



*ugr* | Universidad  
de **Granada**

TRABAJO FIN DE MÁSTER  
MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA

## Integración de Información Geográfica en la Web Semántica

---

Ontología GEOARES

### Autora

Gema Correa Fernández

### Director

José Samos Jiménez



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA  
Y DE TELECOMUNICACIÓN

—  
Granada, 13 de septiembre de 2019









Gema Correa Fernández

## Integración de Información Geográfica en la Web Semántica

---

### Ontología GEOARES

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: Ontología GEOARES**

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 y tecnologías que ofrece la Web Semántica mediante el desarrollo de una prueba de concepto con Información Geográfica procedente de la provincia de Granada, a través de la creación de la ontología GEOARES. Para entender el ámbito de aplicación, se contextualizan y desarrollan los conceptos necesarios en el ámbito de los Sistemas de Información Geográfica y Web Semántica, esta última área ofrece mecanismos muy útiles para la representación de estándares e incorporación de información geoespacial. Gracias a la capa semántica que ofrecen los Sistemas de Información Geográfica es posible agregar conocimiento de dominio sobre la estructura de los datos de la Web Semántica y consultar dicha información haciendo uso de los estándares definidos. Por el enfoque que se realiza, este trabajo puede resultar de utilidad para aquellas personas que quieren seguir aprendiendo sobre los Sistemas de Información Geográfica en un ámbito distinto y quieren adentrarse en la Web Semántica.

**Palabras-clave:** Sistemas de Información Geográfica, SIG, Web actual, Web Semántica, OWL, RDF, SPARQL, GeoSPARQL, Shapefile, QGIS, R, Protegé, GraphBD.



# **Integration of Geographic Information in the Semantic Web: GEOARES Ontology**

Gema Correa Fernández

## **Abstract**

The purpose of this project is study the tools of the Semantic Web that can be used to represent and integrate Geographic Information. In particular, it is focused on assessing the possible tools and technologies offered by the Semantic Web by developing a proof of concept with Geographic Information from the province of Granada, through the creation of the GEOARES ontology. To understand the scope, the necessary concepts are contextualized and developed in the field of Geographic Information Systems and Semantic Web, the latter area offers very useful mechanisms for the representation of standards and incorporation of geospatial information. Thanks to the semantic layer offered by the Geographic Information Systems it is possible to add domain knowledge about the structure of the data of the Semantic Web and consult this information using the defined standards. Due to the approach carried out, this work can be useful for those who want to continue learning about Geographic Information Systems in a different area and want to enter the Semantic Web.

**Key-words:** Geographic Information Systems, GIS, Current Web, Semantic Web, OWL, RDF, SPARQL, GeoSPARQL, Shapefile, QGIS, R, Protégé, GraphBD.



---

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 13 de septiembre 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, Ontología GEOARES*, 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 13 de septiembre de 2019.

**El director:**

**José Samos Jiménez**



# Agradecimientos

A toda esa gente que ha estado acompañandome durante este etapa, familia, amigos y profesores. Gracias a vosotros he crecido tanto personal como profesionalmente. Pero sobretodo quiero agradecer a mi apoyo incondicional durante casi 17 años, quién me ha enseñado a luchar en los momentos más difíciles y a no rendirme bajo ningún concepto.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Definición del problema . . . . .	3
1.3. Motivación . . . . .	3
1.4. Premisas e hipótesis . . . . .	4
1.5. Objetivos . . . . .	5
1.5.1. Objetivo general . . . . .	5
1.5.2. Objetivos específicos . . . . .	5
1.6. Estructura de la monografía . . . . .	6
<b>2. Sistemas de Información Geográfica</b>	<b>9</b>
2.1. Sistemas de Información Geográfica . . . . .	9
2.1.1. Conceptos generales . . . . .	9
2.1.2. Representación de datos geoespaciales . . . . .	12
2.2. ¿Qué no es un SIG? . . . . .	16
2.3. Tipos de software SIG . . . . .	17
2.4. Conclusiones . . . . .	19
<b>3. Web Semántica</b>	<b>21</b>
3.1. Web Semántica . . . . .	21
3.1.1. Conceptos generales . . . . .	21
3.1.2. Concepto de Web Semántica . . . . .	26
3.2. Arquitectura de la Web Semántica . . . . .	31
3.2.1. URI/IRI . . . . .	32
3.2.2. XML . . . . .	33
3.2.3. RDF . . . . .	34
3.2.4. Ontología . . . . .	38
3.2.5. Lenguajes de consulta . . . . .	41
3.3. ¿Qué no es la Web Semántica? . . . . .	45
3.4. Aplicaciones de la Web Semántica . . . . .	45
3.5. Conclusiones . . . . .	46

<b>4. Estándares de Consulta de Información Geográfica y GeoS- PARQL</b>	<b>49</b>
4.1. Introducción . . . . .	49
4.2. Estándar OGC para la Información Geográfica . . . . .	49
4.2.1. Operadores booleanos de relaciones espaciales . . . . .	52
4.2.2. Métodos de apoyo al análisis espacial . . . . .	54
4.3. GeoSPARQL . . . . .	55
4.4. Conclusiones . . . . .	58
<b>5. Web Semántica Geoespacial</b>	<b>59</b>
5.1. Introducción a la prueba de concepto . . . . .	59
5.2. Selección y obtención de los datos geográficos . . . . .	60
5.3. Creación de la ontología GEOARES . . . . .	69
5.4. Poblar la ontología . . . . .	72
5.4.1. Edificaciones . . . . .	73
5.4.2. Curvas de Nivel . . . . .	75
5.4.3. Puntos de Cota . . . . .	76
5.5. Realización de consultas GeoSPARQL . . . . .	79
5.5.1. Activar los plugin necesarios para manejar GeoSPARQL	79
5.5.2. Consultas donde intervienen solo polígonos . . . . .	81
5.5.3. Consultas donde intervienen solo puntos . . . . .	83
5.5.4. Consultas donde intervienen solo líneas . . . . .	84
5.5.5. Consultas donde intervienen líneas y puntos . . . . .	85
5.5.6. Consultas donde intervienen polígonos y puntos . . . . .	86
5.5.7. Consultas donde intervienen polígonos y líneas . . . . .	88
5.5.8. Consultas donde intervienen polígonos, líneas y puntos	88
5.6. Visualización de la Información Geográfica . . . . .	89
5.7. Conclusiones . . . . .	92
<b>6. Conclusiones y Trabajo Futuro</b>	<b>93</b>
6.1. Conclusiones . . . . .	93
6.2. Proyectos relacionados . . . . .	95
6.3. Trabajo futuro . . . . .	97
<b>A. Planificación del TFM</b>	<b>99</b>
<b>B. Problemas y Soluciones</b>	<b>101</b>
<b>C. Instalación de GraphDB</b>	<b>105</b>
<b>Bibliografía</b>	<b>109</b>

# Índice de figuras

1.1. Ejemplo de búsqueda en Google . . . . .	2
2.1. Elementos que conforman el sistema SIG [18] . . . . .	10
2.2. Sistemas de coordenadas UTM [2] . . . . .	11
2.3. Sistemas de coordenadas WGS84 [44] . . . . .	11
2.4. Posibles capas de información en un mapa [36] . . . . .	12
2.5. Representación del modelo Ráster y Vectorial [39, 41] . . . . .	13
2.6. Clases de geometrías en WKT [23] . . . . .	14
2.7. Ejemplos de geometrías representadas en WKT [38] . . . . .	15
2.8. Ejemplo de GML [8] . . . . .	16
2.9. Entorno de trabajo de una aplicación CAD [26] . . . . .	17
3.1. Ejemplo de la estructura HTML [13] . . . . .	22
3.2. Evolución de la Web [10, 20] . . . . .	23
3.3. Ejemplo de la búsqueda “ <i>la buena lectura</i> ” en Google . . . . .	24
3.4. Entrada en Wikipedia para la WWW . . . . .	24
3.5. Principios básicos de la Web Semántica [30] . . . . .	26
3.6. Paso de la Web actual a la Web Semántica [42] . . . . .	28
3.7. Web Sintáctica vs. Web Semántica [40] . . . . .	28
3.8. Búsqueda semántica de “Apple” en DuckDuckGo . . . . .	29
3.9. Búsqueda clásica de “Apple” en Google . . . . .	30
3.10. Modelo multicapa de la Web Semántica [42] . . . . .	31
3.11. Estructura de la Web Semántica [35] . . . . .	32
3.12. Conjunto que englobas las IRI [37] . . . . .	33
3.13. Ejemplo de XML [32] . . . . .	33
3.14. Ejemplo de grafo RDF [30] . . . . .	34
3.15. Estructura de un grafo RDF [45] . . . . .	35
3.16. Descripción de un recurso musical en RDF [7] . . . . .	36
3.17. Ejemplo de triplets [31] . . . . .	36
3.18. Diagrama Linked Open Data ( <a href="https://lod-cloud.net">https://lod-cloud.net</a> ) . . . . .	37
3.19. Ejemplo de una ontología [35] . . . . .	38
3.20. Nomenclatura básica ontología OWL [35] . . . . .	40
3.21. Estructura de una consulta en SPARQL . . . . .	42

3.22. Ejemplo de consulta de SPARQL [38] . . . . .	43
3.23. Respuesta a la salida de la consulta de la figura 3.22 . . . . .	43
3.24. Tecnologías de la Web Semántica explicadas [35] . . . . .	44
3.25. Mapa conceptual de la Web Semántica [24] . . . . .	44
4.1. Jerarquía de clases [28] . . . . .	50
4.2. <i>OpenGIS Simple Features Access</i> [4] . . . . .	51
4.3. Operaciones de la clase <i>Geometry</i> [28] . . . . .	52
4.4. Ejemplos del operador <i>Within</i> [28] . . . . .	53
4.5. Ejemplos del operador <i>Touches</i> [28] . . . . .	54
4.6. Ejemplos de los métodos [34] . . . . .	54
4.7. Clases y propiedades básicas de GeoSPARQL [38] . . . . .	56
5.1. Portal del Instituto de Estadística y Cartografía [22] . . . . .	60
5.2. Diferentes archivos que componen el formato Shapefile . . . . .	61
5.3. Sistema de referencia usado en los Shapefiles . . . . .	62
5.4. Visualización del elemento <i>Edificaciones</i> en QGIS . . . . .	63
5.5. Visualización del elemento <i>Curvas de Nivel</i> en QGIS . . . . .	64
5.6. Visualización del elemento <i>Punto Cota</i> en QGIS . . . . .	65
5.7. Visualización de las tres geometrías en QGIS . . . . .	65
5.8. Seleccionar opciones para guardar el Shapefile . . . . .	67
5.9. Obtener la información de los Shapefiles en CSV . . . . .	68
5.10. Guardando el fichero CSV de los Shapefiles . . . . .	68
5.11. Inicio de Protege . . . . .	69
5.12. Estructura para una ontología geoespacial . . . . .	70
5.13. Definición de URIs para la ontología . . . . .	70
5.14. Ontología de prototipo . . . . .	71
5.15. Crear el tipo de dato geo:wktLiteral . . . . .	72
5.16. Crear las propiedades . . . . .	72
5.17. Poblar ontología para los datos de Edificaciones . . . . .	73
5.18. Creación de las subclases para TipoEdificaciones y EstadoEdificaciones de Edificaciones . . . . .	74
5.19. Reglas para poblar la ontología con Edificaciones . . . . .	75
5.20. Creación de las subclases para TipoCurvaNivel, ProcedenciaCurvaNivel y CategoriaCurvaNivel de CurvaNivel . . . . .	76
5.21. Reglas para poblar la ontología con CurvaNivel . . . . .	76
5.22. Creación de las subclases para TipoPuntoCota y ContextoPuntoCota de PuntoCota . . . . .	77
5.23. Reglas para poblar la ontología con CurvaNivel . . . . .	78
5.24. Ontología GEOARES . . . . .	78
5.25. Información sobre la ontología GEOARES . . . . .	79
5.26. Cómo importar una ontología en GraphDB . . . . .	80
5.27. Visualizar la ontología completa en GraphDB . . . . .	80
5.28. Salida de la consulta para el Ejemplo 1 . . . . .	81

5.29. Salida de la consulta para el Ejemplo 2 . . . . .	82
5.30. Información de la consulta para el Ejemplo 2 . . . . .	82
5.31. Salida de la consulta para el Ejemplo 3 . . . . .	83
5.32. Salida de la consulta para el Ejemplo 4 . . . . .	84
5.33. Salida de la consulta para el Ejemplo 5 . . . . .	84
5.34. Salida de la consulta para el Ejemplo 6 . . . . .	85
5.35. Salida de la consulta para el Ejemplo 7 . . . . .	86
5.36. Salida de la consulta para el Ejemplo 8 . . . . .	87
5.37. Salida de la consulta para el Ejemplo 9 . . . . .	87
5.38. Salida de la consulta para el Ejemplo 10 . . . . .	88
5.39. Salida de la consulta para el Ejemplo 11 . . . . .	89
5.40. Página principal del mapa Web . . . . .	89
5.41. Obtención de la información del GID para Edificaciones . . . . .	90
5.42. Obtención de la información del GID para Curvas de Nivel . . . . .	91
5.43. Obtención de la información del GID para Puntos de Cota . . . . .	91
6.1. Página de inicio de XOSM . . . . .	96
6.2. Hacer consulta en XOSM . . . . .	96
6.3. Salida de una consulta de ejemplo en XOSM . . . . .	97
B.1. Comprobación del Plugin Oracle Database instalado . . . . .	102
B.2. Error de GeoSPARQL en Protegé . . . . .	102
B.3. Comprobación del funcionamiento de GeoSPARQL en GraphDB	103
C.1. Página para la descarga de GraphDB Free . . . . .	105
C.2. Formulario para iniciar la descarga de GraphDB Free . . . . .	106
C.3. Iniciando GraphDB en el terminal . . . . .	107
C.4. Página de inicio de GraphDB en <i>Workbench</i> . . . . .	107
C.5. Página para crear un nuevo repositorio en GraphDB . . . . .	108



# Índice de cuadros

2.1. Tipos de Software SIG . . . . .	18
3.1. Desafíos que presenta la Web actual [31] . . . . .	25
3.2. Ejemplos de URIs . . . . .	32
3.3. Clases del vocabulario RDF . . . . .	35
3.4. Propiedades del vocabulario RDF . . . . .	35
3.5. Propiedades del vocabulario RDFS . . . . .	39
3.6. Clases del vocabulario RDFS . . . . .	40
4.1. Operadores Booleanos de relaciones espaciales . . . . .	53
4.2. Métodos de apoyo al análisis espacial . . . . .	55
4.3. Relaciones topológicas con sus significados . . . . .	56
4.4. Funciones para comparar y manipular geometrías . . . . .	57
5.1. Esquema del modelo de datos descargado . . . . .	61
5.2. Parte del CSV para Curvas de Nivel I . . . . .	66
5.3. Parte del CSV para Curvas de Nivel II . . . . .	66
5.4. Parte del CSV para Puntos de Cota . . . . .	67
5.5. Parte del CSV para Edificaciones . . . . .	67



# Capítulo 1

## Introducción

**RESUMEN:** Este capítulo define los límites del proyecto: problema a resolver, motivos de su elección, objetivos previstos y estructura. Teniendo como propósito convencer al lector y adentrarlo en el área definida como Web Semántica y, darle a conocer la capacidad que esta presenta para integrar información procedente de los Sistemas de Información Geográfica (SIG).

