



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Grado en Ingeniería Informática

Trabajo Fin de Grado

**Actualizar las Herramientas de Linked
Data Geográfico utilizadas por el Grupo
de Ingeniería Ontológica**

Autor: Beñat Agirre Arruabarrena
Tutor(a): Oscar Corcho García

Madrid, Junio - 2021

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ingeniería Informática

Título: Actualizar las Herramientas de Linked Data Geográfico utilizadas por el Grupo de Ingeniería Ontológica

Junio - 2021

Autor: Beñat Agirre Arruabarrena
Tutor: Oscar Corcho García
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

El Ontology Engineering Group lleva más de una década trabajando con datos geográficos enlazados españoles. En 2010[1] se definió un caso de uso y en 2019[2] se refinó el proceso de generación y publicación de los datos abiertos utilizando el dataset BTN100 como caso de estudio. En los últimos años han se han popularizado nuevas herramientas y formatos que ofrecen ventajas no disponibles en las usadas hasta el momento. Entre ellas se encuentran el programa de transformaciones de datos PDI9 Kettle, que reemplaza a GEOKettle; el formato GeoPackage, más práctico que el shapefile; y Apache Maven, más extenso que Apache Ant. Por consiguiente, también será necesario actualizar tripleGeo, el plugin para GEOKettle desarrollado por el OEG, para integrarlo en el nuevo toolbox.

Abstract

The Ontology Engineering Group has been working with Spanish geographical linked data for over a decade. In 2010[1] a use case was defined, and in 2019[2] the open data generation and publication process was refined in a case study with BTN100. New tools and formats have gained popularity over the past few years, which offer more advantages over the ones used to date. This includes the data transformation tool PDI9 Kettle, which replaces GEOKettle; Geopackage, which is more practical than the Shapefile format; and Apache Maven, more fully-featured than Apache Ant. Thus, the plugin developed for GEOKettle by the OEG, tripleGeo, will also need to be updated.

Tabla de contenidos

1. Introducción	1
1.1. Objetivos	1
1.1.1. Objetivos iniciales	1
1.1.2. Objetivos actualizados	1
1.2. Estado del Arte	2
1.2.1. GIS	2
1.2.1.1. Shapefile	3
1.2.1.2. GeoPackage	4
1.2.2. Datos enlazados	4
1.2.3. Portales de Datos abiertos	5
1.2.4. Map4RDF	5
1.2.5. GeoKettle y tripleGEO	6
1.3. Pentaho Data Integration Spoon	7
1.3.1. Pentaho GIS Plugins	7
1.3.2. Apache Ant	8
1.3.3. Apache Maven	8
2. Desarrollo	9
2.1. GeoKettle	9
2.1.1. Funcionamiento PDI	10
2.1.1.1. Shapefile File Input	11
2.1.1.2. Switch case	12
2.1.1.3. Dummy Plugin	13
2.1.1.4. Valor JavaScript Modificado	13
2.1.1.5. Selecciona/Renombrar valores	14
2.1.1.6. SRS Transformation	15
2.1.1.7. TripleGeoKettle	15
2.1.1.8. Text file output	16
2.1.1.9. Resultado de la transformación	17
2.2. Port a PDI9	18
2.2.1. Tests con pentaho-gis-plugins	18
2.2.2. Entorno de desarrollo	18
2.2.2.1. Dependencias e instalación	18
2.2.2.2. Pentaho sdk plugins	21
2.3. Proceso de porteo	21
2.3.1. Diseño del plugin	21

2.3.2. Resultado parcial	24
2.3.3. Debugging	24
2.3.4. Problemas de dependencias	26
2.4. Adaptación de las transformaciones	28
2.5. Soporte GeoPackage	29
2.6. Problemas encontrados	31
2.6.1. Out of memory	31
2.6.2. Línea eléctrica	31
3. Resultados y conclusiones	33
4. Análisis de impacto	35
4.1. Personal	35
4.2. Empresarial	36
4.3. Social	36
4.4. Medioambiental	36
Bibliografía	39
Anexos	40
A. Anexos	41
A.1. Glosario de términos	41

Capítulo 1

Introducción

1.1. Objetivos

El objetivo principal del trabajo es modernizar las herramientas de Linked Data Geográfico desarrolladas por el Grupo de Ingeniería Ontológica. En el OEG se ha venido tradicionalmente trabajando con el Instituto Geográfico Nacional para la exportación de algunos de sus datos geográficos a formato Linked Data. Un ejemplo se puede encontrar en la web del Instituto Geográfico Nacional. [7]

1.1.1. Objetivos iniciales

Cuando se presentó la primera propuesta del TFG, los objetivos eran los siguientes:

- Dar soporte GeoPackage a la herramienta Map4RDF
- Dar soporte GeoPackage a la herramienta GeoKettle y su plugin para transformar a RDF
- Realizar un procesamiento completo de todos los datos del IGN para generar este tipo de formato.

1.1.2. Objetivos actualizados

Desde que se publicó “A sustainable process and toolbox for geographical linked data generation and publication: a case study with BTN100” en 2019[2], GeoKettle ha dejado de estar soportado. La página oficial y de documentación ya no están disponibles. Algunas funcionalidades de GeoKettle se integraron en PDI directamente y otras desaparecieron. Actualmente, el soporte GIS de Pentaho está dentro de PDI Spoon y además hay algunas funcionalidades más en el plugin llamado pentaho-gis-plugins[25].

Por otro lado, recientemente, el Open Geospatial Consortium ha publicado el formato GeoPackage, que tiene el objetivo de convertirse en un estándar para la representación de datos geográficos. El siguiente objetivo de este trabajo es el de dar soporte GeoPackage para las herramientas normalmente utilizadas para

este tipo de tareas. Ya que el 3 de abril de 2021 pentaho-gis-plugins añadió soporte geopackage a su plugin, PDI9 ya tiene soporte para geopackage.

En cuanto a Map4RDF, tras analizar su funcionamiento se concluyó que no requiere soporte para GeoPackage explícito, ya que tras las transformaciones realizadas los datos son iguales, independientemente del formato de donde se hayan obtenido.

En resumen, los objetivos actualizados son los siguientes:

1. Replicar la funcionalidad y las transformaciones de GeoKettle + TripleGeo en la nueva suite PDI.
2. Dar soporte GeoPackage al plugin para transformar a RDF y las transformaciones existentes.
3. Realizar un procesado completo de todos los datos del IGN con las nuevas transformaciones.

1.2. Estado del Arte

1.2.1. GIS

Los sistemas de información geográfica son herramientas que permiten almacenar y analizar datos geoespaciales. Los sistemas digitales actuales permiten realizar consultas interactivas, añadir entradas a las bases de datos y visualizarlos de manera intuitiva. La información geográfica se puede aplicar a todo tipo de áreas, entre las que se encuentran la ingeniería, transporte, telecomunicación, economía, sociología... Debido a la gran importancia tanto en el sector público como el privado[5], los estándares abiertos cobran importancia por estar disponibles al público, no tener que pagar licencias y ser consensuados por organizaciones de estándares internacionales. Entre ellas se encuentra el Open Geospatial Consortium(OEG) que se creó en 1994 y agrupa a 521 (en marzo de 2021) miembros de organizaciones públicas y privadas.[6] El OGC trabaja junto con las principales organizaciones de estándares de su ámbito (ISO/TC 211, W3C, IETF...) [4]

Existen diversos formatos de fichero GIS, divididos en **raster** y **vector**. La diferencia es equivalente a la que existe entre imágenes con resolución limitada por el número de pixels (raster) y las imágenes vectoriales formadas por puntos, líneas y polígonos; con resoluciones infinitas. Cada tipo de formato tiene sus ventajas y desventajas y la elección dependerá del caso de uso. Existen varios formatos vectoriales pero para este trabajo sólo se se considerarán el formato *shapefile* y el *GeoPackage* para cumplir los objetivos.

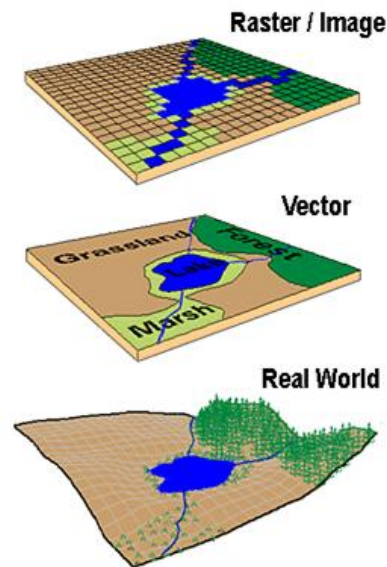


Figura 1.1: Representación del terreno mediante vectores y raster

1.2.1.1. Shapefile

El formato ESRI Shapefile (SHP) es un formato de archivo de datos espaciales vectorial desarrollado por la compañía ESRI a principios de la década de 1990. A pesar de ser propietario, la especificación es abierta, y se considera un estándar de facto. Debido a su popularidad, goza de gran compatibilidad con SIG. Gracias al uso de un fichero índice, se obtiene una velocidad de lectura alta, y su eficiencia de tamaño produce archivos relativamente pequeños.

Sin embargo, tiene varias desventajas, algunas derivadas del uso del estándar dBase [11] [14]:

1. No tiene definición de sistema de referencia de coordenadas ¹, se puede usar uno pero no es parte estándar de la especificación.
2. Se reparte en múltiples ficheros: es incómodo y lleva a errores al compartirlos.
3. Los nombres de atributos están limitados a 10 caracteres ASCII
4. El número máximo de campos de atributo es 255.
5. Solo admite float, integer, date y text con un máximo de 254 caracteres.
6. No se puede especificar el conjunto de caracteres de la BBDD.
7. El tamaño está limitado a 4GB.
8. No admite valores NULL

¹Un sistema de coordenadas es un sistema de referencia que se utiliza para representar la ubicación de entidades geográficas, imágenes y observaciones (como las localizaciones GPS) dentro de un marco geográfico común. Los sistemas de coordenadas permiten a los datasets geográficos utilizar ubicaciones comunes para la integración de datasets. [15]

9. No hay forma de describir las relaciones topológicas en el formato.
10. Solamente puede almacenar una geometría por archivo.
11. Utiliza una estructura de datos de tabla plana, sin jerarquías, relaciones ni estructura en árbol.
12. El soporte 3D es muy limitado.

1.2.1.2. GeoPackage

GeoPackage es un formato GIS implementado en SQLite publicado por el OEG en 2014. [12]

El formato geopackage tiene las siguientes ventajas [14]:

- Es abierto, no propietario, basado en estándares, independiente de plataformas, portable y compacto.
- Gracias a SQLite puede almacenar datos grandes (hasta 140TB)[17] y los atributos de las geometrías pueden contener nombres muy largos.
- Dispone de índices espaciales basados en R-trees [16] que incrementan la velocidad de búsquedas espaciales y su visualización en los SIG de escritorio.
- Todo el contenido se almacena en un único archivo .gpkg que puede almacenar multitud de tipos de geometrías
- Soporta el uso directo, para acceder a los datos de GeoPackage de forma «nativa» sin traducciones de formato intermedio.
- GeoPackage es soportado por GDAL[18], la librería de conversión de datos utilizada por multitud de programas GIS, y los principales programas GIS.

1.2.2. Datos enlazados

El objetivo de los datos enlazados es utilizar la web como una única base de datos global. Tim Berners Lee, creador de la World Wide Web, quien acuñó el término linked data[13], definió sus 4 principios fundamentales:

1. Utilizar URIs para identificar los recursos publicados en la Web.
2. Utilizar URIs HTTP para que las personas puedan consultar esos recursos.
3. Cuando alguien acceda a una URI, proporcionar información útil mediante estándares (RDF*, SPARQL).
4. Incluir enlaces a otras URIs para facilitar el descubrimiento de más información relacionada.

Los datos enlazados posibilitan la web semántica, extensión de la web tradicional en la que la información tiene significado bien definido[19] y fundamentada en:

- URIs: cadena de caracteres que identifica los recursos de una red de forma unívoca.

Introducción

- RDF: método para la descripción conceptual o modelado de la información.
- HTTP: protocolo de comunicación.

RDF modela información mediante triples o tripletas de sujeto-predicado-objeto. El sujeto hace referencia al recurso y el predicado a sus rasgos o aspectos y relación entre el sujeto y el objeto. SPARQL es el lenguaje para la consulta de grafos RDF.

