



*ugr* | Universidad  
de **Granada**

TRABAJO FIN DE MÁSTER  
MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA

## Integración de Información Geográfica en la Web Semántica

---

### SUBTÍTULO

#### Autora

Gema Correa Fernández

#### Director

José Samos Jiménez



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA  
Y DE TELECOMUNICACIÓN

—  
Granada, 31 de agosto de 2019









Gema Correa Fernández

# Integración de Información Geográfica en la Web Semántica

---

## SUBTÍTULO

Trabajo de Fin de Máster para la integración de Información Geográfica  
en la Web Semántica del Departamento de Lenguajes y Sistemas  
Informáticos de la Escuela Técnica Superior de Ingenierías  
Informáticas y de Telecomunicaciones de la UGR

Director: José Samos Jiménez

Septiembre de 2019



# **Integración de Información Geográfica en la Web Semántica:**

## **SUBTÍTULO**

Gema Correa Fernández

### **Resumen**

Este trabajo tiene como objetivo estudiar las herramientas de la Web Semántica que se pueden utilizar para representar e integrar Información Geográfica. En particular, está centrado en valorar las posibles herramientas que ofrece la Web Semántica y en el desarrollo de una prueba de concepto con Información Geográfica procedente de la provincia de Granada. Para entender el ámbito, se contextualizan y desarrollan los conceptos necesarios del área de los Sistemas de Información Geográfica, además de comprender los conceptos clave de la Web Semántica, que ofrece mecanismos muy útiles para la integración de información y estándares de representación de Información Geográfica mediante diversas herramientas. Por el enfoque que se realiza, este trabajo puede resultar de utilidad para aquellas personas que quieren seguir aprendiendo sobre los Sistemas de Información en un ámbito distinto.

**Palabras-clave:** Sistema de Información Geográfica, SIG, Web actual, Web Semántica, OWL, RDF, GEOSPARQL, Shapefile, Protégé, GraphBD.



**TÍTULO EN INGLÉS:  
SUBTÍTULO EN INGLÉS**

Gema Correa Fernández

**Abstract**

**Key-words:**



---

Yo, **Gema Correa Fernández**, alumna del Máster Universitario de Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 75572158-T, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Máster en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Asimismo, el código fuente del proyecto y esta documentación pueden consultarse en la dirección <https://github.com/Gecofer/TFM> una vez defendido el TFM, para que aquellos que lo deseen puedan conocer el proyecto.

Fdo: Gema Correa Fernández

Granada a 31 de agosto de 2019



---

D. **José Samos Jiménez**, Profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

**Informa:**

Que el presente trabajo, titulado *Integración de Información Geográfica en la Web Semántica, SUBTÍTULO*, ha sido realizado bajo su supervisión por **Gema Correa Fernández**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 31 de agosto de 2019.

**El director:**

**José Samos Jiménez**



# Agradecimientos



# Índice general

<b>1. Prueba de Concepto</b>	<b>1</b>
1.1. Web Semántica Geoespacial . . . . .	1
1.1.1. Selección y obtención de los datos geográficos . . . . .	2
1.1.2. Creación de la ontología . . . . .	12
1.1.3. Poblar la ontología . . . . .	16
1.1.4. Realización de consultas GeoSPARQL . . . . .	22
1.1.5. Visualización de la Información Geográfica . . . . .	34
1.2. Evaluación de la prueba de concepto . . . . .	34
1.3. Resumen del capítulo . . . . .	35
<b>Bibliografía</b>	<b>36</b>



# Índice de figuras

1.1.	Portal del Instituto de Estadística y Cartografía . . . . .	3
1.2.	Diferentes archivos que componen el formato Shapefile . . . . .	4
1.3.	. . . . .	5
1.4.	Visualización del elemento <i>Edificaciones</i> en QGIS . . . . .	5
1.5.	Visualización del elemento <i>Curvas de Nivel</i> en QGIS . . . . .	6
1.6.	Visualización del elemento <i>Punto Cota</i> en QGIS . . . . .	7
1.7.	Visualización de las tres geometrías en OpenStreetMap . . . . .	8
1.8.	. . . . .	9
1.9.	. . . . .	10
1.10.	. . . . .	11
1.11.	. . . . .	13
1.12.	. . . . .	13
1.13.	. . . . .	14
1.14.	Ontología de prototipo . . . . .	15
1.15.	. . . . .	15
1.16.	. . . . .	16
1.17.	. . . . .	17
1.18.	. . . . .	18
1.19.	. . . . .	18
1.20.	. . . . .	19
1.21.	. . . . .	20
1.22.	. . . . .	20
1.23.	. . . . .	21
1.24.	. . . . .	21
1.25.	. . . . .	22
1.26.	. . . . .	23
1.27.	. . . . .	24
1.28.	. . . . .	24
1.29.	. . . . .	25
1.30.	. . . . .	25
1.31.	. . . . .	26
1.32.	. . . . .	27
1.33.	. . . . .	27

---

1.34. . . . .	28
1.35. . . . .	29
1.36. . . . .	30
1.37. . . . .	30
1.38. . . . .	31
1.39. . . . .	31
1.40. . . . .	32
1.41. . . . .	33
1.42. . . . .	33

# Índice de cuadros

1.1. Esquema del modelo de datos descargado . . . . .	3
---	---



# Capítulo 1

## Prueba de Concepto

**RESUMEN:** Este capítulo presenta la prueba de concepto realizada para la incorporación de Información Geográfica, procedente de Ogíjares (Granada), mediante herramientas de la Web Semántica. En el ejemplo se estudian y proponen herramientas de la Web Semántica que se pueden utilizar para representar e integrar datos geoespaciales con diferentes geometrías.

### 1.1. Web Semántica Geoespacial

Como hemos comentado, uno de nuestros objetivos es estudiar las herramientas de la Web Semántica que se pueden utilizar para representar e incorporar Información Geográfica, valorarlas y desarrollar una prueba de concepto. Para ello es necesario haber entendido los capítulos dedicados a los sistemas SIG y a la Web Semántica, y así comprender que el principal nexo de unión entre ambas tecnologías se obtiene a través de las consultas realizadas con GeoSPARQL. Sabiendo todos estos conocimientos es posible comenzar con el desarrollo de la ontología, sin embargo, la primera tarea imprescindible es la selección del conjunto de datos geoespaciales que se desea hacer accesible mediante la Web Semántica.

Durante la prueba de concepto contaremos con la presencia de las herramientas destinadas a la generación de información geoespacial (**QGIS** para permitir obtener la información de los *Shapefile* a hojas de cálculo), generación de documentos *RDF* (**Protegé** para permitir llevar la información de los *Shapefile* hacia documentos en formato *RDF* de manera gráfica y su posterior visualización), consumo (**GraphDB** para permitir la visualización de archivos en formato *RDF* y la realización de las consultas del lenguaje estándar de consulta geoespacial *GEOSPARQL*) y visualización de la información geoespacial obtenida de las consultas (**R** para permitir ubicar la información geográfica en un mapa interactivo mediante la librería *Shiny*).

*Nota:* La versión usada de QGIS es la 3.8.0-Zanzibar para macOS 10.14

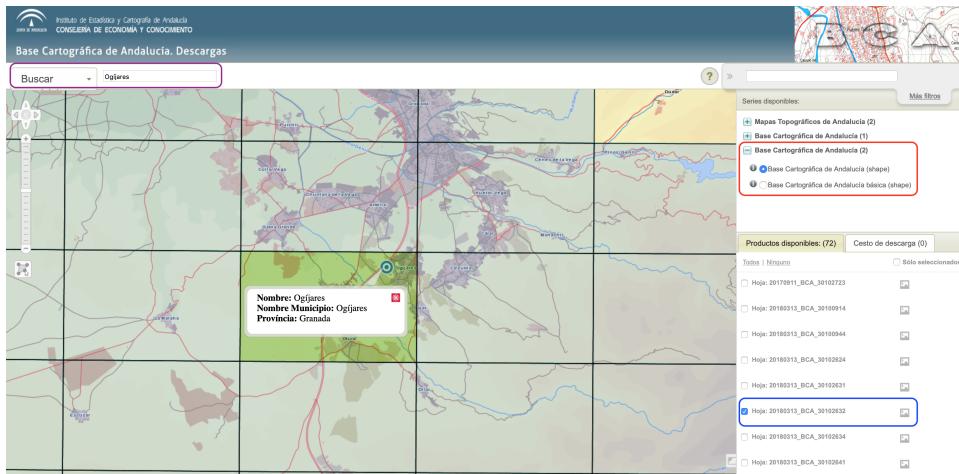
### 1.1.1. Selección y obtención de los datos geográficos

En la actualidad, disponemos de diversas fuentes oficiales y no oficiales que nos proporcionan mapas de calidad con los que poder trabajar. No obstante, al querer hacer uso de Información Geográfica de España, es importante destacar dos fuentes principales:

- **Instituto Geográfico Nacional (IGN)**, es la fuente oficial para todo el territorio español y la descarga de mapas se puede realizar a través de la dirección <http://centrodedescargas.cnig.es/CentroDescargas/buscador.do#>.
- **Instituto de Estadística y Cartografía de Andalucía**, es la fuente oficial para todo el territorio andaluz y la descarga de mapas se puede realizar a través de la dirección <https://www.juntadeandalucia.es/institutodeestadisticaycartografia/bcadescargas/>.

Sin embargo, como vamos a trabajar con datos procedentes de la provincia de Granada, en concreto de mi pueblo Ogíjares, he optado por escoger los que nos proporciona el *Instituto de Estadística y Cartografía de Andalucía* [1]. A continuación, se muestran los pasos para su obtención:

1. La descarga se realiza a través de la plataforma del Instituto de Estadística y Cartografía de Andalucía, de la Consejería de Economía y Conocimiento (URL mencionada en el punto anterior). Una vez dentro, seleccionamos la opción *Base Cartográfica de Andalucía* (2), escogemos *Base Cartográfica de Andalucía (shape)* y buscamos el municipio de *Ogíjares* (figura 1.1). Nos aparecerán varios cuadrantes, escogemos con el que nosotros queramos trabajar; yo me he descargado el cuadrante que aparece seleccionado y remarcado en color.