### 1.1. Introducción

El trabajo que se presenta a continuación es un estudio sobre la posibilidad de **integrar Información Geográfica en la Web Semántica mediante el uso de herramientas específicas, a través de la creación de la ontología GEOARES**. A partir de la Información Geográfica disponible en el Instituto de Estadística y Cartografía de Andalucía y centrándonos en la información geoespacial de edificaciones, curvas de nivel y puntos de cotas de Ogíjares (Granada), se realiza una prueba de concepto gracias a las posibilidades de integración que los SIG presentan en la Web Semántica.

Actualmente, la gran cantidad de datos presentes en la Web pueden suponer un problema para la búsqueda de información. Este hecho ha ocasionado que organismos tanto privados como públicos hayan optado por la utilización de la Web Semántica. De hecho, la Web Semántica surge como una extensión de la actual Web, para dar solución al problema de que la mayoría de los contenidos de la Web están diseñados para los humanos y no para que los programas de software procesen la semántica de los datos. Para que esto ocurra, es necesario hacer uso de herramientas y tecnologías específicas de la Web Semántica. Dichas tecnologías tratan de aportar información extra a los recursos Web, proporcionando contenidos con significado que permitan mejorar la interoperabilidad entre los Sistemas Informáticos e incluso

la aparición de agentes capaces de realizar procesos inteligentes de captura y tratamiento de la información. Por otro lado, los motores de búsqueda clásicos cuando manejan grandes cantidades de datos, en su mayoría no estructurados y generalmente escritos en lenguajes naturales, suelen tener que hacer frente a la imprecisión, es decir, usan similitudes de palabras o estadísticas para encontrar fuentes relevantes, como es el caso de la búsqueda a la que estamos acostumbrados con Google, Yahoo! o Bing.

En la figura 1.1 se observa como los motores de búsquedas clásicos (Google) no interpretan la información, sino que simplemente se limitan a devolver resultados a partir de las palabras clave de la búsqueda. En el ejemplo expuesto, se devuelven primero aquellos resultados que contienen en su título la frase “sistemas de información geográfica”. Esto se realiza así, debido a que la coincidencia de esa frase en el título del documento tiene más relevancia que si lo encontráramos en el contenido textual de dicha página Web.

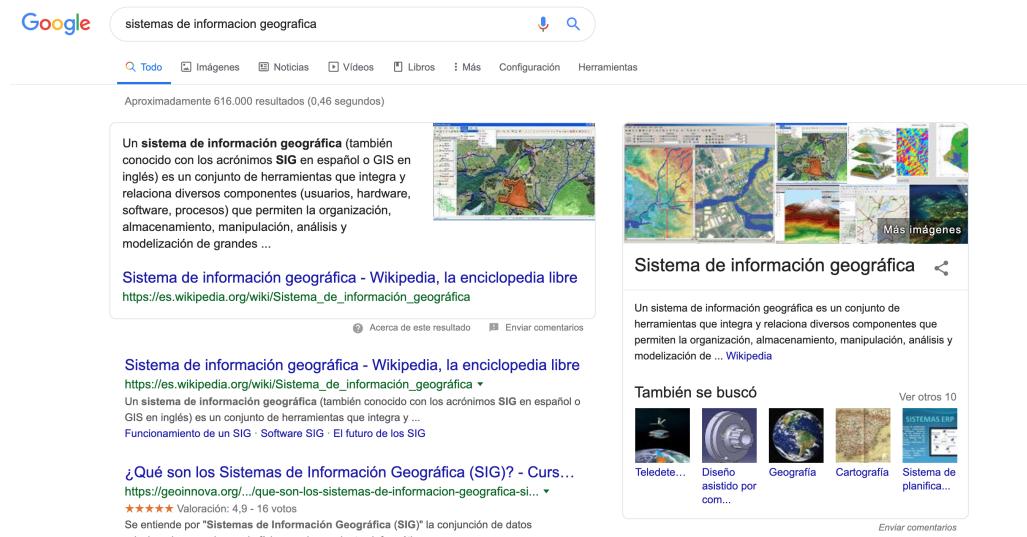


Figura 1.1: Ejemplo de búsqueda en Google

Por contraposición, en el mundo de los SIG, los datos geoespaciales requieren más esfuerzo técnico. Al ser los datos SIG muy relevantes para muchos aspectos de la vida humana, diversos organismos tanto públicos como privados construyen fuentes geoespaciales (mapas) de alta calidad para distintas necesidades: modelos digitales del terreno, mapas de ciudades o mapas meteorológicos, entre otros; además, dictan cómo deben publicarse dichos datos de manera pública. El carácter estructurado de los datos SIG hace posible describirlos con técnicas semánticas, por ejemplo, los objetos espaciales de una capa de un mapa pueden transformarse en instancias de una clase y los

atributos pueden convertirse en una propiedad de datos.

La capa semántica de los SIG brinda la ventaja de agregar conocimiento de dominio sobre la estructura de los datos (por ejemplo, se puede definir Edificio como una subclase de Sistema Urbano) y de vincular diversos sistemas SIG (por ejemplo, Edificio en un SIG es una subclase de Construcción en otro). Principalmente, por la complejidad de fusionar ambas tecnologías y sus posibilidades, pretendo realizar una prueba de concepto que integre Información Geográfica procedente de la provincia de Granada en la Web Semántica y plasmar dicha información en la creación de una ontología llamada GEOARES.

## 1.2. Definición del problema

La Web Semántica permite incorporar información muy diversa, entre ella cabe destacar la Información Geográfica. Con los conocimientos que tengo adquiridos durante el Grado en Ingeniería Informática en el ámbito de los SIG y los de Web Semántica adquiridos durante el Máster Universitario en Ingeniería Informática, pretendo hacer uso de ambas áreas para desarrollar un ejemplo práctico que agregue conocimiento geográfico mediante las herramientas de la Web Semántica.

A partir de los mecanismos para la integración de información procedente de diversas fuentes que ofrece la Web Semántica, se han definido estándares de representación de Información Geográfica mediante herramientas de la Web Semántica. En este proyecto se estudian algunas de las herramientas que se pueden utilizar para representar e integrar Información Geográfica, valorarlas y desarrollar una prueba de concepto. Se trabajará con información geoespacial, concretamente con archivos en formato *Shapefile* de la provincia de Granada, en particular de mi pueblo, Ogíjares, y nos centraremos en la información geoespacial obtenida de edificaciones, puntos de cota y curvas de nivel. En la prueba de concepto, se hará uso de las herramientas destinadas a la generación de información geoespacial propias del área SIG (*QGIS* para transformar los *Shapefile*) junto con herramientas de la Web Semántica (*Protegé*, *GraphBD*) y otras para representar la información obtenida como resultado (*R* y *Shiny*).

## 1.3. Motivación

*¿Qué motivación y razones me han llevado a comenzar un proyecto como este?* Cómo Ingeniera Técnica Informática y estudiante del Máster Universitario en Ingeniería Informática, he adquirido a lo largo del Grado y del Máster conocimientos y usos muy diversos que se le puede dar a la tecno-

logía. Después de haber realizado el Trabajo de Fin de Grado en *Análisis de Información Geográfica mediante QGIS y R centrado en el Modelo Digital del Terreno*, he querido seguir cumpliendo el objetivo que me propuse: conocer más el mundo en el que se mueven los SIG, puesto que en estos dos años me es imposible abarcar todas las áreas relacionadas con el mismo. Además de haberme matriculado en el Grado en la asignatura de *Sistemas de Información Geográficos*, destacar que durante la realización del Máster Universitario en Ingeniería Informática, he adquirido algunos conocimientos en la temática de Web Semántica a partir de la asignatura de *Desarrollo de Software basado en Componentes y Servicios* impartida por Manuel Ignacio Capel Tuñón. Es más, las ganas de adentrarme en este mundo, han hecho que durante este año haya asistido al curso de *Web Semántica* del Máster de Desarrollo de Software, impartido por diversos profesores, entre ellos el tutor de este TFM. Por esto, con la idea de pasar a un siguiente nivel mi Trabajo de Fin de Grado, surgió este Trabajo de Fin de Máster en el que intento profundizar en la Web Semántica Geoespacial, con el fin de enriquecerme tanto personal como profesionalmente. Con este TFM quiero permitir al lector iniciarse en este tema tan interesante.

La idea de este proyecto comenzó a tomar forma a principios de curso, tras mantener varias reuniones con el tutor del TFG y ver que posibilidades había de seguir estudiando y ampliando los SIG de manera más práctica y profesional (en el Apéndice A se encuentra la planificación del TFM). Entonces, partiendo de todos los conocimientos que tengo antes de la realización de este trabajo, me pregunto lo siguiente: *¿existe interoperabilidad<sup>1</sup> en los Sistemas de Información Geográfica?* Este tipo de preguntas ha llevado a cabo el inicio del desarrollo de la prueba de concepto, en donde, gracias a la capa semántica que ofrecen los SIG, es posible integrarlos de manera adecuada en la Web Semántica. En los sucesivos capítulos, iremos detallando cada uno de los conceptos que, a lo largo de esta introducción, se han mencionado.

## 1.4. Premisas e hipótesis

La realización de este trabajo ha partido de varios supuestos básicos: el primero parte de la posibilidad de la interoperabilidad que presentan los datos con Información Geográfica. Asumiendo, que los datos geoespaciales interoperables pueden ser utilizados por diferentes programas y aplicaciones permitiendo así el intercambio de información y conocimiento entre ellos. El segundo supuesto es un hecho y es la utilidad de integrar información estructurada de diversas fuentes en la Web Semántica gracias a la estructura

---

<sup>1</sup>La interoperabilidad es la capacidad de los Sistemas de Información y de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos.

que estos presentan mediante su representación en una ontología para así permitir que la información contenida en las páginas Web sea entendible tanto por humanos como por máquinas. Y por último, considero que este tema puede resultar de interés en el ámbito tecnológico en general, y de los Ingenieros en Informática en particular.

## 1.5. Objetivos

Este apartado recoge los objetivos iniciales marcados para la realización del TFM, especificando los propósitos que se esperan conseguir del mismo. A continuación, se detallan tanto los objetivos generales como específicos.

### 1.5.1. Objetivo general

El objetivo de este proyecto es profundizar **en las herramientas de la Web Semántica que se pueden utilizar para representar e integrar Información Geográfica, valorarlas y desarrollar una prueba de concepto con la creación de la ontología GEOARES**. Para ello, se hará uso de las distintas posibilidades que existen actualmente en la Web Semántica Geoespacial. Así que, mediante este trabajo se adquirirá experiencia en la Web Semántica y en el uso de sus herramientas para combinar información procedente de los SIG.

### 1.5.2. Objetivos específicos

En el subapartado anterior, se mencionan a grandes rasgos los objetivos que vamos a lograr con dicho estudio y aplicación. A continuación, se detallan los objetivos más específicos del proyecto:

1. **Comprender la relación existente entre los Sistemas de Información Geográfica y la Web Semántica**, a priori dos áreas completamente diferentes.
2. **Hacer uso de fuentes de información geoespacial**, con las que obtener datos públicos geográficos de buena calidad.
3. **Localizar diversas herramientas de la Web Semántica** que se puedan utilizar para representar e integrar Información Geográfica.
4. Aprender a **agregar conocimiento de dominio geoespacial sobre la estructura de los datos** mediante el empleo de herramientas de la Web Semántica como Protegé.
5. Aprender a **crear ontologías y poblarlas** mediante información geográfica existente.

6. Apreciar las carencias o mejoras que supone usar **Protegé** frente a **otro tipo de herramientas**.
7. Conocer cuál es la **estructura de los datos geoespaciales y sus especificaciones para la realización de las consultas con el lenguaje GeoSPARQL**.
8. Aprender a **ubicar información geoespacial en un mapa mediante el uso de la biblioteca Leaflet**.
9. Mostrar algunos **métodos de trabajo**, estrechamente adaptados al tratamiento de Información Geográfica en la Web Semántica.
10. **Aportaciones a la comunidad o al lector**, para que el proyecto sirva como puerta de acceso al mundo de la Web Semántica y, en particular, al de la Web Semántica Geoespacial, y facilite el acceso a parte de los conocimientos actuales disponibles.
11. **Aportaciones hacia mi persona** en la puesta en práctica y adquisición de conocimiento de los anteriores puntos.