1.2.3. Portales de Datos abiertos

Los datos abiertos parten de la idea de que los datos deberían estar disponibles de forma libre para todo el mundo, libre de derechos de autor, patentes o de otros mecanismos de control. Los portales de datos abiertos proporcionan una manera sencilla de buscar y obtener estos datos. Los datos pueden tener cualquier procedencia, pero han cobrado especial importancia los datos ligados a las políticas de Gobierno abierto, que persigue que los datos y la información, especialmente las que poseen las administraciones públicas, se publiquen de forma abierta. [20]

El tercer objetivo de este trabajo se centra en realizar un procesado completo de todos los datos del IGN. *datos.ign.es* es una iniciativa del Instituto Geográfico Nacional (IGN) para la generación de la información semántica de sus recursos[21]. Actualmente el dataset disponible es la Base Topográfica Nacional 1:100.000 (BTN100), un catálogo de datos geográficos agrupados por temáticas.

1.2.4. Map4RDF

Map4rdf es una herramienta para la navegación y visualización de datasets RDF con información geoespacial mediante facetas[8]. Algunos ejemplos de las facetas y sus contenidos que permiten clasificar los elementos del BTN100:

- Altimetría: Cerro, Cordillera, Montaña...
- Hidrografía: Bahía, Cabo, Playa...
- Transporte: Aeropuerto, Calle, Faro...
- ...

El funcionamiento de Map4rdf es el siguiente:

1. El componente *DAO*² se conecta a una *triplestore*³ mediante el *endpoint SPARQL*⁴ para responder a las consultas de facetas.
2. La interfaz de navegación facetada obtiene la lista de facetas y las visualiza.
3. El usuario selecciona una faceta y el componente DAO realiza una consulta en el triplestore mediante el endpoint SPARQL para recuperarlas la información pedida.

²Data Acces Object: proporciona una interfaz abstracta a una base de datos.

³Triplestore: base de datos de tripletas

⁴SPARQL endpoint: url capaz de recibir y procesar peticiones del protocolo SPARQL

4. La interfaz recibe toda esta información y la visualiza en el mapa.

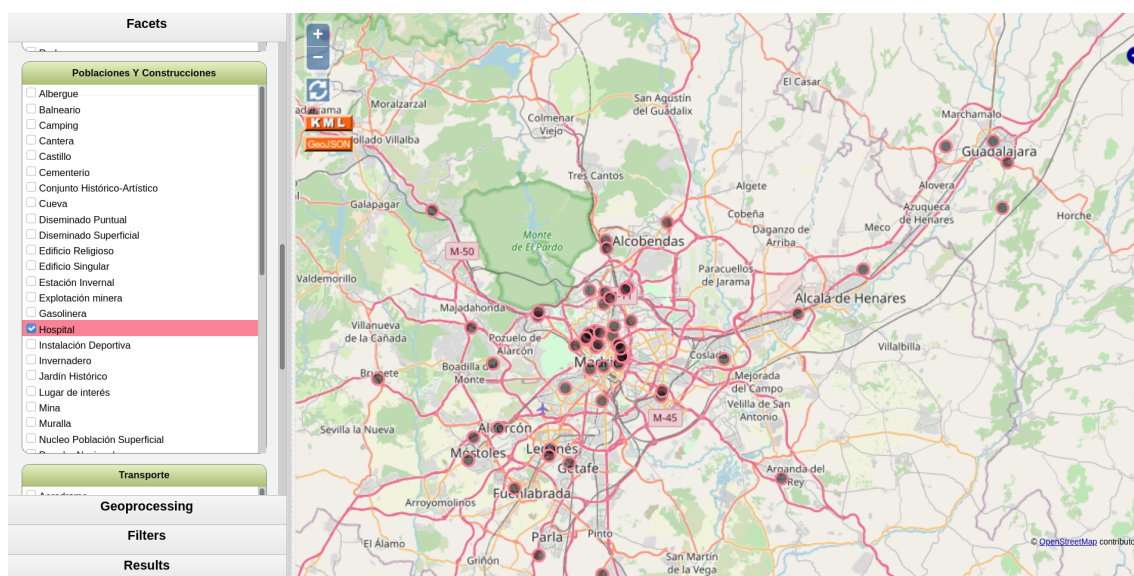


Figura 1.2: <http://certidatos.ign.es/map/> que implementa Map4Rdf

1.2.5. GeoKettle y tripleGEO

GeoKettle es una (antigua) versión de Pentaho Data Integration (Kettle)[23] con capacidad de tratamiento de datos espaciales. Es una potente herramienta ETL: extracción, transformación y carga orientada al uso de metadatos y con funcionalidades espaciales dedicada a la integración de diversos orígenes de datos para la construcción y/o actualización de bases de datos espaciales y almacenes de datos espaciales. [22] TripleGeo es un plugin para GeoKettle que transforma datos geoespaciales en tripletas RDF siguiendo el estándar GeoSPARQL [24]

Introducción

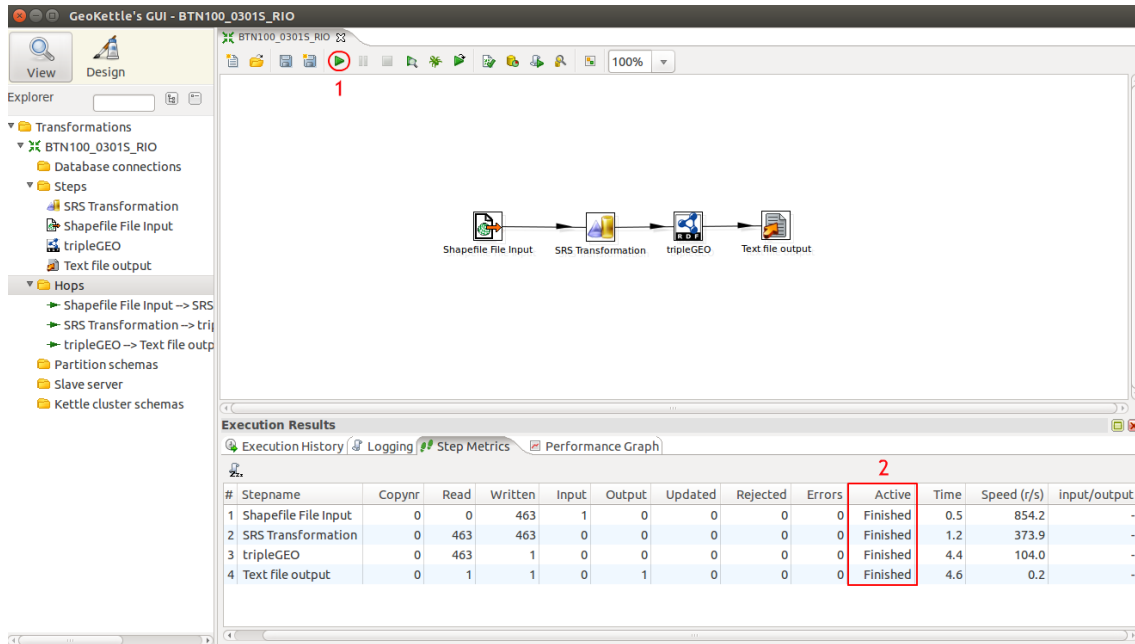


Figura 1.3: TripleGeoKettle en funcionamiento, TripleGeoKettle wiki

1.3. Pentaho Data Integration Spoon

PDI Spoon[27] es la herramienta que reemplaza a GEOKettle. La GUI, el funcionamiento y el SDK para desarrollar plugins es parecido. Sin embargo, en cuanto al desarrollo de plugins, cambia la manera de gestionar las dependencias, las cabeceras de algunas interfaces que se deben implementar, los iconos, metadatos, estructura de carpetas...

1.3.1. Pentaho GIS Plugins

Es un plugin desarrollado por Atol Conseils et Développements[26] para PDI9. Proporciona parte de la funcionalidad GIS que tenía GEOKettle. fig.1.4

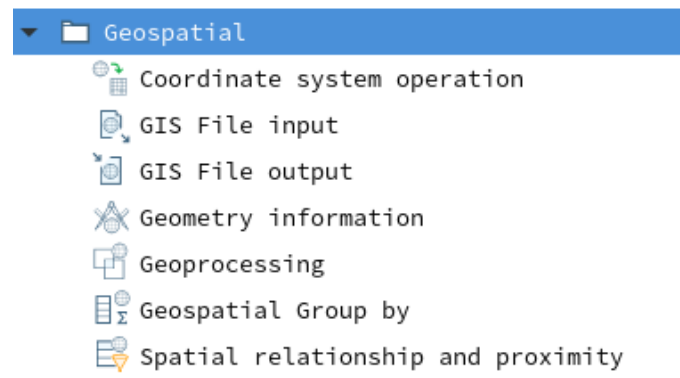


Figura 1.4: Steps incluidos en gis-plugins

1.3.2. Apache Ant

Apache Ant[28] es una librería Java y herramienta de línea de comandos para construir aplicaciones java (compilar, ensamblar y ejecutarlas). Los scripts de configuración se escriben en formato XML y son muy flexibles.

1.3.3. Apache Maven

Apache Maven [29] es una herramienta de gestión de proyectos y dependencias. Está basado en el concepto Project Object Model (POM). Maven es capaz de construir las aplicaciones, descargar las dependencias y gestionarlas, ejecutar tests, crear documentación...

El fichero principal, pom.xml detalla la configuración. Se pueden incluir repositorios externos y dependencias. Maven se encarga de descargar las dependencias y sus subdependencias desde los repositorios para no tener que hacerlo a mano.

Capítulo 2

Desarrollo

2.1. GeoKettle

Dado que se trata de replicar la funcionalidad que proporcionaba GEOKettle, se analizarán las transformaciones realizadas por el OEG en el repositorio de GitHub BTN100[3]. Como se puede ver en la figura 2.1, partes de la transformación fallan. Es lo que se pretende solucionar.

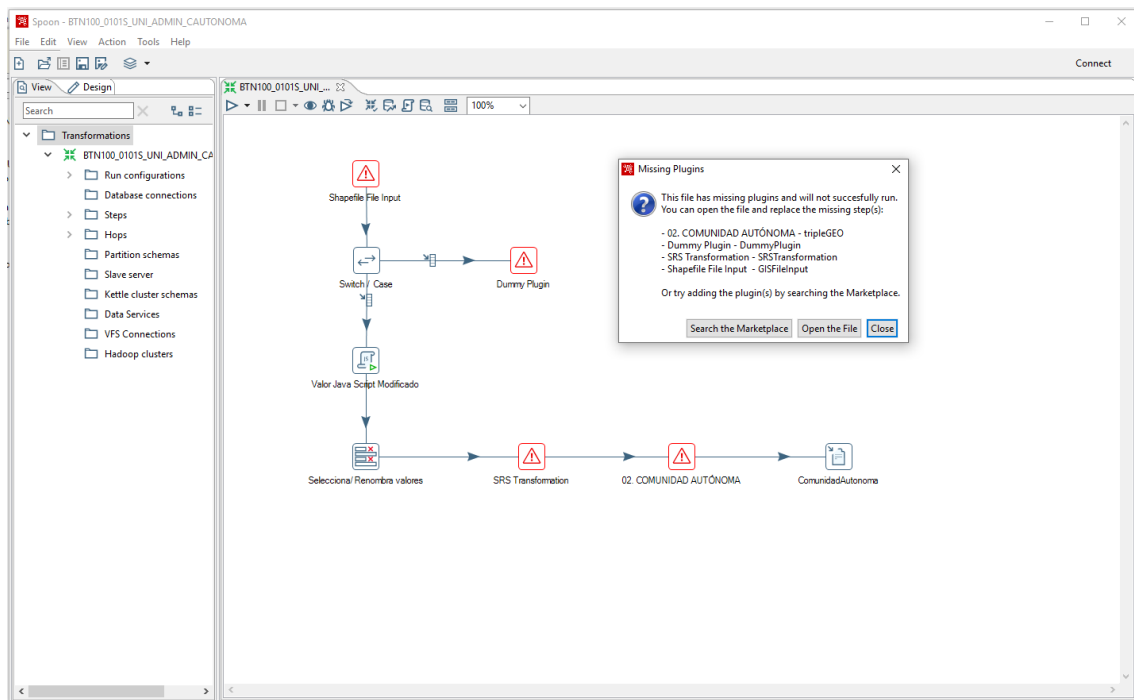


Figura 2.1: Transformación GEOKettle importada en la nueva suite PDI9

2.1.1. Funcionamiento PDI

Para poder realizar el “port” de GeoKettle a Spoon, primero es necesario entender el funcionamiento y transformaciones actuales de GeoKettle observando la entrada y salida de cada paso. Además esta manera se observará mejor el flujo de datos y será más fácil añadir soporte a GeoPackage en el futuro. No todas las transformaciones tienen los mismos pasos, pero son pero todas tienen pasos en común y una estructura parecida. Como ejemplo se utilizarán los datos de BTN100_0101S_UNI_ADMIN y la transformación BTN100_0101S_UNI_ADMIN_CAUTONOMA, ambos en la ruta /transformaciones-shape/1-UnidadesAdministrativas del repositorio github btn100. La transformación contiene los siguientes pasos:

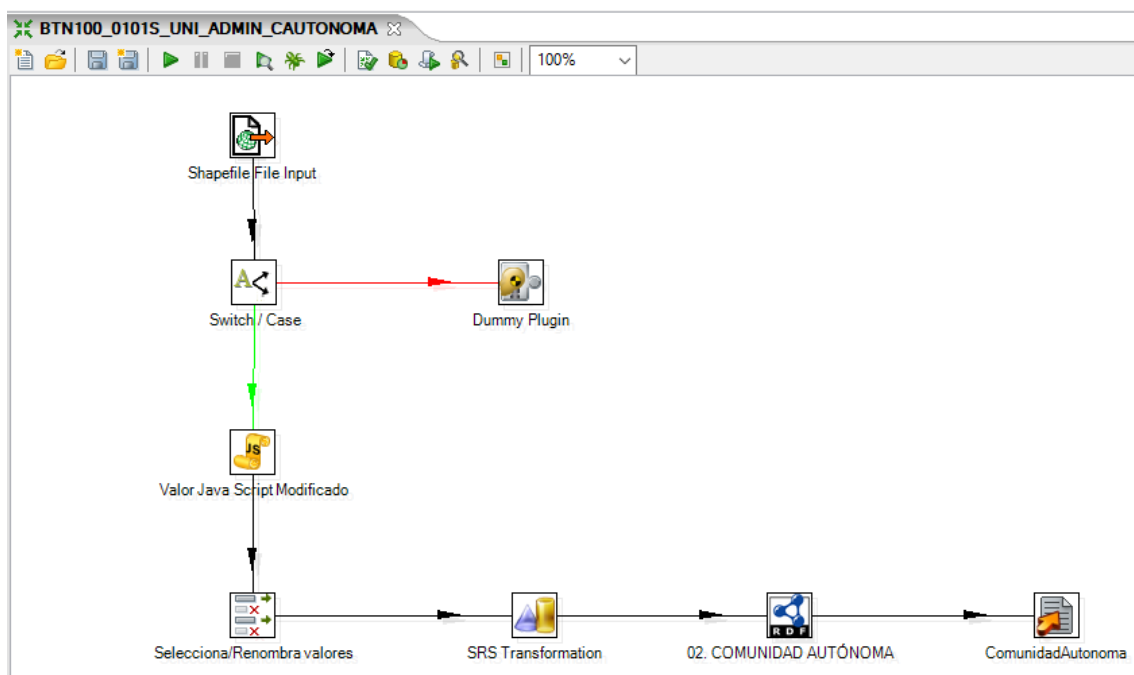


Figura 2.2: Transformación BTN100_0101S_UNI_ADMIN_CAUTONOMA

1. Shapefile File Input
2. Switch Case
3. Dummy Plugin
4. Valor Java Script Modificado
5. Selecciona/Renombrar valores
6. SRS Transformation
7. TripleGeo
8. Text Output

Desarrollo

2.1.1.1. Shapefile File Input

Lee el fichero shapefile.

1. *.shp*: Geometría fig.2.3

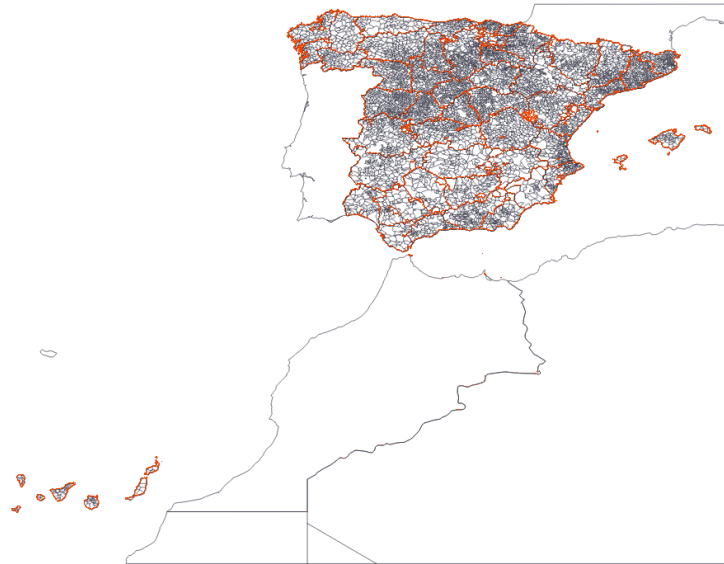
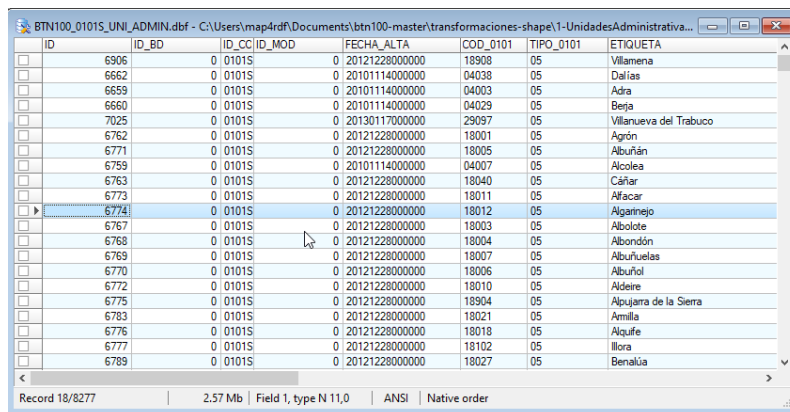


Figura 2.3: Geometría contenida en el shapefile

2. *.dbf*: Datos asociados en columnas fig.2.4

Una captura de pantalla de un archivo de datos dbf. La ventana muestra una lista de registros con columnas como ID, ID_BD, ID_CC, ID_MOD, FECHA_ALTA, COD_0101, TIPO_0101 y ETIQUETA. El registro 18/8277 está seleccionado.

ID	ID_BD	ID_CC	ID_MOD	FECHA_ALTA	COD_0101	TIPO_0101	ETIQUETA
6906	0	0101S	0	20121228000000	18908	05	Villanueva
6662	0	0101S	0	20101114000000	04038	05	Dalias
6659	0	0101S	0	20101114000000	04003	05	Adra
6660	0	0101S	0	20101114000000	04029	05	Benja
7025	0	0101S	0	20130117000000	29097	05	Villanueva del Trabuco
6762	0	0101S	0	20121228000000	18001	05	Agón
6771	0	0101S	0	20121228000000	18005	05	Albuñán
6759	0	0101S	0	20101114000000	04007	05	Alcolea
6763	0	0101S	0	20121228000000	18040	05	Cáñar
6773	0	0101S	0	20121228000000	18011	05	Afacar
6774	0	0101S	0	20121228000000	18012	05	Algarnejo
6767	0	0101S	0	20121228000000	18003	05	Albolote
6768	0	0101S	0	20121228000000	18004	05	Albondón
6769	0	0101S	0	20121228000000	18007	05	Albuñuelas
6770	0	0101S	0	20121228000000	18006	05	Albuñol
6772	0	0101S	0	20121228000000	18010	05	Aldeire
6775	0	0101S	0	20121228000000	18904	05	Alpujarra de la Sierra
6783	0	0101S	0	20121228000000	18021	05	Amilla
6776	0	0101S	0	20121228000000	18018	05	Alquife
6777	0	0101S	0	20121228000000	18102	05	Ilora
6789	0	0101S	0	20121228000000	18027	05	Benalúa

Figura 2.4: Datos columnares dbf asociados a la geometría

3. *.shx*: Índice para acelerar búsquedas

4. *.prj*: Sistema de coordenadas

A4 Switch / case

Step name

Field name to switch

Use string contains comparison ☐

Case value data type

Case value conversion mask

Case value decimal symbol

Case value grouping symbol

Case values

#	Value	Target step
1	02	Valor Java Script Modificado

Default target step

OK Cancel

Figura 2.5: Paso switch

Desarrollo

	ID	ID_BD	ID_CC	ID_MOD	FECHA_ALTA	COD_0101	TIPO_0101	ETIQUETA
<input type="checkbox"/>	8259	0	0101S	0	20130702000000	00	01	MARRUECOS
<input type="checkbox"/>	8260	0	0101S	0	20130702000000	00	01	ARGELIA
<input type="checkbox"/>	8370	0	0101S	0	20140617000000	00	01	ESPAÑA
<input type="checkbox"/>	8257	0	0101S	0	20130702000000	00	01	SÁHARA OCCIDENTAL
<input type="checkbox"/>	8258	0	0101S	0	20130702000000	00	01	MAURITANIA
<input type="checkbox"/>	8264	0	0101S	0	20130702000000	00	01	FRANCIA
<input type="checkbox"/>	8263	0	0101S	0	20130703000000	00	01	ANDORRA
<input type="checkbox"/>	8394	0	0101S	0	0	00	01	PORTUGAL
<input checked="" type="checkbox"/>	8251	0	0101S	0	20141009120621	14	02	Región de Murcia
<input checked="" type="checkbox"/>	8253	0	0101S	0	20141009120553	10	02	Comunitat Valenciana
<input checked="" type="checkbox"/>	8347	0	0101S	0	20141009120534	08	02	Castilla-La Mancha
<input checked="" type="checkbox"/>	8345	0	0101S	0	20141009120526	07	02	Castilla y León
<input checked="" type="checkbox"/>	8343	0	0101S	0	20140617000000	01	02	Andalucía
<input checked="" type="checkbox"/>	8341	0	0101S	0	20141009120600	11	02	Extremadura
<input checked="" type="checkbox"/>	8240	0	0101S	0	20141009120503	05	02	Canarias
<input checked="" type="checkbox"/>	8256	0	0101S	0	20141009120637	16	02	País Vasco/Euskadi
<input checked="" type="checkbox"/>	8244	0	0101S	0	20130206000000	03	02	Principado de Asturias
<input checked="" type="checkbox"/>	8250	0	0101S	0	20141009120607	12	02	Galicia
<input checked="" type="checkbox"/>	8241	0	0101S	0	20141009120548	09	02	Cataluña/Catalunya
<input checked="" type="checkbox"/>	8249	0	0101S	0	20120314000000	02	02	Aragón
<input checked="" type="checkbox"/>	8255	0	0101S	0	20141009120612	13	02	Comunidad de Madrid
<input checked="" type="checkbox"/>	8245	0	0101S	0	20141009120513	06	02	Cantabria
<input checked="" type="checkbox"/>	8246	0	0101S	0	20141009120445	04	02	Illes Balears
<input checked="" type="checkbox"/>	8243	0	0101S	0	20141009120630	15	02	Comunidad Foral de Navarra
<input checked="" type="checkbox"/>	8242	0	0101S	0	20141009120643	17	02	La Rioja
<input type="checkbox"/>	539	0	0101S	0	20120216000000	03	03	Alacant/Alicante
<input type="checkbox"/>	8136	0	0101S	0	20120718000000	30	03	Murcia
<input type="checkbox"/>	8330	0	0101S	0	20140617000000	02	03	Albacete
<input type="checkbox"/>	8314	0	0101S	0	20140617000000	45	03	Toledo
<input type="checkbox"/>	2114	0	0101S	0	20120514000000	09	03	Burgos
<input type="checkbox"/>	8271	0	0101S	0	20130715000000	01	03	Araba/Álava
<input type="checkbox"/>	8326	0	0101S	0	20140617000000	34	03	Palencia
<input type="checkbox"/>	7376	0	0101S	0	20130215000000	41	03	Sevilla
<input type="checkbox"/>	8278	0	0101S	0	20130613000000	14	03	Córdoba
<input type="checkbox"/>	8300	0	0101S	0	20140617000000	23	03	Jaén
<input type="checkbox"/>	8310	0	0101S	0	20140617000000	04	03	Almería

Figura 2.6: La filas correspondientes a las CCAA

2.1.1.3. Dummy Plugin

No hace ninguna transformación, su propósito es recoger los datos innecesarios del switch.