**Figura 1.1: Portal del Instituto de Estadística y Cartografía**

- La carpeta descargada contiene mapas de áreas muy diversas, las cuales hacen uso de geometría de punto, de línea o de área. Entre las capas que se nos proporciona nos podemos encontrar diversos modelos de datos (tabla 1.1). *Si queremos saber más sobre de cada uno de ellos debemos acceder al siguiente [enlace](#).*

**Cuadro 1.1: Esquema del modelo de datos descargado**

MODELO DE DATOS	ELEMENTOS
INFRAESTRUCTURAS GEOGRÁFICAS	Puntos geodésicos, líneas administrativas
TOPONIMIA	Topónimos
RELIEVE	Curvas de nivel, puntos de cota
SISTEMA URBANO	Edificaciones, curvas artificiales
SERVICIOS	Centros educativos, instalaciones deportivas
RED HIDROGRÁFICA	Corriente artificial, punto fluvial
RED VIARIA	Carreteras, carril bici
INFRAESTRUCTURAS ENERGÉTICAS Y DE TELECOMUNICACIONES	Instalación de energía eléctrica, explotación minera
INFRAESTRUCTURAS HIDRÁULICAS	Deósitos hidráulicos, presa
INFRAESTRUCTURAS DE TRANSPORTES	Área de servicio de descanso
INFRAESTRUCTURAS MEDIOAMBIENTALES	Instalación de tratamiento de aguas.
CUBIERTA TERRESTRE	Lindes



**Figura 1.2: Diferentes archivos que componen el formato Shapefile**

3. El fichero descargado para cada uno de los modelos de datos que acabamos de comentar tiene como formato principal *Shapefile (sh)*, el formato más usado para almacenar información geoespacial. Es un formato de archivos que almacena información no topológica con características espaciales de elementos geográficos que soportan geometrías como puntos, líneas y polígonos [2]. Es originario de *Environmental Systems Research Institute* (ESRI)<sup>1</sup> y consta fundamentalmente de un archivo principal, un archivo de índice y una tabla dBase. Estos archivos suelen requerir poco espacio de almacenamiento en disco y se pueden leer y escribir con facilidad. Los diferentes archivos que componen el formato Shapefile tienen el mismo nombre cada uno con su respectiva extensión, como se aprecia en la figura 1.2:
  - **Archivo principal (\*.shp):** archivo de longitud variable en el que cada registro describe una forma con sus respectivos vértices.
  - **Archivo de índice (\*.shx):** acompaña al archivo principal (\*.shp) que almacena la posición de los identificadores de entidades individuales en el archivo .shp
  - **Tabla dBase (\*.dbf):** almacena la información de atributos de las entidades.
4. La visualización del archivo Shapefile se puede realizar a través de cualquier software GIS, en nuestro caso vamos a utilizar el software de análisis geoespacial QGIS. Respecto los modelos de datos disponibles que se desean escoger para hacer accesible mediante la Web Semántica, vamos a centrarnos en la información geoespacial de edificaciones, curvas de nivel y cotas de altura de Ogíjares (Granada) [3]:

---

<sup>1</sup> Empresa que desarrolla y comercializa software para SIG.

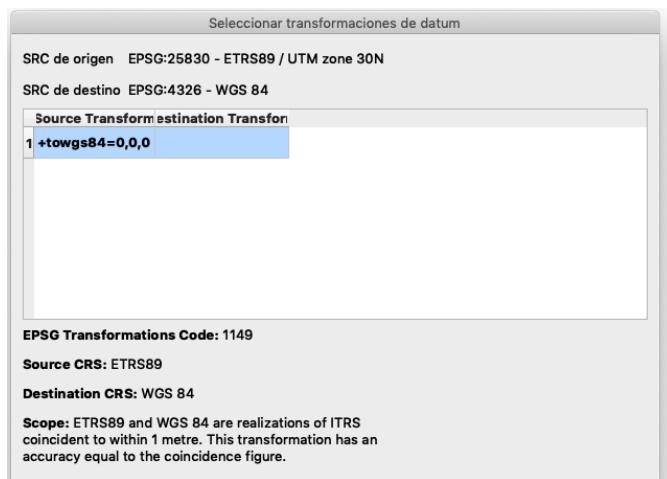
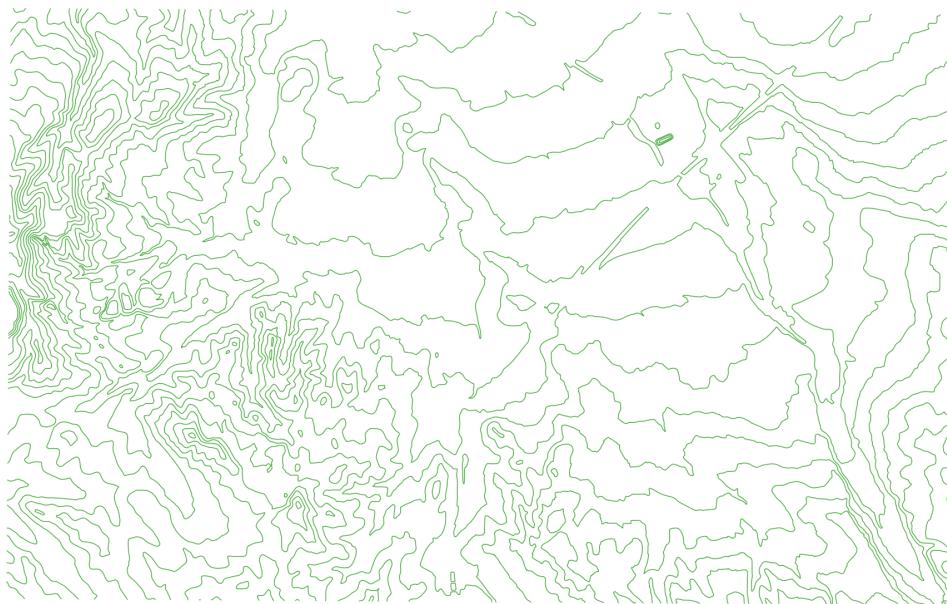


Figura 1.3

Figura 1.4: Visualización del elemento *Edificaciones* en QGIS

- **Edificaciones:** se agrupan en esta capa varios elementos que conforman edificaciones con geometría de polígono (figura 1.4). Dispone de la información:
  - GID.
  - ID DE LA HOJA.
  - ESTADO: estado de uso de la entidad o el tramo de la entidad.

- En construcción (CON), en ruinas (RUI), sin clasificar (SCL) o en uso (USO).
- TIPO: tipo de elemento según su función.
  - Casetas o cobertizos (CAS), caso genérico (CGN), chabolas (CHA), chimeneas (CHI), edificación (EDI), industrial (IND), invernadero (INV), marquesina (MAR), nave abierta (NAB), nicho (NIC), patio (PAT), sin clasificar (SCL), tentadero (TEN), torre genérica (TGN), transformador (TRF) o torre de vigía (TVG).



**Figura 1.5:** Visualización del elemento *Curvas de Nivel* en QGIS

- **Curvas de nivel:** línea imaginaria de altitud constante que sirve para describir la forma tridimensional de la superficie terrestre con geometría de línea (figura 1.5). Dispone de la información:
  - GID.
  - ID DE LA HOJA.
  - COTA: recoge la coordenada altura ortométrica del elemento capturado en metros.
  - CATEGORÍA: categoría de la curva de nivel.
    - Auxiliar (AUX), maestra (MAE), normal (NOR) o sin clasificar (SCL).
  - PROCEDENCIA: procedencia de la curva de nivel.
    - Combinado (CMB), elementos terreno (ETE), lidar (LID), MDT (MDT), restitución (RES) o sin clasificar (SCL).

- TIPO: tipo de la curva de nivel.
  - Caso genérico (CGN), depresión (DEP) o sin clasificar (SCL).