## 1.6. Estructura de la monografía

En este primer capítulo, se ha desarrollado una pequeña introducción al contexto en el que desarrolla este trabajo, detallando los motivos de la elección del mismo. A continuación, se resumen los contenidos del resto de capítulos de este documento:

- En el *capítulo 2 (Sistemas de Información Geográfica)*, se presentan los conceptos básicos relevantes sobre SIG. Se muestran los aspectos relacionados con el proyecto, necesarios para entender los datos geográficos a usar.
- En el *capítulo 3 (Web Semántica)*, se expone el concepto de Web Semántica, y las capas o tecnologías que conforman su arquitectura, necesarias para comprender su uso y aplicación en las herramientas y tecnologías usadas en la prueba de concepto del *capítulo 5*.
- En el *capítulo 4 (Estándares de Consulta de Información Geográfica y GeoSPARQL)*, se presenta el nexo de unión entre las áreas expuestas en los *capítulos 2* y *3*, conocimientos indispensables para la realización de la prueba de concepto de la Web Semántica Geoespacial en el *capítulo 5*.
- En el *capítulo 5 (Web Semántica Geoespacial)*, es la principal aportación de este TFM, en donde se lleva a cabo la prueba de concepto

para integrar Información Geográfica procedente de la Junta de Andalucía, en concreto de la provincia de Granada, en la Web Semántica a través de la creación de la ontología GEOARES, gracias a las herramientas disponibles, las cuales se estudian y valoran para permitir desarrollar dicha prueba de concepto de manera adecuada y de acuerdo con los estándares establecidos.

- En el *capítulo 6 (Conclusiones y Trabajo Futuro)*, se evalúan las propuestas realizadas, recopilando tanto lo que se ha hecho a lo largo del desarrollo de este trabajo como las conclusiones y resultados finales obtenidos de esta experiencia. Además, se comentan proyectos relacionados y posibles mejoras para el futuro que se podrían aplicar, pero que por diversos motivos caen fuera del ámbito de este trabajo.
- En el *Apéndice A (Planificación del TFM)*, se presenta la planificación realizada para el desarrollo del Trabajo de Fin de Máster.
- En el *Apéndice B (Problemas y Soluciones)*, se exponen los problemas que han ido ocurriendo durante la creación de la prueba de concepto y cómo se han ido solventando.
- En el *Apéndice C (Instalación de GraphDB)*, se explica el procedimiento realizado para la instalación del software GraphDB.



## Capítulo 2

# Sistemas de Información Geográfica

**RESUMEN:** Este capítulo presenta los SIG como una herramienta que permite manejar Información Geográfica. De esta forma, explicaremos y nos adentraremos en el mundo de los SIG, hablaremos sobre los posibles modelos y representaciones de los datos geográficos, y de algunas herramientas existentes en el mercado para el tratamiento de datos e Información Geográfica.

### 2.1. Sistemas de Información Geográfica

A continuación, vamos a introducir la definición de SIG, para ello se va a tratar de contextualizar los elementos clave de los SIG.

#### 2.1.1. Conceptos generales

Los **Sistemas de Información Geográfica (SIG)** o **GIS** (*Geographic Information System*), hacen referencia a una amplia familia de estándares, tecnologías y herramientas de software para almacenar, editar, analizar, mostrar e integrar datos espaciales. En el presente trabajo, nos vamos a centrar mayoritariamente en los aspectos de *búsqueda* e *integración* [5]. De tal forma que los SIG son sistemas complejos que integran un conjunto de elementos interrelacionados entre sí [29] (figura 2.1). Sin embargo, desde otro punto de vista, un SIG consiste en la unión de dos ciencias: la Geografía y la Informática, es decir, consiste en un Sistema de Información (SI) en donde dicha información incluye su posición en el espacio, utilizando para ello un sistema de coordenadas estandarizado resultado de una proyección cartográfica.



**Figura 2.1:** Elementos que conforman el sistema SIG [18]

Para trabajar con información espacial georreferenciada es necesario el establecimiento de un sistema de referencia con el que poder expresar la situación de un punto dado, es decir, se necesita establecer un sistema de codificación para cada una de las posiciones sobre la superficie terrestre y la asignación de estas a sus correspondientes coordenadas [29]. De entre todos los posibles sistemas de referencia que hay, nos vamos a centrar en el sistema de coordenadas *UTM*, *WGS84* y *ETRS89* los cuales son los sistemas usados en los datos geográficos escogidos para la realización de la prueba de concepto. Antes de entrar en detalle, es importante saber que cada país o continente usa un modelo diferente (el que mejor se aproxima a su situación), es por eso que debemos prestar especial atención a cuál se está usando.

- **UTM (Universal Transversal de Mercator)** es un sistema de proyección geodésica en el cual se construye geométricamente el mapa de manera que los meridianos y paralelos se transforman en una red rectangular, de tal forma que se conservan los ángulos pero se distorsionan todas las superficies sobre los objetos originales [2]. En este sistema se divide al globo terráqueo en 60 zonas septentrionales y meridionales, en donde cada zona tiene su propio meridiano central. En la figura 2.2 podemos ver cómo es la división mundial.
- **WGS84 (World Geodetic System 1984)** es un sistema de coordenadas geográficas mundial que permite localizar cualquier punto de la Tierra por medio de tres unidades dadas, es el más empleado en la actualidad. Este sistema de coordenadas agrega a *Greenwich* como el punto de inicio (primer meridiano) para la longitud ( $0^{\circ}$ ) y establece las unidades en grados. Además, el sistema GPS lo emplea como sistema geodésico de referencia [29] (figura 2.3).

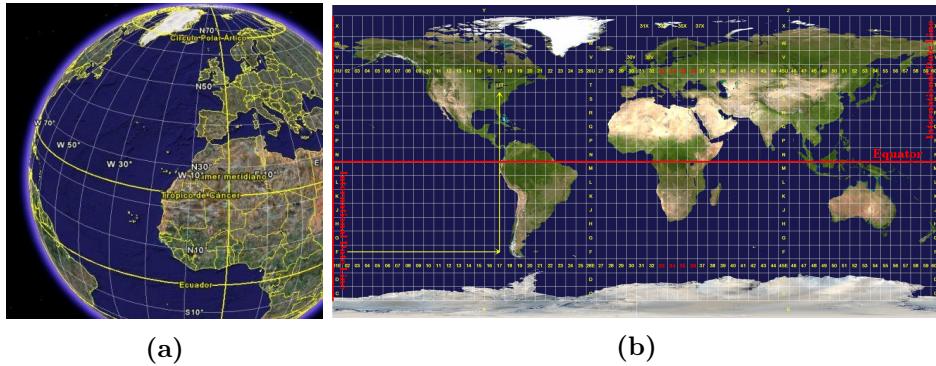


Figura 2.2: Sistemas de coordenadas UTM [2]

- **ETRS89 (*European Terrestrial Reference System 1989*)**. En España hasta 2008 se utilizaba el sistema oficial *ED50 (European Datum 1950)*, hasta que apareció el *ETRS89* y sustituyó al *ED50* [34]. *ETRS89* es un sistema de referencia geocéntrico, muy cercano al *WGS84* del GPS para ser usado no sólo en Topografía, sino en dispositivos de navegación por toda Europa. Además, está regulado por el real decreto 1071/2007 como sistema de referencia geodésico para España [21].

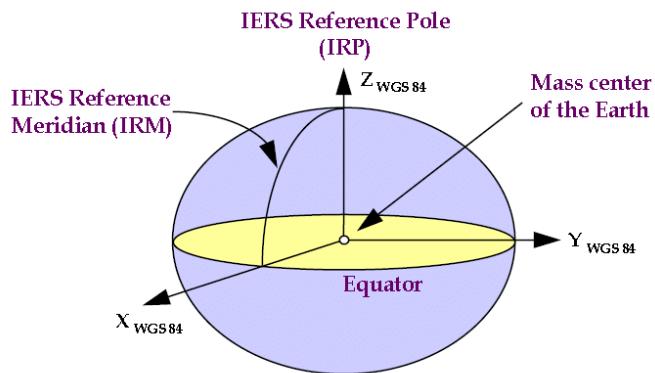
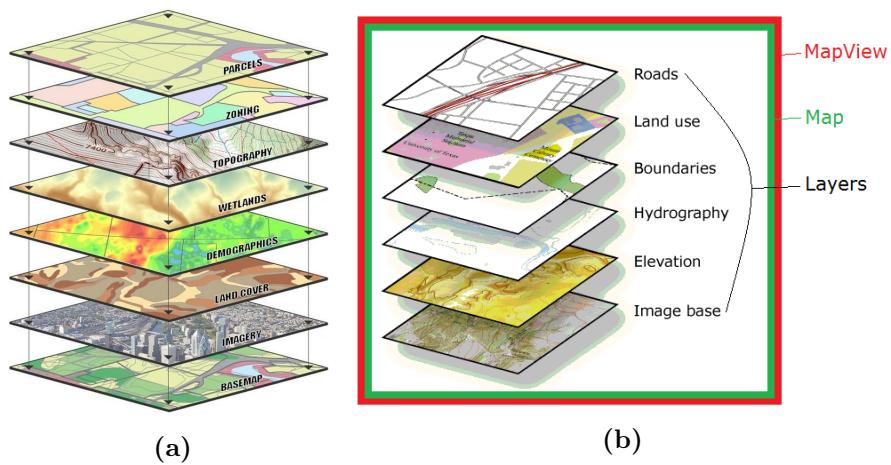


Figura 2.3: Sistemas de coordenadas WGS84 [44]

Tradicionalmente, la salida principal de los sistemas SIG es en formato de mapa<sup>1</sup>, lo que permite contener diferentes elementos relevantes como una leyenda (que explica lo que significan todos los símbolos utilizados), una escala (para mostrar la escala del mapa) o elementos interactivos (herramientas

<sup>1</sup>Un mapa “es un conjunto de capas con información y datos, en donde los datos son codificados como alfanuméricos y contienen coordenadas que permiten ubicarlo” [12].

para hacer zoom, seleccionar capas o buscar), entre otros [5]. Asimismo, un mapa puede estar constituido por varias capas, representadas en un orden determinado, una encima de otra. La primera capa (representada primero) generalmente se llama capa base, el resto de capas superpuestas. Las capas pueden ser muy diversas, desde fotografías aéreas, carreteras, parcelas de tierra o ríos hasta nombres de ciudades o de calles. En la figura 2.4 podemos ver dos ejemplos de las distintas capas que puede tener un mapa [29].



**Figura 2.4:** Posibles capas de información en un mapa [36]

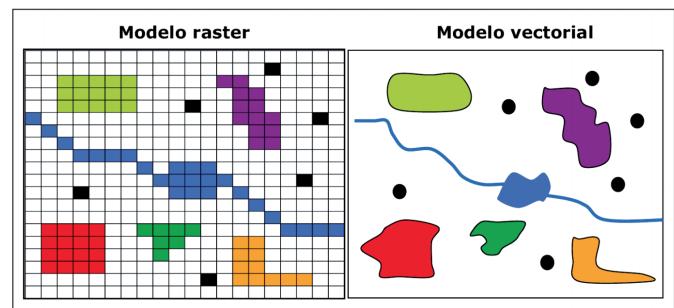
En la figura 2.4 se aprecia como cada capa aporta información relevante para la construcción del mapa, sin embargo es necesario conocer su representación. Entonces, gracias a esa información y a la Web Semántica, veremos cómo es posible transformar la capa de un mapa a instancias de una clase.

### 2.1.2. Representación de datos geoespaciales

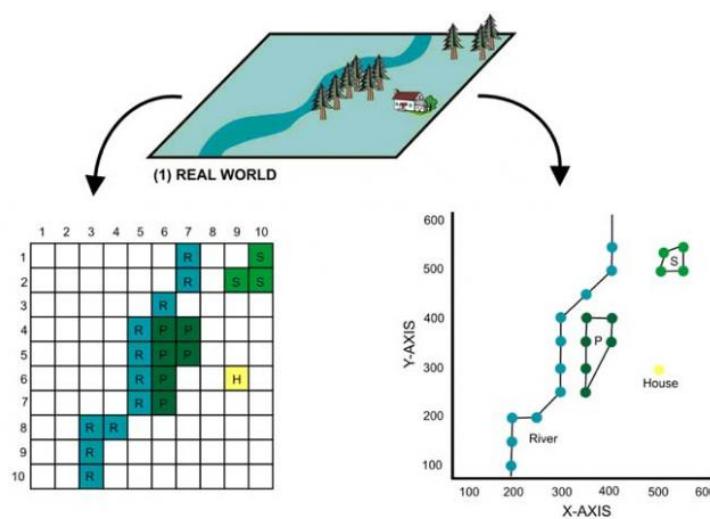
Otra consideración importante de los mapas, es su representación. El mapa se puede representar con el modelo de datos vectorial o ráster y ambos pueden presentar la misma información de manera distinta [9]:

1. **Modelo de datos Vectorial:** compuesto por formas geométricas, es decir, puntos (para modelizar ocurrencias o eventos como ciudades, aeropuertos o casos de una enfermedad), líneas (para representar fenómenos estáticos como carreteras o ríos, o datos dinámicos como rutas por carreteras) o polígonos (para representar bordes como continentes, países o urbanizaciones) (figura 2.5) [34]. Los formatos de archivos para datos vectoriales no son tan abundantes como para los datos ráster, pero entre ellos destacamos: *Shapefile (shp)* y *GeoJSON*.