2.1.1.4. Valor JavaScript Modificado

El script cambia el formato de la fecha para facilitar la lectura: de YYYYMMDDHHMMSS a YYYY-MM-DD. También crea un nuevo campo llamado identificador a partir del campo etiqueta, cambiando espacios por barras bajas, mayúsculas por minúsculas, quitando tildes y signos de puntuación. fig.2.7 y 2.8

2.1. GeoKettle

```
Script1
//Script here

var aux = FECHA_ALTA.match(/^(\\d{4})(\\d{2})(\\d{2})$/);

//Here we fix the date
if(!aux || aux.length < 4){
    FECHA_ALTA = "";
}
else{
    FECHA_ALTA = new Date(aux[1], aux[2]-1, aux[3]);
    FECHA_ALTA = date2str(FECHA_ALTA, "yyyy-MM-dd");
}

var etiq = ETIQUETA.toLowerCase().replace(' ','-');
etiq = etiq.replace(/s/g, '-').replace('á', 'A').replace('é', 'e').replace('É', 'E').replace('í', 'i')
.replace('i', 'I').replace('ó', 'o').replace('Ó', 'O').replace('ú', 'u').replace('Ú', 'U').replace('ñ', 'n');
etiq = etiq.replace(' ','').replace(' ','');
identificador = etiq
```

Figura 2.7: Script Javascript

Examine preview data

Standard view

Geographic view

Rows of step: Valor Java Script Modificado (17 rows)

#	the_geom	ID	ID_BD	ID_CODIGO	ID_MOD	FECHA_ALTA	COD_0101	TIPO_0101	ETIQUETA	identificador
1	MULTIPO...	8251	0	0101S	0	2014-10-09	14	02	Región de Murcia	region-de-murcia
2	MULTIPO...	8253	0	0101S	0	2014-10-09	10	02	Comunitat Valenciana	comunitat-valenciana
3	MULTIPO...	8347	0	0101S	0	2014-10-09	08	02	Castilla-La Mancha	castilla-la-mancha
4	MULTIPO...	8345	0	0101S	0	2014-10-09	07	02	Castilla y León	castilla-y-leon
5	MULTIPO...	8343	0	0101S	0	2014-06-17	01	02	Andalucía	andalucia
6	MULTIPO...	8341	0	0101S	0	2014-10-09	11	02	Extremadura	extremadura
7	MULTIPO...	8240	0	0101S	0	2014-10-09	05	02	Canarias	canarias
8	MULTIPO...	8256	0	0101S	0	2014-10-09	16	02	País Vasco/Euskadi	pais-vasco/euskadi
9	MULTIPO...	8244	0	0101S	0	2013-02-06	03	02	Principado de Asturias	principado-de-asturias
10	MULTIPO...	8250	0	0101S	0	2014-10-09	12	02	Galicia	galicia
11	MULTIPO...	8241	0	0101S	0	2014-10-09	09	02	Cataluña/Catalunya	cataluna/catalunya
12	MULTIPO...	8249	0	0101S	0	2012-03-14	02	02	Aragón	aragon
13	MULTIPO...	8255	0	0101S	0	2014-10-09	13	02	Comunidad de Madrid	comunidad-de-madrid
14	MULTIPO...	8245	0	0101S	0	2014-10-09	06	02	Cantabria	cantabria
15	MULTIPO...	8246	0	0101S	0	2014-10-09	04	02	Illes Balears	illes-balears
16	MULTIPO...	8243	0	0101S	0	2014-10-09	15	02	Comunidad Foral de Navar...	comunidad-foral-de-nav...
17	MULTIPO...	8242	0	0101S	0	2014-10-09	17	02	La Rioja	la-rioja

Figura 2.8: Resultado del cambio de formato de fecha

2.1.1.5. Selecciona/Renombrar valores

Cambia los metadatos de la columna FECHA_ALTA para que sea reconocida como fecha. fig.2.9

Desarrollo

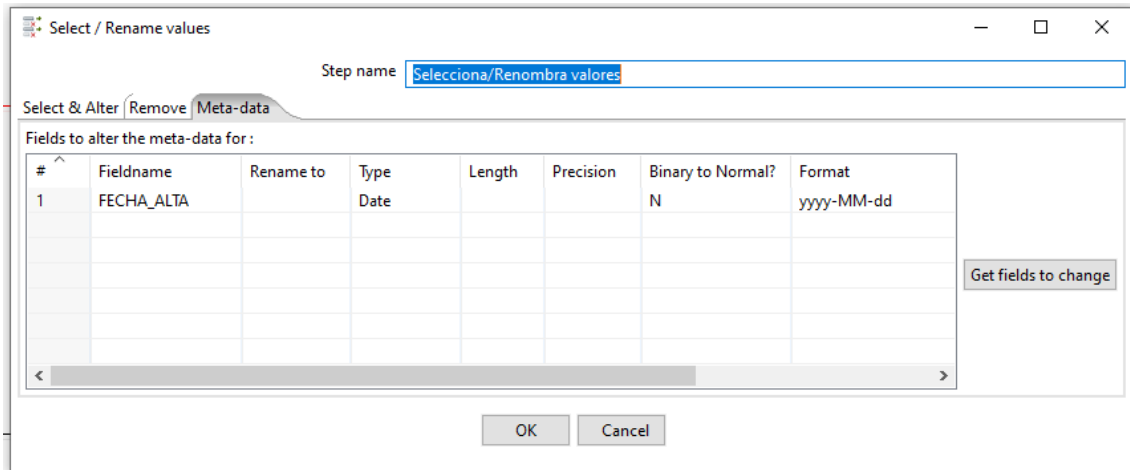


Figura 2.9: Selecciona/renombra valores

2.1.1.6. SRS Transformation

Realiza la reproyección del sistema de coordenadas de ETRS89 a WGS84. fig.2.10

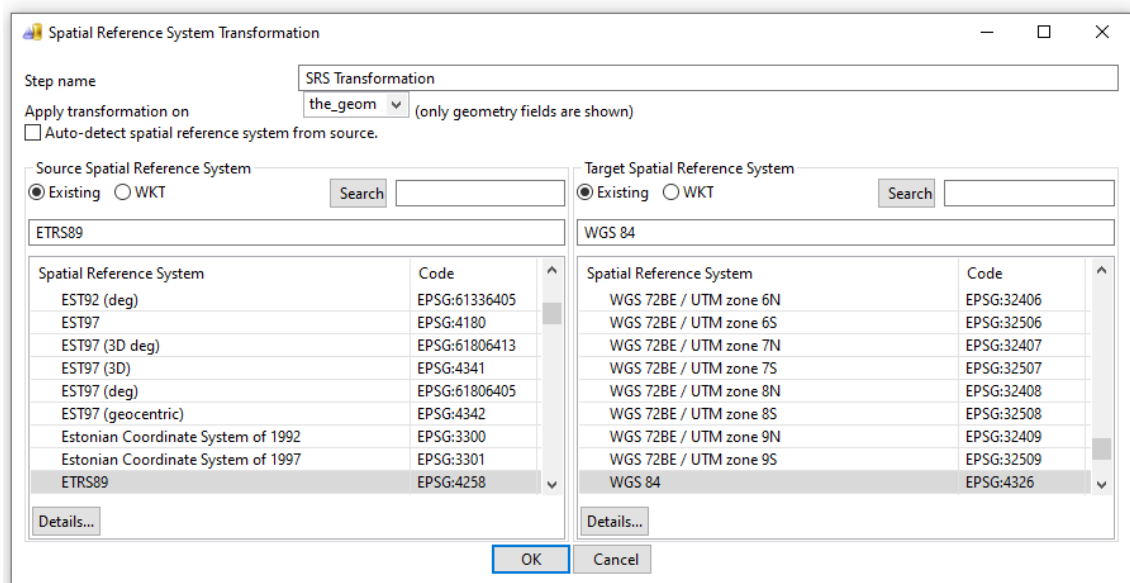


Figura 2.10: Transformación SRS

2.1.1.7. TripleGeoKettle

Transforma el shapefile del paso anterior en RDF en formato .ttl (Turtle). Se pueden configurar los parámetros asociados a la ontología y decidir si se muestran ciertas columnas o no. fig.2.11

tripleGEO step

Step name: 02. COMUNIDAD AUTÓNOMA

General Information

Attribute *: identificador

Feature *: ComunidadAutonoma

Ontology namespace URI *: http://vocab.linkeddata.es/datosabiertos/def/sector-publico/territori

Ontology namespace prefix *: esadm

Resource namespace URI *: https://datos.ign.es/recurso/btn100/comunidad-autonoma/

Resource namespace prefix *: georec

Language: null

Path CSV: null

Generate UUIDs: ☐

#	Other Prefix	Other URI
1		

Prefix and URI for the columns

Show?	Column	Prefix	URI
YES	the_geom	n/a	n/a
NO	identifier	dc	http://purl.org/dc/terms/
NO	ID_BD		
NO	ID_CODIGO		
YES	replaces	dc	http://purl.org/dc/terms/
YES	created	dc	http://purl.org/dc/terms/
YES	codigolNE	esadm	http://vocab.linkeddata.es/datosabiertos/def/sector-publico/ter
NO	TIPO_0101		
YES	title	dc	http://purl.org/dc/terms/
YES	identifier	dc	http://purl.org/dc/terms/

OK Restart Columns Cancel

Figura 2.11: tripleGeoKettle

2.1.1.8. Text file output

Escribe los datos RDF en un fichero de texto con formato .ttl. fig.2.12

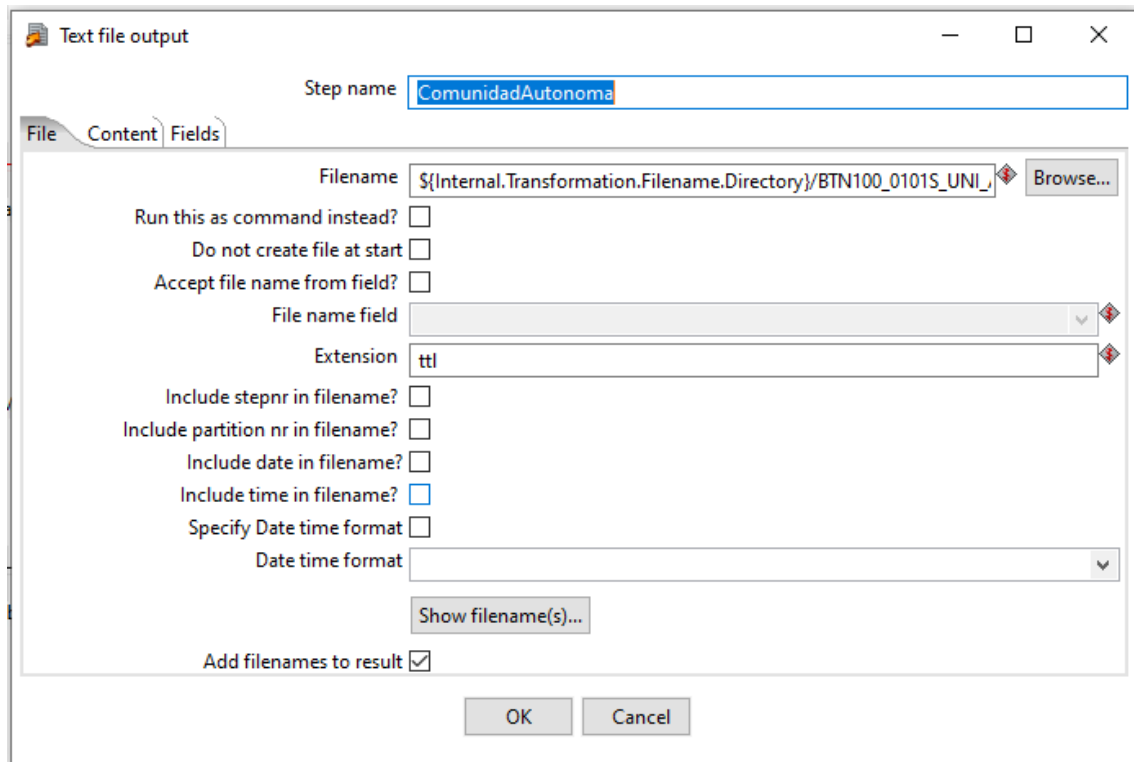


Figura 2.12: text-file-output

2.1.1.9. Resultado de la transformación

```
@prefix geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix geosparql: <http://www.opengis.net/ont/geosparql#> .
@prefix sf:      <http://www.opengis.net/ont/sf#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix georec:  <https://datos.ign.es/recurso/btn100/comunidad-autonoma/> .
@prefix esadm:   <http://vocab.linkeddata.es/datosabiertos/def/sector-publico/territorio#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix dc:      <http://purl.org/dc/terms/> .

georec:comunidad-de-madrid
  a
    rdfs:label "comunidad-de-madrid" ;
    dc:created "2014-10-09"^^xsd:date ;
    dc:identifier "comunidad-de-madrid" ;
    dc:title "Comunidad de Madrid" ;
    esadm:codigoINE "13"^^xsd:int ;
    geosparql:hasGeometry <https://datos.ign.es/recurso/btn100/comunidad-autonoma/...

georec:region-de-murcia
  a
    rdfs:label "region-de-murcia" ;
    dc:created "2014-10-09"^^xsd:date ;
    dc:identifier "region-de-murcia" ;
```

```

dc:title "Region de Murcia" ;
esadm:codigoINE "14"^^xsd:int ;
geosparql:hasGeometry <https://datos.ign.es/recurso/btn100/comunidad-autonoma/...

<https://datos.ign.es/recurso/btn100/comunidad-autonoma/aragon/geometry>
a sf:Polygon ;
geosparql:asWKT "POLYGON ((-1.6174492000010632 40.94373283914169, -1.62366030000...
...