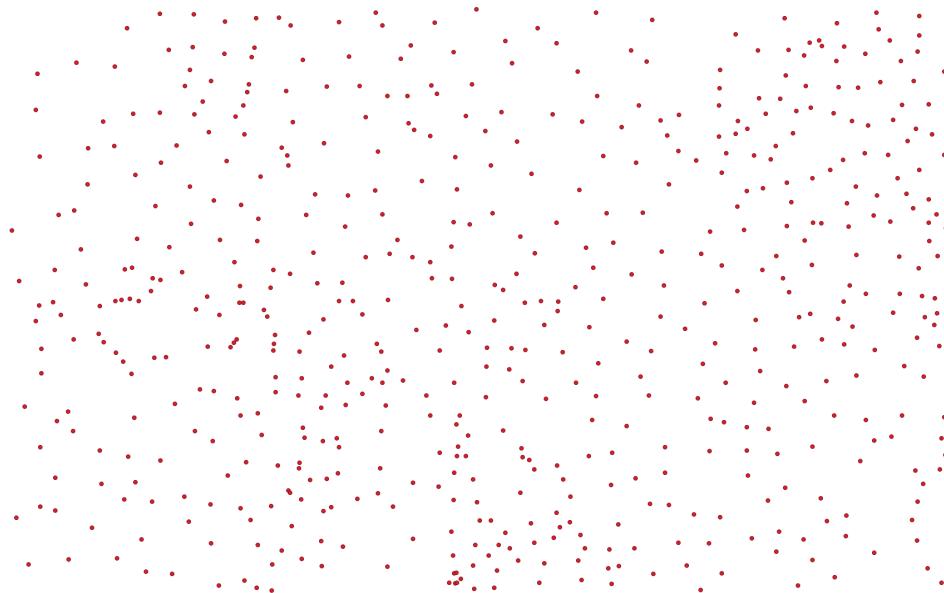
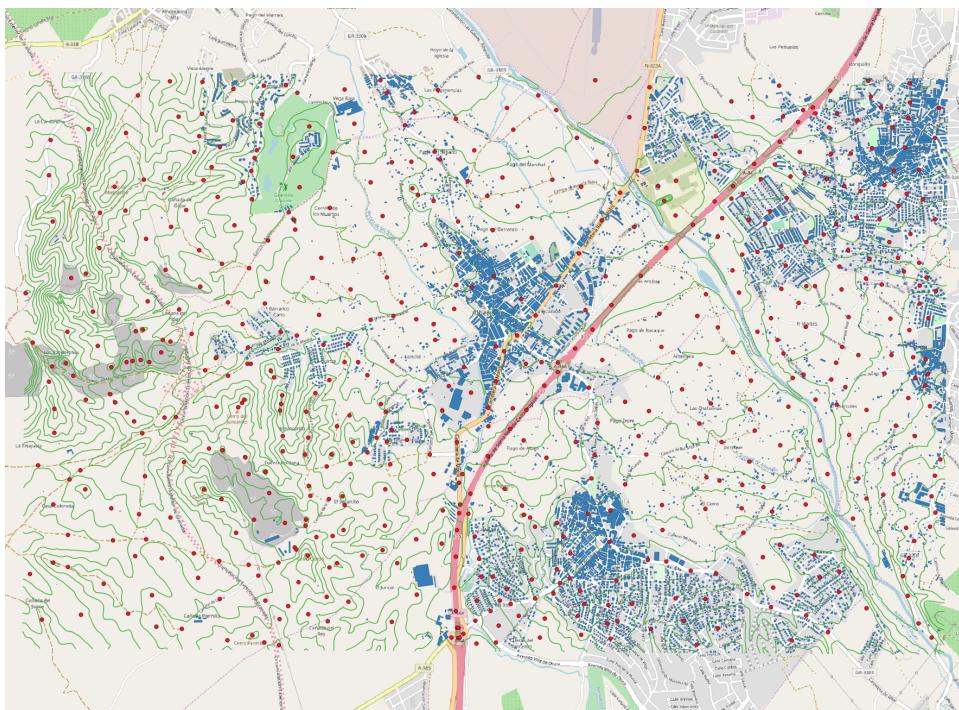


Figura 1.6: Visualización del elemento *Punto Cota* en QGIS

- **Punto cota:** punto situado sobre la superficie terrestre del cual se conoce su altitud sobre el nivel medio del mar, y que se representa para facilitar la interpretación gráfica de la morfología del terreno con geometría de punto (figura 1.6). Dispone de la información:
  - GID.
  - ID DE LA HOJA.
  - COTA: recoge la coordenada altura ortométrica del elemento capturado en metros.
  - CONTEXTO: contexto del punto de cota.
    - Caso genérico (CGN), cima (CIM), collado (COL), depresión (DEP), edificación (EDI) o sin clasificar (SCL).
- TIPO: tipo de elemento según su función.
  - Punto cota construcción elevada (CON) o punto cota terreno (TER).

Con esto hemos terminado de explicar los datos que vamos hacer accesibles a través de la Web Semántica. En la figura 1.7 es posible visualizar las tres geometrías juntas y localizadas en OpenStreetMap, en donde las líneas verdes hacen referencia a las curvas de nivel, los puntos rojos a los puntos de cota y los polígonos azules a las edificaciones.



**Figura 1.7: Visualización de las tres geometrías en OpenStreetMap**

Una vez decididos los datos que vamos a usar, debemos pasarlo a un formato que sea adecuado, es decir, para poblar la ontología haciendo uso de Protegé se necesita Excel worbook, formato que nos ofrece QGIS. Por otra manera,

Se han propuesto otros esquemas para codificar datos de geometría simple en RDF. El vocabulario de Geografía básica del W3C (<http://www.w3.org/2003/01/geo/>) es un vocabulario popular. Estos vocabularios simples tienen limitaciones, por ejemplo, la incapacidad de especificar diferentes datos y sistemas de coordenadas, y por lo tanto no se usaron en GeoSPARQL. Tenga en cuenta que la mayoría de los datos de geometría existentes codificados con estos vocabularios se pueden convertir fácilmente en representaciones GeoSPARQL. La consulta SPARQL a continuación crea valores geo: wktLiteral a partir de geometrías de geografía básica del W3C.

Tenga en cuenta que la serialización elegida afecta fuertemente la conceptualización de la geometría. La serialización WKT alinea los tipos de geometría con las características simples ISO 19125 [ISO 19125-1], y la serialización GML alinea los tipos de geometría con el esquema espacial ISO 19107 [6].

Esta sección establece los requisitos para representar datos de geometría en RDF basados en WKT según lo definido por Características simples [ISO 19125-1].

Todos los literales RDFS de tipo geo: wktLiteral consistirán en un URI opcional que identifica el sistema de referencia de coordenadas seguido de Texto bien conocido de características simples (WKT) que describe un valor geométrico. Geo válido: wktLiterals se forman concatenando un URI absoluto válido como se define en [RFC 2396], uno o más espacios (carácter Unicode U + 0020) como separador y una cadena WKT como se define en Características simples [ISO 19125-1]

```
1 "Point(-83.38 33.95)"^^<http://www.opengis.net/ont/geosparql#wktLiteral>
```

Para obtener el formato CSV en QGIS se realizar el mismo procesa en las tres capas: click derecho sobre la capa en cuestión, Exportar >Guardar objetos como, y nos aparecerá una ventana como la siguiente.

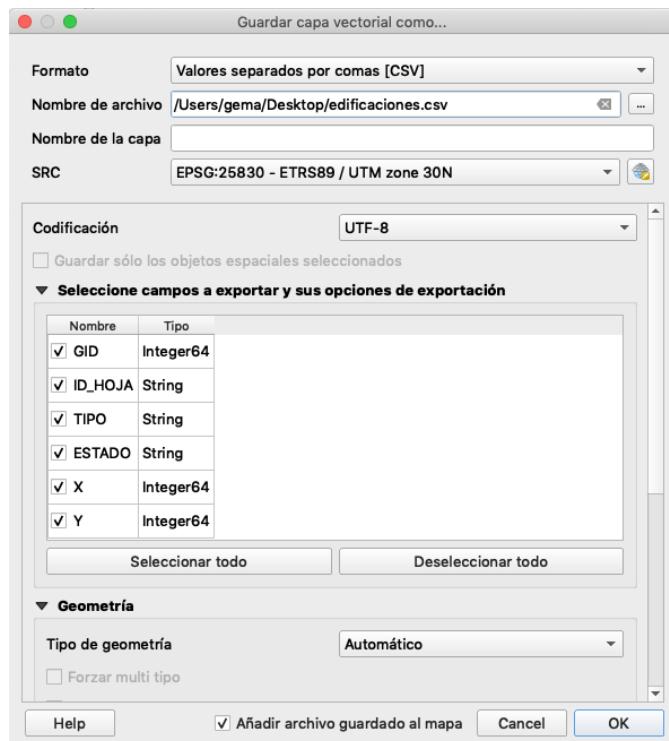


Figura 1.8

Como hemos comentado anteriormente, para manejar la geometría es necesario hacer uso de la codificación WKT, así que seleccionamos dicha opción, para que a parte de obtener los atributos del modelos de datos en cuestión, podamos obtener la información pertinente a WKT.

Para cualquier forma más compleja que un punto de latitud / longitud, WKT es nuestra única opción. Se admiten muchas formas; Estos son algunos de los que vemos más comúnmente:

Punto (LONG LAT): un punto único como se describió anteriormente  
Tenga en cuenta la falta de una coma Cadena de líneas (LONG1 LAT1, LONG2 LAT2, ..., LONGN LATN): una línea que conecta los puntos especificados Comas entre cada punto Sobre (minLong, maxLong, maxLat, minLat): un rectángulo con las esquinas especificadas Tenga en cuenta las comas entre cada Especialmente tenga en cuenta el orden algo extraño de (min, max, max, min). Para formas más complejas, Stardog admite JTS. Al descargar y habilitar esta biblioteca, obtiene acceso a estas formas, especialmente:

Polígono (LONG1 LAT1, LONG2 LAT2, ..., LONGN LATN, LONG1 LAT1): una forma rellena con los puntos especificados Tenga en cuenta que un polígono debe comenzar y terminar con el mismo punto, es decir, estar cerrado



Figura 1.9

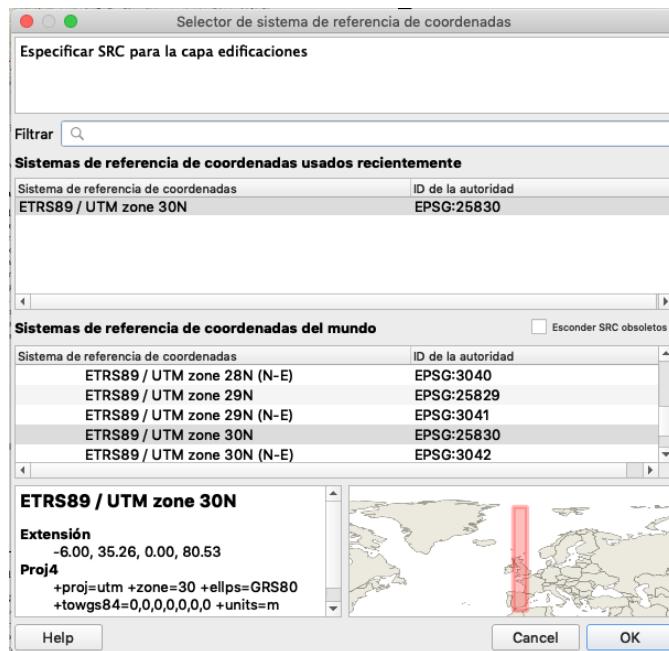


Figura 1.10

Realizando el mismo proceso para las tres capas obtenemos la siguiente información en Hoja de Excel, a continuación ponemos un trocito de cada uno de ellos.

