MANUEL DE JESUS CALLE"	http://geo.linkeddata.ec/resource/centroeducativo/f6248f4b2af0c24bd951a6c3	http://www.opengis.net/ont/sf#Point
"HARVEST CHRISTIAN SCHOOL" @es	http://geo.linkeddata.ec/resource/centroeducativo/ed5784b0a2b6d83c40b2a2f	http://www.opengis.net/ont/sf#Point
"ZOILA GLORIA JARRE DE VERA" @es	http://geo.linkeddata.ec/resource/centroeducativo/c3eb39a992bb8b0ab78ff3ff	http://www.opengis.net/ont/sf#Point
"EMILIO HIDALGO" @es	http://geo.linkeddata.ec/resource/centroeducativo/61b19e7c379bf48c59e346a	http://www.opengis.net/ont/sf#Point
"MANUEL ADRIAN NAVARRO" @es	http://geo.linkeddata.ec/resource/centroeducativo/b4cd2fdd52ee3ef9506ecd49	http://www.opengis.net/ont/sf#Point

Figura 4.45: Tabla de resultado de consulta

En la figura 4.46 se observa la consulta SPARQL mencionada anteriormente en la figura 4.44 y un frangmento de datos de los resultados devueltos por la consulta indicando la relación que existe entre las variables establecidas en la consulta SPARQL y las columnas del resultado obtenido.

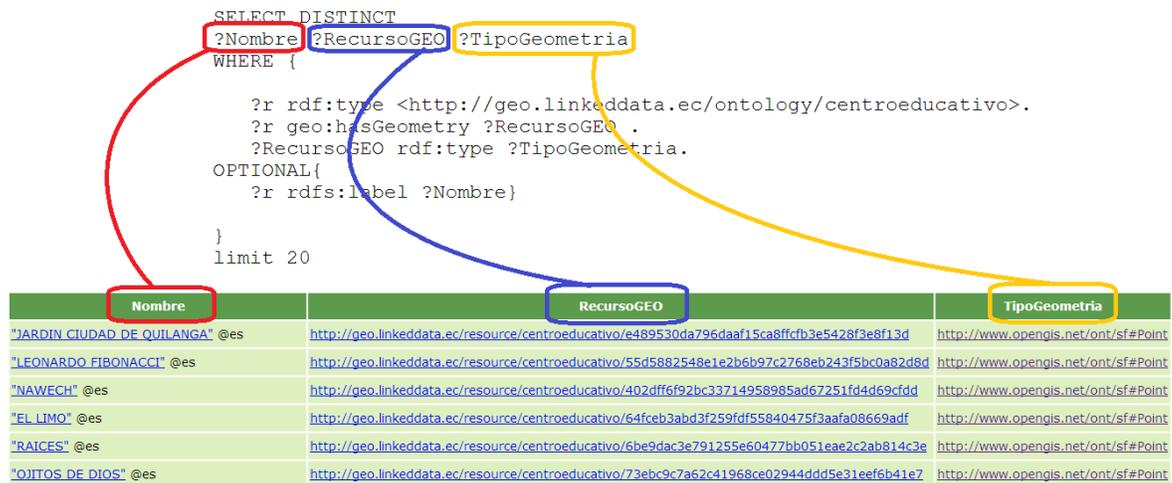


Figura 4.46: Relación Visual entre consulta SPARQL y Resultado

Visualizando Información en MAP4RDF

MAP4RDF es una aplicación Web que permite el gráfico de los recursos con información Geoespacial en formato RDF que previamente ha sido cargada al Triple Store Parliament (servidor que soporta consultas aplicando funciones GEOSPARQL). Esta versión de la aplicación actualmente permite recuperar información geográfica en formato WKT mediante consultas SPARQL y GEOSPARQL y como resultado se obtiene la geometría como puntos, líneas o polígonos así como sus respectivas etiquetas para la identificación textual de cada uno de los recursos y finalmente son graficados en un mapa a través de OpenLayers. El procedimiento para la utilización de esta herramienta es fácil para los usuarios, selecciona el tipo de recursos que desee observar, la aplicación transforma la petición del usuario a consulta SPARQL o GEOSPARQL según sea el caso, la aplicación envía la consulta al Triple Store y cuando este devuelve los datos se grafican en el mapa como se observa en la figura 4.47, en este caso los recursos seleccionados para la visualización son “Cantones”, es decir se graficaron a todos los cantones del Ecuador.

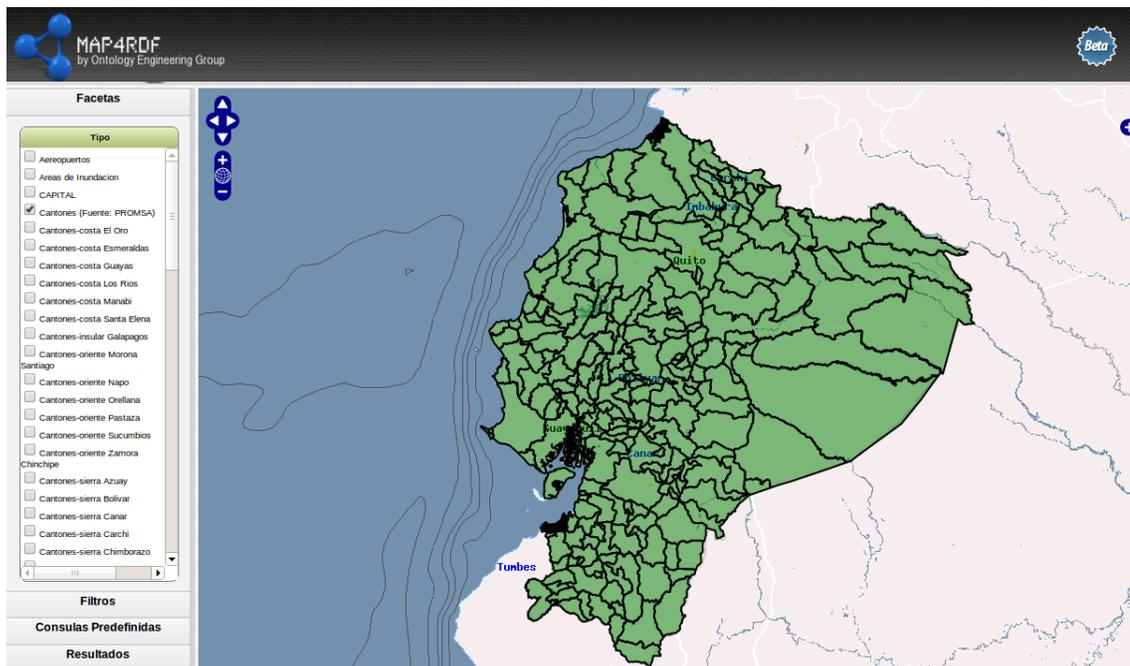


Figura 4.47: Visualización de Información Geométrica en formato RDF (Cantones del Ecuador)

MAP4RDF también cuenta con cinco consultas GEOSPARQL predeterminadas, a continuación se muestran las consultas + que fueron implementadas en la aplicación.

1. **ENUNCIADO:** Obtener a las poblaciones que se encuentran dentro de la Provincia del AZUAY.

CONSULTA GEOSPARQL IMPLEMENTADA

```
SELECT distinct ?r ?label ?Figure ?r2 ?Figure2
WHERE {
  ?r rdf:type <http://geo.linkeddata.ec/ontology/poblaciones_aeorg>.
  ?r rdfs:label ?label.
  ?r geo:hasGeometry ?geo.
  ?geo rdf:type ?geoType .
  ?geo geo:asWKT ?Figure.

  ?r2 rdf:type <http://geo.linkeddata.ec/ontology/provincias_promsa>.
  ?r2 rdfs:label "AZUAY"@es.
  ?r2 geo:hasGeometry ?geo2.
```

```

?geo2 rdf:type ?geoType2 .
?geo2 geo:asWKT ?Figure2 .

FILTER ( geof:sfContains( ?Figure2 , ?Figure) ) .
}

```

RESULTADO: En la figura 4.48 se muestra el resultado de la primera consulta predeterminada de MAP4RDF.

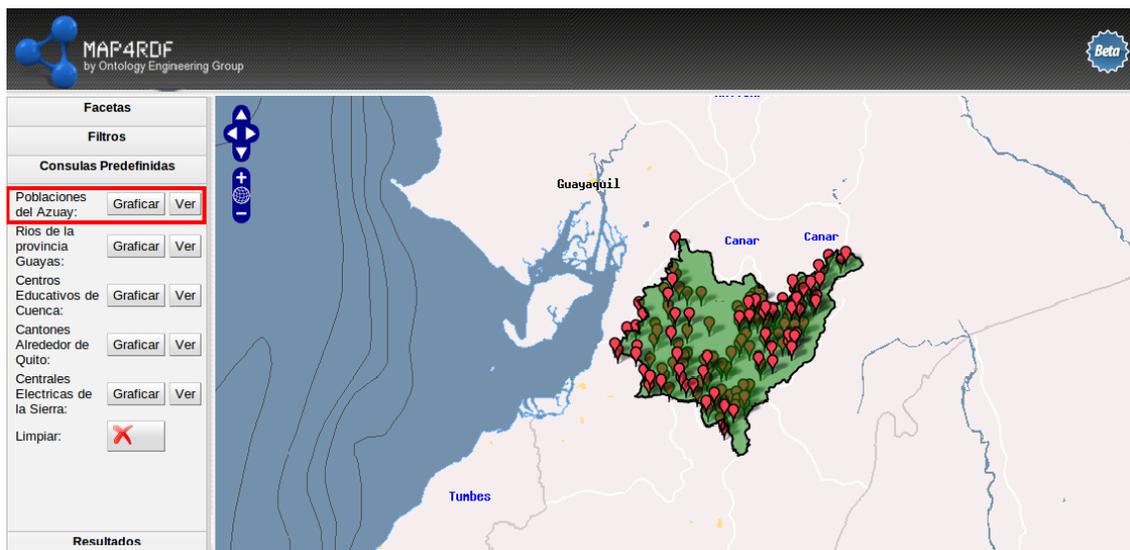


Figura 4.48: Poblaciones que están dentro de la provincia del Azuay.

2. **ENUNCIADO:** Ríos que intersectan la Provincia del GUAYAS.

CONSULTA GEOSPARQL IMPLEMENTADA

```

SELECT distinct ?r ?label ?Figure ?r2 ?Figure2
WHERE {
  ?r rdf:type <http://geo.linkeddata.ec/ontology/riosdobles_promsa>.
  ?r rdfs:label ?label.
  ?r geo:hasGeometry ?geo.
  ?geo rdf:type ?geoType .
  ?geo geo:asWKT ?Figure.

  ?r2 rdf:type <http://geo.linkeddata.ec/ontology/provincias_promsa>.
  ?r2 rdfs:label "GUAYAS"@es.
  ?r2 geo:hasGeometry ?geo2.
}

```

```

?geo2 rdf:type ?geoType2 .
?geo2 geo:asWKT ?Figure2 .

FILTER ( geof:sfIntersects (? Figure2 , ?Figure) ) .
}

```

RESULTADO: En la figura 4.49 se muestra el resultado de la segunda consulta predeterminada de MAP4RDF.

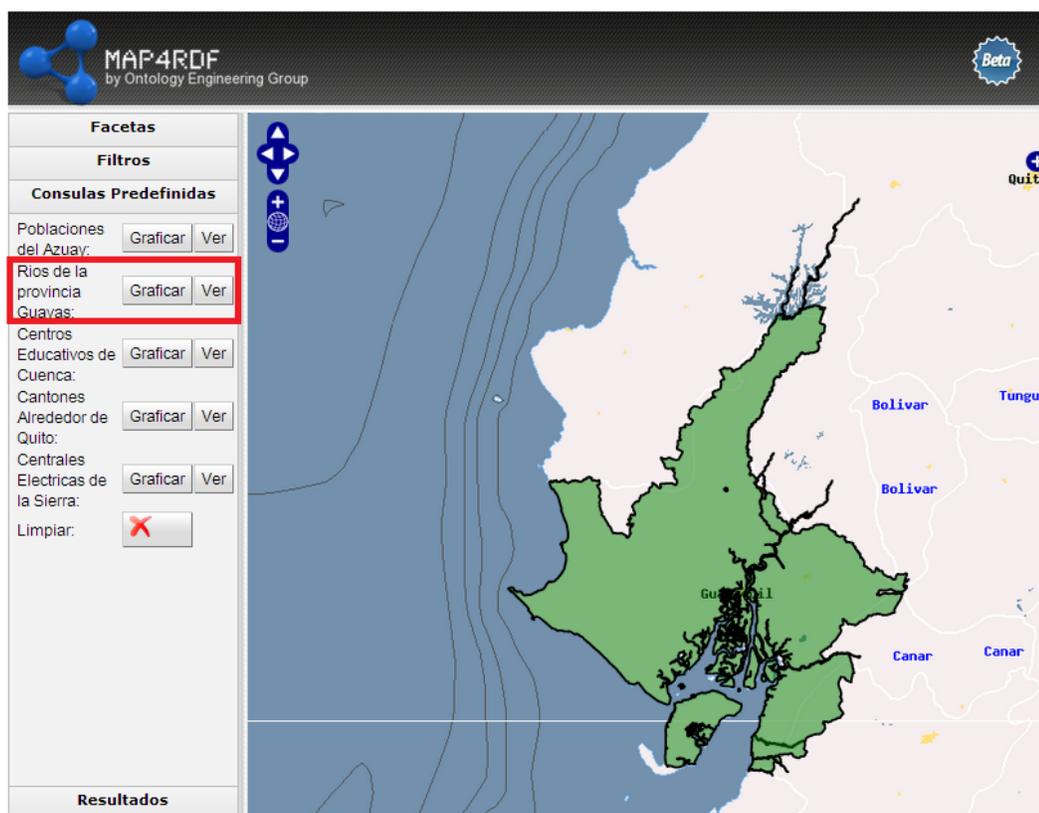


Figura 4.49: Ríos que intersectan la Provincia del Guayas.

3. **ENUNCIADO:** Centros Educativos que se encuentran dentro de un área de 5 Km a la redonda de la ciudad de Cuenca.

CONSULTA GEOSPARQL IMPLEMENTADA

```

SELECT distinct ?r ?label ?Figure ?r2 ?Figure2
WHERE {
    ?r rdf:type <http://geo.linkeddata.ec/ontology/centroeducativo> .
}

```

```
?r rdfs:label ?label .
?r geo:hasGeometry ?geo .
?geo rdf:type ?geoType .
?geo geo:asWKT ?Figure .

?r2 rdf:type <http://geo.linkeddata.ec/ontology/
ciudadesPatrimoniales> .
?r2 rdfs:label "CUENCA"@es .
?r2 geo:hasGeometry ?geo2 .
?geo2 rdf:type ?geoType2 .
?geo2 geo:asWKT ?Figure2 .

FILTER(geof:distance(?Figure , ?Figure2 , units:metre) <= 5000 ) .
}
```

RESULTADO: En la figura 4.50 se muestra el resultado de la tercera consulta predeterminada de MAP4RDF.