```

2.2. Port a PDI9

2.2.1. Tests con pentaho-gis-plugins

El 3 de Marzo de 2021 el plugin añadió soporte para el formato GeoPackage. Con el siguiente test se comprueba que los pasos o steps que se utilizaran para reemplazar las transformaciones antiguas funcionan correctamente.

The screenshot displays the Pentaho Spoon IDE interface. On the left, a tree view shows the 'Transformations' folder containing the 'trasformacion-laguna-lee-geopackage' transformation. The main canvas shows a flow diagram with the following steps: 'GIS File input', 'Modified JavaScript value', 'Select values', 'Value mapper', 'Coordinate system operation', and 'GIS File output'. A configuration dialog for the 'GIS File output' step is open, showing the following details:

- Step name: GIS File output
- Type: GeoPackage
- Pass output to servlet: ☐
- Do not create file at start: ☐
- Filename: C:\Users\map4rdf\Desktop\test-lagu... (with a 'Browse...' button)
- Geometry field: the_geom
- Output options: Encoding: UTF-8
- Other static parameters table:

#	Parameter	Required	Value
1	Replace file	Yes	Yes
2	Table name	Yes	the-ta...
3	Commit a...	Yes	1000
4	Replace ta...	Yes	No
5	Identifier	No	the-id...
6	Description	No	the-d...
7	EPSG code	No	4326
8	Geometry ...	No	GEOM...
9	Force to 2D	Yes	Yes

Below the dialog, the 'Execution Results' pane shows a log of the transformation execution, indicating that all steps completed successfully.

Figura 2.13: test-laguna-geopackage

2.2.2. Entorno de desarrollo

2.2.2.1. Dependencias e instalación

En los últimos años Pentaho ha pasado de utilizar Apache Ant a utilizar Apache Maven. TripleGeoKettle también utilizaba Ant, y se ha decidido utilizar Maven

Desarrollo

por las ventajas que ofrece. El proyecto TripleGeoKettle contenía una carpeta lib en la que se encontraban los .jar con las dependencias necesarias. Ahora, las dependencias se administran con Maven y el pom.xml. Para ello se incluyen los repositorios de Pentaho y OSGeo y las dependencias que se quieren incluir. Las properties funcionan a modo de “variable” para poder cambiar la versión de pdi fácilmente en un solo lugar. maven-assembly-plugin compila el plugin y genera el .jar.

```
1 <?xml version="1.0"?>
2 <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
  xsd/maven-4.0.0.xsd" xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>oeg-upm</groupId>
6   <artifactId>tripleGeoKettle-oeg</artifactId>
7   <version>1</version>
8   <name>tripleGeoKettle</name>
9
10  <properties>
11    <pentaho-metadata.version>9.1.0.0-324</pentaho-metadata.version>
12    <pdi.version>9.1.0.0-324</pdi.version>
13  </properties>
14
15  <repositories>
16    <repository>
17      <id>pentaho-releases</id>
18      <url>https://nexus.pentaho.org/content/groups/omni</url>
19    </repository>
20    <repository>
21      <id>osgeo</id>
22      <url>https://repo.osgeo.org/repository/release/</url>
23    </repository>
24  </repositories>
25
26  <dependencies>
27    <dependency>
28      <groupId>org.pentaho</groupId>
29      <artifactId>pentaho-metadata</artifactId>
30      <version>${pentaho-metadata.version}</version>
31      <scope>provided</scope>
32    </dependency>
33    <dependency>
34      <groupId>pentaho-kettle</groupId>
35      <artifactId>kettle-core</artifactId>
36      <version>${pdi.version}</version>
37      <scope>provided</scope>
38    </dependency>
39    <dependency>
40      <groupId>pentaho-kettle</groupId>
41      <artifactId>kettle-engine</artifactId>
42      <version>${pdi.version}</version>
43      <scope>provided</scope>
44    </dependency>
45    <dependency>
46      <groupId>pentaho-kettle</groupId>
47      <artifactId>kettle-ui-swt</artifactId>
48      <version>${pdi.version}</version>
49      <scope>provided</scope>
50    </dependency>
51
52    <dependency>
53      <groupId>org.apache.jena</groupId>
54      <artifactId>jena-arq</artifactId>
55      <version>3.0.0</version>
```

```

56     </dependency>
57     <dependency>
58         <groupId>com.vivid solutions</groupId>
59         <artifactId>jts</artifactId>
60         <version>1.11</version>
61     </dependency>
62     <dependency>
63         <groupId>org.geotools</groupId>
64         <artifactId>gt-geometry</artifactId>
65         <version>2.7-M0</version>
66     </dependency>
67
68 </dependencies>
69
70 <build>
71     <plugins>
72         <plugin>
73             <artifactId>maven-assembly-plugin</artifactId>
74             <executions>
75                 <execution>
76                     <id>distro-assembly</id>
77                     <phase>package</phase>
78                     <goals>
79                         <goal>single</goal>
80                     </goals>
81                     <configuration>
82                         <appendAssemblyId>false</appendAssemblyId>
83                         <descriptors>
84                             <descriptor>src/main/assembly/assembly.xml</descriptor>
85                         </descriptors>
86                     </configuration>
87                 </execution>
88             </executions>
89         </plugin>
90     </plugins>
91 </build>
92 </project>

```

A continuación se muestran los pasos para configurar el entorno de desarrollo en Arch Linux:

```

1 # Instalar PDI desde AUR (los mirrors son mucho mas rapidos que los de SourceForge)
2 yay -S pdi-ce
3 # Instalar Java 8
4 sudo pacman -S jdk8-openjdk
5 # Cambiar la version de java con
6 sudo archlinux-java set java-8-openjdk
7 # Instalar el Plugin de desarrollo
8 https://sourceforge.net/projects/pentaho/files/Pentaho%209.1/plugins/kettle-sdk-plugin-assembly-9.1.0.0-324.zip/download
9 # Instalar Maven
10 sudo pacman -S maven
11 # Descargar el settings.xml del repositorio de github en ~/.m2
12 https://raw.githubusercontent.com/pentaho/maven-parent-poms/master/maven-support-files/settings.xml
13 # Desde el directorio del plugin compilar y empaquetar el plugin
14 mvn clean package
15 # Instalarlo en PDI9 copiando el contenido del .zip generado en el directorio target
16 sudo cp target/tripleGeoKettle-oeg-1.jar /opt/pdi/plugins/steps/tripleGeoKettle-oeg-1/

```

2.2.2.2. Pentaho sdk plugins

Pentaho ofrece plugins[30] muy sencillos de ejemplo que muestran como implementar un plugin. Para el port de TripleGeoKettle se ha utilizado como referencia kettle-sdk-step-plugin, que simplemente escribe Hello World. Con él se ha probado la compilación, el pom.xml customizado, y se han realizado pruebas varias como cambiar el icono o el nombre del plugin.

2.3. Proceso de porteo

2.3.1. Diseño del plugin

Se comienza con el sdk de Pentaho y se realizan los cambios necesarios para integrar TripleGeoKettle en el nuevo entorno. Se muestran las imágenes del antes y después de la estructura de carpetas. fig.2.14 y 2.15

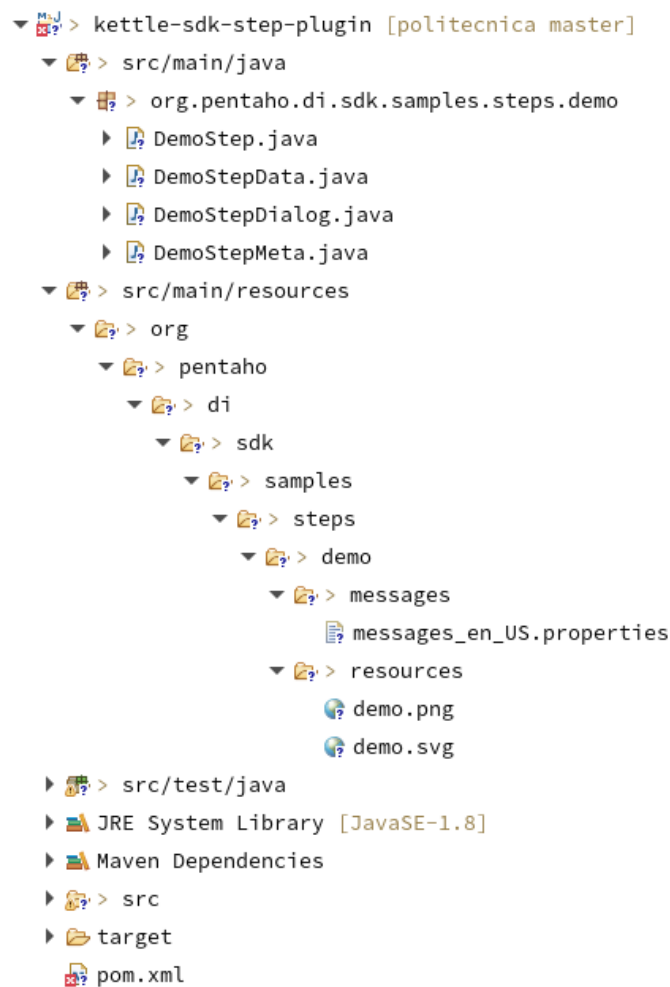


Figura 2.14: Estructura de carpetas del sdk

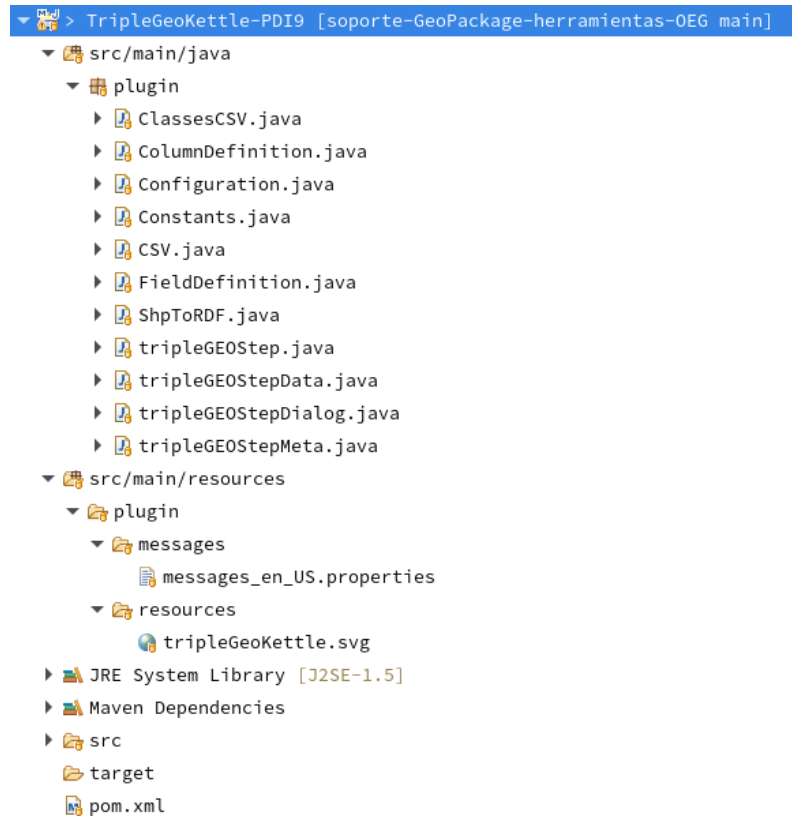


Figura 2.15: Estructura de carpetas de TripleGeoKettle portado

Pasos seguidos:

1. Cambiar el nombre del paquete a “plugin”. Es importante que el de java y el de resources tengan el mismo nombre para que los diálogos lean el fichero properties correctamente.
2. Actualizar el fichero messages.properties, que contiene el texto de la GUI del plugin y varios enlaces.

```
1 #sdk properties
2 tripleGEO.FieldName.Label=Output field name
3 tripleGEO.CheckResult.ReceivingRows.OK=Step is receiving input from other steps.
4 tripleGEO.CheckResult.ReceivingRows.ERROR=No input received from other steps!
5
6 tripleGEOStep.Name=TripleGeoKettle
7 tripleGEOStep.TooltipDesc=An ETL Tool for Transforming Geospatial Data into RDF
   under the GeoSPARQL standard.
8 tripleGEOStep.DocumentationURL=https://github.com/oeg-upm/geo.linkeddata.es-
   TripleGeoKettle/wiki
9 tripleGEOStep.CasesURL=https://github.com/oeg-upm/geo.linkeddata.es-TripleGeoKettle
   /issues
10 tripleGEOStep.ForumURL=https://github.com/oeg-upm/geo.linkeddata.es-TripleGeoKettle
   /issues
11 tripleGEOStep.Linenr=Linenr {0}
12 tripleGEOStep.Error.NoOutputField=Could not find Output Field in row
13
14 # tripleGeoKettle custom properties
15 tripleGEOStepDialog.Shell.Title=tripleGEO step
```

```
16 tripleGEOSTepDialog.Tab.MainTab=General Information
17 tripleGEOSTepDialog.AttributeName.Label=Attribute *
18 tripleGEOSTepDialog.Feature.Label=Feature *
19 tripleGEOSTepDialog.OntologyNS.Label=Ontology namespace URI *
20 tripleGEOSTepDialog.OntologyNSPrefix.Label=Ontology namespace prefix *
21 tripleGEOSTepDialog.ResourceNS.Label=Resource namespace URI *
22 tripleGEOSTepDialog.ResourceNSPrefix.Label=Resource namespace prefix *
23 tripleGEOSTepDialog.Language.Label=Language
24 tripleGEOSTepDialog.PathCSV.Label=Path CSV
25 tripleGEOSTepDialog.PathCSVButton.Label=Browse CSV file
26 tripleGEOSTepDialog.PathCSVButtonTooltip.Label=Browse for CVS file
27 tripleGEOSTepDialog.uuids.Label=Generate UUIDs
28 tripleGEOSTepDialog.Fields.Label=Other prefix and URI
29 tripleGEOSTepDialog.other.Label=Other URI
30 tripleGEOSTepDialog.otherPrefix.Label=Other Prefix
31 tripleGEOSTepDialog.Column.Label=Column
32 tripleGEOSTepDialog.Columns.Label=Prefix and URI for the columns
33 tripleGEOSTepDialog.otherColumns.Label=URI
34 tripleGEOSTepDialog.otherPrefixColumns.Label=Prefix
35 tripleGEOSTepDialog.ShowColumns.Label=Show?
36 tripleGEOSTepDialog.RestartFields.Button=Restart Columns
```

3. Modificar la annotation @Step de DemoStepMeta.java. Se utiliza para que el programa reconozca y categorice el plugin correctamente.

```
1 @Step(
2     id = "TripleGeoKettle",
3     name = "tripleGEOSTep.Name",
4     description = "tripleGEOSTep.TooltipDesc",
5     image = "plugin/resources/tripleGeoKettle.svg",
6     categoryDescription = "i18n:org.pentaho.di.trans.step:BaseStep.Category.
7         Transform",
8     i18nPackageName = "tripleGeoKettle",
9     documentationUrl = "tripleGEOSTep.DocumentationURL",
10    casesUrl = "tripleGEOSTep.CasesURL",
11    forumUrl = "tripleGEOSTep.ForumURL"
12 )
```

4. Cambiar los nombres de las clases de Demo a TripleGeoKettle.
5. Importar las clases java auxiliares de tripleGeoKettle.
6. Existe un método deprecado en tripleGeoStepMeta.java: Valuemeta(). Se utiliza el constructor sin parámetros y luego se le asigna el tipo 2. Para sustituirlo, como el tipo es 2, que según la siguiente tabla[31] significa string, es necesario cambiarlo a ValueMetaString().
7. Error en los métodos readRep y saveRep. Es necesario importar ObjectID y sustituir el tipo long por ObjectID en la cabecera de la función.
8. Cambiar la siguiente línea para que la clase Dialog lea correctamente los contenidos del fichero properties.

```
1     main/java/plugin/tripleGEOSTepDialog.java:69
2     private static String PKG = tripleGEOSTepDialog.class.getPackage().getName();
```

9. PDI9 utiliza iconos .svg y proporciona una guía de diseño[32]. Por tanto se ha actualizado el icono antiguo de .png a .svg con los nuevos colores

fig.2.16



Figura 2.16: Nuevo icono svg

2.3.2. Resultado parcial

La base del porteo se ha realizado correctamente como se puede ver en la figura 2.17. El dialogo se abre y se pueden cambiar los parámetros de configuración. También es capaz de leer la salida del paso anterior de SRS. Sin embargo, todavía no funciona correctamente. Probablemente se debe a que el paso anterior (transformación de coordenadas) pasa su información de manera distinta al de GeoKettle (en PDI le llegan 9 campos y en GeoKettle 8). Será necesario seguir los pasos de la documentación para conectar el debugger y solucionar el problema de `NullPointerException`.

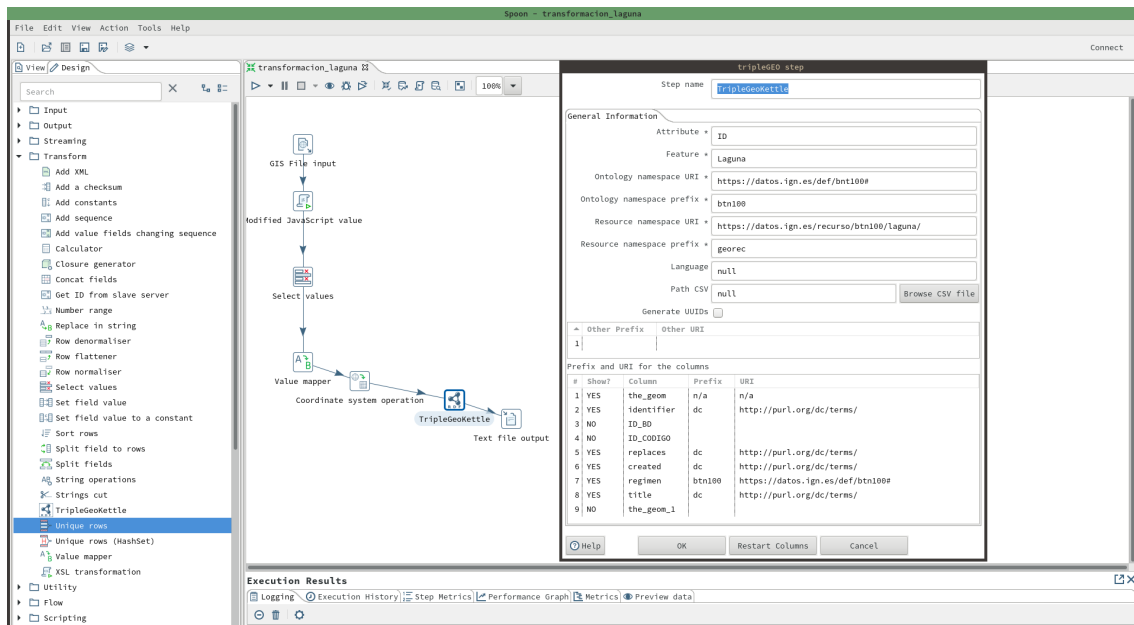


Figura 2.17: TripleGeoKettle corriendo en PDI9

2.3.3. Debugging

Para depurar el plugin se ha creado una nueva configuración de depuración eclipse en localhost y el puerto 1044 (fig.2.18). Por otro lado se ha creado una copia del script de inicio de PDI9 spoon.sh, llamada debug-spoon.sh, y se le han añadido los siguientes parámetros de ejecución.

Desarrollo

```
1 if [ -z "$PENTAHO_DI_JAVA_OPTIONS" ]; then
2     PENTAHO_DI_JAVA_OPTIONS="-Xms1024m -Xmx2048m -Xdebug -Xnoagent -Djava.compiler=NONE
3     -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=1044"
fi
```

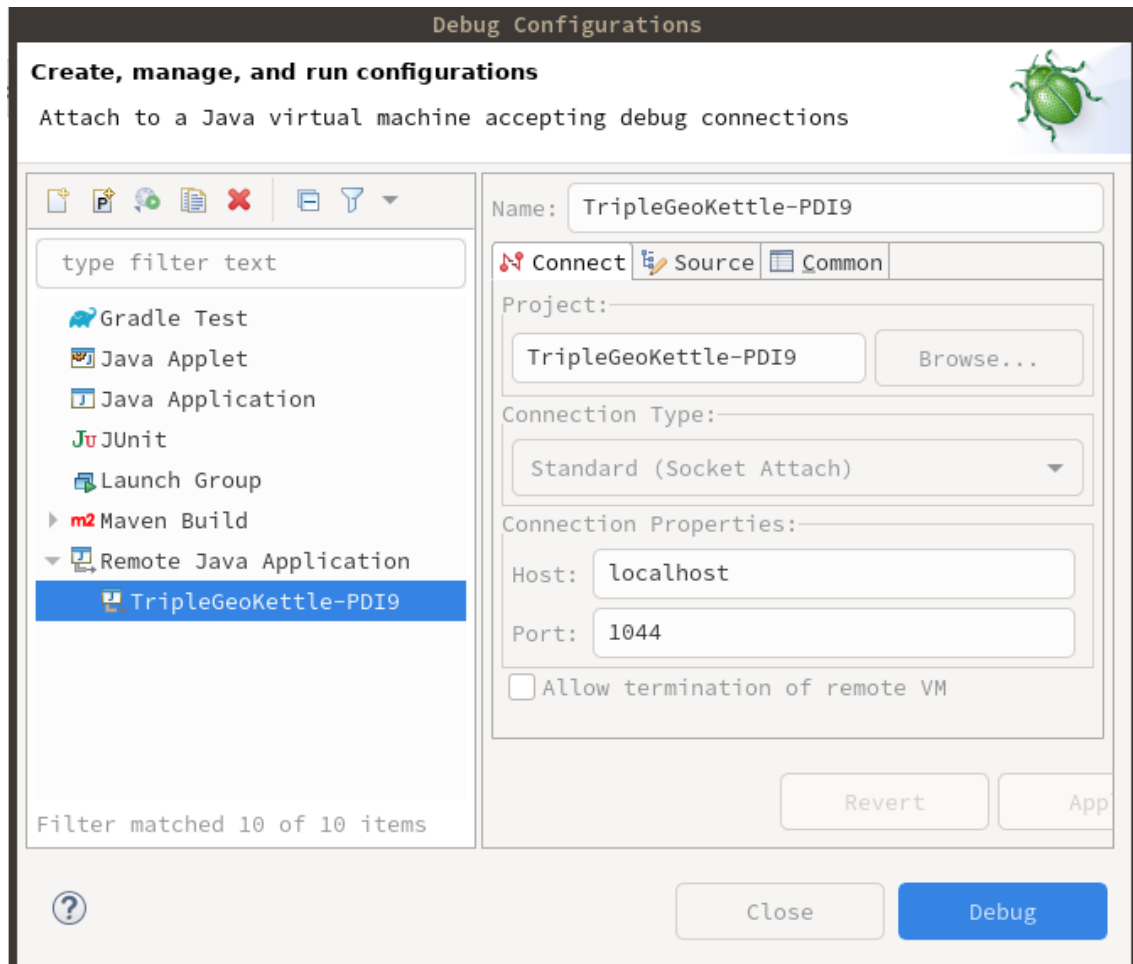


Figura 2.18: Configuración de depuración en Eclipse

El trazado del error `NullPointerException` obtenido tras ejecutar la transformación es el siguiente:

```
1 1. Con la siguiente llamada se inicia el modelo RDF:
2   tripleGEOSTep.java 160: this.shpToRDF.getModelFromConfiguration();
3
4 2. Desde la funcion se crea el modelo con:
5   Model modelAux = ModelFactory.createOntologyModel(OntModelSpec.RDFS_MEM);
6
7 3. En ese momento salta una exception que no detiene el programa y la variable modelAux
   se mantiene en null.
8   Schema factory class org.apache.xerces.impl.dv.xs.SchemaDVFactoryImpl does not
   extend from SchemaDVFactory.
9
10 4. Despues se utiliza en la funcion
11   ShpToRDF.java 176: setModel_rdf(modelAux);
```

```
12
13 5. Y como modelAux se ha inicializado a null, model_rdf tambien lo es.
14   ShpToRDF.java 712: public void setModel_rdf(Model model_rdf) { this.model_rdf =
      model_rdf; }
15
16 6. Finalmente, al llamar a la siguiente funcion salta NullPointerException ya que
      model_rdf es null:
17 ShpToRDF.java 550:
18   private void insertResourceTypeResource(String r1, String r2) {
19       this.model_rdf.add(this.model_rdf.createResource(r1), RDF.type, this.
      model_rdf.createResource(r2));
20   }
```

La causa es una incompatibilidad de dependencias. Apache Jena y vividsolutions.jts utilizan Xerces, pero versiones distintas.