Dicir el tamaño que tiene cada CSV

WKT	GID	ID_HOJA	TIPO	CA
LINESTRING (440675.4 4106319.93, ..., 440675.4 4106319.93)	122602	102632	CGN	NO
LINESTRING (440972.52 4106237.28, ..., 440972.52 4106237.28)	172684	102632	CGN	NO
LINESTRING (440976.2 4106311.24, ..., 440976.2 4106311.24)	186391	102632	CGN	NO
...	...	...	...	...
LINESTRING (442507.66 4105386.75, ..., 442507.66 4105386.75)	471150	102632	DEP	MA

WKT	GID	ID_HOJA	TIPO	CONTEXTO	COTA
MULTIPOINT ((446228.92 4108432.53))	890947	102632	TER	CGN	721.77
MULTIPOINT ((446265.6 4108675.83))	890948	102632	TER	CGN	715.40
MULTIPOINT ((445197.65 4107245.48))	890949	102632	TER	CGN	761.02
...	...	...	...	...	...
MULTIPOINT ((442401.99 4104501.95))	891529	102632	TER	CGN	771.75

WKT	GID	ID_HOJA	T
POLYGON ((446020.74 4107035.28, ... ,446020.74 4107035.28))	181062	102632	E
POLYGON ((446050.16 4107127.71, ... ,446050.16 4107127.71))	181064	102632	E
POLYGON ((441430.69 4106827.08, ... ,441430.69 4106827.08))	758271	102632	E
...	...	...	...
POLYGON ((446168.83 4108720.87, ... , 446168.83 4108720.87))	1330488	102632	P

La información que está contenida en la muestra de las anteriores tablas, contiene la información para poblar nuestra ontología.

### 1.1.2. Creación de la ontología

Una vez seleccionado el conjunto de datos que se desea hacer accesible mediante la Web Semántica, es hora de pasar al desarrollo y creación de la ontología. Gracias a la ontología es posible romper la barrera de la interoperabilidad, en donde los datos geoespaciales pueden ser utilizados por diferentes tipos de programas y aplicaciones. La interoperabilidad de los datos geoespaciales es extremadamente importante para las aplicaciones geoespaciales, ya que existen grandes cantidades de datos espaciales en diferentes formatos geográficos. La interoperabilidad de los datos geoespaciales elimina las barreras para el intercambio de datos y permite a los usuarios acceder, mapear, visualizar y analizar directamente datos con diferentes formatos de datos espaciales. Los datos geoespaciales interoperables hacen posible la distribución rápida de información y el intercambio entre departamentos [4].

No obstante, escribir en lenguajes como RDF y OWL resultan sumamente difícil y propensos a errores. Afortunadamente, existen en el mercado entornos gráficos para visualizar y construir ontologías de forma más fácil, como **Protegé**. Es por eso que vamos hacer uso de este software para la creación de nuestra ontología.

software desarrollado por la Universidad de Stanford. Permite editar ontologías con una interfaz sencilla y en un entorno de menús, botones, cuadros de diálogo o representaciones gráficas fáciles de usar, tan importantes para una tarea con tanta de abstracción y síntesis.

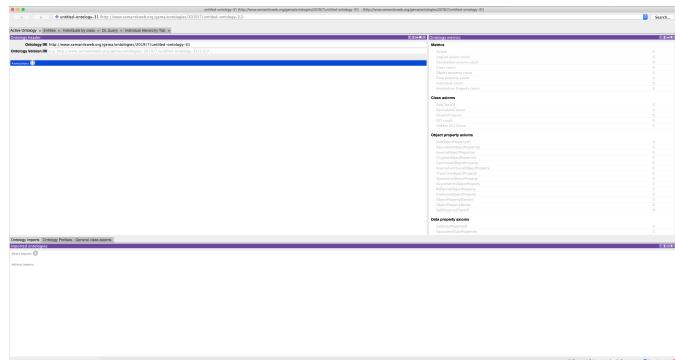


Figura 1.11

3. Nos vamos a Protegé, como comentamos previamente Protegé es una herramienta de fácil uso que nos permite crear ontologías de manera más fácil y de forma gráfica, lo que facilita su creación en un mayor tiempo.

Hay que tener en cuenta que la ontología está enfocada para ser manejada en España y los datos son procedentes de Andalucía, es por eso que los nombres aparecerán en Español, a excepción de los que se nos especifique en GeoSPARQL.

Hay que crearlo con los URIS determinados, siguiendo las especificaciones de GeoSPARQL.

Básicamente una ontología geoespacial se basa en:

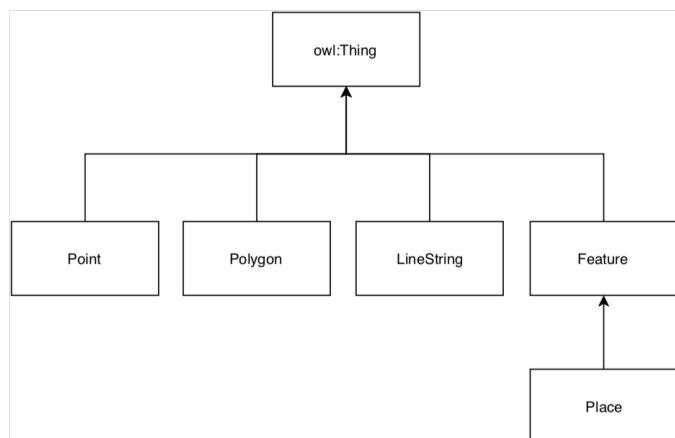


Figura 1.12

A parte de hacer uso de las especificaciones que tiene GeoSPARQL, es interesante añadir información específica de nuestros datos geográficos, con el fin de enriquecer más nuestra ontología.

Lo primero que tenemos que hacer es definir los prefijos con los que

vamos a trabajar, como vamos hacer uso de GeoSPARQL, necesitaremos los siguientes.

El de Applicattion, es para manejar los datos que hemos metido y no son de ningun estandar de GeoSPARQL.

Ontology Imports   Ontology Prefixes   General class axioms	
Ontology prefixes:	
Prefix	Value
geo	# http://www.opengis.net/ont/geosparql#
my	http://example.org/ApplicationSchema#
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
sf	http://www.opengis.net/ont/sf#
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema#

Figura 1.13

- Creamos el tipo Polygon con IRI: <http://www.opengis.net/ont/sf> -
- Creamos el tipo Point con IRI: <http://www.opengis.net/ont/sf> -
- Creamos el tipo LineString con IRI: <http://www.opengis.net/ont/sf> -
- Creamos el tipo Feature con IRI: <http://www.opengis.net/ont/geosparql> -
- Creamos el subtipo GID con IRI: <http://example.org/ApplicationSchema>

- Creamos el tipo Edificaciones con IRI: <http://example.org/ApplicationSchema>
- Creamos el subtipo TipoEdificaciones con IRI: <http://example.org/ApplicationSchema>
- Creamos el subtipo EstadoEdificaciones con IRI: <http://example.org/ApplicationSchema>
  
- Creamos el tipo PuntoCota con IRI: <http://example.org/ApplicationSchema>
- Creamos el subtipo TipoPuntoCota con IRI: <http://example.org/ApplicationSchema>
- Creamos el subtipo ContextoPuntoCota con IRI: <http://example.org/ApplicationSchema>
  
- Creamos el tipo CurvaNivel con IRI: <http://example.org/ApplicationSchema>
- Creamos el subtipo CategoriaCurvaNivel con IRI: <http://example.org/ApplicationSchema>
- Creamos el subtipo ProcedenciaCurvaNivel con IRI: <http://example.org/ApplicationSchema>
- Creamos el subtipo TipoCurvaNivel con IRI: <http://example.org/ApplicationSchema>

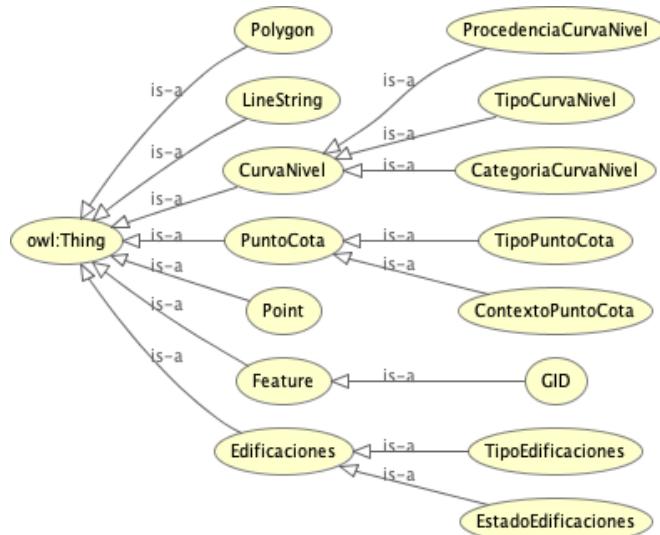


Figura 1.14: Ontología de prototipo

Si usamos WKT es necesario crear el tipo de dato

RDFS Datatype: geo:wktLiteral

- Creamos en Datatype: wktLiteral con IRI <http://www.opengis.net/ont/geosparql>

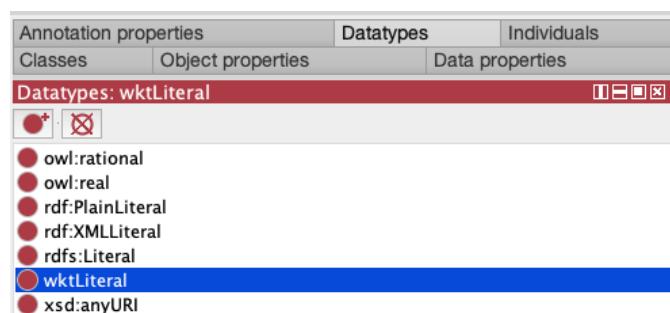


Figura 1.15

Si usamos WKT es necesario crear la propiedad

- Creamos la propiedad asWKT con IRI: <http://www.opengis.net/ont/geosparql>
- Creamos la propiedad tieneCota con IRI: <http://example.org/ApplicationSchema>

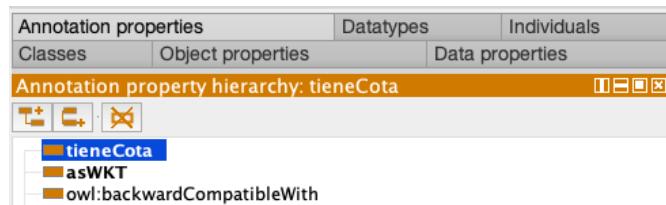


Figura 1.16

Con esto ya hemos terminado de definir nuestra ontología. A continuación, pasamos a poblarla.