2. **Modelo de datos Ráster:** compuesto por celdas con información (matriz regular de celdas o píxeles) en la que cada celda contiene un valor que representa información del mundo real (figura 2.5). Los ráster son imágenes de satélite, fotografías aéreas digitales, imágenes digitales o mapas escaneados [3] y sus formatos de archivo son muy usados en el ámbito SIG: *Tagged Image File Format (tif)*, *Joint Photographic Experts Group (jpg o jpeg)* y *ArcInfo ASCII (asc)*.



(a)



(b)

Figura 2.5: Representación del modelo Ráster y Vectorial [39, 41]

Sin embargo, no es de extrañar que a veces veamos los dos modelos combinados, ya que las herramientas actuales permiten trabajar con ambos a la vez. Un ejemplo de ello es si queremos representar en un mapa con puntos

las oficinas de correos de la ciudad. Además, existe una gran flexibilidad para transformar los datos de un modelo a otro.

A parte de los modelos de datos que acabamos de explicar, existen formatos estandarizados y de uso común para datos espaciales, entre ellos destacamos *GML* y *WKT* (usado en la prueba de concepto). La codificación **WKT** (*Well Known Text*) es una sintaxis en formato ASCII estandarizada, definida por el *Open Geospatial Consortium* (OGC)<sup>2</sup> para el intercambio de información espacial en formato vectorial. WKT se puede usar para representar geometrías, sistemas de referencia de coordenadas y transformaciones entre sistemas de referencia de coordenadas. Este estándar se concentra en formas de representar y manipular características simples. Las geometrías representadas usando WKT tienen las siguientes propiedades [23]:

- Todas las geometrías están cerradas topológicamente, lo que significa que todos los puntos que comprenden el límite de la geometría pertenecen a la geometría.
- Todas las coordenadas dentro de una geometría están en el mismo sistema de referencia de coordenadas.

Además, está basado en un estándar que define cómo representar las geometrías de los vectores. En la figura 2.6 se presenta la jerarquía de clases para geometrías de características simples representadas en WKT. Este estándar se detallará en el capítulo 4, sección 4.2 Estándar OGC para la Información Geográfica.

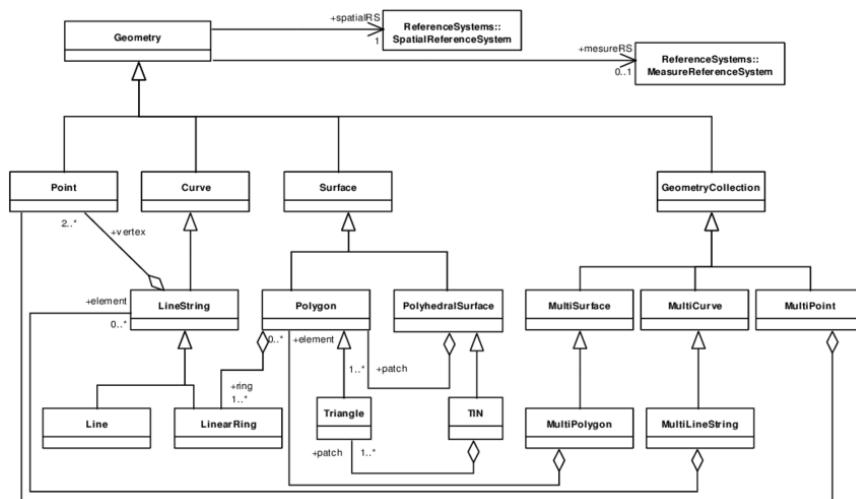


Figura 2.6: Clases de geometrías en WKT [23]

<sup>2</sup>OGC es una organización internacional para crear estándares geoespaciales abiertos que define varios estándares de servicios SIG para diferentes propósitos.

En la figura 2.7 se aprecian los objetos que es capaz de describir WKT [17]. Por ejemplo, la geometría de punto `POINT(37.96 23.71)` podría usarse para representar la ubicación de la ciudad de Atenas; donde 37.96 es la latitud de Atenas y 23.71 su longitud [23].

Geometry type	WKT representation	Geometry
Point	<code>Point(5 5)</code>	.
LineString	<code>LineString(5 5,28 7,44 14,47 35,40 40,20 30)</code>	
Polygon	<code>Polygon((5 5,28 7,44 14,47 35,40 40,20 30,5 5))</code>	
Polygon	<code>Polygon((5 5,28 7,44 14,47 35,40 40,20 30,5 5), (28 29,14.5 11,26.5 12,37.5 20,28 29))</code>	
MultiPoint	<code>MultiPoint((5 5),(28 7),(44 14), (47 35),(40 40),(20 30))</code>	. . . .
Geometry Collection	<code>GeometryCollection( Point(5 35), LineString(3 10,5 25,15 35,20 37,30 40), Polygon((5 5,28 7,44 14,47 35,40 40,20 30,5 5), (28 29,14.5 11,26.5 12,37.5 20,28 29)) )</code>	

Figura 2.7: Ejemplos de geometrías representadas en WKT [38]

Por otro lado, la codificación **GML** (*Geography Markup Language*) es el estándar de codificación basado en XML más común para la representación de datos geoespaciales (figura 2.8). GML fue desarrollado por OGC y proporciona esquemas XML para definir una variedad de conceptos que son útiles en Geografía: características geográficas, geometría, sistemas de referencia de coordenadas, topología, tiempo y unidades de medida [23]. Por eso, este elemento de Geografía codificado en XML (figura 2.8), es más ventajoso que WKT, ya que permite expresar más información (sistemas de referencia de coordenadas, unidades de medida, tiempo y más tipos de geometría) [38].

```

<?xml version="1.0" encoding="UTF-8"?>
<Cities xmlns:gml="http://www.opengis.net/gml">
  <gml:featureMember>
    <City fid="City0">
      <ID>1</ID>
      <Name>Point1</Name>
      <TheGeometry>
        <gml:Point>
          <gml:pos>-83.688, 43.019</gml:pos>
        </gml:Point>
      </TheGeometry>
    </City>
  </gml:featureMember>
</Cities>

```

Figura 2.8: Ejemplo de GML [8]

La aparición de los SIG y las posibilidades que ofrecen para operar sobre un mapa pueden servir como impulso para la creación de diversas aplicaciones, por ejemplo, en el ámbito ambiental, la planificación de tuberías subterráneas o la gestión de los servicios de emergencia [14]. Sin embargo, sigue siendo un gran problema compartir estos datos geoespaciales y utilizarlos para el desarrollo de aplicaciones SIG, no porque los datos no estén disponibles sino porque hay una gran cantidad de datos geográficos almacenados en diferentes lugares y con diferentes formatos, lo que dificulta la reutilización de datos para nuevas aplicaciones e imposibilita el intercambio de datos. De tal forma que dichas tareas son difíciles debido a la heterogeneidad de los sistemas existentes en términos de conceptos de modelado de datos, técnicas de codificación de datos y estructuras de almacenamiento. Entonces, gracias a su integración con la Web Semántica, se pueden solventar gran parte de las carencias que acabamos de comentar [5].

## 2.2. ¿Qué no es un SIG?

Hasta ahora hemos visto que los SIG trabajan con Información Geográfica, no obstante existen herramientas que hacen uso y explotan de distinta manera este tipo de información, las cuales pueden incluir tecnologías o sistemas más complejos similares a los SIG pero con un enfoque totalmente distinto. Por este motivo, es imprescindible mencionar los programas de **Diseño Asistido por Ordenador (CAD)**, estas herramientas se pueden dividir básicamente en programas de dibujo 2D y de modelado 3D. Los CAD comparten características comunes con los SIG, pero no lo son [29], ya que

aparte de ser muy usados en áreas como la arquitectura para crear planos en 3D, presentan diferencias con los SIG [26]:

- Los SIG tienen un abanico infinito de posibilidades (agricultura, hidrología, teledetección, arqueología). Sin embargo, un CAD está limitado a la producción de planos, centrándose sobre todo en el ámbito de generación de planos para ingeniería (ingeniería civil y urbanismo), instalaciones mecánicas, diseño de plantas o arquitectura.
- En un SIG siempre debemos trabajar con información georreferenciada, es decir, debemos tener esta información ubicada en el espacio y vinculada a un sistema de referencia espacial.
- Un SIG refleja la realidad y un CAD diseña algo que no existe aún, por tanto con un SIG podemos realizar operaciones que nos ayuden a tomar decisiones, lo que con un CAD no sería posible.
- Un dato SIG se almacena como un dato geográfico complejo, mientras que un dato CAD se almacena como un *dibujo* (figura 2.9).
- En un CAD lo habitual es trabajar con un único fichero que contenga distinta información, mientras que en un SIG tradicionalmente se trabaja con un fichero por cada tipo de geometría.

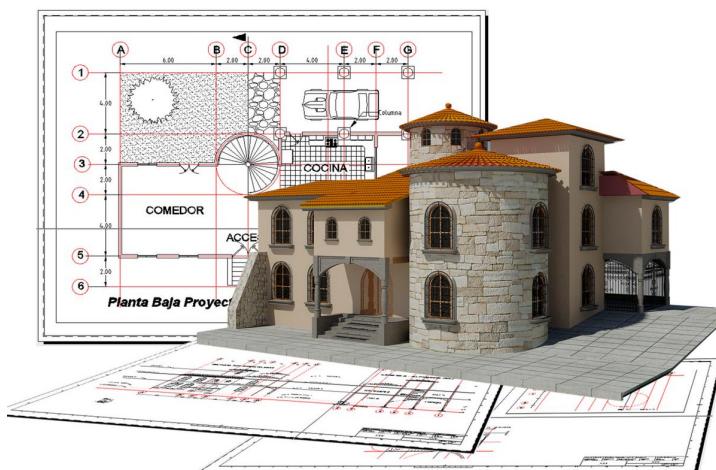


Figura 2.9: Entorno de trabajo de una aplicación CAD [26]

### 2.3. Tipos de software SIG

Dentro de los tipos de software disponibles para la manipulación de datos geoespaciales, existen sistemas de software SIG de escritorio tanto comercia-

les como *open-source*, es decir, libres y no libres. Dentro de los softwares no libres, podemos ubicar los softwares no comerciales, como los creados por empresas u organismos gubernamentales, algunos de los cuales son accesibles desde su plataforma Web. Por otro lado, los diferentes proveedores tienen sus propios diseños de software propietario, modelos de datos y estructuras de almacenamiento de bases de datos. Por lo tanto, las bases de datos geográficas basadas en estos diseños no pueden comunicarse sin la conversión de datos. Entonces, para intercambiar información y compartir recursos entre sistemas heterogéneos, se deben desarrollar herramientas de conversión para transferir datos de un formato a otro y así mejorar dicha accesibilidad [45].

Por otra parte, también podemos encontrar otro tipo de software relacionado con la posibilidad de compartir datos a través de la Web, conocidos como Web SIG (*Esri's ArcGIS Server* o *MapInfo's MapExtreme*). Estos programas ofrecen mejores herramientas para compartir datos en la Web, no obstante tienen los mismos problemas de los diseños del software propietario, los modelos de datos y las estructuras de almacenamiento de bases de datos [45]. El intercambio de datos, facilitado por los avances en las tecnologías de red, se ve obstaculizado por la incompatibilidad de la variedad de modelos y formatos de datos utilizados en diferentes sitios.

En la tabla 2.1 mostramos los software SIG más usados a nivel nacional, sin olvidarnos de que cada país u organismo puede hacer uso de otros.

Cuadro 2.1: Tipos de Software SIG

Software	Descripción
ESRI ArcGIS	Conjunto de productos de software no libre SIG producido y comercializado por ESRI
QGIS	Software libre de la fundación OSGeo para plataformas GNU/Linux, Unix, MacOS, Microsoft Windows y Android, que permite manejar formatos ráster y vectoriales a través de las bibliotecas GDAL y OGR
SAGA GIS	Programa libre que se utiliza para editar datos espaciales, desarrollado por el Dep. de Geografía Física de la Univ. de Göttingen (Alemania). Actualmente mantenido por una comunidad de desarrolladores
GRASS GIS	Software libre SIG bajo licencia GPL, que puede soportar información tanto ráster como vectorial y posee herramientas de procesado digital de imágenes
gvGIS	Proyecto de desarrollo de software libre para SIG, impulsado por el gobierno regional de la Comunidad Valenciana (Generalidad Valenciana) de España

## 2.4. Conclusiones

Este capítulo ha introducido los SIG como a un término amplio que hace referencia a una amplia familia de estándares, tecnologías y herramientas de software para almacenar, editar, analizar, mostrar e integrar datos espaciales. Dichos datos espaciales necesitan hacer uso de un sistema de referencia con el que poder expresar la situación de un punto dado; entre los sistemas que existen en la actualidad debemos destacar *UTM* y *WGS4*. No obstante, en España se utiliza el oficial en Europa, *ETRS89 (European Terrestrial Reference System 1989)*. Entre los SIG de código abierto, es importante subrayar que QGIS es a día hoy el software libre con más proyección. Además este software nos permitirá obtener la codificación WKT para la representación de las geometrías, debido a que este formato es el empleado en la representación de la estructura para los datos geográficos de la Web Semántica en las consultas con GeoSPARQL.



# Capítulo 3

# Web Semántica

**RESUMEN:** En este capítulo se presenta la Web Semántica como una extensión de la Web actual que nos ayuda a encontrar respuestas a preguntas en Internet haciendo uso de la información semántica contenida en la búsqueda. Los conocimientos que adquiramos a lo largo de este capítulo sirven como base para el posterior desarrollo del ejemplo práctico. De esta manera, dividiremos el capítulo en tres grandes apartados: el primero contextualiza y explica la Web Semántica, el segundo nos adentra en las capas de la arquitectura de la Web Semántica, y el tercero habla sobre aplicaciones que presenta la Web Semántica.

## 3.1. Web Semántica

A continuación, vamos a introducir la definición de Web Semántica, para ello es necesario realizar un breve estudio del arte sobre la Web actual, para así comprender su origen y contexto en el que nace.