2.3.4. Problemas de dependencias

Para el pom.xml de Maven se ha decidido dejar de utilizar maven-assembly-plugin ya que en algunos casos no incluía los jars necesarios y saltaban errores en tiempo de ejecución. En su lugar se utilizará maven-shade-plugin que es más apropiado para proyectos grandes con muchas dependencias que pueden tener conflictos entre sí. Además crea un “super jar” que incluye todas las dependencias, tal y como se hacía antiguamente con el build.xml de Ant.

Las versiones detalladas en el pom.xml son necesarias por compatibilidad. Por ejemplo, no es posible utilizar la versión de jts que se utilizaba anteriormente (1.11), ni la de gt-geometry, ya que hay muchos problemas de compatibilidad debido a Xerces. Xerces también se utiliza en Jena, y debido a que todas las dependencias (incluso las de pentaho) utilizan Xerces, ha sido un verdadero rompecabezas solucionar las incompatibilidades. Es más, fue tan complicado, que durante unos días se pasó todo el código de Jena del fichero ShpToRDF.java a otra librería (RDF4J) para evitar las incompatibilidades. Pero como llevó al detenido estudio del código fuente, ayudó a encontrar las incompatibilidades que tenía Jena con Xerces y los métodos deprecados.

Fue necesario realizar dos cambios:

```
1 ShpToRDF:347, debido al error de cast, cambiar
2   geometry = reader.read((String)row[this.posGeometry]);
3 por
4   geometry = reader.read(row[this.posGeometry].toString());
5
6 tripleGEOStep:73, debido a error unsupported operationexception, cambiar
7   rm.remove(0)
8 por
9   rm.removeValueMeta(0);
```

Finalmente el contenido del pom.xml es el siguiente:

```
1 <?xml version="1.0"?>
2 <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
   xsd/maven-4.0.0.xsd" xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
   .w3.org/2001/XMLSchema-instance">
3   <modelVersion>4.0.0</modelVersion>
```

```
4 <groupId>oeg-upm</groupId>
5 <artifactId>tripleGeoKettle-oeg</artifactId>
6 <version>1</version>
7 <name>tripleGeoKettle</name>
8 <properties>
9   <pentaho-metadata.version>9.1.0.0-324</pentaho-metadata.version>
10  <pdi.version>9.1.0.0-324</pdi.version>
11  <java.version>1.8</java.version>
12 </properties>
13 <repositories>
14   <repository>
15     <id>pentaho-releases</id>
16     <url>https://nexus.pentaho.org/content/groups/omni</url>
17   </repository>
18   <repository>
19     <id>osgeo</id>
20     <url>https://repo.osgeo.org/repository/release/</url>
21   </repository>
22 </repositories>
23 <dependencies>
24   <dependency>
25     <groupId>org.pentaho</groupId>
26     <artifactId>pentaho-metadata</artifactId>
27     <version>${pentaho-metadata.version}</version>
28     <scope>provided</scope>
29   </dependency>
30   <dependency>
31     <groupId>pentaho-kettle</groupId>
32     <artifactId>kettle-core</artifactId>
33     <version>${pdi.version}</version>
34     <scope>provided</scope>
35   </dependency>
36   <dependency>
37     <groupId>pentaho-kettle</groupId>
38     <artifactId>kettle-engine</artifactId>
39     <version>${pdi.version}</version>
40     <scope>provided</scope>
41   </dependency>
42   <dependency>
43     <groupId>pentaho-kettle</groupId>
44     <artifactId>kettle-ui-swt</artifactId>
45     <version>${pdi.version}</version>
46     <scope>provided</scope>
47   </dependency>
48
49   <dependency>
50     <groupId>org.apache.jena</groupId>
51     <artifactId>apache-jena-libs</artifactId>
52     <version>3.0.0</version>
53     <type>pom</type>
54     <exclusions>
55       <exclusion>
56         <groupId>xerces</groupId>
57         <artifactId>xercesImpl</artifactId>
58       </exclusion>
59       <exclusion>
60         <groupId>xerces</groupId>
61         <artifactId>xml-apis</artifactId>
62       </exclusion>
63       <exclusion>
64         <groupId>xerces</groupId>
65         <artifactId>xml-apis-xerces</artifactId>
66       </exclusion>
67     </exclusions>
68   </dependency>
69
70
```

```
71     <dependency>
72         <groupId>com.vividsolutions</groupId>
73         <artifactId>jts</artifactId>
74         <version>1.13</version>
75     </dependency>
76     <dependency>
77         <groupId>org.geotools</groupId>
78         <artifactId>gt-geometry</artifactId>
79         <version>11.2</version>
80     </dependency>
81
82 </dependencies>
83
84 <build>
85     <plugins>
86         <plugin>
87             <artifactId>maven-compiler-plugin</artifactId>
88             <version>3.8.1</version>
89             <configuration>
90                 <encoding>UTF-8</encoding>
91                 <target>${java.version}</target>
92                 <source>${java.version}</source>
93             </configuration>
94         </plugin>
95         <plugin>
96             <groupId>org.apache.maven.plugins</groupId>
97             <artifactId>maven-shade-plugin</artifactId>
98             <version>3.2.4</version>
99             <executions>
100                 <execution>
101                     <phase>package</phase>
102                     <goals>
103                         <goal>shade</goal>
104                     </goals>
105                 </execution>
106             </executions>
107         </plugin>
108     </plugins>
109 </build>
110 </project>
```

2.4. Adaptación de las transformaciones

Una vez el plugin ha sido portado, el siguiente paso es adaptar todas las transformaciones .ktr para que puedan ejecutarse en PDI9. La mayoría de los pasos existentes se pueden copiar y pegar directamente desde GeoKettle a PDI9 (switch/case, valor javascript modificado, salida de fichero de texto, selecciona/renombrar valores, filtrar...). Es necesario sustituir el paso que lee el fichero shp y el de transformación de coordenadas por los respectivos de pentaho-gis-plugins. El paso de GeoKettle no hace falta modificarlo, pero hace falta precederlo por un paso de seleccionar/renombrar valores, ya que además de la geometría transformada, el paso de SRS nuevo también pasa la geometría original. Es necesario quitar la original del pipeline y sustituirla por la original en el mismo orden para que GeoKettle la lea sin problemas. El modo de trabajo ha sido el siguiente para los 104 ficheros .ktr:

1. Crear la nueva transformación
2. Copiar los pasos compatibles de GeoKettle a PDI9 (Ctrl+C,V)

3. Añadir los 3 pasos nuevos (input, transformación, renombrado)
4. Ejecutar la transformación y comprobar que los ficheros output coinciden con los originales.

Fue considerado realizar estos cambios mediante un script, pero las transformaciones no son homogéneas, algunas tienen unos pasos y otras no, y por tanto habría sido necesario revisarlas todas manualmente. Por tanto, hacer las transformaciones se ha considerado más rápido y seguro.

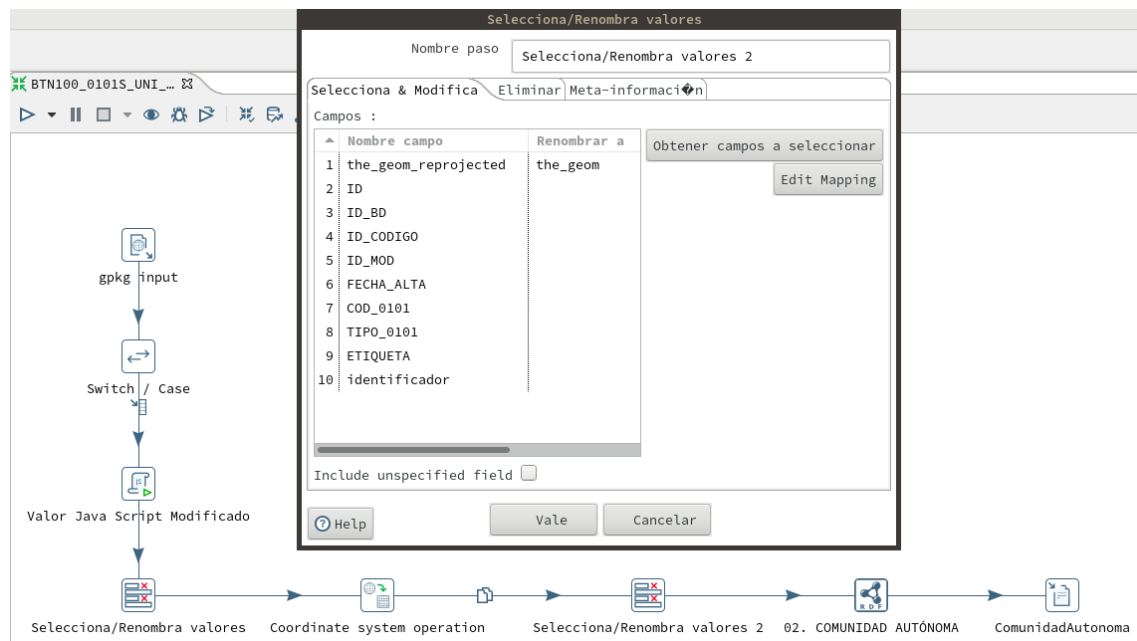


Figura 2.19: Nuevos pasos y renombrado/reordenado de los valores

2.5. Soporte GeoPackage

El primer paso para añadir soporte a GeoPackage es convertir todos los .shp y ficheros relacionados a .gpkg. Para ello se hace uso del comando ogr2ogr de GDAL[33]. El parámetro -nln Table es el nombre de la capa de geometría, que debe coincidir con el del step gpkg input.

A continuación se modifica el primer step de la transformación, el de lectura de .shp a .gpkg, utilizando sustituciones sed en el en el .ktr, ya que en el fondo es un fichero .xml.

Todo esto lo realiza el script shpTogpkg.sh:

```
1 #!/bin/bash
2 shopt -s globstar
3 start="../btn100-master-geopackage/transformaciones-shape"
4 echo "converting shp files to gpkg..."
5 for f in $start/**/*.shp ; do
6     echo "$f"
7     ogr2ogr -f "GPKG" "${f%.shp}.gpkg" "$f" -nln Table -nlt MULTIPOLYGON
```

2.5. Soporte GeoPackage

```
8 done
9 echo "done"
10
11 echo "converting ktr to gpkg..."
12 for f in $start/**/*.ktr ; do
13     echo "$f"
14     sed -e 's/<from>shp/<from>gpkg/g' \
15         -e 's/<name>shp/<name>gpkg/g' \
16         -e 's/<inputFormat>ESRI_SHP/<inputFormat>GEOPACKAGE/g' \
17         -e 's:<key>FORCE_TO_2D:<key>DB_TABLE_NAME</key> <value>Table</value> </param> <
18             param> <key>FORCE_TO_2D:g' \
19         -e 's:shp/<inputFileName>:gpkg/<inputFileName>:g' \
20         -e 's/<encoding>ISO-8859-1/<encoding>UTF-8/g' \
21         -i $f
22 done
23 echo "done"
```

A continuación, es necesario volver a ejecutar todas las transformaciones y comprobar que los resultados son los mismos que con los .shp. Mediante el script `gpkg-tests.sh` se ejecutan las transformaciones con la herramienta pan de pen-taho y se comparan los ficheros .ttl originales con los recién generados.

```
1 #!/bin/bash
2 shopt -s globstar
3
4 if [ $# -ne 2 ]; then
5     echo "Incorrect arguments"
6     echo "  gpkg-tests.sh - run ktr transformations and/or check ttl results"
7     echo "  gpkg-tests.sh [run/check] [DIRECTORY]"
8     exit
9 fi
10
11 #start="./btn100-master-geopackage/transformaciones-shape/1-UnidadesAdministrativas/2-
12   ZonaProtegida"
13 start="$2"
14
15 if [ $1 == "run" ]; then
16     rm gpkg-tests-pan.log
17     echo "counting transformations..."
18     i=0
19     for f in $start/**/*.ktr ; do
20         i=$((i+1))
21     done
22     echo "$i ktr files to process"
23
24     j=0
25     echo "running gpkg transformations..."
26     for f in $start/**/*.ktr ; do
27         pan -file=$f >> gpkg-tests-pan.log 2>&1
28         j=$((j+1))
29         echo "$j/$i $f"
30     done
31     echo "done"
32 fi
33
34 echo "finding differences in gpkg transformation results..."
35 for f in $start/**/*.ttl ; do
36     cmp $f $(sed 's/btn100-master-geopackage/btn100-master/g' <<< $f)
37 done
38 echo "done"
```

2.6. Problemas encontrados

2.6.1. Out of memory

Al ejecutar varias transformaciones se genera el error outofmemoryerror. Para solucionarlo, es necesario añadir el parámetro `-Xmx6g` a la línea 250 de `spoon.sh`, como indica la guía de pentaho [34]. Simplemente aumenta el límite de memoria que puede utilizar el programa.