### 1.1.3. Poblar la ontología

Como comentamos, Protegé nos permite poblar una ontología haciendo uso de las herramientas que ofrece, es por eso que la elección de dicho software ha sido principalmente por esta característica, ya que en todo caso, haber creado a mano las miles de instancias que tenemos no hubiera sido rentable ni eficiente. Para poblar una ontología en Protegé con la información procedente del CSV del Shapefile, nos vamos a Tools > Create axioms for Excel Workbook y abrimos el fichero en cuestión. La única dificultad que presenta esta manera es que debemos aprender unas reglas para poblar, sin embargo, desde mi punto de vista esta manera es más eficiente que hacerlo a mano.

5. Se nos abre una pestaña con la que tenemos que crear reglas para poblar. Seleccionamos todo el libro de Excel y empezamos a crear las reglas:

Para guardar las instancias de cada una de las capas, vamos hacer uso del GID, y no del ID HOJA; puesto que el GID es único para cada uno.

#### Edificaciones

Geometría polígono

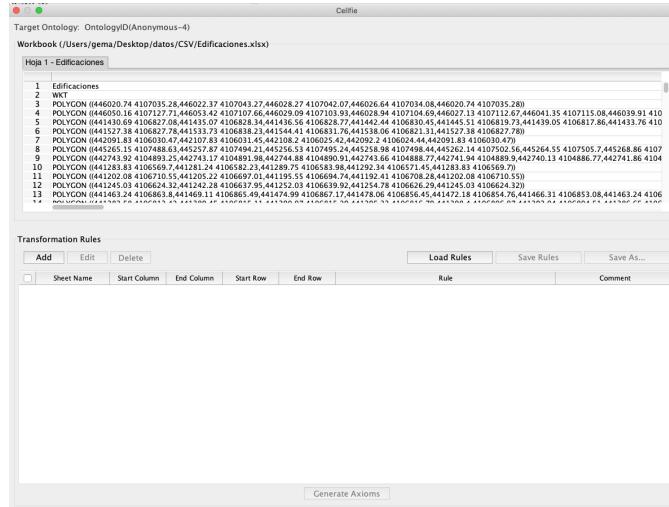


Figura 1.17

REGLA PARA AÑADIR INSTANCIAS A LA CLASE GID y Polígono

```

1 Individual: @B*
2 Types: GID, Polygon

```

REGLA PARA AÑADIR LAS SUBCLASES A LA CLASE tipoEdificaciones y su asignación con sus respectivos GID

```

1 Class: @D*
2 SubClassOf: TipoEdificaciones
3
4 Individual: @B*
5 Types: @D*

```

REGLA PARA AÑADIR LAS SUBCLASES A LA CLASE EstadoEdificaciones y su asignación con sus respectivos GID

```

1 Class: @E*
2 SubClassOf: EstadoEdificaciones
3
4 Individual: @B*
5 Types: @E*

```

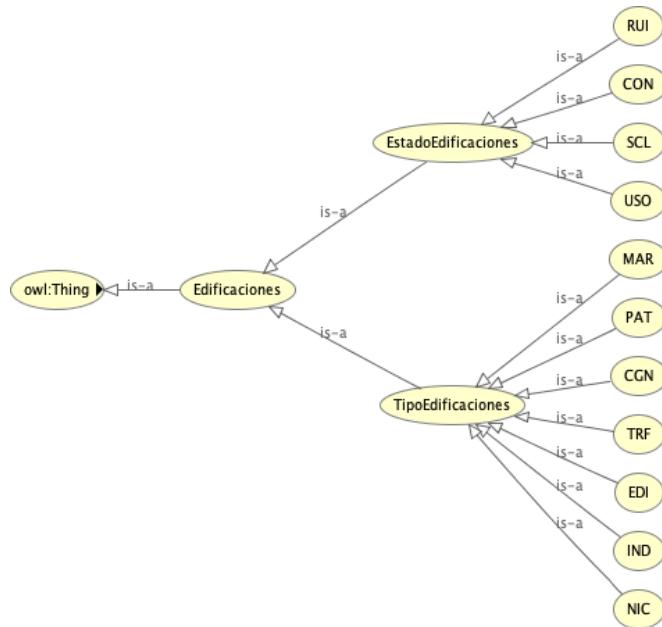


Figura 1.18

Regla para añadir a asWKT la codificación del Polígono

- 1 `Individual: @B*`
- 2 `Annotations: asWKT @A* ( wktLiteral )`

Sheet Name	Start Column	End Column	Start Row	End Row	Rule	Comment
Hoja 1 - Edificacion-A	E	3	7303		Individual: @B*	Types: GID, Polygon
Hoja 1 - Edificacion-A	E	3	7303		Individual: @B*	Types: GID, Polygon
Hoja 1 - Edificacion-A	E	3	7303		Individual: @B*	Annotations: asWKT @A*
Hoja 1 - Edificacion-A	E	3	7303		Class: @C	SubClassOf: TipoEdificaciones
Hoja 1 - Edificacion-A	E	3	7303		Class: @C	SubClassOf: EstadoEdificaciones
Hoja 1 - Edificacion-A	E	3	7303		Individual: @B*	Types: GID

Figura 1.19

### CurvaNivel

Geometría Linea

REGLA PARA AÑADIR INSTANCIAS A LA CLASE GID y LineString

- 1 `Individual: @B*`
- 2 `Types: GID, LineString`

REGLA PARA AÑADIR LAS SUBCLASES A LA CLASE TipoCurva-Nivel y su asignación con sus respectivos GID

- 1 `Class: @D*`

```

2 SubClassOf: TipoCurvaNivel
3
4 Individual: @B*
5 Types: @D*
```

REGLA PARA AÑADIR LAS SUBCLASES A LA CLASE ProcedenciaCurvaNivel y su asignación con sus respectivos GID

```

1 Class: @F*
2 SubClassOf: ProcedenciaCurvaNivel
3
4 Individual: @B*
5 Types: @F*
```

REGLA PARA AÑADIR LAS SUBCLASES A LA CLASE CategoriaCurvaNivel y su asignación con sus respectivos GID

```

1 Class: @E*
2 SubClassOf: CategoriaCurvaNivel
3
4 Individual: @B*
5 Types: @E*
```

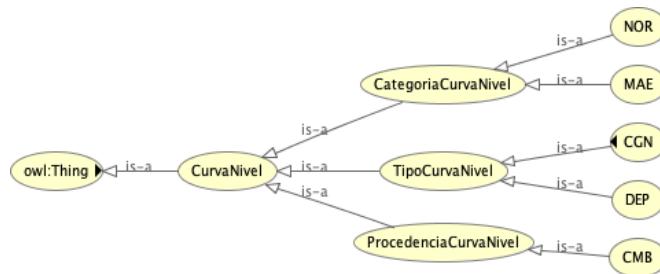


Figura 1.20

Regla para añadir a asWKT la codificación del LineString

```

1 Individual: @B*
2 Annotations: asWKT @A* ( wktLiteral )
```

Regla para añadir a tieneCota la codificación del LineString

```

1 Individual: @B*
2 Annotations: tieneCota @G* ( xsd:double )
```

	Sheet Name	Start Column	End Column	Start Row	End Row	Rule	Comment
5	Hoja 1 - curvanevel	A	G	3	155	Individual: @B* Types: @D*	
6	Hoja 1 - curvanevel	A	G	3	155	Individual: @B* Types: @E*	
7	Hoja 1 - curvanevel	A	G	3	155	Type: GDI_LineString	
8	Hoja 1 - curvanevel	A	G	3	155	Individual: @B* Types: @E*	
9	Hoja 1 - curvanevel	A	G	3	155	Individual: @B* Types: @F*	
10	Hoja 1 - curvanevel	A	G	3	155	Individual: @B* Types: @G*	
11	Hoja 1 - curvanevel	A	G	3	155	Individual: @B* Types: @H*	
12	Hoja 1 - curvanevel	A	G	3	155	Individual: SWKT (@A*) Class: @D*	
13	Hoja 1 - curvanevel	A	G	3	155	SubClassOf: TipoCurvaNivel	
14	Hoja 1 - curvanevel	A	G	3	155	SubClassOf: CategoríaCurvaNivel	
15	Hoja 1 - curvanevel	A	G	3	155	Individual: @B* Annotations: tieneCota @C* (xsd:double)	
16	Hoja 1 - curvanevel	A	G	3	155	Class: @E*	
17	Hoja 1 - curvanevel	A	G	3	155	SubClassOf: ProcedenciaCurvaNivel	

Figura 1.21

PuntoCota

## Geometría Punto

## REGLA PARA AÑADIR INSTANCIAS A LA CLASE GID y LineString

1 Individual: @B\*  
2 Types: GID, Point

## REGLA PARA AÑADIR LAS SUBCLASES A LA CLASE TipoPunto-Cota y su asignación con sus respectivos GID

```
1 Class: @E*
2 SubClassOf: TipoPuntoCota
3
4 Individual: @B*
5 Types: @D*
```

## REGLA PARA AÑADIR LAS SUBCLASES A LA CLASE Contexto-PuntoCota y su asignación con sus respectivos GID

```
1 Class: @E*
2 SubClassOf: ContextoPuntoCota
3
4 Individual: @B*
5 Types: @E*
```