### 3.1.1. Conceptos generales

La revolución informática, acaecida durante el siglo XX, llevó consigo la introducción de cambios severos en distintos aspectos de la sociedad. De entre todas estas transformaciones, fueron muchos los que vaticinaron un nuevo mundo, fundamentado principalmente por la aparición de la informática personal, en donde los seres humanos tendrían acceso a grandes repositorios de información [11]. Este hecho trajo consigo el desarrollo, por parte del físico del CERN<sup>1</sup> Tim Berners-Lee, de un sistema de vinculación y transferencia de documentos en red que acabaría convirtiéndose en la World Wide Web (WWW) o Red Global Mundial, actualmente conocida como la Web [5].

---

<sup>1</sup>El CERN, Organización Europea para la Investigación Nuclear, es uno de los centros de investigación científica más grandes y respetados del mundo ubicado en Suiza.

A lo largo de la literatura, muchos son los autores que han definido la Web Semántica como la Web del futuro. Sin embargo, para poder comprenderla es necesario entender bien cuál es la Web actual [42]. En términos generales, la Web actual es una red informática compuesta por multitud de documentos [31]. Estos documentos son básicamente páginas Web que hacen uso del lenguaje natural para expresar sus contenidos (texto, imágenes, audios, enlaces) y de etiquetas HTML (*HyperText Markup Language*) para su interpretación en los navegadores Web (figura 3.1).

### Estructura básica



Figura 3.1: Ejemplo de la estructura HTML [13]

Una de las principales causas del éxito de la Web se debe a la implantación como formato universal del lenguaje HTML (figura 3.1). Gracias a este lenguaje, los ordenadores pueden analizar la estructura de las páginas Web, determinar cuál es la cabecera o indicar dónde hay un enlace a otra página. Sin embargo, no tienen una manera fiable de procesar la semántica de las páginas, lo que convierte a la Web actual en una Web Sintáctica cargada de documentos HTML diseñados sólo para ser leídos por humanos [35].

Adicionalmente, la Web surgió con navegadores básicos que sólo interpretaban texto, después apareció HTML haciendo las páginas más amigables y de fácil acceso. En la figura 3.2 podemos ver que la Web ha pasado por distintas fases desde su primera generación, en donde las personas con conocimiento específico de diseño y composición de páginas Web crean documentos con contenido y definen los hipervínculos que los entrelazan para que los usuarios no expertos consuman esta información [20]; pasando por la Web 2.0 que busca que los usuarios interactúen con las páginas Web que visitan y se sientan como parte de la experiencia; hasta la evolución de la Web 2.0 conocida como Web 3.0, y en ocasiones confundida con la Web Semántica, es decir, la Web 3.0 es lo que se experimenta a diario en el dispositivo móvil al abrir la aplicación de Twitter, que te permite consultar la información de la red social sin necesidad de abrir el navegador [1].

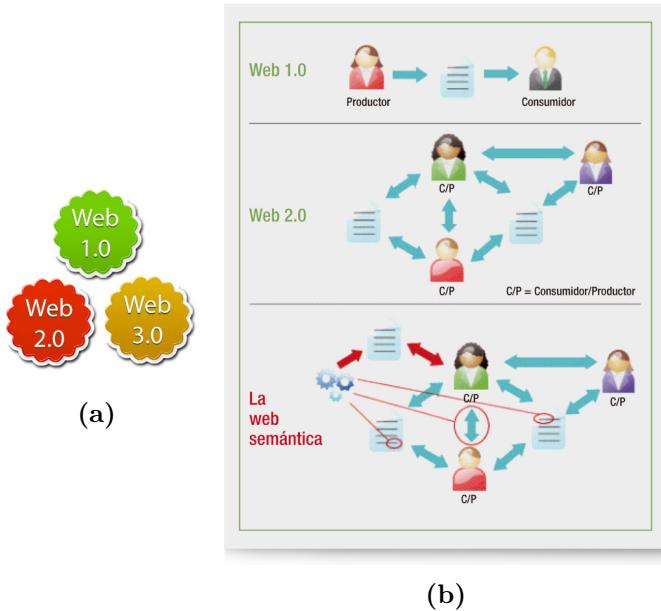


Figura 3.2: Evolución de la Web [10, 20]

La Web Sintáctica, la que usamos actualmente, hace alusión a la búsqueda de información sin interpretación del significado. Por ejemplo, si se escribe en Google la frase “*la buena lectura*”, el navegador buscará en qué páginas aparecen esas tres palabras, es decir, dicha búsqueda se hace sin tener en cuenta el significado que pueda tener la frase. Por ejemplo, en la figura 3.3 podemos apreciar cómo dicha búsqueda nos devuelve páginas en cuyo título aparecen al menos dos de las tres palabras de nuestra frase. Esto se debe a que el enfoque clásico de buscar en la Web se basa en similitudes y sinónimos generales de texto y palabras, además de en el uso de estadísticas, lo que dificulta limitar los resultados que un usuario desea. De esta manera, el usuario debe seguir un proceso iterativo para encontrar las palabras clave adecuadas para la obtención de los resultados esperados.

Otro ejemplo para la Web podría ser el mostrado en la figura 3.4, en donde podemos ver una página Web (Wikipedia) con la descripción de la WWW [31]. La mayoría de nosotros estamos familiarizados con Wikipedia, y cuando accedemos a ella no nos supone mucha dificultad distinguir su contenido textual, sus imágenes o sus enlaces en color azul para redirigirnos a otras páginas Web con otros contenidos totalmente distintos. Entonces *¿cómo podría una aplicación consumir datos de dos Webs distintas?* o *¿cómo sabría qué contenido leer dentro de cada página Web?* Ambas preguntas podrán ser contestadas una vez finalizado el presente trabajo.

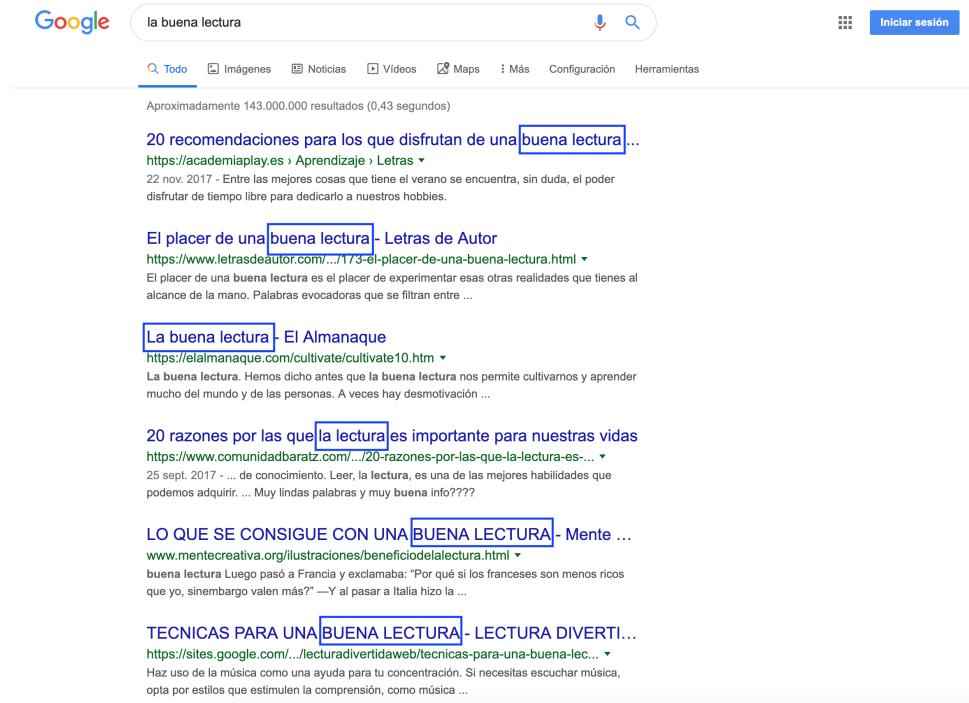


Figura 3.3: Ejemplo de la búsqueda “*la buena lectura*” en Google

Figura 3.4: Entrada en Wikipedia para la WWW

Seguidamente, una vez entendida la Web actual, es necesario comprender los desafíos que presenta (tabla 3.1). Para ello debemos empezar diciendo que

la Web actual es masiva (*¿cuántos millones de páginas Web existen en la red?*), cambiante (*¿cuántos tweets se generan en un segundo?*), heterogénea (*¿cuántos millones de dispositivos independientes generan datos en un día?*) y está hecha para humanos [31].

**Cuadro 3.1: Desafíos que presenta la Web actual [31]**

<b>Heterogénea</b>	Múltiples organizaciones generan datos de forma independiente
<b>Masiva</b>	La cantidad de información existente es enorme
<b>Cambia muy rápido</b>	Cada día son publicados y borrados enormes volúmenes de información
<b>Hecha para humanos</b>	En general, una persona puede interpretar la información de una página Web

En consecuencia, podemos concluir que la Web se ha impuesto como el instrumento de uso cotidiano más potente y rápido para el intercambio y/o difusión de información en nuestra sociedad, accesible toda ella a través de Internet. No obstante, su capacidad para satisfacer necesidades específicas es limitada. Entonces, *¿es realmente accesible esta información?* Hoy en día, existen numerosas herramientas como Google, Yahoo! o Bing que nos facilitan el acceso a esos datos [11]. Sin embargo, estas herramientas tienen dificultades a la hora de entender la información que está contenida en ellas. Asimismo, la abrumadora obtención de resultados, tanto relevantes como irrelevantes a partir de una búsqueda, denota una importante falta de precisión en la Web [31]. Este inconveniente se debe principalmente a que la Web actual carece de capacidad para expresar significados. En particular, estas herramientas miran las páginas Web como si fueran un conjunto de palabras, limitándose a recoger cadenas de caracteres indexadas en grandes bases de datos o a buscar palabras clave, y presentarlas en la pantalla del ordenador para su posterior visualización (figura 3.3) [7]. Por lo cual, estas herramientas no son capaces de entender los elementos que forman parte de ella (lugares geográficos, actores, películas) y las relaciones existentes entre estos elementos, lo que dificulta que el ordenador no sepa realmente lo que significa la información, puesto que la mayoría de los contenidos de la Web actual están diseñados para ser leídos por humanos [35].

Por esta razón surge la Web Semántica, para proporcionar estructura al contenido semántico de las páginas Web, con la finalidad de que los contenidos puedan ser consumidos por máquinas de manera más eficiente. En el siguiente subapartado entramos en detalle en el concepto de Web Semántica.

### 3.1.2. Concepto de Web Semántica

Para comprender qué es la Web Semántica ([www.semanticweb.org](http://www.semanticweb.org)), es necesario establecer los principios básicos sobre los que se asienta (figura 3.5). En el subapartado anterior se estableció el contexto en el que nace, a continuación se procede a explicar su definición.



Figura 3.5: Principios básicos de la Web Semántica [30]

La **Web Semántica** es una corriente promovida también por Tim Berners-Lee, cuyo fin es lograr que las máquinas puedan entender y, por tanto, utilizar lo que la Web contiene. De ahí que se haya tomado el término *semántica*, que desde el punto de vista lingüístico “*es la disciplina que estudia el significado de los términos*” [20]. Esta idea surge a raíz de un artículo publicado por la revista *Scientific American* en mayo de 2001, en donde Tim Berners-Lee propone una nueva forma de organizar el contenido en la Red para superar el problema de la heterogeneidad semántica y proporcionar a las computadoras contenidos Web significativos [7, 45].

En palabras de Tim Berners-Lee [42]: “*La Web Semántica es una extensión de la actual Web en la que a la información disponible se le otorga un significado bien definido que permite a los ordenadores y a las personas trabajar en cooperación. Está basada en la idea de proporcionar en la Web datos definidos y enlazados, permitiendo que aplicaciones heterogéneas lo-calicen, integren, razonen y reutilicen la información presente en la Web*”. Considerando la definición aquí aportada, la Web Semántica se manifiesta como una evolución de la actual Web, no una sustitución ya que mantiene sus características principales (descentralización, facilidad de acceso), en donde los ordenadores son capaces de interpretar los documentos; sin hacer uso de Inteligencia Artificial, ya que la semántica se encuentra en las páginas [11].

Pero, ¿cómo se traduce lo que acabamos de definir a la práctica? En la práctica, la Web Semántica es un conjunto de recomendaciones<sup>2</sup> desarrolla-

<sup>2</sup>Una recomendación es una descripción formal de una tecnología que debe ser utilizada por todos.

das por el **World Wide Web Consortium (W3C)** que es el organismo encargado de velar por la normalización en Internet y dictar los distintos estándares para la Web. El objetivo principal de este organismo reside en “*guiar a la Web hacia su máximo potencial mediante el desarrollo de protocolos y pautas comunes que promuevan su evolución y garanticen su interactividad*” [31] [7]. Dentro de la página oficial del W3C (<https://www.w3.org>), podemos encontrar otra definición para la Web Semántica: “*La Web Semántica es la representación de datos en la Web. Es un esfuerzo colaborativo liderado por W3C con la participación de un gran número de investigadores y socios industriales. Se basa en el uso de RDF, que integra una gran variedad de aplicaciones mediante el uso de XML, para la sintaxis y el uso de URLs para su identificación*” [11]. Considerando la definición aquí expuesta, los términos mencionados se escapan de los conocimientos presentados hasta ahora, en donde las tecnologías nombradas serán tratadas en sucesivos apartados, ya que la W3C recomienda el uso de RDF (*Resource Description Framework*) y OWL (*Ontology Web Language*) para la construcción de la Web Semántica.