2.6.2. Línea eléctrica

El archivo de transformación original `BTN100_0702L_LIN_ELEC.ktr` era incorrecto. En el paso `switch/case` no se hacía correctamente la división entre líneas de alta y baja tensión; en los dos case dejaba pasar el los mismos valores (01) que corresponden a líneas de baja tensión. Deben dividirse entre 01 y 02.

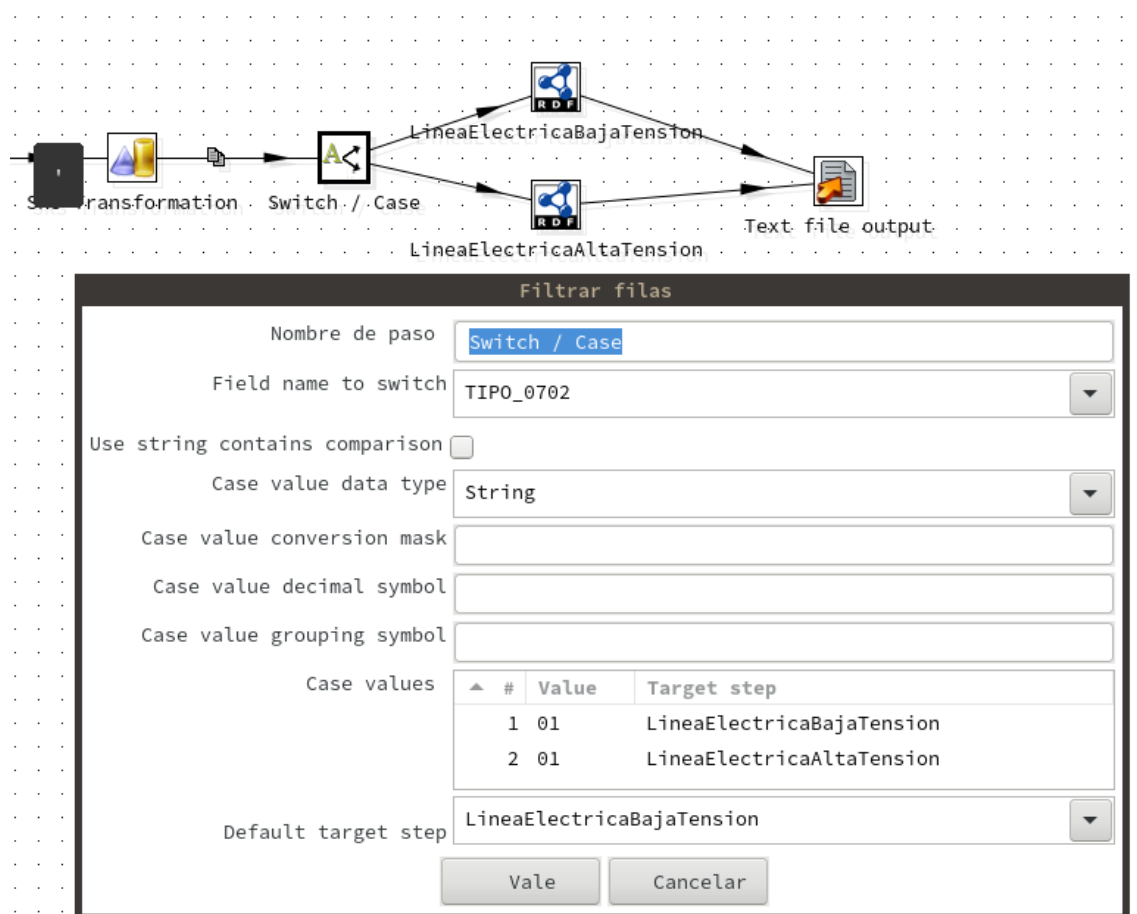
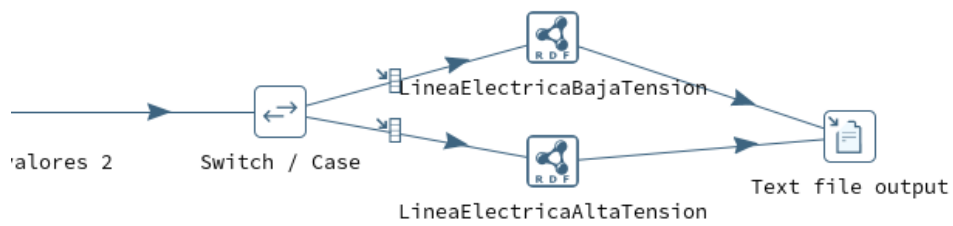


Figura 2.20: Transformación original incorrecta

2.6. Problemas encontrados



Filtrar filas

Nombre de paso:

Field name to switch:

Use string contains comparison: ☐

Case value data type:

Case value conversion mask:

Case value decimal symbol:

Case value grouping symbol:

Case values:

	Value	Target step
1	01	LineaElectricaAltaTension
2	02	LineaElectricaBajaTension

Default target step:

Figura 2.21: Transformación corregida

Capítulo 3

Resultados y conclusiones

En conclusión, se han cumplido los objetivos planteados y por tanto se ha obtenido:

1. Replicar la funcionalidad y las transformaciones de GeoKettle + TripleGeo en la nueva suite PDI.
 - Plugin actualizado compatible con la nueva versión de PDI (9).
 - Transformaciones actualizadas compatibles con la nueva suite.
2. Dar soporte GeoPackage a la herramienta GeoKettle y su plugin para transformar a RDF.
 - Archivos en formato GeoPackage a partir de los Shapefile originales.
 - Transformaciones nuevas que lean archivos GeoPackage en vez de Shapefile.
3. Realizar un procesado completo de todos los datos del IGN para generar este tipo de formato.
 - Archivos .ttl obtenidos tras ejecutar las nuevas transformaciones.

Capítulo 4

Análisis de impacto

En este capítulo se realizará un análisis del impacto potencial de los resultados obtenidos durante la realización del TFG, en los diferentes contextos para los que se aplique:

- Personal
- Empresarial
- Social
- Económico
- Medioambiental
- Cultural

En dicho análisis se destacarán los beneficios esperados, así como también los posibles efectos adversos.

Se recomienda analizar también el potencial impacto respecto a los Objetivos de Desarrollo Sostenible (ODS), de la Agenda 2030, que sean relevantes para el trabajo realizado (ver enlace)

Además, se harán notar aquellas decisiones tomadas a lo largo del trabajo que tienen como base la consideración del impacto.

4.1. Personal

El Trabajo de Fin de Grado ha tenido un gran impacto personal. Marca el fin de mi grado en informática. Ha sido un trabajo constante durante meses en el que he aprendido cosas nuevas y en el que he aplicado lo aprendido durante estos cuatro años.

4.2. Empresarial

Los recursos generados en este TFG pueden ser utilizados por cualquier empresa que así lo desee.

4.3. Social

Los recursos también pueden ser utilizados por particulares, que ahora podrán disfrutar de herramientas más actualizadas para sus consultas.

4.4. Medioambiental

La visualización de datos geográficos facilita estudio de sucesos medioambientales en España.

Bibliografía

- [1] A. de León, V. Saquicela, LM. Vilches-Blázquez, B. Villazón-Terrazas, F. Priyatna and Oscar Corcho, “Geographical Linked Data: a Spanish Use Case”, in *Proceedings of the In I-SEMANTICS '10 6th International Conference on Semantic Systems*
- [2] Espinoza-Arias, P., García-Delgado, M., Corcho, O. et al. “A sustainable process and toolbox for geographical linked data generation and publication: a case study with BTN100.”, in *Open geospatial data, softw. stand.* 4, 2 (2019).
- [3] Repositorio GitHub BTN100, <https://github.com/oeg-upm/btn100>
- [4] OGC, “OGC’s Role in the Spatial Standards World”, in *An Open GIS Consortium (OGC) White Paper*
- [5] OGC, ISO, TC/ 211, IHO, “A Guide to the Role of Standards in Geospatial Information Management”, in *Fifth Session of the United Nations Committee of Experts on Global Geospatial Information Management (UN-GGIM). Held from 3-7 August 2015 at the United Nations Headquarters in New York.*
- [6] Miembros del OGC, “<https://www.ogc.org/ogc/members>”
- [7] Web de datos del Instituto Geográfico Nacional, <http://datos.ign.es/>
- [8] Leon, Alexander de; Wisniewki, Filip; Villazón-Terrazas, Boris y Corcho, Oscar “ Map4rdf - Faceted Browser for Geospatial Datasets”, in *PMOD workshop, 2012*
- [9] Web de Map4rdf, “<https://oeg-upm.github.io/map4rdf/>”,
- [10] “ESRI Shapefile Technical Description”, *An ESRI White Paper—July 1998*
- [11] ESRI, “Geoprocessing considerations for shapefile output”, *Arcgis Desktop 9.3 Help*
- [12] OGC® GeoPackage Encoding Standard 1.3, “<http://www.geopackage.org/spec/>”,
- [13] Tim Berners-Lee “Linked Data Design Issues”, in *W3C 2009*
- [14] Aurelio Morales “Di no al Shapefile y sí al GeoPackage”, in *mappingis.com 2018*

-
- [15] ESRI ArcMap 10.8 Docs, “¿Qué son las proyecciones cartográficas?”, <https://desktop.arcgis.com/es/arcmap/latest/map/projections/what-are-map-projections.htm>
- [16] Antonin Guttman “1984. R-trees: a dynamic index structure for spatial searching”, in *Proceedings of the 1984 ACM SIGMOD*
- [17] “Limits In SQLite”, <https://www.sqlite.org/limits.html>
- [18] “GDAL”, in <https://gdal.org/>
- [19] Tim Berners-Lee, James Hendler, Eric Miller, “Integrating Applications on the Semantic Web”, in *Journal of the Institute of Electrical Engineers of Japan*, Vol 122(10), October, 2002, p. 676-680.
- [20] “¿Qué son Datos Abiertos?”, <https://datos.madrid.es>
- [21] Iniciativa del Instituto Geográfico Nacional (IGN) “<http://datos.ign.es/>”,
- [22] GEOKettle, OSGeo Live Wiki
“https://live.osgeo.org/archive/10.0/es/overview/geokettle_overview.html”,
- [23] Pentaho Data Integration 9.1 Documentation
“https://help.pentaho.com/Documentation/9.1/Products/Pentaho_Data_Integration”,
- [24] TripleGeo Documentation
“https://github.com/oeg-upm/geo.linkeddata.es-TripleGeoKettle/wikiTripleGeo_Documentation”,
- [25] pentaho-gis-plugins
“<https://github.com/atoled/pentaho-gis-plugins>”,
- [26] Sitio web de atoled
“<https://www.atoled.com/expertise/solutions-geographiques-open-source-sig>”,
- [27] Documentación PDI Spoon
“https://help.pentaho.com/Documentation/8.2/Products/Data_Integration/PDI_Client”,
- [28] Sitio web de Apache Ant
“<https://ant.apache.org/>”,
- [29] Sitio web de Apache Maven
“<https://maven.apache.org/>”,
- [30] SDK de plugins PDI
“<https://github.com/pentaho/pdi-sdk-plugins>”,
- [31] Tabla con equivalencias
<https://javadoc.pentaho.com/kettle/constant-values.html>

BIBLIOGRAFÍA

[32] Guía de diseño de plugins

https://help.pentaho.com/Documentation/8.2/Developer_Center/PDI/Extend/035ns,

[33] Documentación de la herramienta ogr2ogr

<https://gdal.org/programs/ogr2ogr.html>,

[34] Solución OutOfMemoryError

<https://help.pentaho.com/Documentation/5.2/OP0/100/090/040>,

Apéndice A

Anexos

A.1. Glosario de términos

- *OGC*: Open Geospatial Consortium
- *GIS*: Geographic Information System
- *SIG*: Sistema de Información Geográfica
- *ESRI*: Environmental Systems Research Institute
- *IGN*: Instituto Geográfico Nacional
- *GDAL*: Geospatial Data Abstraction Library
- *URI*: Uniform Resource Identifier
- *RDF*: Resource Description Framework
- *SPARQL*: SPARQL And Rdf Query Language
- *ETL*: Extract, Transform and Load
- *KETTLE*: Kettle Extraction Transformation Transport Load Environment
- *PDI*: Pentaho Data Integration
- *ab*: cd