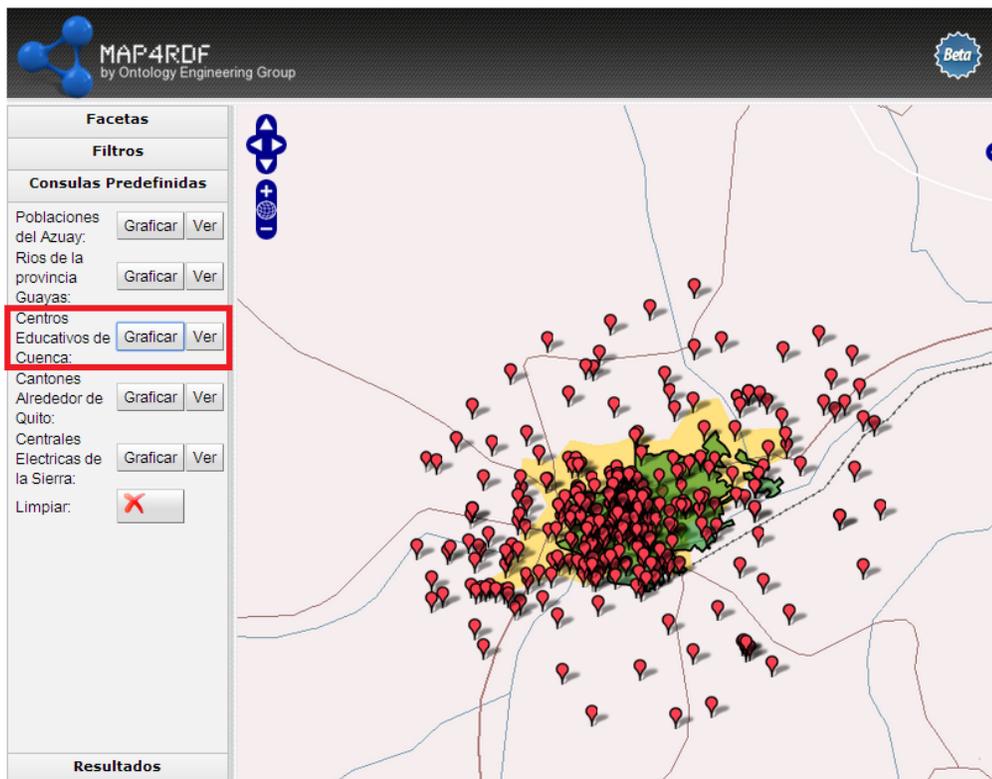


Figura 4.50: Centros Educativos a 500 metros a la redonda de la ciudad de Cuenca.

4. **ENUNCIADO:** Cantones que se encuentran alrededor del cantón Quito.

CONSULTA GEOSPARQL IMPLEMENTADA

```
SELECT distinct ?r ?label ?Figure
WHERE {
  ?r rdf:type <http://geo.linkeddata.ec/ontology/cantones_inec>.
  ?r rdfs:label ?label.
  ?r geo:hasGeometry ?geo.
  ?geo rdf:type ?geoType .
  ?geo geo:asWKT ?Figure.

  ?r2 rdf:type <http://geo.linkeddata.ec/ontology/cantones_inec>.
  ?r2 rdfs:label "QUITO"@es.
  ?r2 geo:hasGeometry ?geo2.
  ?geo2 rdf:type ?geoType2 .
  ?geo2 geo:asWKT ?Figure2.

  FILTER (geof:sfTouches(?Figure , ?Figure2)).
}
```

RESULTADO: En la figura 4.51 se muestra el resultado de la cuarta consulta predeterminada de MAP4RDF.

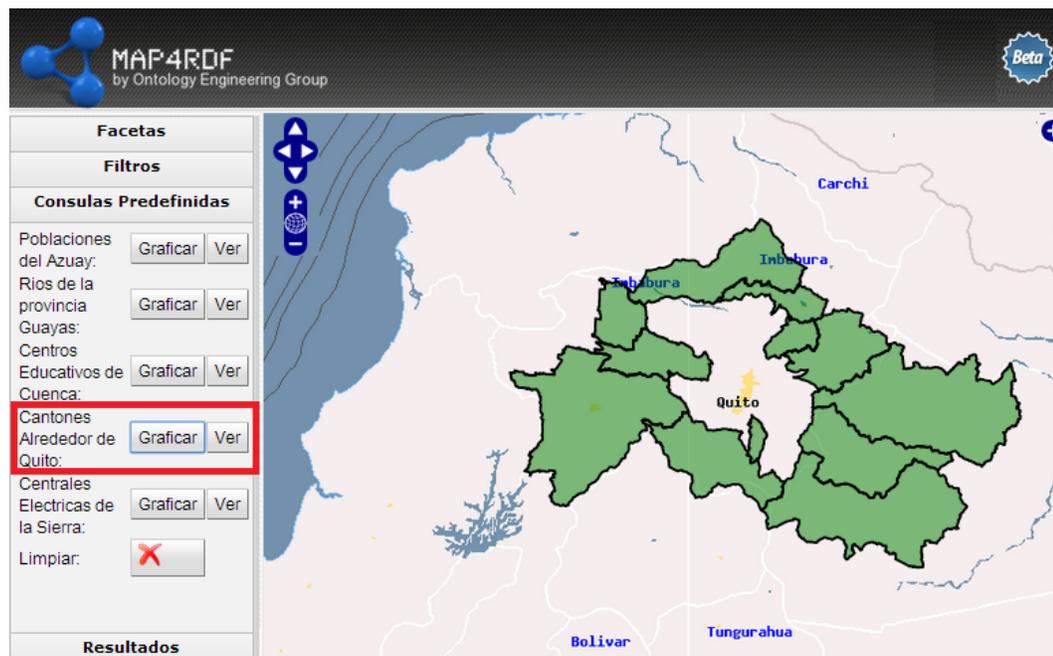


Figura 4.51: Cantones que rodean al cantón Quito

5. ENUNCIADO: Centrales Eléctricas ubicadas en la región SIERRA

CONSULTA GEOSPARQL IMPLEMENTADA

```
SELECT distinct ?r ?label ?Figure ?r2 ?Figure2
WHERE {
  ?r rdf:type <http://geo.linkeddata.ec/ontology/central_Electrica>.
  ?r rdfs:label ?label.
  ?r geo:hasGeometry ?geo.
  ?geo geo:asWKT ?Figure.

  ?r2 rdf:type <http://geo.linkeddata.ec/ontology/regiones>.
  ?r2 rdfs:label "SIERRA"@es.
  ?r2 geo:hasGeometry ?geo2.
  ?geo2 geo:asWKT ?Figure2.

  FILTER (geof:sfWithin(?Figure , ?Figure2)).
}
```

RESULTADO: En la figura 4.52 se muestra el resultado de la quinta consulta predeterminada de MAP4RDF.

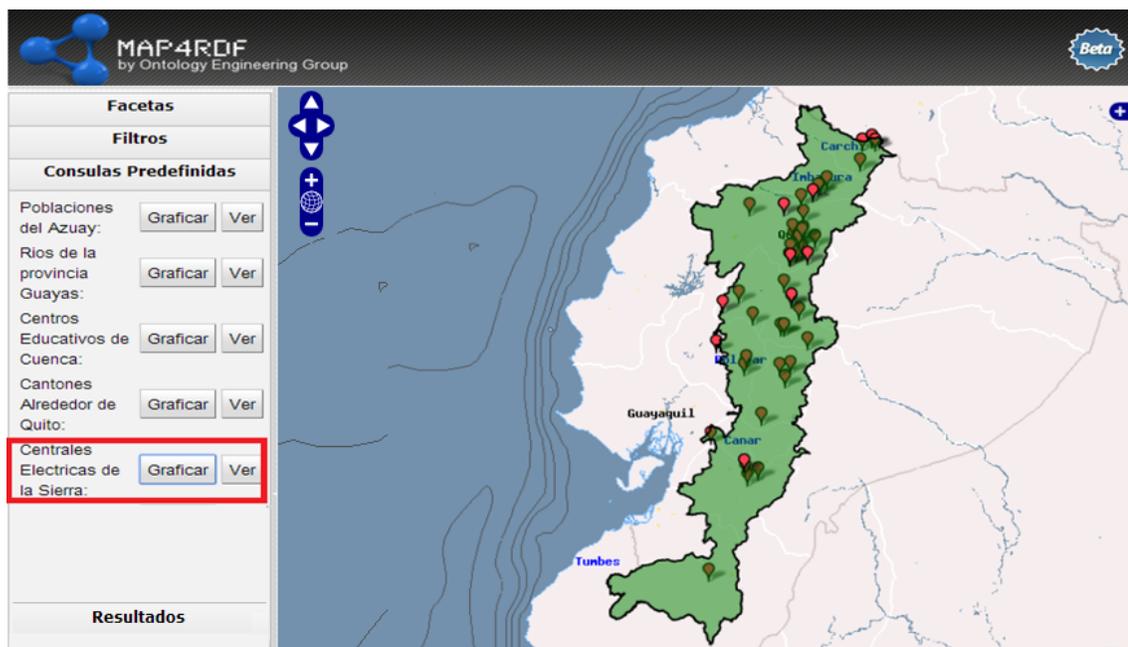
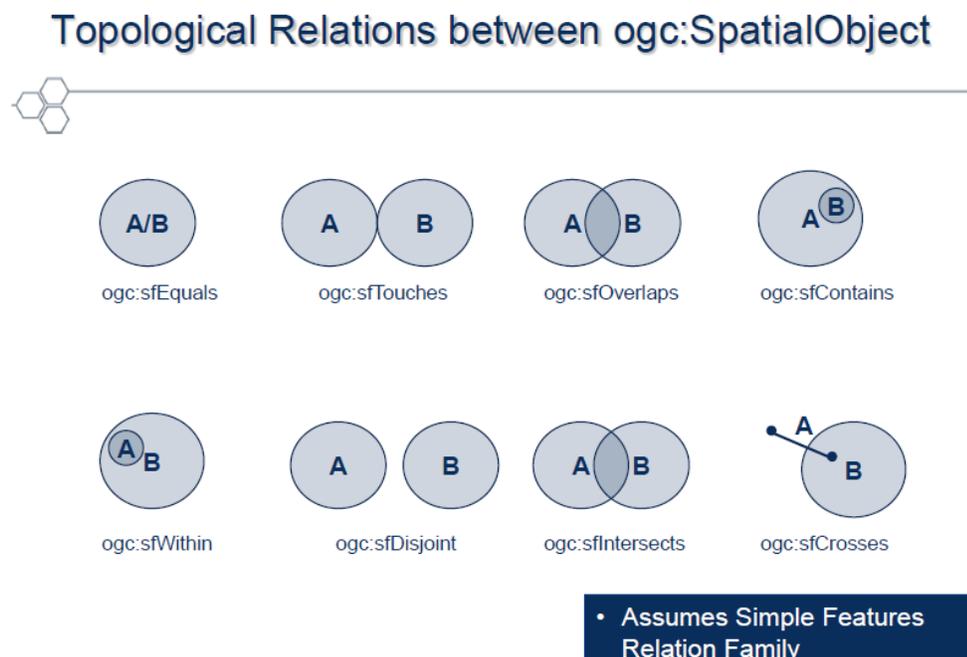


Figura 4.52: Centrales Eléctricas ubicadas en la región SIERRA

4.3. GENERACIÓN DE RELACIONES GEOESPACIALES

Una vez realizadas todas las actividades de la metodología para la generación de datos enlazados geoespaciales se ha considerado importante el análisis de geometrías entre los diferentes recursos publicados. Al haber seguido el modelo estándar GEOSPARQL es posible realizar análisis geoespacial entre los recursos, es decir se puede realizar cálculo de distancias, intersecciones, determinar si elemento contiene a otro elemento y otros tipos de cálculos según sus geometrías.

GEOSPARQL agrega una serie de funciones al tradicional vocabulario SPARQL, estas funciones son las que permiten realizar al análisis geoespacial de los recursos. En la figura 4.53 se ilustra las posibles relaciones que permiten realizar las funciones de análisis geoespacial, en la figura 4.54 se muestra las funciones y su respectiva sintaxis GEOSPARQL.



OGC®

Figura 4.53: Relaciones Espaciales - Fuente: [12], Página 18.

GeoSPARQL Topological Query Functions



- `ogcf:sfEquals`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean
- `ogcf:sfDisjoint`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean
- `ogcf:sfIntersects`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean
- `ogcf:sfTouches`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean
- `ogcf:sfCrosses`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean
- `ogcf:sfWithin`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean
- `ogcf:sfContains`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean
- `ogcf:sfOverlaps`(geom1: ogc:wktLiteral, geom2: ogc:wktLiteral): xsd:boolean



Assumes Simple Features
Relation Family

29

Figura 4.54: Funciones GeoSPARQL - Fuente: [12], Página 29.

En la figura 4.55 se presenta un ejemplo de consulta GEOSPARQL, implementada en MAP4RDF como predeterminada, que busca a los Ríos que interseca la provincia del Guayas, en la mencionada figura se detalla que parte de la consulta permite extraer las geometrías de los ríos del Ecuador y la geometría de la provincia del Guayas respectivamente para su posterior análisis geoespacial aplicando la función “`geof:sfIntersects`” y en la figura 4.56 se presenta los resultados en MAP4RDF.

```
Consulta 2
La siguiente consulta muestra a los Ríos que intersectan a la Provincia del GUAYAS
A B
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX sf: <http://www.opengis.net/ont/sf#>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
SELECT distinct ?r ?label ?Figure ?r2 ?Figure2
WHERE {
?r rdf:type <http://geo.linkeddata.ec/ontology/riosdobles_promsa>.
?r rdfs:label ?label.
?r geo:hasGeometry ?geo.
?geo rdf:type ?geoType .
?geo geo:asWKT ?Figure. Geometrias de A

?r2 rdf:type <http://geo.linkeddata.ec/ontology/provincias_promsa>.
?r2 rdfs:label "GUAYAS"@es.
?r2 geo:hasGeometry ?geo2.
?geo2 rdf:type ?geoType2 .
?geo2 geo:asWKT ?Figure2. Geometrias de B

FILTER (geof:sfIntersects(?Figure2, ?Figure)).
```

FUNCIONES GeoSPARQL

Figura 4.55: Consulta GeoSPARQL - Intersección: Ríos y la Provincias del Guayas

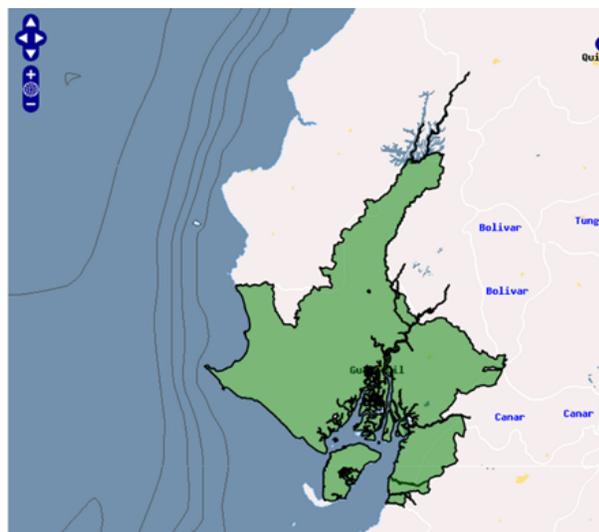


Figura 4.56: Visualización GeoSPARQL - Intersección: Ríos y la Provincias del Guayas



5. RESULTADOS

En este capítulo se muestran los resultados obtenidos con la finalidad de cumplir cada uno de los objetivos planteados al inicio del presente trabajo de tesis. Como se describió en los capítulos anteriores, se partió desde un análisis de las fuentes de información, posteriormente se desarrolló un plugin (Geometry2RDF) para la generación de RDF en el entorno ETL, luego se actualizó la herramienta (MAP4RDF) para consumo de datos geospaciales RDF cumpliendo con el estándar GEOSPARQL, y finalmente se desarrolló todo el proceso de generación, publicación y consumo de Datos Enlazados Geospaciales, logrando así el cumplimiento del objetivo principal, **“Creación de un prototipo de Repositorio de Datos Enlazados Geospaciales del Ecuador, utilizando las herramientas mejoradas para generación de RDF”**, obteniendo los resultados esperados.