De igual manera, nos encontramos a lo largo de la literatura más definiciones de Web Semántica, si bien todas guardan un nexo en común y es el de dotar de mayor significado a la Web actual a través de lenguajes universales que resuelvan los problemas ocasionados por una Web carente de semántica en la que a veces el acceso a la información se convierte en una tarea difícil [16]. Para ello, se desarrollan y usan lenguajes que facilitan la introducción en la Web de contenido legible por las máquinas [5]. Entonces, ¿cuáles son los requisitos esenciales para que una Web de datos pueda ser accedida y entendida tanto por ordenadores como por personas? [31]:

1. Se necesita disponer de un **lenguaje que permita especificar los recursos de la Web y las relaciones que existen entre ellos**. Con esto, lo que se pretende es desarrollar una Web más cohesionada en donde sea más fácil localizar, compartir e integrar información para sacar un mayor partido a los recursos disponibles. Para conseguir este objetivo es necesario que un sistema automático sea capaz de sacar sus propias conclusiones respecto a las búsquedas realizadas.
2. Se necesita poder consultar esos datos mediante aplicaciones computacionales. Para ello es necesario disponer de un **lenguaje para describir consultas procesables por un computador** para su entendimiento y ser capaz de sacar conclusiones a partir de los datos de manera automática. Con esto se pretende explorar la Web de forma más automática y obtener resultados más enfocados.

En la figura 3.6 se muestra cómo puede llegar al conocimiento, articulado en las ontologías, con la aplicación de técnicas de extracción de información en los documentos expresados en lenguaje natural de la Web actual.

A partir de aquí pueden crearse contenidos con la arquitectura propia de la Web Semántica y generar directamente contenidos en lenguaje natural en diferentes idiomas. Además, se recoge la necesidad de que la extracción de información de los textos debe estar vinculada directamente a las ontologías, ya que la Web Semántica necesita que la información tenga una estructura jerárquica para poder realizar las inferencias [42].

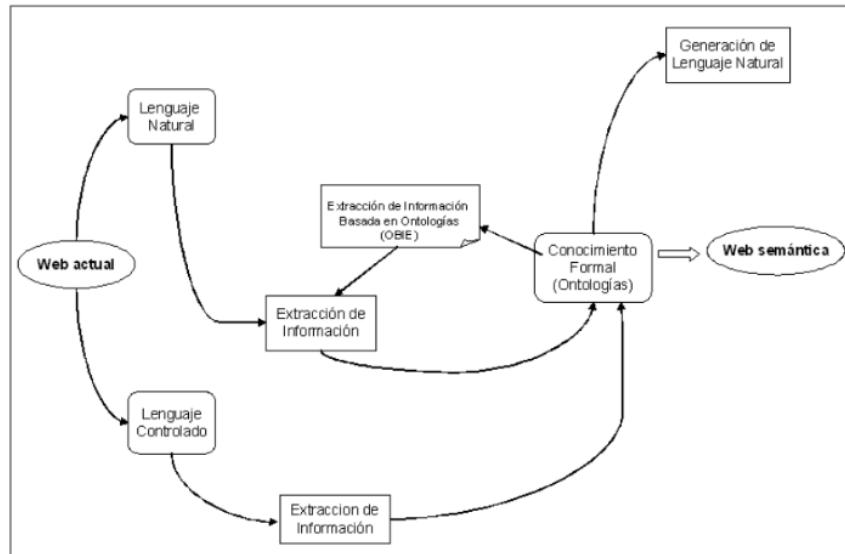


Figura 3.6: Paso de la Web actual a la Web Semántica [42]

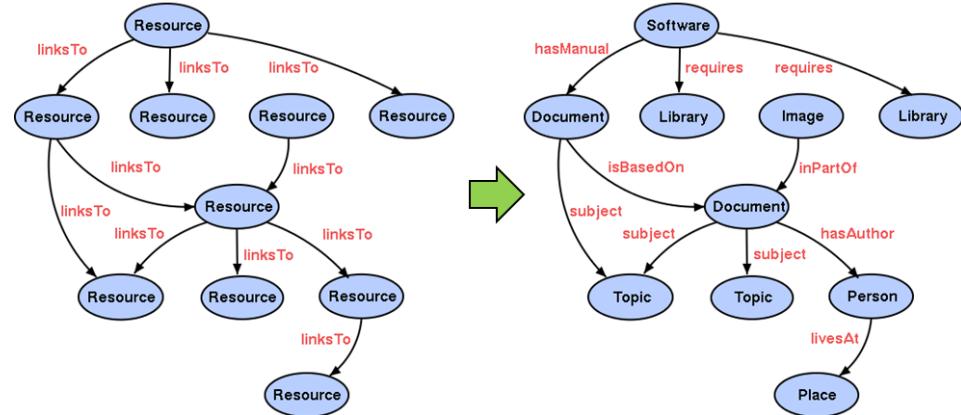


Figura 3.7: Web Sintáctica vs. Web Semántica [40]

A partir de los conceptos explicados hasta ahora, la implantación de la Web Semántica frente a la actual Web supone un cambio de paradigma, ya

que tiene que pasarse de una Web basada y creada en lenguaje natural a una Web estructurada y organizada, en donde los contenidos etiquetados semánticamente serán el elemento principal [42]. En la figura 3.7 podemos ver la diferencia que hay entre ambos tipos de Web. Se observa cómo el esquema de la izquierda, el de la actual Web, está basado en los hipervínculos que nos permiten ir saltando de una página Web a otra, mientras que en la Web Semántica (esquema de la derecha) disponemos de información más relevante.

Por otro lado, para comprender los conceptos hasta aquí tratados, vamos a ver un ejemplo de buscador semántico. En Internet existen muchos buscadores de este tipo, de entre todas las posibilidades revisadas [25] nos hemos quedado con dos: *Swoogle* (<http://swoogle.umbc.edu/2006/>) y *DuckDuckGo* ([https://duckduckgo.com/?t=h\\_](https://duckduckgo.com/?t=h_)), se ha hecho una pequeña demostración con el segundo. **DuckDuckGo** es un motor de búsqueda semántico rico en funciones que tiene innumerables razones para dejar atrás a Google. Respecto a las búsquedas, podemos diferenciar entre una búsqueda clásica, búsqueda de información y/o compras, entre otros. Si se busca un término que tiene más de un significado, nos da la oportunidad de elegir lo que estamos buscando originalmente, con sus distintos resultados o significados.

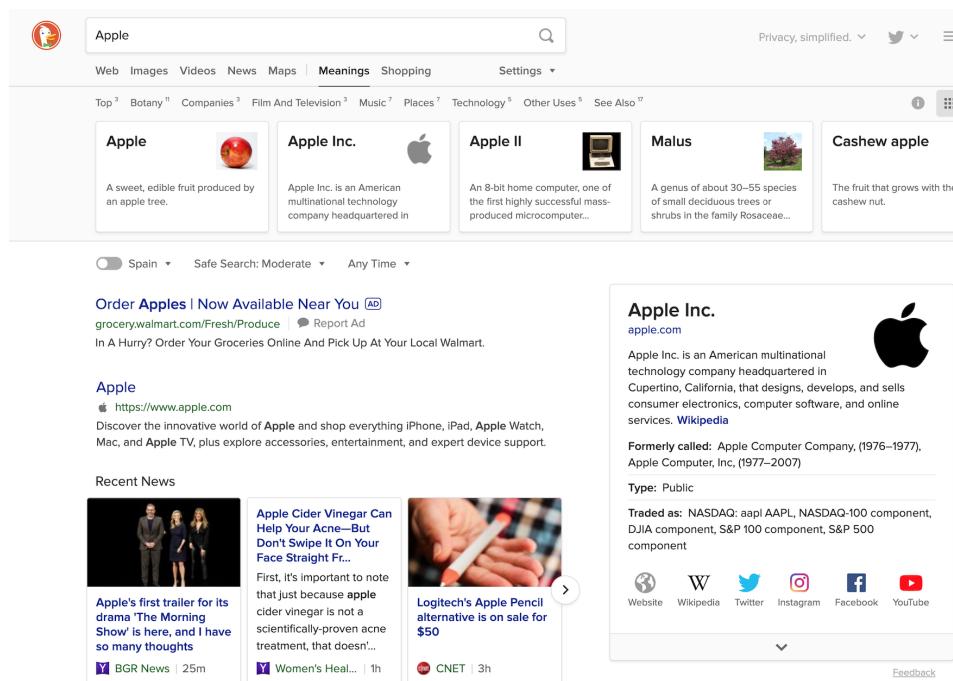


Figura 3.8: Búsqueda semántica de “Apple” en DuckDuckGo

Por ejemplo, la búsqueda del término *Apple*, en un navegador con idioma

inglés, ofrece una larga lista de posibles significados, entre los que podemos destacar principalmente la fruta y la empresa tecnológica, a parte de muchos otros. Además, los clasifica en categorías, como se puede observar en la figura 3.8. Sin embargo, la búsqueda para el término *Apple* en un buscador sintáctico como puede ser Google, nos ofrece de primeras sólo la empresa tecnológica, ya que, debido a estadísticas, se obtiene que esa es la similitud con mayor porcentaje para mostrar antes que otros resultados (figura 3.9).

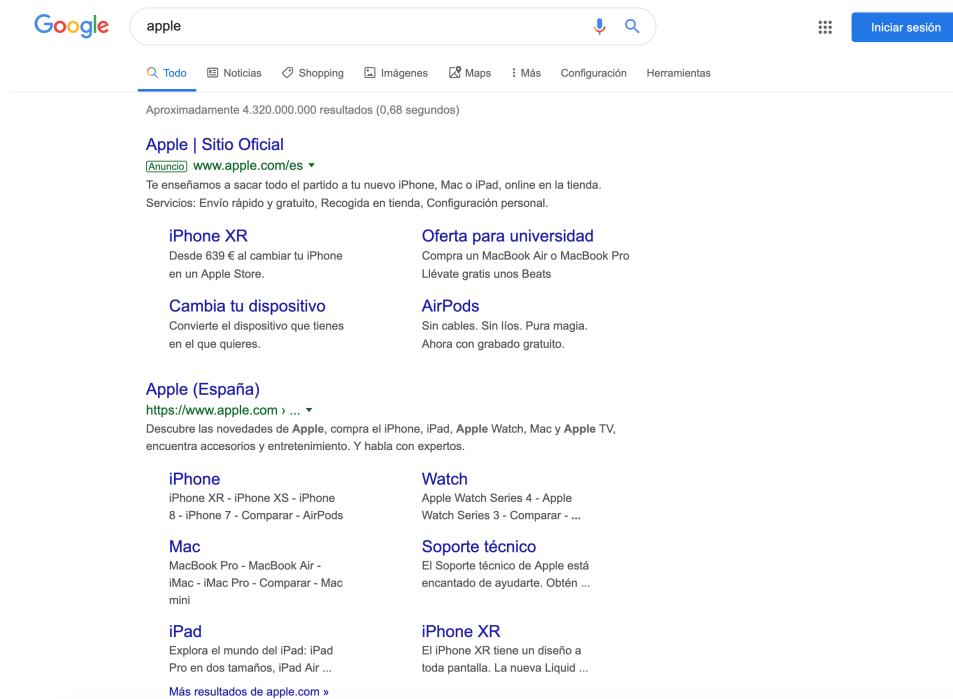


Figura 3.9: Búsqueda clásica de “Apple” en Google

Por tanto, después de haber analizado este ejemplo podemos concluir que la mayoría de los contenidos de la Web de hoy en día están diseñados para que los humanos los lean, no para que los programas de software procesen la semántica de manera significativa. Por eso nace la Web Semántica, para llevar estructuras a los contenidos semánticos de las páginas Web y permitir que los programas de software procesen y comprendan los datos en las páginas. En donde, dichas tecnologías tratan de aportar información extra a los recursos Web, proporcionando contenidos con significado que permitan mejorar la interoperabilidad entre los sistemas informáticos [30]. A través de la Web Semántica, los programas de software pueden usar colecciones estructuradas de información y conjuntos de reglas de inferencia para realizar razonamientos automatizados [45].

A continuación se explican las capas que componen su arquitectura, tecnologías usadas en la prueba de concepto que se verá en el capítulo 5.

### 3.2. Arquitectura de la Web Semántica

¿Cuáles son los estándares para la Web Semántica? En la figura 3.10 se muestra la arquitectura de la Web Semántica tal y como la definió Berners-Lee. Se trata de una estructura de capas, donde cada nivel resulta un requisito previo para el siguiente [42]:

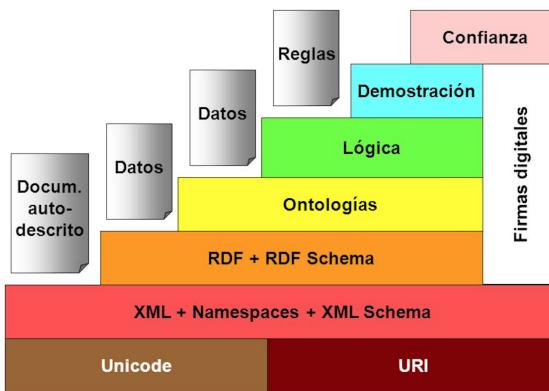


Figura 3.10: Modelo multicapa de la Web Semántica [42]

1. **Unicode, URI, XML y RDF:** los tres primeros niveles hacen referencia a la base y los estándares en los que se sustenta su desarrollo, ya que permiten convertir la Web en una infraestructura global para compartir y reutilizar datos y documentos entre diferentes usuarios.
2. **Ontologías:** capa donde reside el contenido semántico del sistema.
3. **Lógica:** en esta capa, a partir de la estructura semántica generada con las ontologías y los metadatos, se realizan las inferencias lógicas; sin embargo, actúan como freno para la Web Semántica ya que comportan una infraestructura que actualmente es difícil de realizar a gran escala.
4. **Seguridad y Confianza:** los agentes deben ser escépticos sobre la información obtenida, y contrastar minuciosamente las distintas fuentes; podrán utilizar la firma digital para verificar que la fuente es fiable.