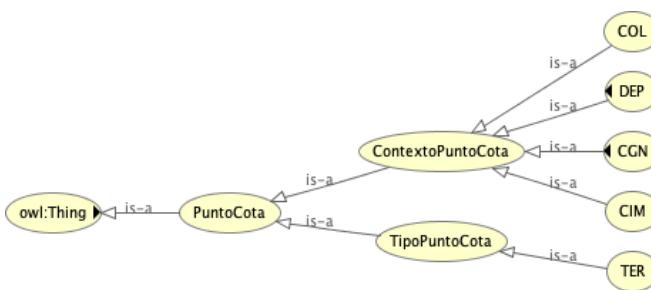


Figura 1.22

Regla para añadir a asWKT la codificación del Polígono

	Sheet Name	Start Column	End Column	Start Row	End Row	Rule	Comment
1	Hoja 1 - PuntoCota	A	F	3	\$85	Individual: @B*	
2	Hoja 1 - PuntoCota	A	F	3	\$85	Annotations: tieneCota @F* (xsd:double)	
3	Hoja 1 - PuntoCota	A	F	3	\$85	Individual: @B*	
4	Hoja 1 - PuntoCota	A	F	3	\$85	Types: @D*	
5	Hoja 1 - PuntoCota	A	F	3	\$85	Individual: @B*	
6	Hoja 1 - PuntoCota	A	F	3	\$85	Types: GID, Point	
7	Hoja 1 - PuntoCota	A	F	3	\$85	Class: @E	
8	Hoja 1 - PuntoCota	A	F	3	\$85	SubClassOf: TipoPuntoCota	
9	Hoja 1 - PuntoCota	A	F	3	\$85	Individual: @B*	
10	Hoja 1 - PuntoCota	A	F	3	\$85	Annotations: asWKT @A*	
11	Hoja 1 - PuntoCota	A	F	3	\$85	Class: @E	
12	Hoja 1 - PuntoCota	A	F	3	\$85	SubClassOf: ContextoPuntoCota	
13	Hoja 1 - PuntoCota	A	F	3	\$85	Individual: @B*	
14	Hoja 1 - PuntoCota	A	F	3	\$85	Types: @E*	

Figura 1.23

1 Individual: @B\*  
 2 Annotations: asWKT @A\* (wktLiteral)

Regla para añadir a tieneCota la codificación del Point

1 Individual: @B\*  
 2 Annotations: tieneCota @F\* (xsd:double)

Y con esto hemos terminado nuestra ontología, para ver la ontología es posible ver el fichero ontologia.owl

Metrics	
Axiom	49160
Logical axiom count	32322
Declaration axioms count	8065
Class count	33
Object property count	0
Data property count	0
Individual count	8029
Annotation Property count	2
Class axioms	
SubClassOf	29
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0

Figura 1.24

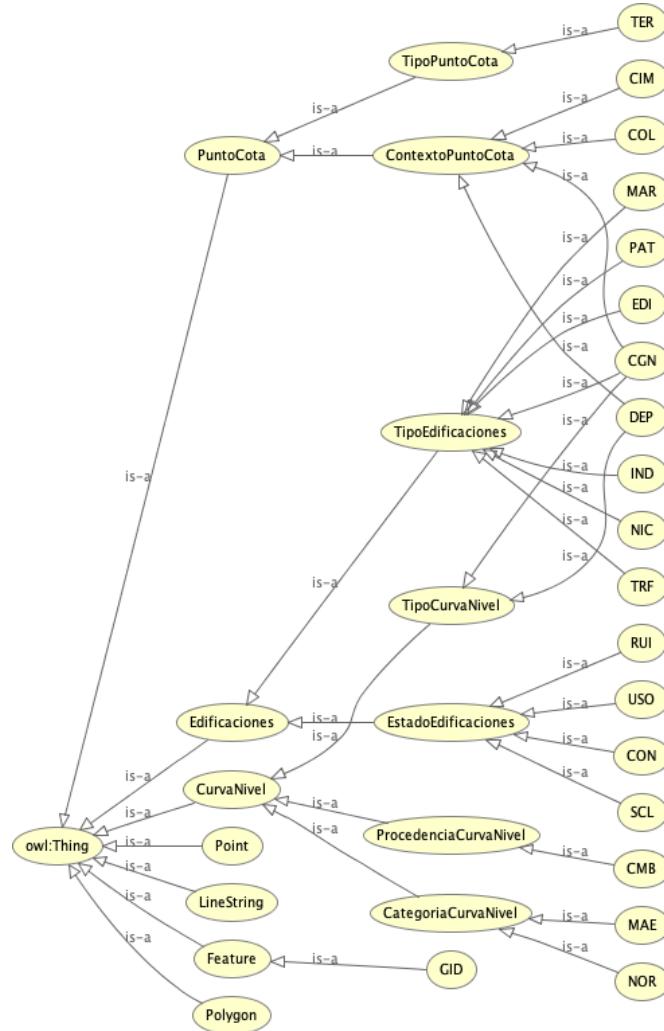


Figura 1.25

#### 1.1.4. Realización de consultas GeoSPARQL

Una vez creada nuestra ontología podemos pasar a hacer las consultas.

Por alguna razón no puedo hacer uso de GeoSPARQL en Protegé 5.5, no detecta los prefijos que tiene GeoSPARQL. Así que he estado buscando si hay algún plugin con dicha librería, pero no ha habido éxito. Lo único que he encontrado. En el Apéndice se puede ver más acerca de los errores encontrados y como he conseguidos solventarlos.

Ya que GeoSPARQL es una extensión de SPARQL, por lo que no viene incluido en el mismo, sino que es necesario hacer uso de plugin

Así que voy hacer uso de GraphDB, que permite hacer uso de GeoS-

PARQL, y una herramienta de fácil uso.

#### Activar los plugin necesarios para manejar GeoSPARQL

Habilitar complemento Cuando el complemento está habilitado, indexa todos los datos GeoSPARQL existentes en el repositorio y reindexa automáticamente cualquier actualización.

```

1 PREFIX : <http://www.ontotext.com/plugins/geosparql#>
2
3 INSERT DATA {
4   :s :enabled "true" .
5 }
```

Ignore errors on indexing

Ignorar errores en la indexación

El predicado ignoreErrors determina si el índice GeoSPARQL continuará construyéndose si ha ocurrido un error. Si el valor se establece en falso, el índice completo fallará si hay un problema con un documento. Si el valor se establece en verdadero, el índice continuará construyéndose y se registrará una advertencia en el registro. Por defecto, el valor de ignoreErrors es falso.

```

1 PREFIX : <http://www.ontotext.com/plugins/geosparql#>
2
3 INSERT DATA {
4   :s :ignoreErrors "true"
5 }
```

Importamos la ontología y ya podemos hacer pruebas con ella.

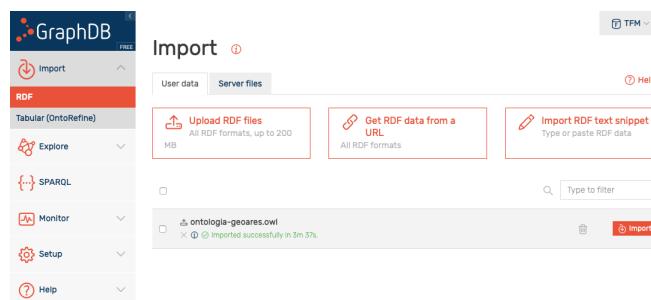


Figura 1.26

Además una ventaja extra que tiene esta herramienta es que nos permite inspeccionar el grafo creado