5.1. ANÁLISIS DE LAS FUENTES DE INFORMACIÓN

En esta sección se muestra los resultados del análisis de las fuentes de información (archivos Shapefile). Los archivos Shapefile se buscaron de forma manual en la Web, con ayuda de buscadores se ingresó a páginas Web que tienen a disposición la descarga de este tipo de información, se logró un total de 121 archivos Shapefile recolectados de diferentes repositorios del Ecuador. En la sección C de ANEXOS, se especifica la ubicación Web de cada uno de los repositorios de donde se obtuvo los archivos shapefile.

Archivos Shapefile Válidos

En la figura 5.57 se observa que se recolectó un total de 121 archivos shapefile de los cuales 80 fueron posibles su transformación a RDF. Mientras que 41 archivos shapefile no pudieron ser procesados puesto que eran inválidos o dio un error durante su transformación, los errores de estos archivos era debido a que se encontraban en sistemas de coordenadas obsoletos, también algunos de esos archivos Shapefile tenían errores en su estructura, es decir no se podían leer con el software GIS y otros archivos poseían poca información o en algunos casos información nula y esto no permitía la generación de RDF.

RDF GENERADOS	TOTAL
SI	80
NO	41
TOTAL ARCHIVOS SHAPEFILE RECOLECTADOS	121

Figura 5.57: Total Archivos RDF Generados

En la figura 5.58 se observa un gráfico de barras con el total de archivos Shapefile validos de los cuales se pudo generar archivos RDF, y los que no pudieron ser generados respecto al total de todos los archivos fuente Shapefile recolectados.

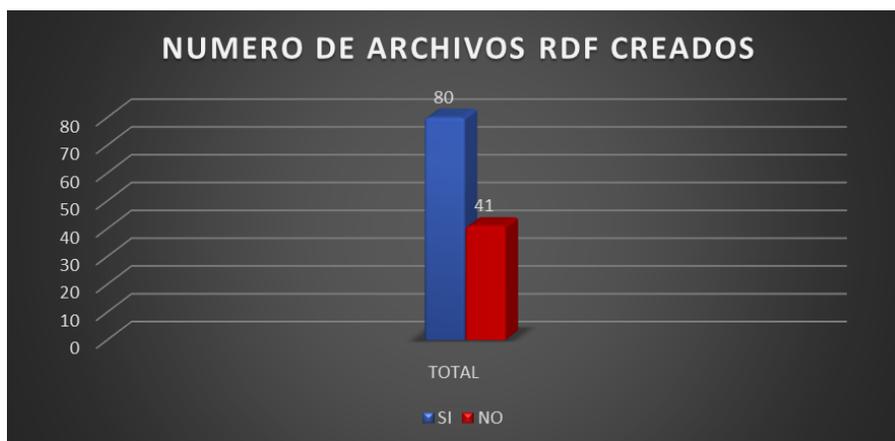


Figura 5.58: Total de Archivos RDF Generados y No Generados

Transformación de Coordenadas de los Shapefile

De los archivos Shapefile válidos para la generación de RDF fue necesario la transformación del sistema de coordenadas de algunos de estos, puesto que se encontraban en sistemas de coordenadas diferentes u obsoletos. Es necesario que los archivos shapefile se encuentren en el Sistema de Coordenadas EPGS:32717(WGS 84 / UTM zone 17S) para no tener ningún problema al generar los archivos RDF con el plugin Geometry2RDF. En la figura 5.59 se presenta los resultados de los archivos Shapefile que necesitaron ser transformados al sistema de coordenadas requerido con ayuda de la herramienta QGIS.

CAMBIO DE SISTEMA DE COORDENADAS	TOTAL
NO	50
SI	30
TOTAL SHAPEFILES APROPIADOS PARA TRANSFORMACION	80

Figura 5.59: Shapefile que sufrieron cambio de sistema de coordenadas

Finalmente, se ha descrito el respectivo análisis de las fuentes de información (archivos Shapefile) y se ha mostrado los respectivos resultados llegando a cumplir el objetivo “**Analizar y definir las fuentes de información (Archivos Shapefile)**” para la correcta generación de información en formato RDF.

5.2. GENERACIÓN DE RDF

En esta sección se describen los resultados obtenidos en la generación de los archivos en formato RDF. La Generación de los archivos RDF se ha realizado correctamente gracias al análisis, corrección de errores y actualización de la herramienta Geometry2RDF al entorno ETL de GeoKettle. También, siguiendo el estándar y modelo ontológico GEOSPARQL los archivos RDF han sido publicados en el Triple Store Parliament sin presentar problema alguno. Además, las consultas SPARQL y GEOSPARQL realizadas a los archivos RDF publicados devuelven información co-



recta como se observa en la sección 4.2.6 de la Explotación y Visualización de la información, lo que indica la correcta utilización y funcionalidad del Plugin Geometry2RDF. Finalmente, se ha logrado cumplir con los objetivos: “**Analizar el modelo ontológico a utilizar (GEOSPARQL)**”, “**Analizar las herramientas existentes para la generación de Datos Enlazados Geoespaciales**” y “**Crear un plugin para la generación de RDF dentro de Geokettle**” del presente trabajo de Tesis.

5.3. MAP4RDF

En esta sección se describen resultados obtenidos al usar la herramienta MAP4RDF. MAP4RDF, ahora modificada, se conecta correctamente al SPARQL Endpoint y sin problema realiza las búsquedas según lo que el usuario seleccione. Siguiendo el modelo estándar GEOSPARQL, MAP4RDF realiza las consultas correctamente devolviendo los resultados en un tiempo en el cual depende del servidor Parliament y del equipo en el que se encuentre instalado el mencionado servidor (Para la configuración del equipo referirse al anexo D). MAP4RDF a mas de graficar puntos también es capaz de graficar geometrías complejas como líneas o polígonos lo que permite que el usuario tenga mejor comprensión del tipo de información que está consultando.

A continuación se describe 3 análisis realizados de consultas de prueba en MAP4RDF.

- Se consultó y graficó a los Centros Educativos del Ecuador siendo 5000 como limite de la consulta, cuya geometría es de tipo PUNTO, en un promedio de 54 segundos.
- Se consultó y graficó a los Cantones del Ecuador siendo 211 en total, cuya geometría es de tipo POLIGONO, en un promedio de 160 segundos.
- Se consultó y graficó a las Carreteras del Ecuador siendo 2300 carreteras en total, cuya geometría es de tipo LINEA, en un promedio de 150 segundos.

Finalmente, el análisis descrito en los puntos anteriores y en la sección 4.2.6 reflejan que la aplicación MAP4RDF genera los resultados esperados en el presente traba-

jo de Tesis, dando cumplimiento al objetivo “**Analizar y extender las herramientas para el consumo de Datos Enlazados Geospaciales (MAP4RDF)**”.

5.4. ENLACE DE DATOS

En esta sección se describe los resultados del proceso de enlaces o interlinking. Fue necesario realizar el proceso de enlace de datos entre algunos archivos RDF que se generaron, porque originalmente los archivos Shapefile contenían información similar, es decir del mismo tipo de recursos; en la figura 5.60 se muestra cuales de estos archivos Shapefile contienen información similar entre si.

Nro.	TIPO DE RECURSO	Fuente A	Nombre de Archivo	Fuente B	ARCHIVOS SHAPE - B
1	provincias	PROMSA	provincias.shp	INEC	nxprovincias.shp
2	volcanes	PROMSA	P_VOLC1.SHP	IGEPN-STGR	PELIGRO VOLCANICO.shp
3	poblaciones	PROMSA	01_pobla.shp	accionecologica.org	setlmtp.shp
4	cantones	PROMSA	cantones.shp	INEC	LIMITES_Cantones.shp
5	cuencas	PROMSA	cuencas.shp	MAGAP	cuencas.shp
6	subcuencas	PROMSA	subcuencas.shp	MAGAP	subcuencas.shp
7	rios	PROMSA	rios dobles finales z17.shp	accionecologica.org	rivera.shp
8	Parque nacional	PROMSA	snap2.shp	MAE	PANE.shp

Figura 5.60: Archivos Shapefile con información similar

En la figura 5.61 se muestra los resultados del proceso de enlace de datos entre los archivos de la figura 5.60.

Nro.	Número de recursos en la fuente A	Número de recursos en la fuente B	ENLACES GENERADOS
1	21	24	21
2	7	13	5
3	4847	2165	1520
4	211	402	211
5	78	79	76
6	143	157	123
7	154	24	20
8	26	45	17

Figura 5.61: Resultados de Generación de Enlaces

También, se realizó el proceso de enlace de datos entre la DBpedia y los datos de

la Fuente A, debido a que esta última tiene mayor número de recursos respecto a la Fuente B y por lo tanto se pudo obtener mayor número de enlaces y los resultados se muestran en la figura 5.62.

Nro.	Número de recursos en la fuente A	Número de enlaces generados con la DBPEDIA
1	21	9
2	7	5
3	4847	18
4	211	2
5	78	0
6	143	0
7	154	1
8	26	1

Figura 5.62: Resultados de Generación de Enlaces con la DBPEDIA

Finalmente, los resultados de esta sección culminan la etapa de enlace de datos la cual forma parte del proceso de publicación de Datos Enlazados Geoespaciales.

5.5. REPOSITORIO WEB

En esta sección se muestra que la información generada en formato RDF se encuentra publicada en un Triple Store al que cualquier usuario podrá conectarse y realizar consultas SPARQL y GEOSPARQL; para ingresar al SPARQL Endpoint diríjase al siguiente enlace “<http://apolo.utpl.edu.ec:8080/parliament/query.jsp>” como se observa en la figura 5.63.

El usuario también podrá acceder a la aplicación MAP4RDF a través de la siguiente página Web “<http://geo.linkeddata.ec/>” y consultar la información geográfica disponible como se observa en la figura 5.64.

Los resultados de esta sección muestran el cumplimiento del objetivo “Aplicar el proceso de publicación de Datos Enlazados Geoespaciales”. Finalmente, en este capítulo se ha verificado el cumplimiento de la generación de los archivos RDF con Geometry2RDF, publicación en el repositorio Web y explotación de datos con MAP4RDF que demuestran el cumplimiento de todos los objetivos específicos de la presente Tesis y, por lo tanto, se ha alcanzado el objetivo general “**Creación de un**

prototipo de Repositorio de Datos Enlazados Geospaciales del Ecuador, utilizando herramientas mejoradas para generación de RDF”, culminando con éxito el presente trabajo de Tesis.