La figura 3.10 no es la única forma de representación de las capas de la arquitectura para la Web Semántica, sino que a lo largo de los años han surgido otros tipos de representaciones similares. En el presente trabajo nos vamos a centrar en otro esquema (figura 3.11), el cual ha sido explicado en los

cursos a los que he asistido durante este año. Así que, a partir del esquema de la figura 3.11, vamos a describir las tecnologías o capas de carácter importante que se utilizan durante el desarrollo del presente trabajo: URI/IRI, XML, RDF, RDF Schema, Ontología (OWL) y SPARQL (GeoSPARQL).

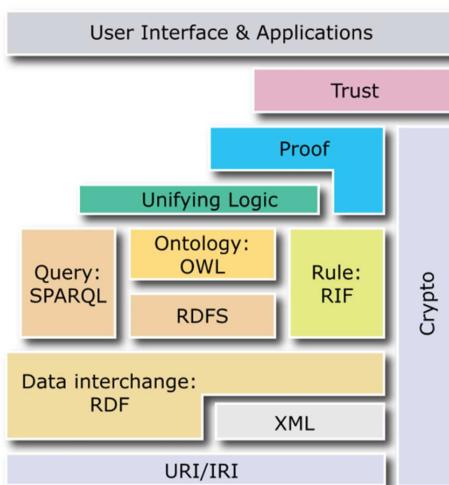


Figura 3.11: Estructura de la Web Semántica | 35

### 3.2.1. URI/IRI

URI/IRI son los identificadores de la Web (figura 3.12). Son el primer elemento necesario para el acceso a los recursos de la Web, los cuales pueden ser identificados únicamente en cualquier idioma, a través del uso de Unicode e identificadores URI [30]. El **URI** (*Uniform Resource Identifier*) es una secuencia compacta de caracteres que identifican un recurso abstracto o físico que no tiene porqué existir en la Web. En la tabla 3.2 se observan algunos ejemplos de URIs [6].

Cuadro 3.2: Ejemplos de URIs

Ejemplo	Descripción
<code>ftp://ftp.is.co.za/rfc/ rfc1808.txt</code>	Esquema FTP para servicios de protocolo de transferencia de archivos
<code>mailto:mduerst@ifi.unizh.ch</code>	Esquema MAILTO para direcciones de correo electrónico
<code>telnet://melvyl.ucop.edu/</code>	Esquema TELNET para servicios interactivos a través del protocolo TELNET

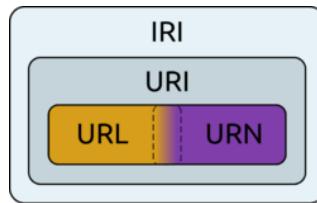


Figura 3.12: Conjunto que englobas las IRI [37]

El **IRI** (*International Resource Identifier*) se puede utilizar para encontrar diferentes tipos de recursos, como documentos, personas, objetos físicos y conceptos abstractos. Las URL que usamos como direcciones Web son una forma de IRI que sí existen en la Web. Además el IRI nace como complemento del URI, debido a que los IRI son identificadores globales y los usuarios pueden reutilizarlos para identificar lo mismo [45].

### 3.2.2. XML

**XML** (*Extensible Markup Language*) es un lenguaje de marcas, derivado de SGML<sup>3</sup>, pensado para ser utilizado en el entorno Web y para ser usado en la descripción sintáctica de los recursos, es decir, es la herramienta utilizada para estructurar y presentar los contenidos Web (figura 3.13). No obstante, ofrece una capacidad limitada para expresar semántica (no tiene etiquetas predefinidas), por lo que es necesario utilizar lenguajes que permitan imponer restricciones semánticas para descripciones completas [30].

```

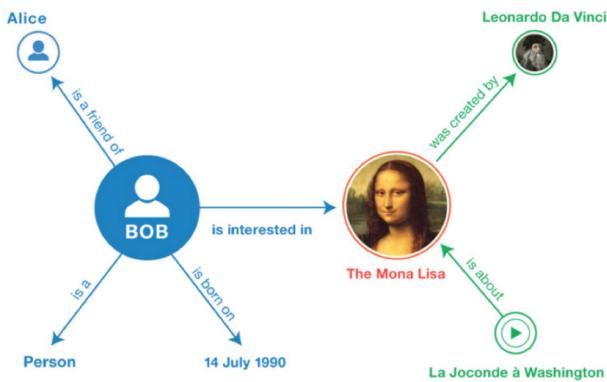
<?xml version="1.0" encoding="iso-8859-1" ?>
- <department>
  - <employee>
    <name>John Doe</name>
    <job>Software Analyst</job>
    <salary>2000</salary>
  </employee>
  - <employee>
    <name>Jane Fletcher</name>
    <job>Designer</job>
    <salary>2500</salary>
  </employee>
</department>
  
```

Figura 3.13: Ejemplo de XML [32]

<sup>3</sup>SGML (*Standard Generalized Markup Language*) es un lenguaje para marcar y describir documentos con independencia total del hardware y software utilizados.

### 3.2.3. RDF

RDF (*Resource Description Framework*) es una familia de especificaciones W3C originalmente diseñada como un modelo de datos de metadatos extendida de XML, pero no es estrictamente un formato XML, y tampoco se trata solo de metadatos. Es un método general para descomponer cualquier tipo de conocimiento en piezas pequeñas, con algunas reglas sobre la semántica o el significado de esas piezas [45]. Es el estándar más popular y extendido en la comunidad Web; proporciona un entorno para expresar la información de un recurso Web de tal forma que pueda ser intercambiada entre aplicaciones sin pérdida de significado, es decir, es una manera de darle la información desmenuizada al ordenador para que la entienda e identifique cada parte de la sentencia esté en el orden que esté [30].



(a) Grafo RDF

```

<Bob> <is a> <person>.
<Bob> <is a friend of> <Alice>.
<Bob> <is born on> <the 4th of July 1990>.
<Bob> <is interested in> <the Mona Lisa>.
<the Mona Lisa> <was created by> <Leonardo da Vinci>.
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>

```

(b) Ejemplo RDF Tripletas

Figura 3.14: Ejemplo de grafo RDF [30]

La estructura central del RDF es un conjunto de triples (tripletas), que se conoce como grafo RDF y que consta de tres componentes, en donde cada miembro puede ser una referencia a un símbolo (generalmente representado por un URI), que tiene un significado bien definido [7]. La figura 3.15 ilustra los tres componentes de un grafo RDF, y la figura 3.14 muestra un ejemplo.

1. **Sujeto:** es el recurso, es decir, todo aquello que puede ser descrito.
2. **Predicado:** introduce la propiedad que va a detallarse sobre el recurso.
3. **Objeto:** es el valor de dicha propiedad.



**Figura 3.15: Estructura de un grafo RDF [45]**

Básicamente RDF es un conjunto de declaraciones que definen clases y propiedades de los recursos. En la tabla 3.3 y 3.4 se muestra el vocabulario empleado por RDF [38].

**Cuadro 3.3: Clases del vocabulario RDF**

CLASE	DESCRIPCIÓN
<code>rdf:XMLLiteral</code>	La clase de los valores literales XML
<code>rdf:Property</code>	La clase de las propiedades RDF
<code>rdf:Statement</code>	La clase de las declaraciones RDF
<code>rdf:Bag</code>	La clase de los contenedores desordenados
<code>rdf:Seq</code>	La clase de los contenedores ordenados
<code>rdf:Alt</code>	La clase de los contenedores de alternativas
<code>rdf&gt;List</code>	La clase de las listas RDF

**Cuadro 3.4: Propiedades del vocabulario RDF**

PROPIEDAD	DESCRIPCIÓN
<code>rdf:type</code>	El sujeto es una instancia de una clase
<code>rdf:first</code>	El primer ítem en la lista RDF del sujeto
<code>rdf:rest</code>	El resto de la lista RDF del sujeto después del primer ítem
<code>rdf:value</code>	Propiedad idiomática usada para valores estructurados
<code>rdf:subject</code>	El sujeto de la declaración RDF del sujeto
<code>rdf:predicate</code>	El predicado de la declaración RDF del sujeto
<code>rdf:object</code>	El objeto de la declaración RDF del sujeto

En la figura 3.16 podemos ver un ejemplo de RDF a partir del vocabulario que acabamos de ver en las tablas anteriores. Otro ejemplo puede verse en la figura 3.17, que muestra un conjunto de triplets que indican la relación entre

un recurso, nombre del recurso y geometría, es decir, en este caso el sujeto es el *Recurso* que tiene la propiedad *Nombre* cuyo valor es *Quito*, además *Recurso* tiene la propiedad *Geometría* cuyo valor es *Punto* y finalmente se indica que el sujeto *Punto*, tiene la propiedad *WKT* que indica el formato de la geometría, cuyo valor es *POINT(-78.52495 -0.22985)* [31].

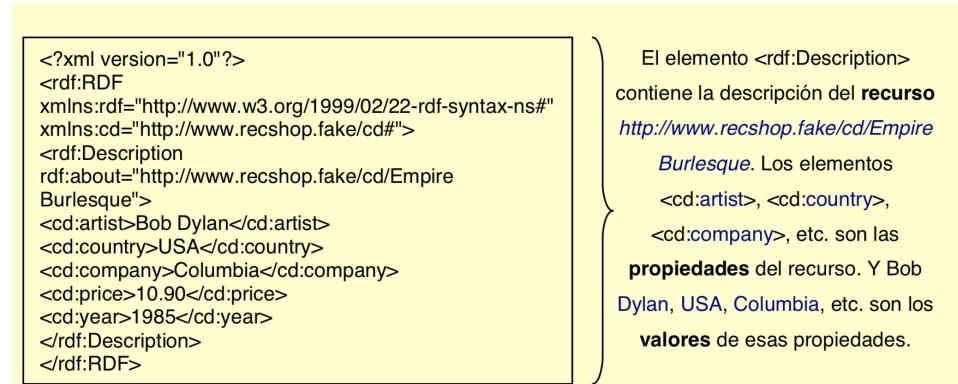


Figura 3.16: Descripción de un recurso musical en RDF [7]

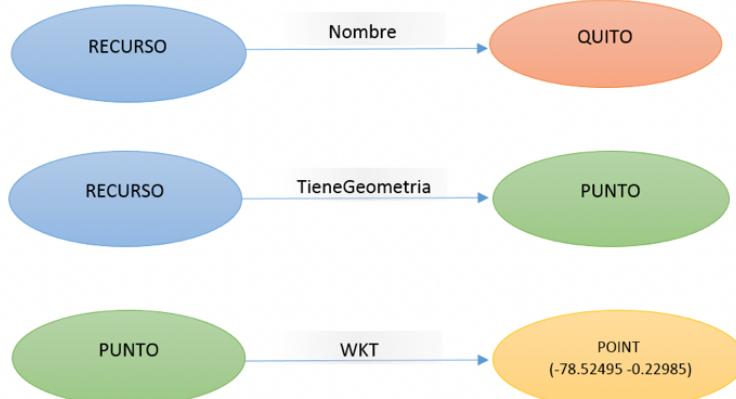


Figura 3.17: Ejemplo de triplets [31]

Por tanto, se entiende que RDF es a la semántica lo que XML es a la sintaxis, ya que XML responde a la necesidad de representar sintácticamente el modelo planteado por RDF para archivos entendibles por ordenador [7].

Adicionalmente y sin entrar mucho en detalle, es necesario nombrar en esta capa el término **Linked Data**, el cual ha permitido introducir el concepto de “Web de datos”. Linked Data es una área de investigación que estudia cómo hacer que los datos RDF estén disponibles en la Web e interconectarlos

con otros datos con el objetivo de aumentar su valor para los usuarios. En resumen, es simplemente el uso de la Web para crear vínculos con tipo entre los datos de diferentes fuentes. Un ejemplo de ello es LinkedGeoData2, en donde los datos de OpenStreetMap están disponibles como RDF y se consultan utilizando el lenguaje de consulta declarativa SPARQL [23, 30].

El diagrama de la figura 3.18 muestra los conjuntos de datos en la nube Linked Open Data así como sus relaciones en 2019. Cada nodo representa un conjunto de datos diferentes publicado en Linked Data y los arcos representan los enlaces RDF que existen entre los ítems de cada par de conjunto de datos relacionados. Como nota a la ilustración puede comentarse que el conjunto que aparece como nodo central es el perteneciente a DBpedia que es el referente a la abstracción semántica de la Wikipedia [30].