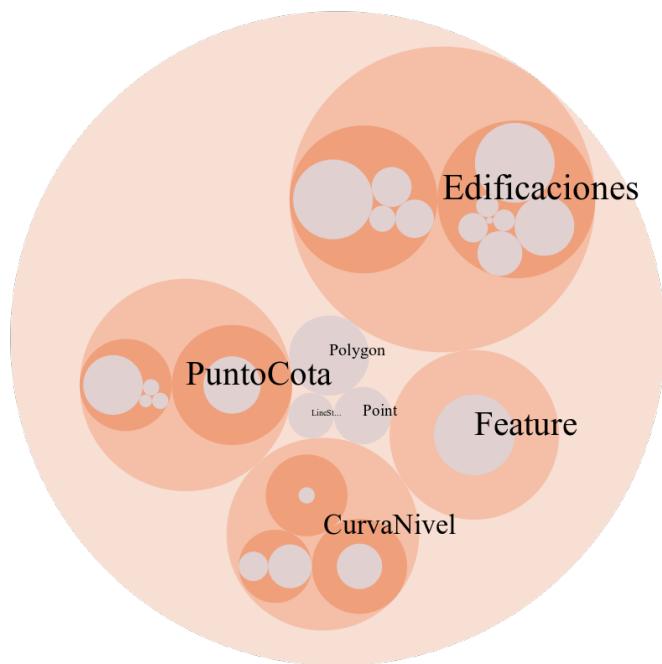


Figura 1.27

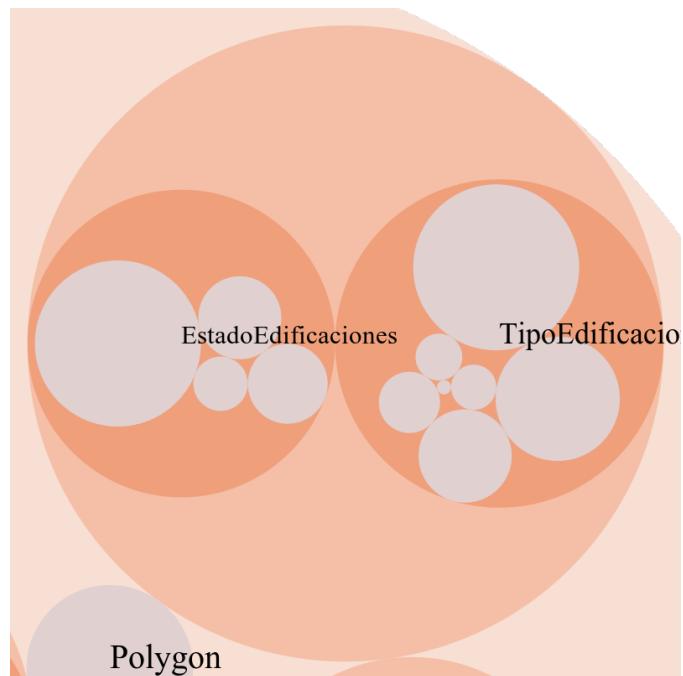


Figura 1.28

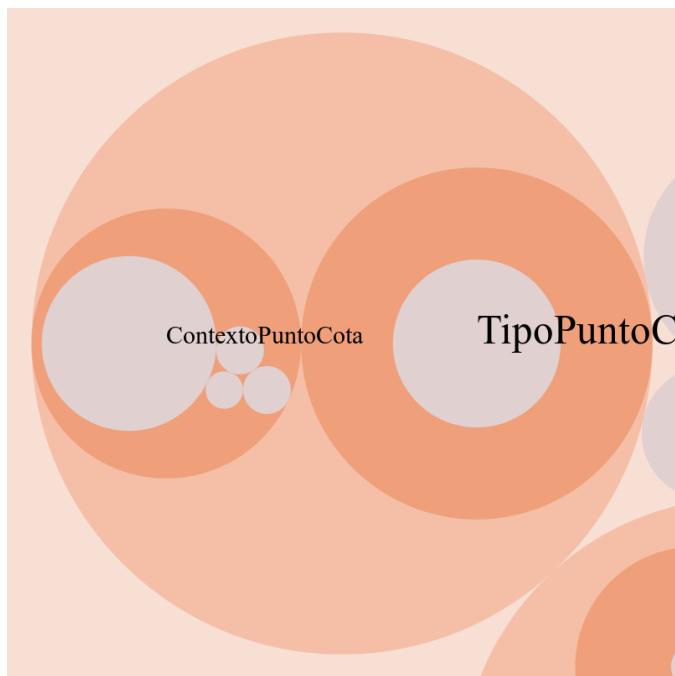


Figura 1.29

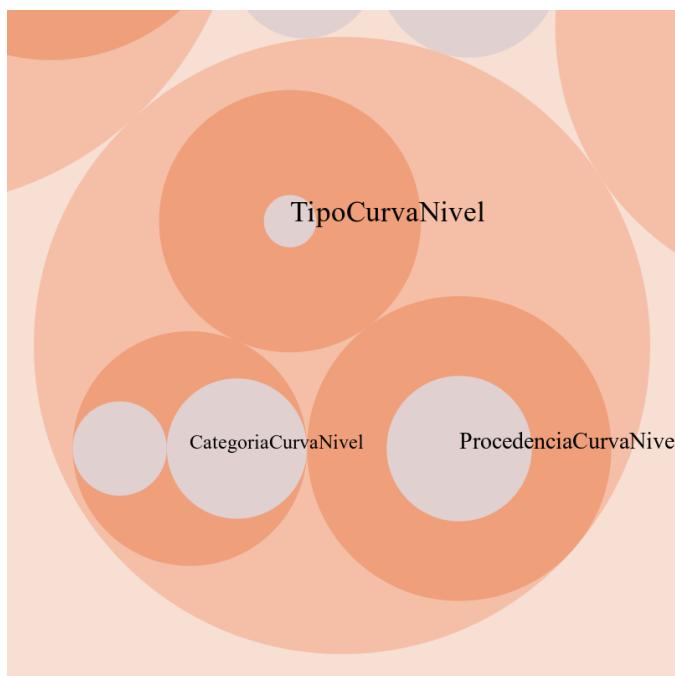


Figura 1.30

Para entrar en contacto con las consultas de SPARQL y GeoSPARQL, vamos a recordar unas cuantas cosas.

Lo primero, disponemos de la información geográfica espacial referente a Ogíjares con respecto edificaciones, curvas de nivel y cotas de altura. Con esta información, que fue seleccionada para un fin, queremos saber las edificaciones que se encuentran a una cierta altura con el fin de obtener un objetivo, para ello, hemos cogido dos geometrías que nos pueden dar la misma aproximación pero con geometrías distintas, una con puntos y otra con líneas.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX my: <http://example.org/ApplicationSchema#>
4 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
5 PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
6
7 SELECT ?x ?p
8 WHERE {
9 ?x rdf:type geo:EDI .
10 ?polygon geo:asWKT ?p
11 }
12 LIMIT 3

```

	x	p
1	geosparql:181062	"POLYGON ((446020.74 4107035.28,446022.37 4107043.27,446028.27 4107042.07,446026.64 4107034.08,446020.74 4107035.28))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>
2	geosparql:181062	"POLYGON ((446050.16 4107127.1,446053.42 4107107.66,446029.09 4107103.93,446028.94 4107104.69,446027.13 4107112.67,446041.35 4107115.08,446039.91 4107125.24,446050.16 4107127.71))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>
3	geosparql:181062	"POLYGON ((441430.69 4106827.08,441435.07 4106828.34,441436.56 4106828.77,441442.44 4106830.45,441445.51 4106819.73,441439.05 4106817.86,441433.76 4106816.36,441430.69 4106827.08))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>

Figura 1.31

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX my: <http://example.org/ApplicationSchema#>
4 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
5 PREFIX sf: <http://www.opengis.net/ont/sf#>
6 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
7
8 SELECT ?f
9
10 WHERE {
11 ?f geo:tieneCota ?fWKT
12 FILTER (?fWKT > 700)
13 }
14 LIMIT 5

```

	f	•
1	geosparql:890971	
2	geosparql:890972	
3	geosparql:890969	
4	geosparql:891145	
5	geosparql:891068	

Figura 1.32

```
1 PREFIX my: <http://example.org/ApplicationSchema#>
2 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
3 SELECT DISTINCT ?point ?pointWKT ?pointCOTA
4 WHERE
5 {
6 # Obtenemos todos los puntos con una cota determinada (>
7 # 720)
8 ?point geo:asWKT ?pointWKT ;
9 geo:tieneCota ?pointCOTA .
10 FILTER (xsd:double(?pointCOTA) > 720)
11 }
```

	point	pointWKT	pointCOTA
1	<a href="#">geosparql:122602</a>	"LINESTRING (440675.4 4106319.93,440667.97 4106322.69,440666.96 4106326.35,440665.15 4106330.44,440664.95 4106335.46,440666.77 4106340.74,440669.25 4106342.74,440674.38 4106344.2,440667.98 4106344.52,440668.15 4106343.42,440670.62 4106330.14,440670.19 410632.6,440670.98 410632.33,440670.62 410632.0,440670.19 410631.93)">"http://www.opengis.net/ont/geosparql#wktLiteral"	"770.0"**xsd:double
2	<a href="#">geosparql:172684</a>	"LINESTRING (440972.52 4106237.28,440972.4 4106232.04,440972.15 4106226.75,440972.76 4106204.44,440964.32 410605.94,440963.48 4106208.23,440962.08 4106213.03,440960.59 4106219.28,440959.19 4106224.26,440958.99 4106227.94,440958.44 410623.81,440958.11 4106240.01,440958.05 4106245.3,440958.04 4106250.86,440958.16 4106255.93,440958.41 4106261.44,440958.21 4106226.96,440964.76 4106272.24,440971.61 4106288.61,440974.39 4106281.06,440974.85 4106275.81,440974.77 4106270.3,440972.52 4106237.28)">"http://www.opengis.net/ont/geosparql#wktLiteral"	"780.0"**xsd:double
3	<a href="#">geosparql:186391</a>	"LINESTRING (440972.6 4106311.24,440973.32 4106318.03,440973.13 4106323.23,440973.29 4106328.26,440973.15 4106333.74,440973.76 4106339.75,440974.28 4106344.93,440974.76 4106350.18,440975.15 4106355.29,440975.73 4106350.54,440975.97 4106356.62,440997.22 4106353.18,440997.56 4106384.03,440998.24 4106382.27,440998.76 4106379.78,440998.63 4106375.83,440998.79 4106370.67,440998.97 4106375.25,440998.75 4106360.0,440998.73 4106354.98,440998.73 4106349.88,440998.53 4106344.41,440998.41 4106339.26,440983.98 4106333.75,440983.14 4106328.19,440982.21 4106323.22,440982.19 4106318.02,440976.2 4106311.24)">"http://www.opengis.net/ont/geosparql#wktLiteral"	"780.0"**xsd:double

Figura 1.33

Hay que tener en cuenta que hay muchas maneras de hacer las consultas, por lo que para obtener una cosa, no es necesario realizar la misma consulta, como acabamos de comprobar.

```
1 PREFIX my: <http://example.org/ApplicationSchema#>
2 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
3
4 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5
```

```

6 SELECT DISTINCT ?point ?pointCota
7 WHERE {
8 # Obtenemos todos los puntos con una cota determinada (>
9    720)
10 ?point rdf:type geo:DEP ;
11 geo:tieneCota ?pointCota ;
12 FILTER (?pointCota > 700)
13 }
14 ORDER BY DESC(?pointCota)
15 LIMIT 5

```

	point	pointCota
1	geosparql:471150	"750.0"^^xsd:double
2	geosparql:460268	"720.0"^^xsd:double
3	geosparql:469869	"710.0"^^xsd:double

Figura 1.34

GEOF:sfEquals

```

1 PREFIX my: <http://example.org/ApplicationSchema#>
2 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
3 PREFIX geof: <http://www.opengis.net/def/function/geosparql
4 />
5
5 SELECT ?f
6 WHERE {
7 ?f geo:asWKT ?fWKT .
8 FILTER (geof:sfEquals(?fWKT, '''
9 LINESTRING (4106237.28,440972.4
10 4106232.04,440972.15 4106226.75,440967.68
11 4106204.44,440964.32 4106205.94,440963.48
12 4106208.23,440962.08 4106213.03,440960.59
13 4106219.28,440959.9 4106224.26,440958.99
14 4106229.78,440958.44 4106234.81,440958.11
15 4106240.01,440958.05 4106245.3,440958.04
16 4106250.86,440958.16 4106255.93,440958.41
17 4106261.44,440959.21 4106266.96,440960.76
18 4106272.24,440971.61 4106288.61,440974.39
19 4106281.06,440974.85 4106275.81,440974.77
20 4106270.3,440972.52 4106237.28)
21 ''''^geo:wktLiteral)))
22 }