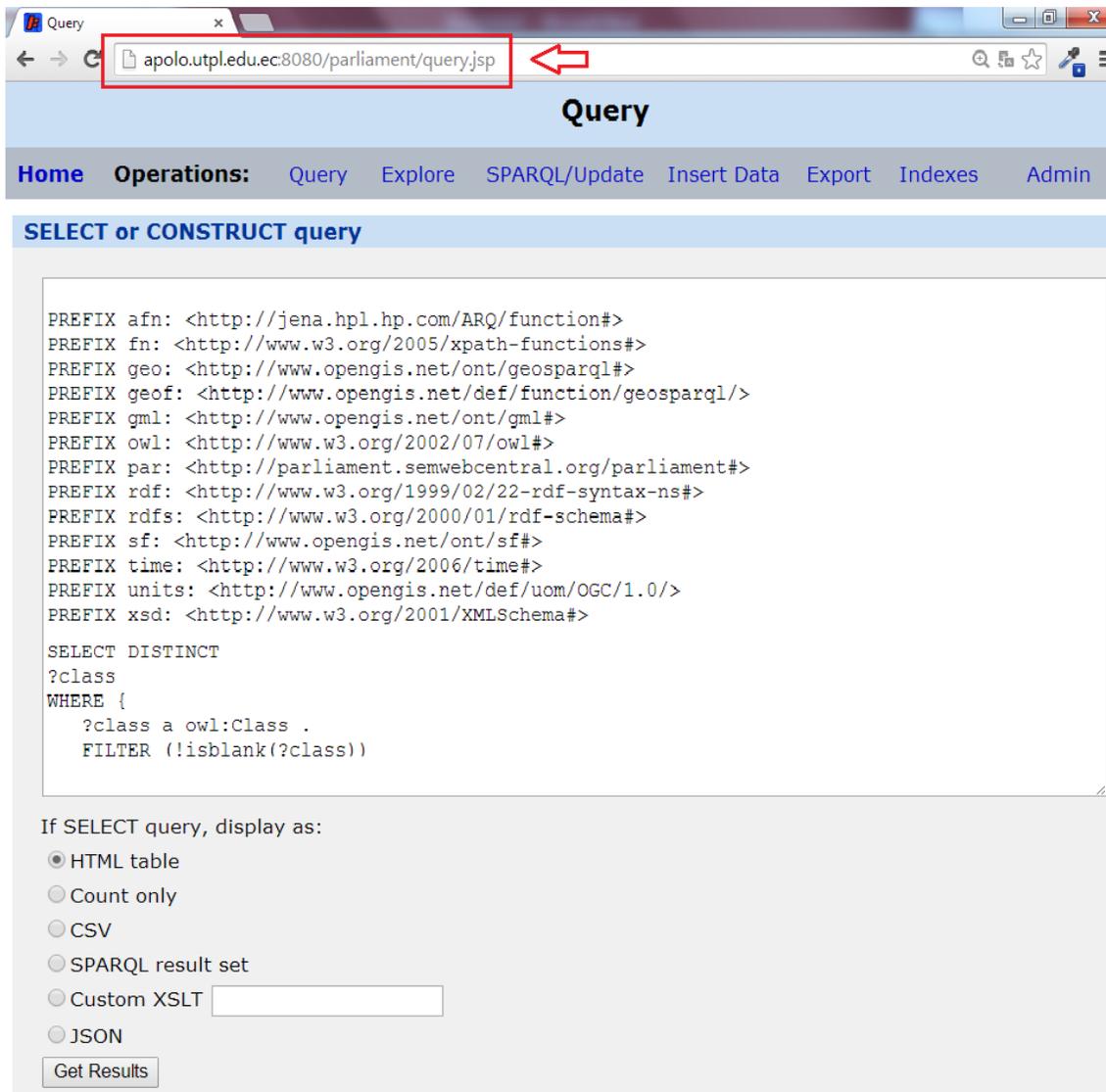


Figura 5.63: Repositorio Geo Linked Data Ecuador

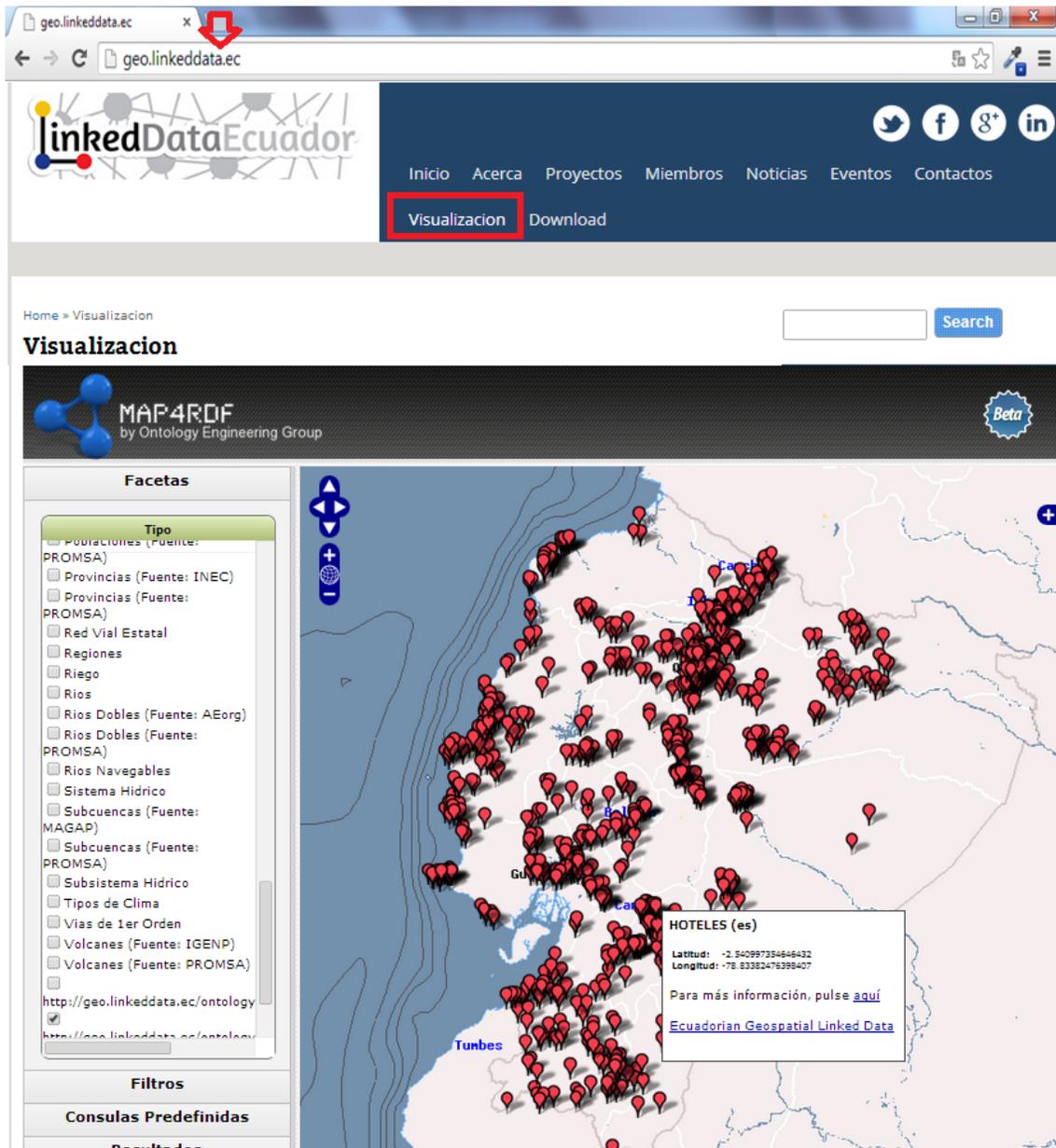


Figura 5.64: Geo Linked Data Ecuador



6. CONCLUSIONES

La presente tesis tuvo como objetivo la publicación de un repositorio de datos geoespaciales del Ecuador, para lo cual se siguió una metodología para la publicación de Linked Data y el estándar GEOSPARQL. A continuación se describen las principales actividades y objetivos que se cumplieron al culminar este trabajo.

1. Como fruto de la investigación, se generaron dos publicaciones que fueron lideradas por los investigadores principales:

W3C

“http://www.w3.org/2014/03/lgd/papers/lgd14_submission_64.pdf”

ASCE

“http://content.asce.org/files/pdf/IPWE2014-PROGRAM-DESIGN-FINAL_LOW.pdf”

2. Se utilizó los estándares GEOSPARQL e ISO 19115, el primero como lenguaje de consulta ya que contiene funciones específicas para manejo de datos espaciales; y el segundo, para el almacenamiento de metadatos, que si bien no se usó completamente este esquema, se dio un inicio para el llenado de estos datos y de esta manera a futuro sea más sencillo completar esta información.
3. Se desarrolló y publicó el plugin GEOMETRY2RDF para GeoKettle, el cual es compatible con GEOSPARQL y pone a disposición al usuario una interfaz amigable que no requiere la modificación de archivos complejos de configuración, ni ser experto en varias áreas para su utilización.
4. Se realizó la mejora de MAP4RDF, que ahora permite graficar de una manera más eficaz y rápida, además de ser compatible con GEOSPARQL.



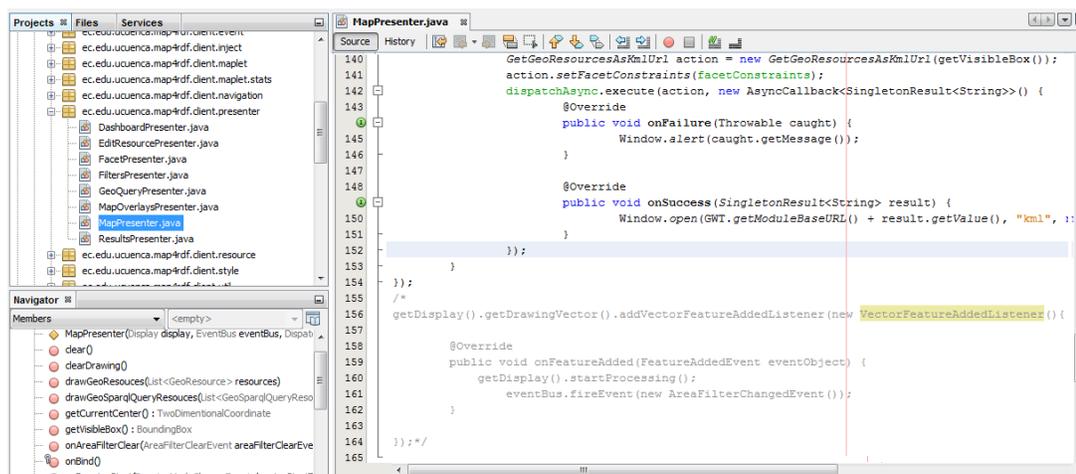
5. Se siguió todos los pasos para la publicación de Datos Enlazados, entre los que se encuentra la realización de un enlazado de datos con la DBPedia, para enriquecer más la información.
6. Se realizó la creación del repositorio linkeddata.ec, de acceso público, el cual contiene todos los Datos Enlazados Geospaciales que han sido generados, los cuales pueden ser visualizados mediante MAP4RDF.

Finalmente, ya con el sitio online accesible por el público en general, se da por cumplido el gran objetivo de esta tesis, que es disponer de un repositorio real de datos geospaciales del Ecuador, pero además, dejando la puerta abierta a nuevos proyectos y mejoras, especialmente en temas de la utilización completa del ISO 19115, del filtrado y validación de información de RDFs, y el desarrollo de plugins para GeoKettle que ayuden a la lectura de nuevos tipos de formato y que utilicen nuevas ontologías además de GEOSPARQL.

Apéndice A

MODIFICACIONES Y CONFIGURACIONES DE MAP4RDF

En esta sección, se presentan los anexos necesarios a los que se ha referenciado en los capítulos anteriores, con el objetivo de mostrar todas las modificaciones, actualizaciones, configuraciones y resultados obtenidos de la aplicación MAP4RDF descritos al pie de cada figura.



```
140 GetGeoResourcesAsKmlUrl action = new GetGeoResourcesAsKmlUrl(getVisibleBox());
141 action.setFacetConstraints(facetConstraints);
142 dispatchAsync().execute(action, new AsyncCallback<SingletonResult<String>>() {
143     @Override
144     public void onFailure(Throwable caught) {
145         Window.alert(caught.getMessage());
146     }
147
148     @Override
149     public void onSuccess(SingletonResult<String> result) {
150         Window.open(GWT.getModuleBaseUrl() + result.getValue(), "kml", 1000, 1000);
151     }
152 });
153
154
155
156
157 getDisplay().getDrawingVector().addVectorFeatureAddedListener(new VectorFeatureAddedListener() {
158
159     @Override
160     public void onFeatureAdded(FeatureAddedEvent eventObject) {
161         getDisplay().startProcessing();
162         eventBus.fireEvent(new AreaFilterChangedEvent());
163     }
164
165 });
```

Figura A.1: Error que impedía filtrar los resultados

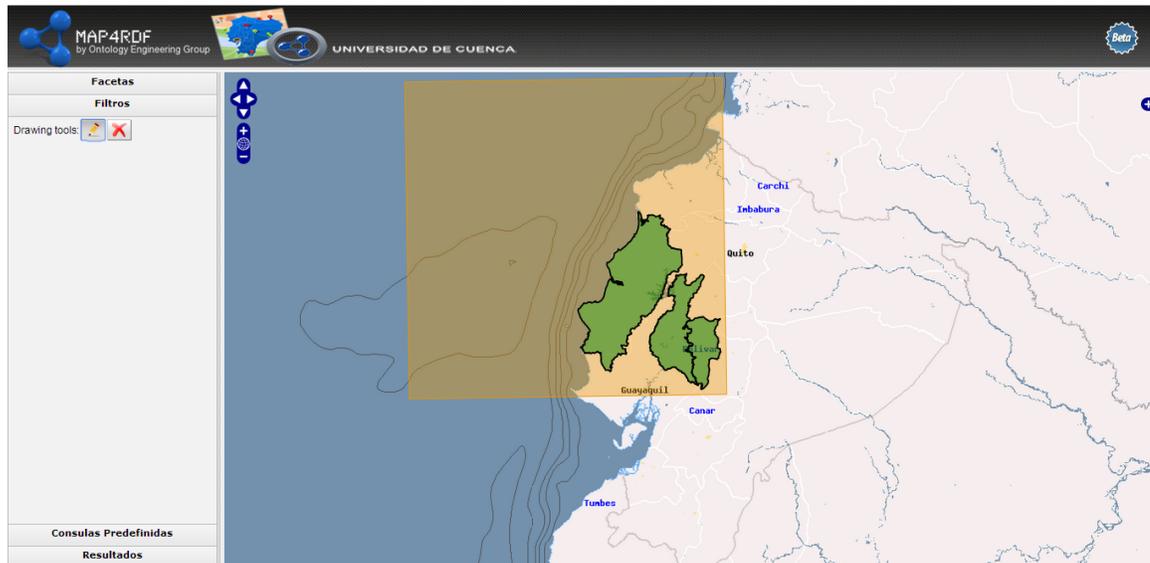


Figura A.2: Error de filtrado de resultados corregido. Filtrado de las provincias del Ecuador

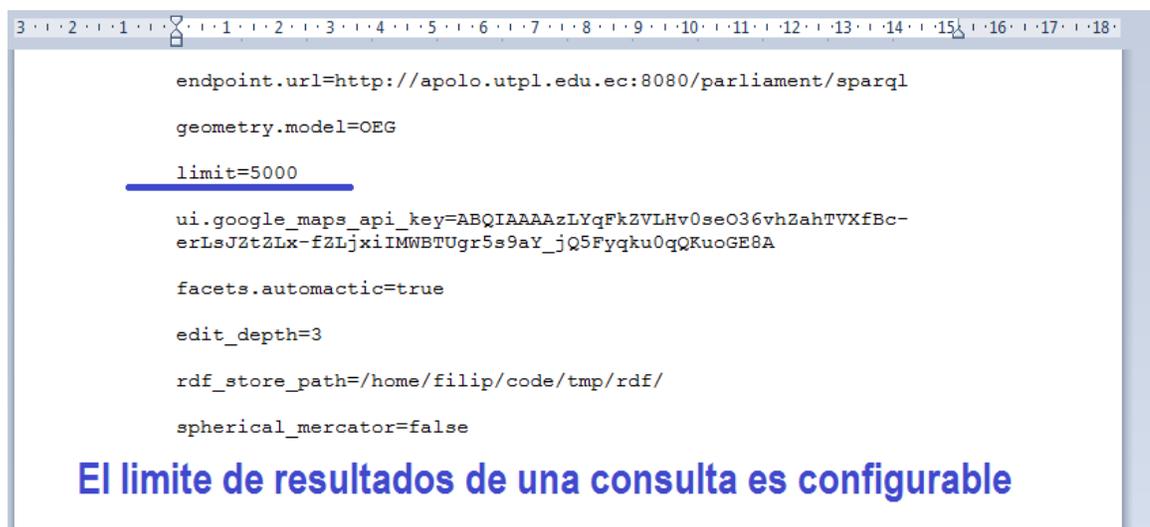
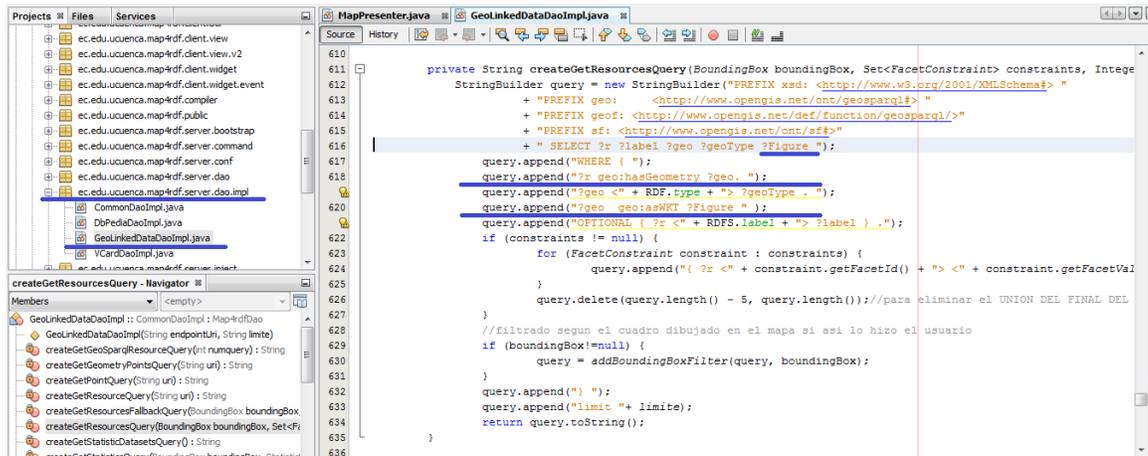
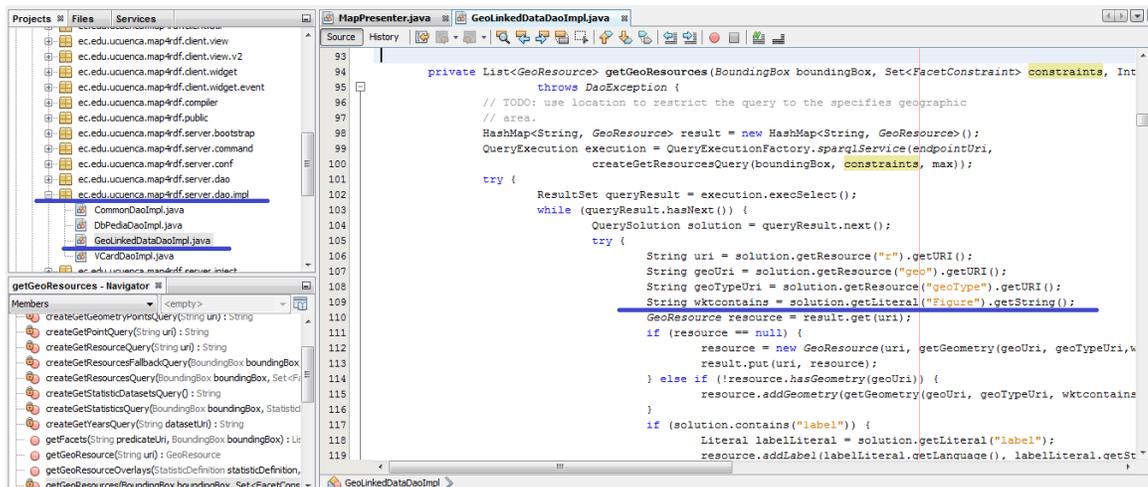


Figura A.3: Limite de resultados máximo de consulta corregido, ahora disponible en archivo de configuración



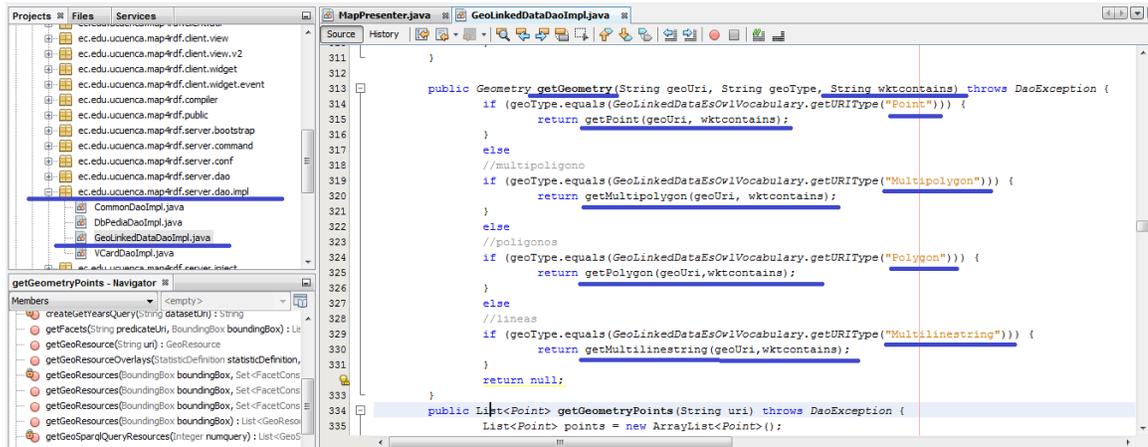
```
private String createGetResourcesQuery(BoundingBox boundingBox, Set<FacetConstraint> constraints, Integer limit) {
    String query = new StringBuilder("PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> "
        + "PREFIX geo: <http://www.opengis.net/ont/geosparql#> "
        + "PREFIX geof: <http://www.opengis.net/def/function/geosparql#> "
        + "PREFIX sf: <http://www.opengis.net/ont/sf#> "
        + "SELECT ?r ?label ?geo ?geoType ?Figure ");
    query.append("WHERE ( ");
    query.append("?r geo:hasGeometry ?geo. ");
    query.append("?geo <\" + RDF.type + \"> ?geoType . ");
    query.append("?geo geo:asWKT ?Figure ");
    query.append("OPTIONAL ( ?r <\" + RDFS.label + \"> ?label ). ");
    if (constraints != null) {
        for (FacetConstraint constraint : constraints) {
            query.append("(! ?r <\" + constraint.getFacetId() + \"> <\" + constraint.getFacetVal() + \"> ");
        }
        query.delete(query.length() - 5, query.length()); //para eliminar el UNION DEL FINAL DEL
    }
    //filtrado segun el cuadro dibujado en el mapa si asi lo hizo el usuario
    if (boundingBox != null) {
        query = addBoundingBoxFilter(query, boundingBox);
    }
    query.append(") ");
    query.append("LIMIT " + limit);
    return query.toString();
}
```

Figura A.4: Modificación de función que construye la cadena de consulta



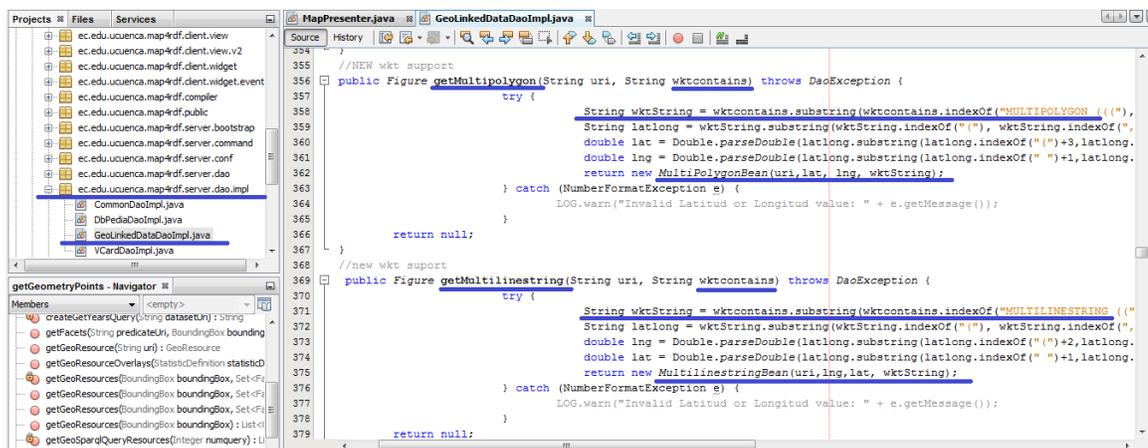
```
private List<GeoResource> getGeoResources(BoundingBox boundingBox, Set<FacetConstraint> constraints, Integer limit) throws DaoException {
    // TODO: use location to restrict the query to the specifies geographic
    // area
    HashMap<String, GeoResource> result = new HashMap<String, GeoResource>();
    QueryExecution execution = QueryExecutionFactory.sparqlService(endpointUri,
        createGetResourcesQuery(boundingBox, constraints, limit));
    try {
        ResultSet queryResult = execution.execSelect();
        while (queryResult.hasNext()) {
            QuerySolution solution = queryResult.next();
            try {
                String uri = solution.getResource("").getURI();
                String geoUri = solution.getResource("geo").getURI();
                String geoTypeUri = solution.getResource("geoType").getURI();
                String wktcontains = solution.getLiteral("Figure").getString();
                GeoResource resource = result.get(uri);
                if (resource == null) {
                    resource = new GeoResource(uri, getGeometry(geoUri, geoTypeUri, wktcontains));
                    result.put(uri, resource);
                } else if (!resource.hasGeometry(geoUri)) {
                    resource.addGeometry(getLiteral("Figure").getString());
                }
                if (solution.contains("label")) {
                    Literal labelLiteral = solution.getLiteral("label");
                    resource.addLabel(labelLiteral.getLanguage(), labelLiteral.getStr());
                }
            } catch (Exception e) {
                // ignore
            }
        }
    }
}
```

Figura A.5: Modificación de función que recupera la propiedad “Figure” de el resultado de consulta



```
311 }
312
313 public Geometry getGeometry(String geoUri, String geoType, String wktcontains) throws DaoException {
314     if (geoType.equals(GeoLinkedDataEsOviVocabulary.getURIType("Point"))) {
315         return getPoint(geoUri, wktcontains);
316     }
317     else
318         //multipoligono
319         if (geoType.equals(GeoLinkedDataEsOviVocabulary.getURIType("Multipolygon"))) {
320             return getMultipolygon(geoUri, wktcontains);
321         }
322     else
323         //poligono
324         if (geoType.equals(GeoLinkedDataEsOviVocabulary.getURIType("Polygon"))) {
325             return getPolygon(geoUri, wktcontains);
326         }
327     else
328         //lineas
329         if (geoType.equals(GeoLinkedDataEsOviVocabulary.getURIType("Multilinestring"))) {
330             return getMultilinestring(geoUri, wktcontains);
331         }
332     }
333     return null;
334 }
335 public List<Point> getGeometryPoints(String uri) throws DaoException {
336     List<Point> points = new ArrayList<Point>();
337 }
```

Figura A.6: Determinación del tipo de Geometría según el WKT recuperado



```
354 //NEW wkt support
355 public Figure getMultipolygon(String uri, String wktcontains) throws DaoException {
356     try {
357         String wktString = wktcontains.substring(wktcontains.indexOf("MULTIPOLYGON ((",
358         String latlong = wktString.substring(wktString.indexOf("(", wktString.indexOf(",
359         double lat = Double.parseDouble(latlong.substring(latlong.indexOf("(")+3, latlong.
360         double lng = Double.parseDouble(latlong.substring(latlong.indexOf("(")+1, latlong.
361         return new MultipolygonBean(uri, lat, lng, wktString);
362     } catch (NumberFormatException e) {
363         LOG.warn("Invalid Latitud or Longitud value: " + e.getMessage());
364     }
365     return null;
366 }
367 //no wkt support
368 public Figure getMultilinestring(String uri, String wktcontains) throws DaoException {
369     try {
370         String wktString = wktcontains.substring(wktcontains.indexOf("MULTILINESTRING ((",
371         String latlong = wktString.substring(wktString.indexOf("(", wktString.indexOf(",
372         double lng = Double.parseDouble(latlong.substring(latlong.indexOf("(")+2, latlong.
373         double lat = Double.parseDouble(latlong.substring(latlong.indexOf("(")+1, latlong.
374         return new MultilinestringBean(uri, lng, lat, wktString);
375     } catch (NumberFormatException e) {
376         LOG.warn("Invalid Latitud or Longitud value: " + e.getMessage());
377     }
378     return null;
379 }
```

Figura A.7: Envío de los recursos a buffer para graficarlos según el tipo de Geometría del WKT