```

	f	♦
1	geospqrql#172684	

Figura 1.35

```

1 <!-- http://www.opengis.net/ont/geosparql#172684 -->
2
3 <owl:NamedIndividual rdf:about="http://www.opengis.net/ont/
   geosparql#172684">
4 <rdf:type rdf:resource="http://www.opengis.net/ont/geosparql
   #CGN"/>
5 <rdf:type rdf:resource="http://www.opengis.net/ont/geosparql
   #CMB"/>
6 <rdf:type rdf:resource="http://www.opengis.net/ont/geosparql
   #GID"/>
7 <rdf:type rdf:resource="http://www.opengis.net/ont/geosparql
   #NOR"/>
8 <rdf:type rdf:resource="http://www.opengis.net/ont/sf#
   LineString"/>
9 <geo:asWKT rdf:datatype="http://www.opengis.net/ont/
   geosparql#wktLiteral">LINESTRING (440972.52
   4106237.28,440972.4 4106232.04,440972.15
   4106226.75,440967.68 4106204.44,440964.32
   4106205.94,440963.48 4106208.23,440962.08
   4106213.03,440960.59 4106219.28,440959.9
   4106224.26,440958.99 4106229.78,440958.44
   4106234.81,440958.11 4106240.01,440958.05
   4106245.3,440958.04 4106250.86,440958.16
   4106255.93,440958.41 4106261.44,440959.21
   4106266.96,440960.76 4106272.24,440971.61
   4106288.61,440974.39 4106281.06,440974.85
   4106275.81,440974.77 4106270.3,440972.52 4106237.28)</
   geo:asWKT>
10 <geo:tieneCota rdf:datatype="http://www.w3.org/2001/
   XMLSchema#double">780.0</geo:tieneCota>
11 </owl:NamedIndividual>
```

```

1 PREFIX my: <http://example.org/ApplicationSchema#>
2 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
3 PREFIX geof: <http://www.opengis.net/def/function/geosparql
   />
4
5 SELECT ?f
6 WHERE {
7 ?f geo:asWKT ?fWKT .
8 FILTER (geof:sfEquals(?fWKT, ''')
9 POLYGON ((446050.16 4107127.71,446053.42
   4107107.66,446029.09 4107103.93,446028.94
```

```

4107104.69,446027.13 4107112.67,446041.35
4107115.08,446039.91 4107125.24,446050.16 4107127.71))
10  ''''^ geo:wktLiteral)))
11 }

```

		f	
1	geosparql:181064		

Figura 1.36

	subject ↗	predicate ↗	object	context ↗
1	geos-parql:181064	geos-parql:asWKT	"POLYGON ((446050.16 4107127.71,446053.42 4107107.66,446029.09 4107103.93,446028.94 4107104.69,446027.13 4107112.67,446041.35 4107115.08,446039.91 4107125.24,446050.16 4107127.71))"	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
2	geos-parql:181064	rdf:type	geosparql:EDI	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
3	geos-parql:181064	rdf:type	geosparql:GID	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
4	geos-parql:181064	rdf:type	geosparql:USO	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
5	geos-parql:181064	rdf:type	sf:Polygon	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
6	geos-parql:181064	rdf:type	owl:NamedIndividual	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>

Figura 1.37

```

1 <!-- http://www.opengis.net/ont/geosparql#181064 -->
2
3 <owl:NamedIndividual rdf:about="http://www.opengis.net/ont/
4   geosparql#181064">
5 <rdf:type rdf:resource="http://www.opengis.net/ont/geosparql
6   #EDI"/>
7 <rdf:type rdf:resource="http://www.opengis.net/ont/geosparql
8   #GID"/>
9 <rdf:type rdf:resource="http://www.opengis.net/ont/geosparql
10  #USO"/>
11 <rdf:type rdf:resource="http://www.opengis.net/ont/sf#
    Polygon"/>
12 <geo:asWKT rdf:datatype="http://www.opengis.net/ont/
13   geosparql#wktLiteral">POLYGON ((446050.16
14   4107127.71,446053.42 4107107.66,446029.09
15   4107103.93,446028.94 4107104.69,446027.13
16   4107112.67,446041.35 4107115.08,446039.91
17   4107125.24,446050.16 4107127.71))</geo:asWKT>
18 </owl:NamedIndividual>

```

GEOF:WITHIN

```

1 PREFIX my: <http://example.org/ApplicationSchema#>
```

```

2 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
3 PREFIX geof: <http://www.opengis.net/def/function/geosparql
4 />
5 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6 SELECT ?f
7 WHERE {
8 ?f geo:asWKT ?fWKT .
9 FILTER (geof:sfEquals(?fWKT, ''
10 MULTIPOINT ((446228.92 4108432.53)
11 ''''^geo:wktLiteral))
12 }

```

	f	♦
1	geosparql:890947	

Figura 1.38

geof:distance

Find the 3 closest features to feature my:C, where computations are based on my:hasExactGeometry.

```

1 PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
2 PREFIX my: <http://example.org/ApplicationSchema#>
3 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
4 PREFIX geof: <http://www.opengis.net/def/function/geosparql
5 />
6 SELECT ?f
7 WHERE {
8 geo:889549 geo:asWKT ?WKT .
9 ?f geo:asWKT ?fWKT .
10 FILTER (?fGeom != ?WKT)
11 }
12
13 ORDER BY ASC(geof:distance(?cWKT, ?fWKT, uom:metre))
14 LIMIT 5

```

	f	♦
1	geosparql:1184768	
2	geosparql:1202787	
3	geosparql:1202801	
4	geosparql:1202820	
5	geosparql:1202821	

Figura 1.39

```

1 PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
2 PREFIX my: <http://example.org/ApplicationSchema#>
3 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
4 PREFIX geof: <http://www.opengis.net/def/function/geosparql
   />
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX sf: <http://www.opengis.net/ont/sf#>
7
8 SELECT ?f
9 WHERE {
10 geo:889549 geo:asWKT ?WKT .
11
12 ?f rdf:type sf:Point ;
13 geo:asWKT ?fWKT .
14 FILTER (?fGeom != ?WKT)
15 }
16
17 ORDER BY ASC(geof:distance(?cWKT, ?fWKT, uom:metre))
18 LIMIT 5

```

	f	
1	geosparql:890947	
2	geosparql:890948	
3	geosparql:890949	
4	geosparql:890950	
5	geosparql:890951	

Figura 1.40

```

1 PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
2 PREFIX my: <http://example.org/ApplicationSchema#>
3 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
4 PREFIX geof: <http://www.opengis.net/def/function/geosparql
   />
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX sf: <http://www.opengis.net/ont/sf#>
7
8 SELECT ?f
9 WHERE {
10 geo:889549 geo:asWKT ?WKT .
11
12 ?f rdf:type sf:LineString ;
13 geo:asWKT ?fWKT .
14 FILTER (?fGeom != ?WKT)
15 }
16
17 ORDER BY ASC(geof:distance(?cWKT, ?fWKT, uom:metre))
18 LIMIT 5

```

	f	♦
1	geosparql:122602	
2	geosparql:172684	
3	geosparql:186391	
4	geosparql:287437	
5	geosparql:387520	

Figura 1.41

```

1 PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
2 PREFIX my: <http://example.org/ApplicationSchema#>
3 PREFIX geo: <http://www.opengis.net/ont/geosparql#>
4 PREFIX geof: <http://www.opengis.net/def/function/geosparql
      />
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX sf: <http://www.opengis.net/ont/sf#>
7
8 SELECT ?p
9 WHERE {
10 ?p rdf:type sf:Polygon ;
11 geo:asWKT ?pWKT .
12
13 ?f rdf:type sf:Point ;
14 geo:tieneCota ?pointCota ;
15 geo:asWKT ?fWKT .
16
17 FILTER (?pointCota > 700)
18
19 }
20
21 ORDER BY ASC(geof:distance(?pWKT, ?fWKT, uom:metre))
22 LIMIT 5

```

	p	♦
1	geosparql:1184768	
2	geosparql:1202787	
3	geosparql:1202801	
4	geosparql:1202820	
5	geosparql:1202821	

Figura 1.42

geo:sfContains

**1.1.5. Visualización de la Información Geográfica**

Sin embargo, con eso sólo podemos obtener los identificadores que queremos, ¿entonces como podemos visualizarlo y ubicarlos en un mapa?

La respuesta es muy simple con Leaflet.

**1.2. Evaluación de la prueba de concepto**

<https://mangomap.com/gis-mapping>

### **1.3. Resumen del capítulo**



# Bibliografía

- [1] Junta de Andalucía. *Base Cartográfica de Andalucía. Descargas.* Disponible en: <https://www.juntadeandalucia.es/institutodeestadisticaycartografia/bcadescargas/>, Acceso en: 27 de Agosto 2019.
- [2] John Fernando Baculima Cumbe & Marcelo Xavier Cajamarca Bermeo. *Diseño e implementació de un repositorio ecuatoriano de datos enlazados geoespaciales.* Disponible en: <http://dspace.ucuenca.edu.ec/handle/123456789/19876>, Acceso en: 13 de Agosto 2019.
- [3] Instituto de Estadística y Cartografía de Andalucía. *Base Cartográfica de Andalucía. Modelo de datos.* Disponible en: [https://www.juntadeandalucia.es/institutodeestadisticaycartografia/prodCartografia/bc/modelo/00\\_Modelo\\_Datos\\_Base\\_Cartografica\\_Aandalucia.pdf](https://www.juntadeandalucia.es/institutodeestadisticaycartografia/prodCartografia/bc/modelo/00_Modelo_Datos_Base_Cartografica_Aandalucia.pdf), Acceso en: 29 de Agosto 2019.
- [4] Weidong Li Chuanrong Zhang, Tian Zhao. *Geospatial Semantic Web.* Enero 2015. Springer.