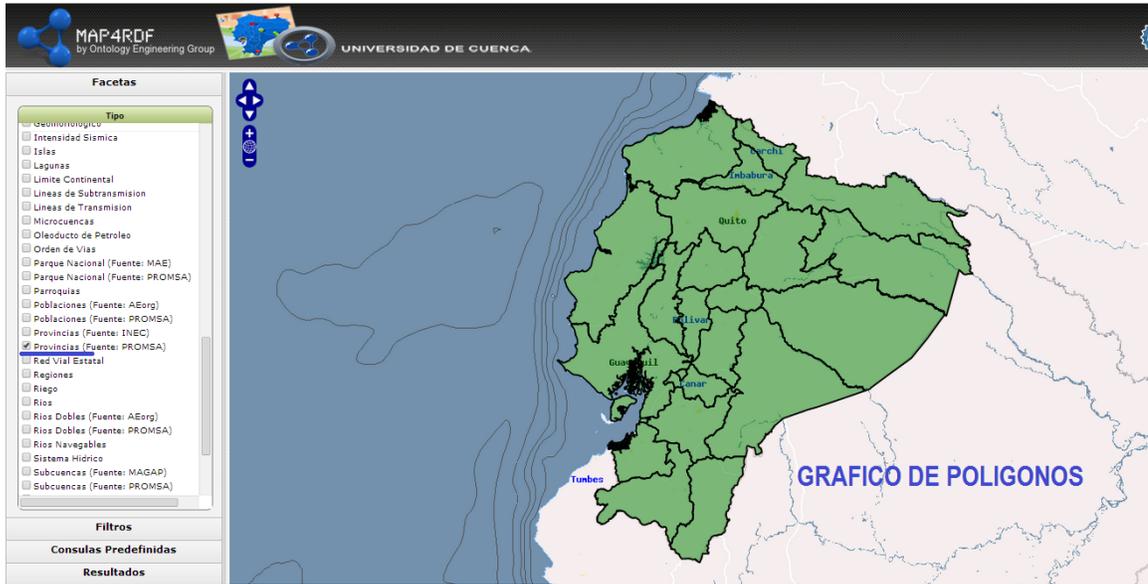


Figura A.8: Gráfico de Polígonos

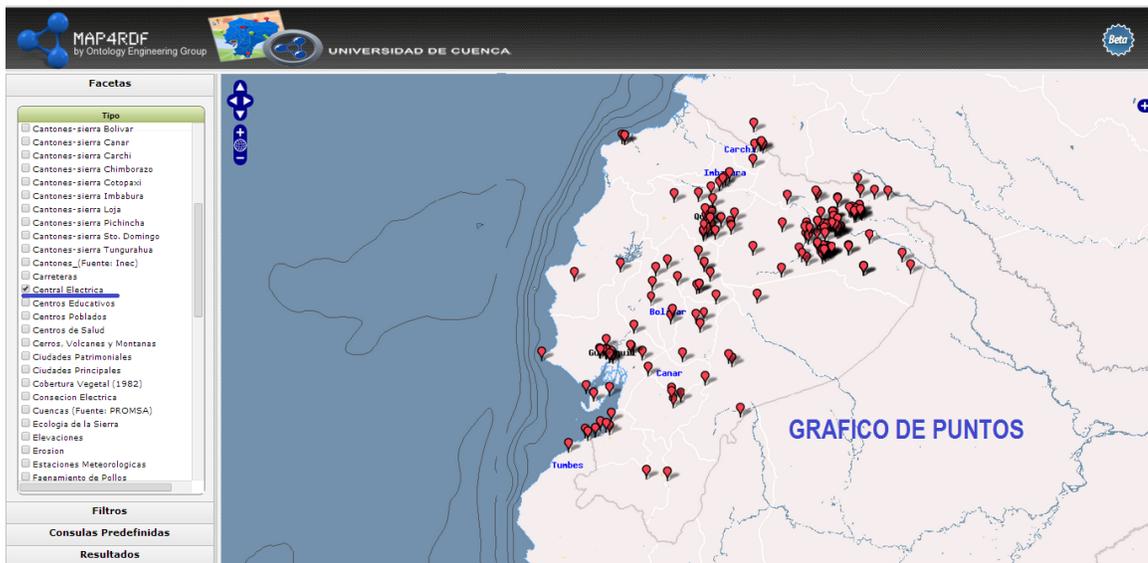


Figura A.9: Gráfico de Puntos

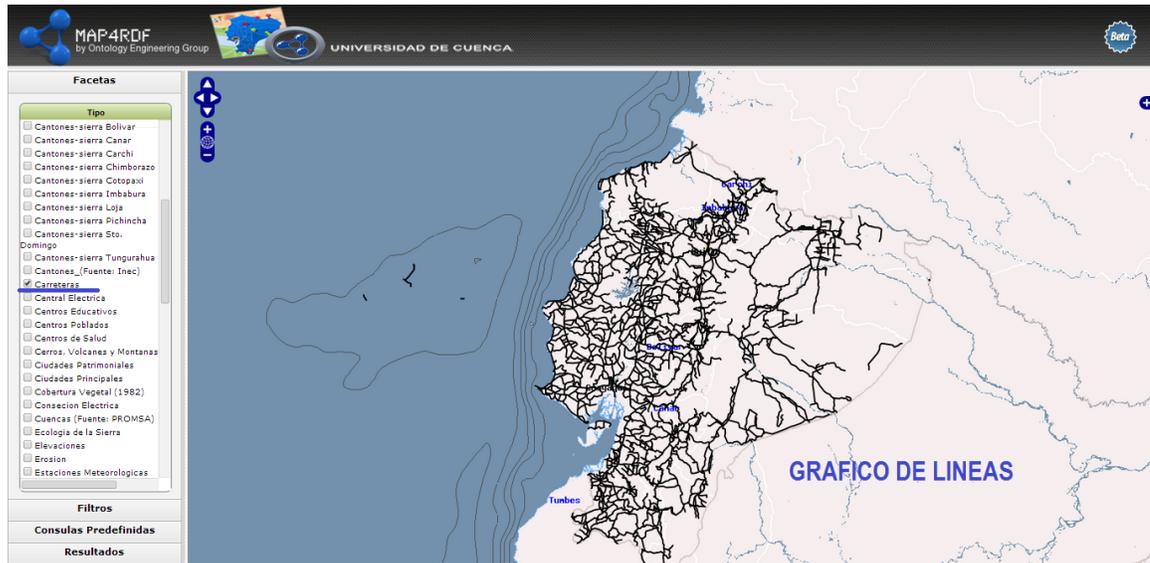
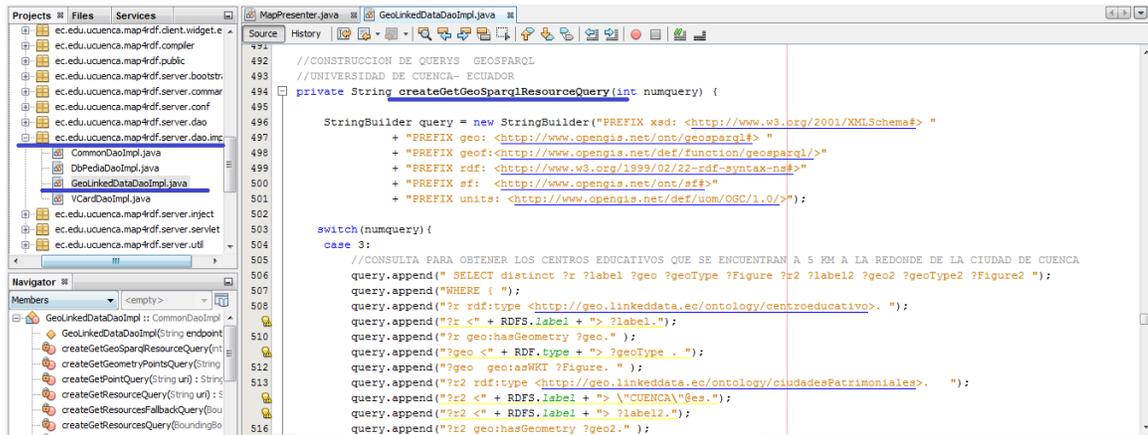


Figura A.10: Gráfico de Líneas

```
MapPresenter.java | GeoLinkedDataDaoImpl.java | OpenLayersMapLayer.java
74
75 public OpenLayersMapLayer(OpenLayersMapView owner, Map map, String name) {
76     this.owner = owner;
77     this.map = map;
78     VectorOptions vectorOptions = new VectorOptions();
79     vectorLayer = new Vector(name + "_vectors", vectorOptions);
80     vectorLayer.setDisplayInLayerSwitcher(false);
81     map.addLayer(vectorLayer);
82 }
83
84 @Override
85 public HasClickHandlers draw(Figure point) {
86     return addFeature(org.gwtopenmaps.openlayers.client.geometry.Geometry.fromWKT(point.getWKT()), getStylePoint());
87 }
88
89 @Override
90 public HasClickHandlers drawMultiPolygon(StyleMapShape<Figure> multipolygon) {
91     return addFeature(org.gwtopenmaps.openlayers.client.geometry.Geometry.fromWKT(multipolygon.getMapShape().getWKT()), c
92 }
93
94 @Override
95 public HasClickHandlers drawPolygon(StyleMapShape<Figure> polygon) {
96     return addFeature(org.gwtopenmaps.openlayers.client.geometry.Geometry.fromWKT(polygon.getMapShape().getWKT()));
97 }
98
99 @Override
100 public HasClickHandlers drawMultilineString(StyleMapShape<Figure> multilinestring) {
101     return addFeature(org.gwtopenmaps.openlayers.client.geometry.Geometry.fromWKT(multilinestring.getMapShape().getWKT()));
102 }
```

Figura A.11: Funciones para el gráfico de los recursos en OpenLayers con el formato WKT



```

492 //CONSTRUCCION DE QUERYS  GEOSPARQL
493 //UNIVERSIDAD DE CUENCA- ECUADOR
494 private String createGetGeoSparqlResourceQuery(int numquery) {
495
496     StringBuilder query = new StringBuilder("PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> "
497     + "PREFIX geo: <http://www.opengis.net/ont/geosparql#> "
498     + "PREFIX geof: <http://www.opengis.net/def/function/geosparql/>"
499     + "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
500     + "PREFIX sf: <http://www.opengis.net/ont/sf#>"
501     + "PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>");
502
503     switch(numquery){
504     case 3:
505         //CONSULTA PARA OBTENER LOS CENTROS EDUCATIVOS QUE SE ENCUENTRAN A 5 KM A LA REDONDE DE LA CIUDAD DE CUENCA
506         query.append(" SELECT distinct ?r ?label ?geo ?geoType ?Figure ?E2 ?label2 ?geo2 ?geoType2 ?Figure2 ");
507         query.append(" WHERE { ");
508         query.append("?r rdf:type <http://geo.linkeddata.ec/ontology/centroeducativo> . ");
509         query.append("?r <+ RDFS.label + '>' ?label.");
510         query.append("?r geo:hasGeometry ?geo . ");
511         query.append("?geo <+ RDFS.type + '>' ?geoType...");
512         query.append("?geo geo:asWKT ?Figure. ");
513         query.append("?r2 rdf:type <http://geo.linkeddata.ec/ontology/ciudadesPatrimoniales> . ");
514         query.append("?r2 <+ RDFS.label + '>' \"CUENCA\"@es.");
515         query.append("?r2 <+ RDFS.label + '>' ?label2.");
516         query.append("?r2 geo:hasGeometry ?geo2. ");

```

Figura A.12: Construcción del texto de la consulta GeoSPARQL

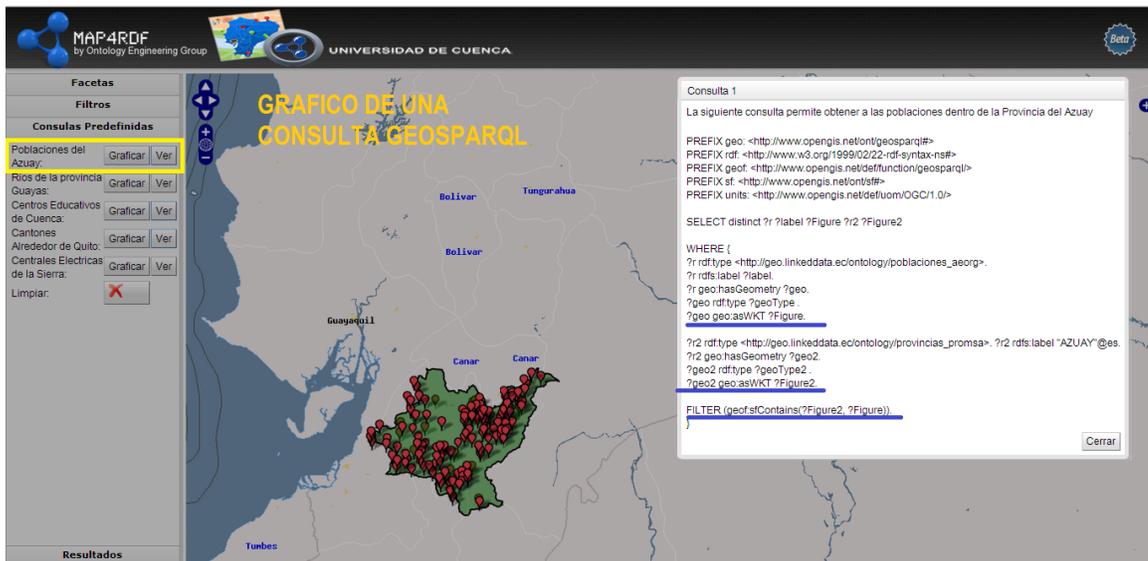


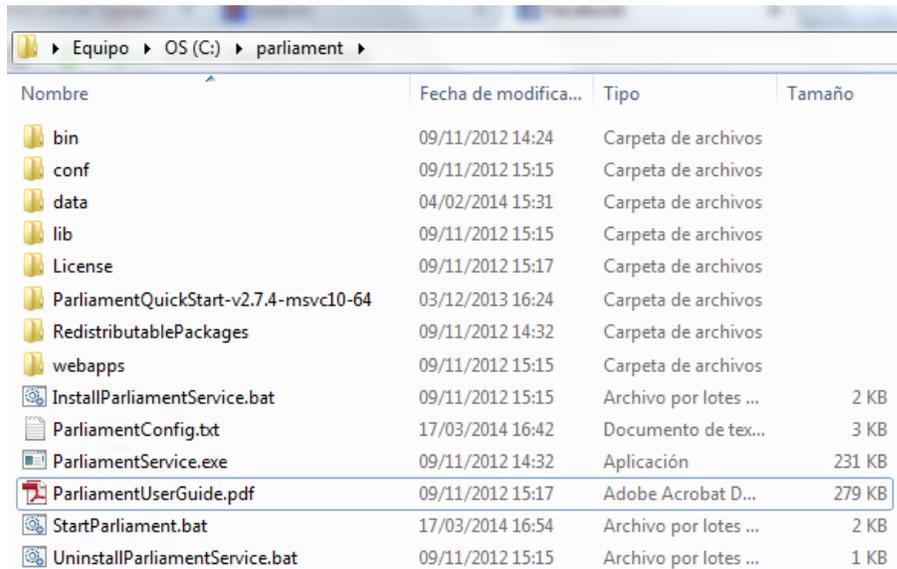
Figura A.13: Gráfico de la Consulta GeoSPARQL

Apéndice B

INSTALACIÓN DEL SERVIDOR PARLIAMENT

A continuación se describe la respectiva instalación del Triple Store Parliament, este programa se instaló sobre el sistema operativo Windows 7.

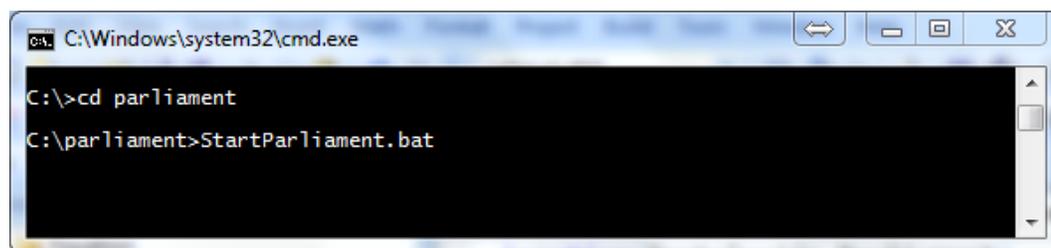
En principio se extrae los archivos y carpetas del archivo descargado como se presenta en la figura B.1 .



Nombre	Fecha de modifica...	Tipo	Tamaño
bin	09/11/2012 14:24	Carpeta de archivos	
conf	09/11/2012 15:15	Carpeta de archivos	
data	04/02/2014 15:31	Carpeta de archivos	
lib	09/11/2012 15:15	Carpeta de archivos	
License	09/11/2012 15:17	Carpeta de archivos	
ParliamentQuickStart-v2.7.4-msvc10-64	03/12/2013 16:24	Carpeta de archivos	
RedistributablePackages	09/11/2012 14:32	Carpeta de archivos	
webapps	09/11/2012 15:15	Carpeta de archivos	
InstallParliamentService.bat	09/11/2012 15:15	Archivo por lotes ...	2 KB
ParliamentConfig.txt	17/03/2014 16:42	Documento de tex...	3 KB
ParliamentService.exe	09/11/2012 14:32	Aplicación	231 KB
ParliamentUserGuide.pdf	09/11/2012 15:17	Adobe Acrobat D...	279 KB
StartParliament.bat	17/03/2014 16:54	Archivo por lotes ...	2 KB
UninstallParliamentService.bat	09/11/2012 15:15	Archivo por lotes ...	1 KB

Figura B.1: Archivos y carpetas descomprimidas de Parliament

A continuación se inicia el Triple Store Parliament desde la ventana de comandos, ubicarse en la carpeta donde se ha extraído todos los archivos, ejecutar el archivo StartParliament.bat como se muestra en la figura B.2 para iniciar los servicios de Parliament.



```
C:\Windows\system32\cmd.exe
C:\>cd parliament
C:\parliament>StartParliament.bat
```

Figura B.2: Comando para ejecución de Parliament

Por defecto el Triple Store Parliament inicia en el puerto 8080: “localhost:8080\parliament”. Posteriormente, cuando la aplicación ya ha iniciado, el siguiente paso, antes de empezar a cargar los archivos RDF, es crear índices espaciales, para esto es necesario dirigirse a la sección “indexes” como se presenta en la figura B.3, y así lograr el soporte de publicación y consulta de datos del estándar GEOSPARQL.

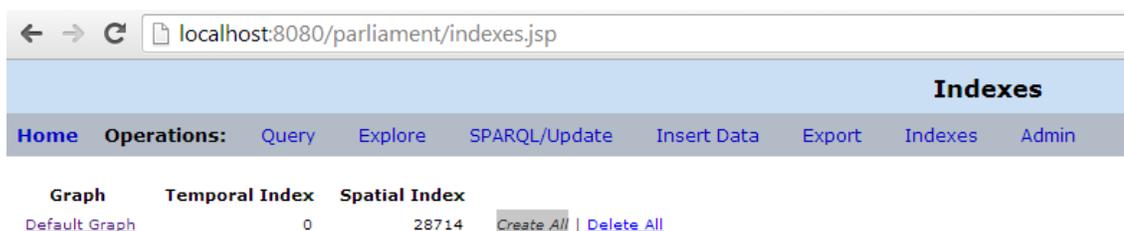


Figura B.3: Crear Indices Espaciales para soporte de consultas GEOSPARQL

Finalmente la aplicación ya está lista para permitir la publicación de archivos RDF y además soportar las consultas de información Geoespacial de RDF. Al realizar el proceso de creación de índices automáticamente se agregan los prefijos necesarios al momento de hacer consultas para el soporte de las funciones de análisis geoespacial del estándar GEOSPARQL como se muestra en la figura B.4.



← → ↻ apolo.utpl.edu.ec:8080/parliament/update.jsp

SPARQL/Update Query

[Home](#) **Operations:** [Query](#) [Explore](#) [SPARQL/Update](#) [Insert Data](#) [Export](#) [Indexes](#) [Admin](#)

SPARQL/Update Query

```
PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX fn: <http://www.w3.org/2005/xpath-functions#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX gml: <http://www.opengis.net/ont/gml#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX par: <http://parliament.semwebcentral.org/parliament#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sf: <http://www.opengis.net/ont/sf#>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

INSERT DATA
{
#triples
}

DELETE DATA
{
#triples
}
```

Prefijos agregados con la creacion de los "Spatial Index"

Figura B.4: Servidor listo para consultas Geoespaciales



Apéndice C

FUENTES DE INFORMACIÓN

En esta sección de anexos se muestran las direcciones Web de donde se obtuvieron las fuentes de información, en la figura C.1 se muestra la dirección Web con el nombre de la entidad o empresa respectivamente.

FUENTES	
PROMSA	http://www.uazuay.edu.ec/promsa/
INEC	http://www.inec.gob.ec/home/
IGEPN-STGR	http://www.igepn.edu.ec/
accionecologica.org	http://www.accionecologica.org/
MAGAP	http://www.agricultura.gob.ec/
MAE	http://www.ambiente.gob.ec/
DBPEDIA	http://dbpedia.org/About

Figura C.1: Fuentes de Información



Apéndice D

CONFIGURACIÓN PARA PUBLICACIÓN Y CONSUMO DE DATOS ENLAZADOS GEOESPACIALES

En esta sección se describen algunas características del equipo sobre el cual se ejecutaron el servidor Parliament y MAP4RDF. Los tiempos de respuesta para las consultas SPARQL y GEOSPARQL de la información generada en formato RDF dependen del Triple Store y del equipo sobre el cual se ejecute. Para realizar las pruebas descritas anteriormente se instaló Parliament en un equipo que tiene las características descritas en la tabla D.1.

Procesador	Intel i7, 2.2 Ghz
RAM	6 GB
Sistema Operativo	Windows 7

Cuadro D.1: Características de Equipo donde se realizo las consultas de prueba

Además, la aplicación MAP4RDF se ejecutó sobre el servidor de aplicación Jboss, para las consultas de prueba se utilizó la versión JBOSS Enterprise 6.0.1.GA como se muestra en la figura D.1 que fue instalada en el equipo mencionado anteriormente

en la tabla D.1.

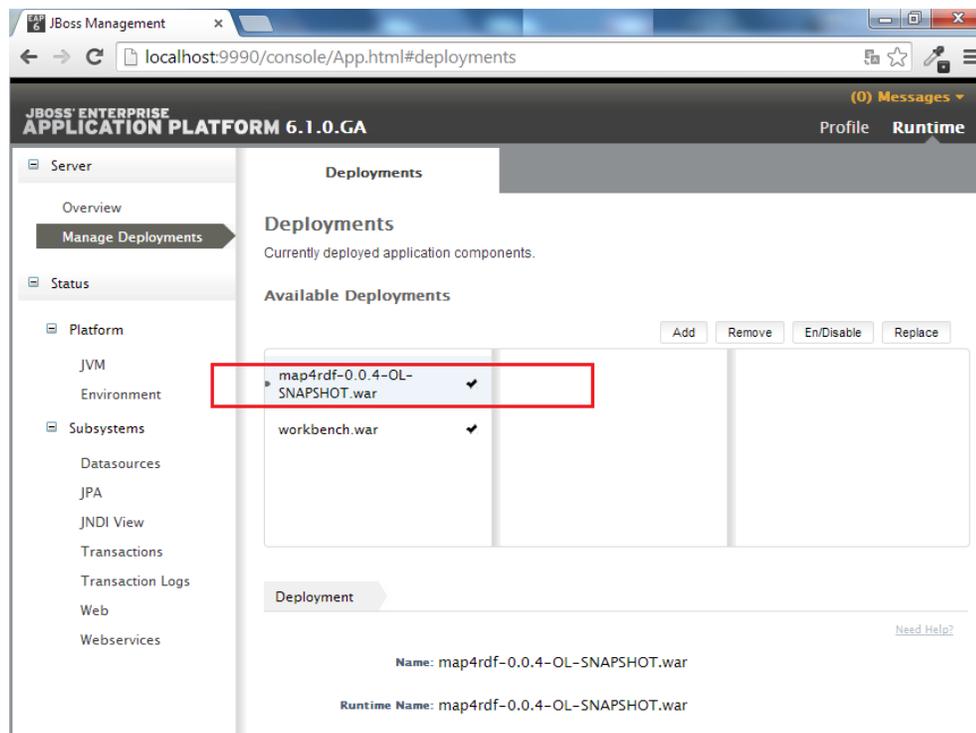


Figura D.1: Aplicación JBOSS sobre el cual se ejecuta MAP4RDF



Apéndice E

CREACIÓN E INSTALACIÓN DE UN PLUGIN EN GEOKETTLE

A continuación se describe el proceso tanto para la creación del plugin Geometry2RDF, como para su respectiva instalación dentro de GeoKettle.

E.0.1. Creación de un plugin para GeoKettle

En este apartado se tratará todos las herramientas requeridas, configuración y partes importantes de código, que es necesario entender para proceder a crear un plugin. Para tratar estos temas, esta sección se divide en lo siguiente:

- Herramientas usadas
- Estructura del proyecto en Eclipse
- Configuración del proyecto
- Funciones principales
- Configuración para depuración del plugin
- Publicación



Herramientas usadas

Para la creación del plugin Geometry2RDF, se usaron las siguientes herramientas:

- Windows 7 64bits.
- Eclipse Juno: Como IDE de desarrollo, se lo utilizó principalmente porque el plugin de ejemplo que proporciona la página web de GeoKettle, tiene un formato compatible con este entorno.
- GeoKettle 2.5: Es hasta este momento, la última versión de GeoKettle.
- Librerías base de GeoKettle: Librerías que estarán disponibles una vez instalado GeoKettle.

Proyecto en Eclipse

El plugin debe tener una estructura determinada, la cual ya ha sido definida por GeoKettle y se indica en la Figura E.1, donde muestra que deben existir las siguientes clases: Data, Dialog y Meta; además de sus archivos PNG y XML para que sea visible en el panel de GeoKettle.

- Clase Meta: Manejo general del plugin, se puede añadir columnas, se indica cuál ventana mostrar, etc.
- Clase Dialog: La venta de configuración.
- Clase Data: Manejo de datos generales, se puede utilizar para indicar los campos de salida.
- Clase Step: Maneja una por una las filas leídas y decide qué hacer con ellas.

Adicionalmente, se tiene el archivo *properties* en el cual se encuentra todo el texto visible en el plugin para que así sea más sencillo su traducción a otros idiomas.

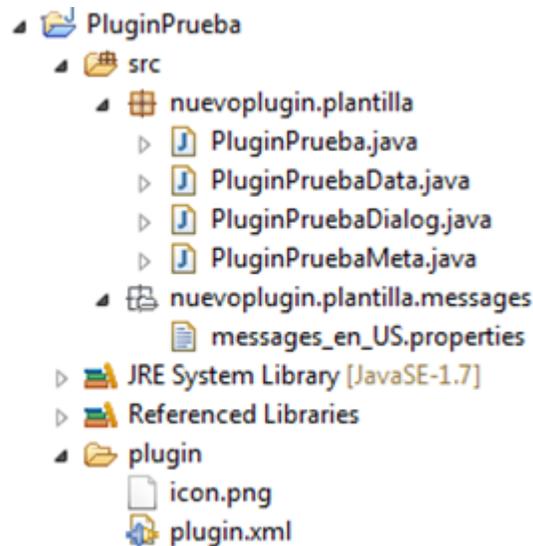


Figura E.1: Estructura del proyecto en Eclipse

Una vez lista la estructura del proyecto, se debe añadir las librerías correspondientes para que Eclipse reconozca las clases de GeoKettle y así poder usar todas las funciones que dispone. En la Figura E.2 se muestra la ventana de configuración de Eclipse (que se la obtiene en las propiedades del proyecto) y mostrando las librerías que ya se han agregado como son kettle-core.jar, kettle-db, kettle-engine.jar, kettle-ui-swt.jar, swt.jar.

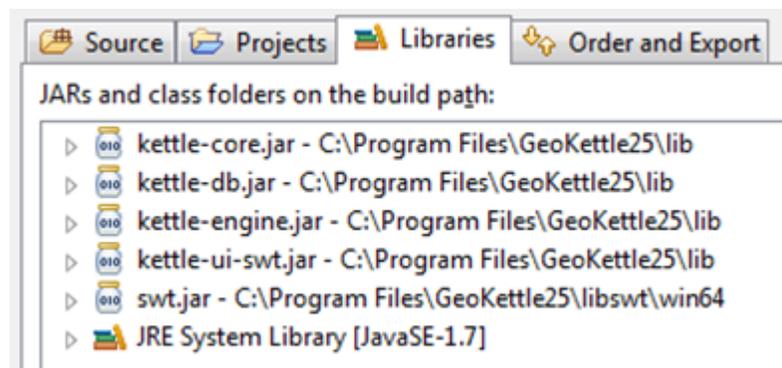


Figura E.2: Librerías de GeoKettle que deben ser agregadas a Eclipse



Lo siguiente a tomar en cuenta, es en el archivo `plugin.xml`, el cual contiene información para que pueda ser publicado dentro del panel de GeoKettle, es importante que toda la información de este archivo sea la correcta, caso contrario no se lo podrá usar el plugin creado. En la Figura E.3 se muestra un ejemplo de este archivo, donde se indican los siguientes parámetros:

- `iconfile`: Indica la ruta del archivo que será usado como ícono dentro del panel de GeoKettle.
- `description`: Alguna descripción de lo que hace el plugin.
- `tooltip`: Texto que se mostrará en GeoKettle al pasar el mouse sobre el ícono del plugin.
- `category`: La categoría de plugins en la que se ubicará, si no existe en las predefinidas en GeoKettle, se creará una nueva.
- `classname`: La ruta de paquetes hasta llegar a la clase Meta del plugin.
- `libraries`: Se debe incluir la ruta del archivo JAR perteneciente al plugin.
- `localized_tooltip`: Es lo mismo que `tooltip`, pero en este caso varía de acuerdo al idioma elegido en GeoKettle.

Funciones principales

En la parte de la programación del plugin, GeoKettle nos brinda funciones que deben ser usadas dentro de sus respectivas clases principales, las cuales servirán desde configurar el plugin, hasta la acción independiente que realice el plugin.

En la clase Meta se debe tomar en cuenta estas funciones:

- `check`: Para revisar que la información de entrada sea válida.
- `getStep`: Para llamar a la clase Step.
- `getStepData`: Para llamar a la clase Data.
- `loadXML`, `readXML`: Para guardar o cargar la configuración.

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin

  id="PruebaPlugin"
  iconfile="icon.png"
  description="PruebaPlugin"
  tooltip="Plugin de prueba"
  category="Demonstration"
  classname="nuevoplugin.plantilla.PluginPruebaMeta">

  <libraries>
    <library name="pruebaplugin.jar"/>
  </libraries>

  <localized_tooltip>
    <tooltip locale="en_US">Test plugin</tooltip>
    <tooltip locale="es_ES">Prueba plugin.</tooltip>
  </localized_tooltip>

</plugin>
```

Figura E.3: Contenido del archivo plugin.xml

- getFields: Para manejar los campos o columnas recibidas.

En la clase Dialog, la funciones principales son:

- Open: Es utilizada para graficar toda la interfaz de la ventana de configuración del plugin.

En la clase Step, se utiliza las siguientes funciones:

- run: Se ejecuta cuando el plugin es ejecutado en algún proceso del ETL.
- getRow: Para obtener la fila obtenida.
- putRow: Para colocar una fila como salida del plugin.



Configuración para la depuración del plugin

Debido a que GeoKettle está originalmente destinado para usarlo como usuario, y no como desarrollador, de forma preterminada no tiene activada ninguna opción de depuración, por lo tanto es necesario ciertas configuraciones tanto en archivos como en Eclipse. Esto principalmente ayudará bastante a detectar errores en el plugin que se esté desarrollando.

En primer lugar, se debe buscar el archivo *spoon.bat* dentro del directorio donde está instalado GeoKettle. Una vez hallado, se modifica las últimas líneas de ese archivo, para que queden como las siguientes:

```
set JAVA_OPTS=-Xdebug -Xrunjwp:transport=dt_socket,address=8000,  
server=y,suspend=n  
start javaw %OPT% %JAVA_OPTS% org.pentaho.di.ui.spoon.Spoon  
%_cmdline%
```

Después de la modificación del archivo, se debe configurar Eclipse para que pueda capturar la ejecución del plugin. En la Figura E.4 muestra la ventana de configuración de Eclipse (se la obtiene en el menú *Run - Debug Configurations...*) donde indica que tiene como *Host* a *localhost* y como puerto al 8000. También se debe revisar que el proyecto al que se aplica la configuración, sea el correcto. Como opción opcional, se tiene la opción *Allow termination of remote VM* la cual permite que si se detiene la depuración en Eclipse, también cierra GeoKettle. Por último, se ejecuta GeoKettle y se da clic en *Debug* para iniciar la depuración.

Publicación

Para publicar el plugin y que éste se pueda distribuir, es necesario generar el archivo JAR respectivo, para lo cual primeramente se da clic derecho sobre la carpeta *src* del proyecto y se elige la opción *Export...*, una vez aparezca la ventana de configuración, se elige la opción *JAR file* tal y como muestra la Figura E.5

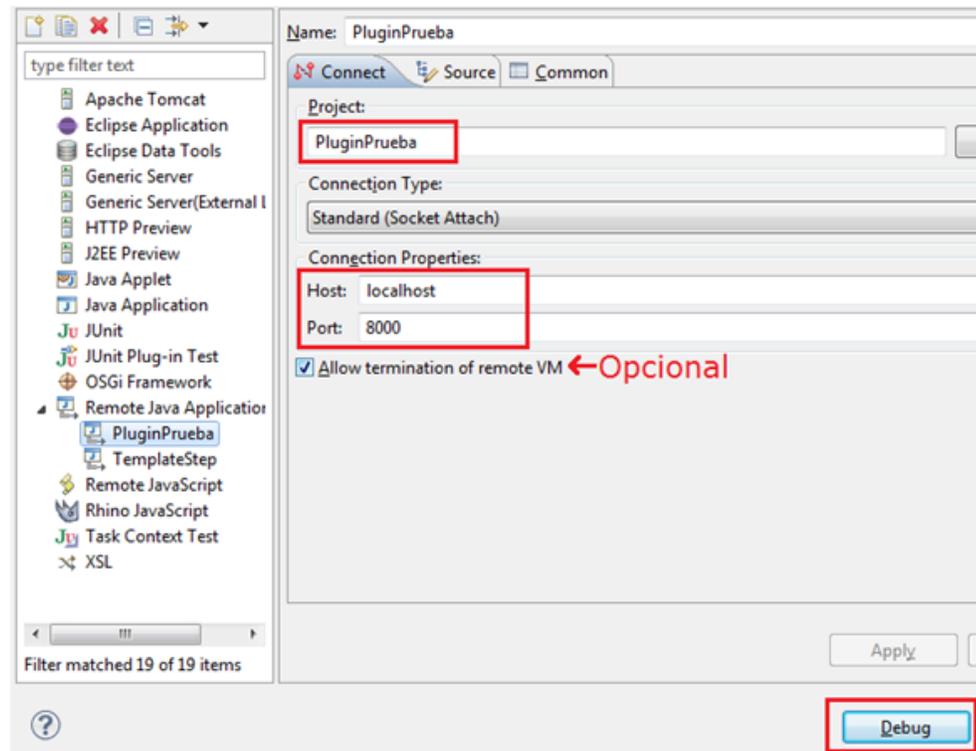


Figura E.4: Configuración de depuración en Eclipse

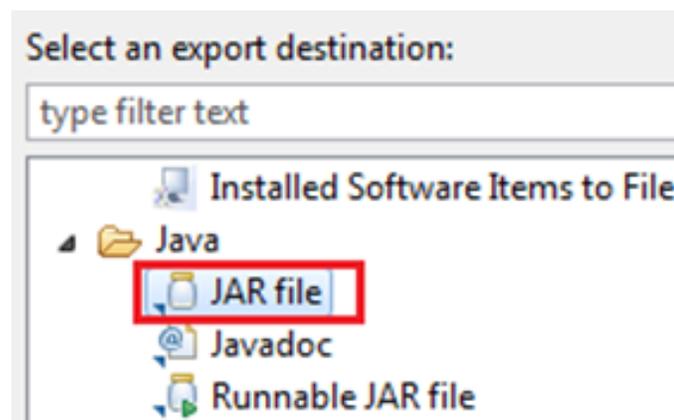


Figura E.5: Opción para generar el archivo JAR

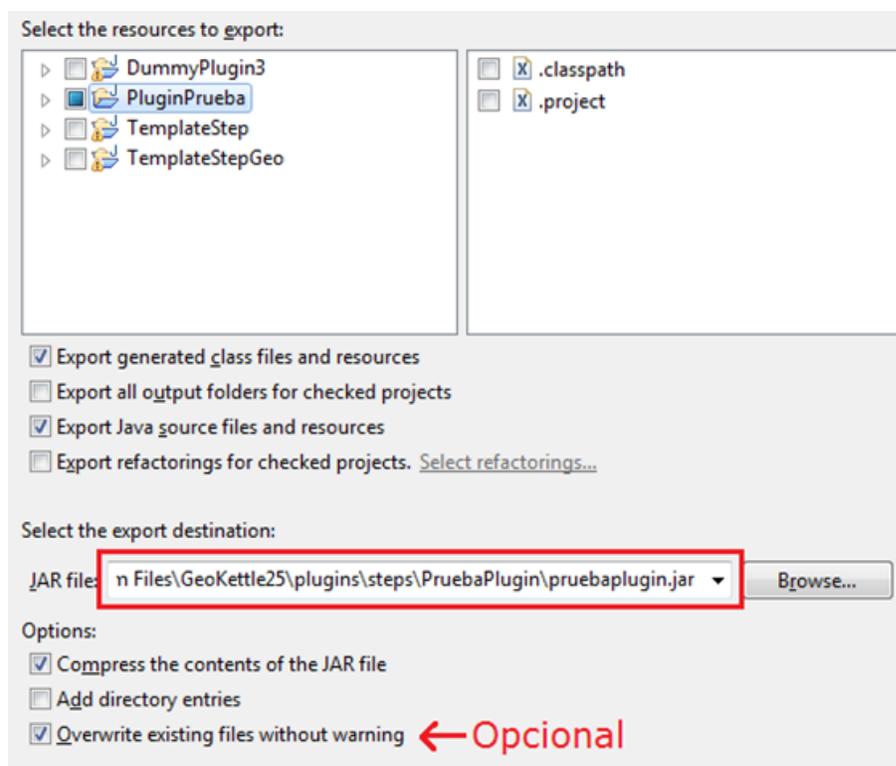


Figura E.6: Configuración para generar el archivo JAR

Una vez elegido el tipo de archivo a generar, se mostrará la ventana de configuración de la Figura E.6, donde se debe especificar la dirección donde se almacenará el archivo JAR y como opcional, si se quiere que sobrescriba el archivo (en caso de ya existir) sin ningún aviso.

Lo último a tener en cuenta es el archivo *plugin.xml* y PNG, ya que ambos deben ir junto al archivo JAR generado, ya que caso contrario, GeoKettle no lo reconocerá como plugin y no lo mostrará en su panel.

E.0.2. Instalación del plugin Geometry2RDF en GeoKettle

Primero, para que el plugin pueda ser visualizado en la lista de GeoKettle, se lo debe colocar dentro de la carpeta plugins en el directorio donde se instaló GeoKettle, en la carpeta respectiva, tal y como muestra la Figura E.7 donde el plugin se ubica en C:\Program Files\GeoKettle25\plugins\steps\Geometry2RDF.

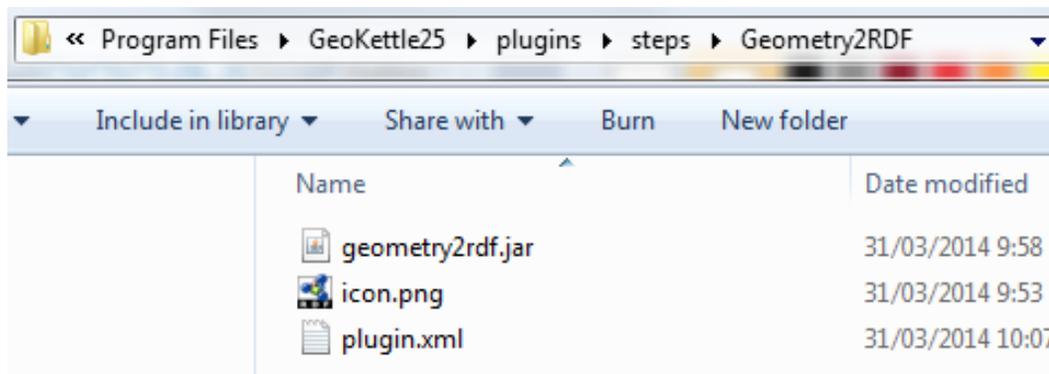


Figura E.7: Directorio donde se debe copiar el plugin Geometry2RDF

Posteriormente, se realiza el armado del esquema para la generación del plugin, lo que da como resultado la estructura de la Figura E.8.

El esquema propuesto consta de los siguientes plugins:

- Lector de archivos SHP y transformador a formato WKT.
- Lector de variables, sirve para colocar en la salida, el path completo del archivo que se está transformando, este dato es necesario en el plugin para la obtención de los metadatos.
- Transformador de coordenadas.
- Plugin Geometry2RDF, para obtener finalmente el RDF.
- Filtrado, para separar los datos de los metadatos generados por Geometry2RDF.
- Salida de archivos, para almacenar el RDF generado, en una determinada ubicación.

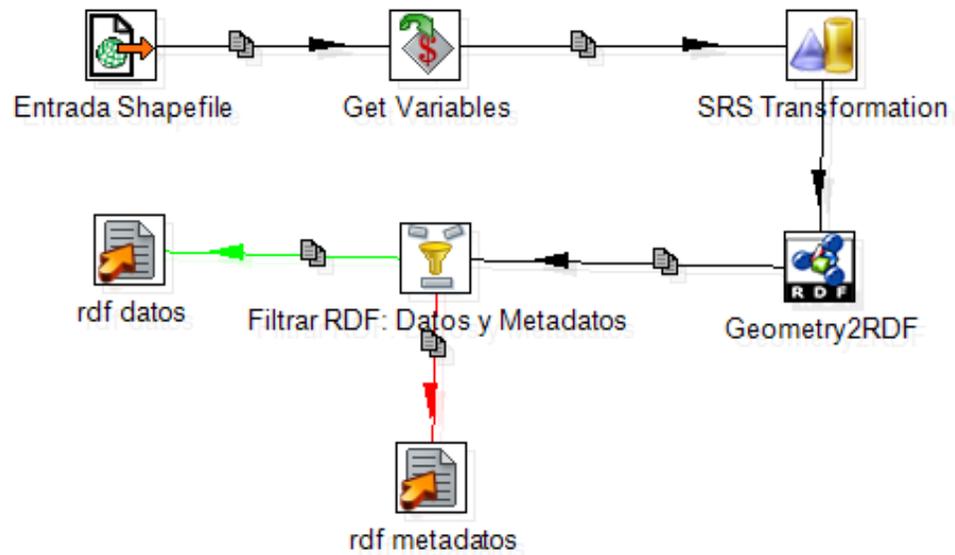


Figura E.8: Modelo de transformación usando el plugin Geometry2RDF



Glosario

D

DBpedia Proyecto para la extracción de datos de Wikipedia para proponer una versión Web Semántica., pág. 21.

E

ETL (Extract, Transform and Load). Extraer, transformar y cargar, es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizarlos, pág. 5.

G

GEOSPARQL Estándar establecido por Open Geospatial Consortium (OGC) para la representación de datos geospaciales en la Web Semántica., pág. 2.

GML Su fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web., pág. 32.

H

HTML (HyperText Markup Language). Hace referencia al lenguaje de programación para la elaboración de paginas Web., pág. 19.

HTTP (Hypertext Transfer Protocol) Protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la World Wide Web., pág. 27.



O

OGC (Open Geospatial Consortium). Su fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web., pág. 28.

OPENLAYERS Biblioteca de JavaScript de código abierto para mostrar mapas interactivos en los navegadores Web, pág. 55.

OWL (Web Ontology Language). Lenguaje de la web semántica diseñado para representar el conocimiento enriquecido y las respectivas relaciones de las cosas., pág. 19.

R

RDF (Resource Description Framework). Recomendación de la W3C, estándar para intercambio de datos en la Web., pág. 2.

RIF (Rule Interchange Format). Recomendación de la W3C, es parte de la estructura de la Web Semántica., pág. 28.

S

SPARQL (SPARQL Protocol and RDF Query Language) Especificación que ofrece un lenguaje y protocolos para consultar y manipular RDF, pág. 2.

W

W3C (World Wide Web Consortium) Es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo., pág. 27.

WKT (Well know Text). Codificación o sintaxis estandarizada, diseñada para describir objetos espaciales expresados de forma vectorial como Puntos Lineas o polígonos., pág. 32.

WWW (Word Wide Web). Red Mundial., pág. 23.



X

XML (eXtensible Markup Language). Lenguaje de marcas extensible, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible., pág. 19.



Bibliografía

- [1] Paul Davidson, CIO, y Sedgemoor District Council. Designing uri sets for the uk public sector. *Klin. Lab. Diagn.*, 1, 2009.
- [2] Alexander de Leon¹, Filip Wisniewski², Boris Villazon-Terrazas², y Oscar Corcho². *MAP4RDF*. 1 Lumata, Madrid, Spain me@alexdeleon.name 2 OEG-DIA, FI, Universidad Politécnica de Madrid, Spain {ffwisniewski,bvillazon,ocorchog}@fi.upm.es, <http://oeg-upm.github.io/map4rdf/>, 2009.
- [3] The Open Source Geospatial Foundation. *GeoTools User Guide*. The Open Source Geospatial Foundation, <http://docs.geotools.org/latest/userguide/>.
- [4] Open Source Geospatial Business Intelligence (GeoBI). *GeoKettle Project*. Open Source Geospatial Business Intelligence (GeoBI), <http://www.spatialytics.org/projects/geokettle/>.
- [5] RDF Working Group. *RDF*. RDF Working Group, <http://www.w3.org/RDF/>, .
- [6] W3C SPARQL Working Group. *SPARQL*. W3C, <http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/>, .
- [7] Jeff Heflin (Lehigh University) heflin@cse.lehigh.edu. *ONTOLOGIA*. W3C, <http://www.w3.org/TR/2004/REC-webont-req-20040210/>.
- [8] Apache Jena. *Getting started with Apache Jena*. Apache, <http://jena.apache.org/>.



- [9] Vilches-Blázquez Luis M., Sevilla Celia, Villalón Miguel, Rodríguez Antonio F., y Gómez-Pérez Asunción. Combinando linked data con servicios geoespaciales.
- [10] Sem Web Central Organization. *ParliamentTM A High-Performance Triple Store, SPARQL Endpoint, and Reasoner*. <http://parliament.semwebcentral.org/>.
- [11] Matthew Perry y John Herring. *Ogc geosparql - a geographic query language for rdf data*. 1.0, 2012.
- [12] Matthew Perry y Xavier Lopez. Bringing location analysis to the semantic web with the ogc geosparql standard.
- [13] silk. *SILK*. silk, silk, 1000.
- [14] Nova Spivack. *Evolucion de la Web Semantica*. <http://www.novaspivack.com/>.
- [15] Luis. M. Vilches-Bazquez, Boris Villazon-Terrazas, Alexander De Leon, Freddy Priyatna, y Oscar Corcho. An approach to publish spatial data on the web: The geolinked data case. 1.
- [16] Boris Villazón-Terrazas y Oscar Corcho. *Methodological Guidelines for Publishing Linked Data*. Universidad Politécnica de Madrid, Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid.
- [17] W3C. *LINKED DATA*. W3C, <http://www.w3c.es/Divulgacion/GuiasBreves/LinkedData>.
- [18] World Wide Web Consortium (W3C). *WEB SEMANTICA*. World Wide Web Consortium (W3C), <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>.