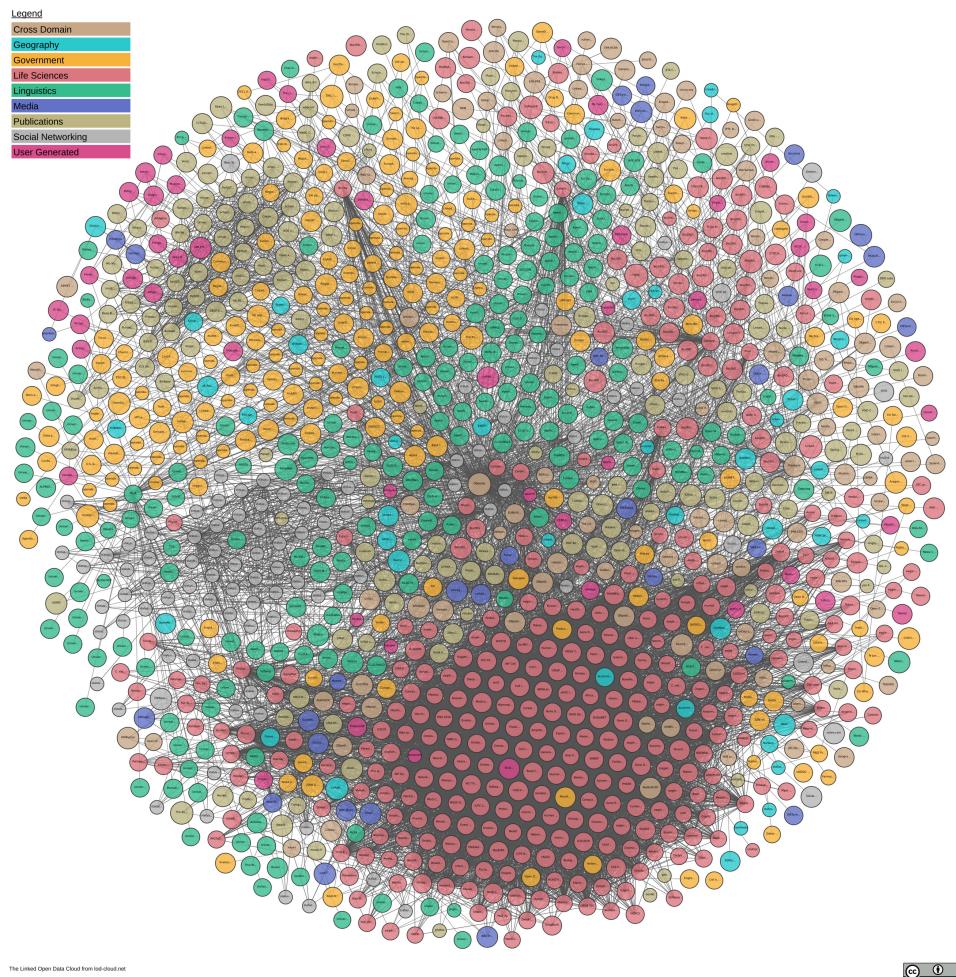


Figura 3.18: Diagrama Linked Open Data (<https://lod-cloud.net>)

### 3.2.4. Ontología

La **ontología** es un concepto filosófico adoptado por la Informática, que parte de la metafísica y que trata del ser en general y de sus propiedades trascendentales [35]. Se trata de un esquema conceptual que define los términos a utilizar para describir y representar un área de conocimiento dado, con el objetivo de facilitar la comunicación entre diferentes entidades, es decir, codifican el conocimiento de un dominio y de esta manera hacen el conocimiento reutilizable [5]. Así, definen de forma estándar y consensuada un vocabulario de conceptos y las relaciones entre ellos dentro de un área concreta del conocimiento, formando redes jerárquicas semánticas y recogiendo reglas lógicas y restricciones para hacer comprender a las máquinas los conceptos de un determinado campo. Por ejemplo, *una ontología de arte establece que todos los escultores son artistas pero no todos los artistas son escultores* [7] o como el ejemplo de la figura 3.19, en donde están estructurados los conceptos, tanto humanos como físicos, que conllevan la realización de un proyecto científico en la Universidad. Asimismo, dentro de una ontología nos podemos encontrar [30, 35]:

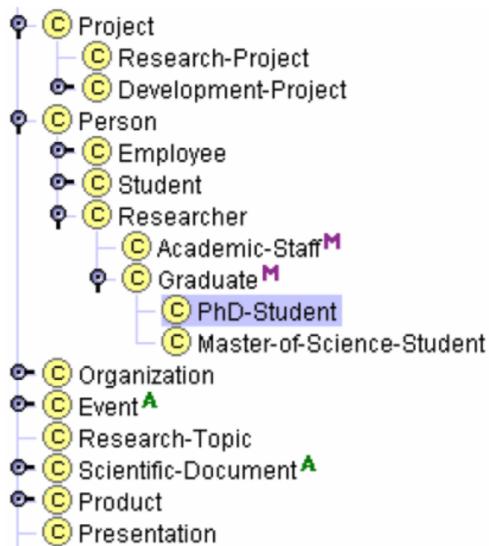


Figura 3.19: Ejemplo de una ontología [35]

- **Conceptos:** sujetos básicos que se intentan formalizar sobre cualquier tipo de clase. En una ontología de deportes, cada clase sería un deporte: tenis, fútbol, baloncesto o natación.
- **Relaciones:** enlace entre los conceptos, cómo interactúan los conceptos del dominio (subclase-de, parte-de).

- **Propiedades:** definen características o atributos de los conceptos.
- **Instancias:** objetos particulares de un concepto.
- **Funciones:** tipo de relación en la que el elemento es el resultado de la aplicación directa de una operación sobre varios conceptos de la ontología.
- **Axiomas:** teoremas que se declaran sobre las relaciones, y que deberán cumplir los elementos de la ontología. Es la clave de la inferencia de conocimiento. Por ejemplo, *si A conoce B, B conoce A*.
- **Constantes:** permiten representar valores primitivos como cadenas de caracteres o valores numéricos
- **Restricciones:** descripciones sobre qué debe cumplirse para que un axioma sea cierto.

Una vez visto el concepto de ontología y su posible aplicación a la Web Semántica, es posible pasar a explicar RDF Schema y OWL.

### RDF Schema

**Esquema RDF o RDFS (*Resource Description Framework Schema*)** es una extensión semántica que enriquece a RDF. Es un vocabulario que permite definir un primer sistema de jerarquías entre las clases de recursos, especificando las propiedades y relaciones admitidas entre ellas, es decir, permite a los usuarios definir características semánticas de los datos RDF y comprobar las restricciones semánticas [7] [45] [30]. Igual que se han definido para RDF, existe un vocabulario empleado por RDFS (tablas 3.6 y 3.5) [38].

Cuadro 3.5: Propiedades del vocabulario RDF

PROPIEDAD	DESCRIPCIÓN
rdfs:subClassOf	El sujeto es una subclase de una clase
rdfs:subPropertyOf	El sujeto es una subpropiedad de una propiedad
rdfs:domain	Un dominio de la propiedad del sujeto
rdfs:range	Un rango de la propiedad del sujeto
rdfs:label	Un nombre para el sujeto legible por personas
rdfs:comment	Una descripción del recurso sujeto
rdfs:member	Un miembro del recurso sujeto
rdfs:seeAlso	Más información sobre el recurso sujeto
rdfs:isDefinedBy	La definición del recurso sujeto

Cuadro 3.6: Clases del vocabulario RDFS

CLASE	DESCRIPCIÓN
<code>rdfs:Resource</code>	La clase de recurso, cada uno
<code>rdfs:Literal</code>	La clase del valor literal, por ejemplo, números enteros
<code>rdfs:Class</code>	La clase de las clases
<code>rdfs:Datatype</code>	La clase de los tipos de datos RDF
<code>rdfs:Container</code>	La clase de los contenedores RDF

Sin embargo, RDFS tiene limitaciones al carecer de expresividad para: información negativa (las mujeres no son hombres), cuantificadores (para que alguien sea considerado madre debe tener al menos un hijo), cardinalidad (un buen estudiante tiene que tener aprobadas más de 4 asignaturas) y no permite atributos de propiedades (transitiva, simétrica, inversa). Por todo esto y más, no se considera lo bastante completo para describir los recursos de la Web con el detalle necesario, pero se utiliza porque se puede emplear en muchos dominios y actuar de puente entre vocabularios [30].

## OWL

Ontologías	OWL	Protégé
Instancias	Individuos	Casos (Individuals)
Relaciones	Propiedades	Slots
Conceptos	Clases	Clases

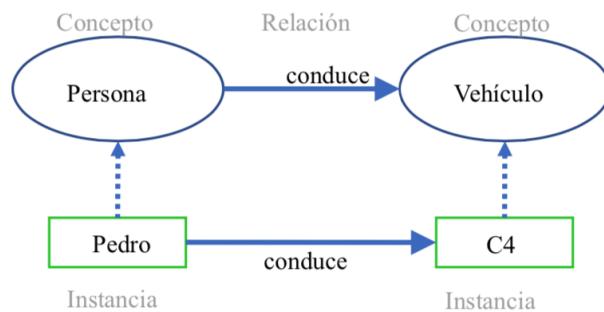


Figura 3.20: Nomenclatura básica ontología OWL [35]

**OWL (Ontology Language Web)** es un lenguaje ontológico para la Web Semántica con significado formalmente definido, sirve para definir ontologías Web estructuradas y provee de más vocabulario para la descripción de propiedades y clases, por ejemplo: relaciones entre clases, cardinalidad,

equivalencia y características de propiedades (figura 3.19). Realmente, OWL es una extensión del lenguaje RDF y emplea tripletas de RDF (figura 3.20), aunque es un lenguaje con más poder expresivo [30].

En Octubre de 2009 nació *OWL 2 Web Ontology Language* como recomendación del W3C para la edición de ontologías, extensión de la primera versión OWL 1 publicada en 2004. OWL 1 proporciona tres lenguajes, cada uno con nivel de expresividad mayor que el anterior, diseñados para ser usados por comunidades específicas de desarrolladores y usuarios [43]. En OWL 2 se definen tres nuevos perfiles y se proporcionan tres sublenguajes que ofrecen ventajas en escenarios particulares, todos más restrictivos que el lenguaje de OWL 1, llamado OWL DL, y cada uno con diferentes aspectos expresivos a cambio de diferentes beneficios computacionales y/o de implementación [30].

- **OWL Lite:** diseñado para aquellos usuarios que necesitan principalmente creación de jerarquías y restricciones simples.
- **OWL DL:** es el lenguaje habitual de ontologías diseñado para aquellos usuarios que quieren la máxima expresividad conservando completitud computacional, es decir, está basado en lógica descriptiva y proporciona la máxima capacidad de expresión para garantizar computabilidad y decidibilidad (todas las conclusiones pueden ser deducidas y todos los cálculos se realizan en un tiempo finito).
- **OWL Full:** diseñado para aquellos usuarios que quieren máxima expresividad y libertad sintáctica de RDF sin garantías computacionales, es decir, permite expresividad de segundo orden, pero sin decidibilidad (no hay garantía computacional).

Por otro lado, las ontologías deben de cumplir unas características: sintaxis bien definida, semántica específica, expresividad, fácilmente traducible entre los lenguajes ontológicos y eficiencia para realizar razonamientos [35]. No obstante, escribir ontologías en lenguajes como RDF y OWL resultan sumamente difícil y propenso a errores. Afortunadamente, existen en el mercado entornos gráficos para visualizar y construir ontologías de forma más sencilla, como **Protegé**, software desarrollado por la Universidad de Stanford que permite editar ontologías con una interfaz simple 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 [33].

### 3.2.5. Lenguajes de consulta

Por último, nos queda presentar el lenguaje de consulta semántica SPARQL, estándar ampliamente utilizado para consultar datos RDF. Además se dispo-

ne de GeoSPARQL, extensión de SPARQL que admite operaciones geoespaciales, y que será introducido en el capítulo 4, sección 4.3 GeoSPARQL.

### SPARQL

SPARQL es un lenguaje estandarizado por la W3C para el desarrollo de la Web Semántica y que permite escribir de manera precisa la información que queremos extraer, es decir, es una especificación que ofrece un lenguaje y protocolos para consultar y manipular RDF [38]. En la figura 3.21, podemos encontrar una consulta básica de selección SPARQL y comprobar que la estructura de este lenguaje se asemeja a SQL. Entre las características de este lenguaje de consulta SPARQL, resaltan:

- Formato XML, JSON, CSV y TSV para resultados en una consulta.
- Actualización de grafos RDF y protocolo para RDF.
- Descubrir información acerca de la información almacenada (*dataset*).
- Manejo de grafos mediante el protocolo HTTP.

```
Select <a sequence of variables>
Where
{
  <tripelpattern>
  Filter (<Boolean expression>)
}
```

Figura 3.21: Estructura de una consulta en SPARQL

Los resultados de una consulta están restringidos por una sección WHERE, que proporciona el patrón de grafo básico para que coincida con el gráfico de datos. El patrón gráfico consiste en un conjunto de triples, que forman un grafo, donde un elemento de un triple puede ser un URI de un recurso, un valor de datos (en el caso del objeto triple) o una variable. Una variable tiene el prefijo con el signo de interrogación y puede aparecer en la sección SELECT (la variable de salida). La sección WHERE también puede contener cláusulas FILTER, que filtran por ejemplo, cadenas y valores numéricos usando varias funciones (predicados) [38].

Un ejemplo simple de una consulta SELECT se encuentra en la figura 3.22, en donde se quieren obtener de la Dbpedia, los ingenieros que residen en ciudades con una población de más de 10000000 habitantes. En la figura 3.23 se puede observar la salida a dicha consulta [38].

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?eng ?bigcity
WHERE {
    ?eng dbo:residence ?bigcity.
    ?eng a dbo:Engineer.
    ?bigcity dbo:populationTotal ?population.
    FILTER (?population > 1000000).
}
```

Figura 3.22: Ejemplo de consulta de SPARQL [38]

eng	bigcity
<http://example.org/people/Joe>	<http://dbpedia.org/resource/Prague>

Figura 3.23: Respuesta a la salida de la consulta de la figura 3.22

Como hemos comentado anteriormente, la herramienta Protegé nos permite construir ontologías de manera más sencilla y rápida, sin embargo, también ofrece funciones para realizar consultas con SPARQL. No obstante, aunque SPARQL sea un lenguaje ampliamente utilizado para consultar ontologías RDF, carece de las construcciones necesarias para consultar datos espaciales.

Con esto hemos terminado de explicar las tecnologías de la Web Semántica. En la figura 3.24 podemos apreciar el recorrido que hemos hecho por las distintas capas que tiene la arquitectura para la Web Semántica. Como el resto de capas no resultan relevantes para nuestro proyecto, se ha obtenido por no exponerlas, para así enfocar el presente trabajo en el objetivo planteado al principio del mismo.

En suma, el objetivo de la Web Semántica es que la Web pase de ser una colección de documentos a convertirse en una base de conocimiento. En la figura 3.25 se puede ver en forma de grafo parte de los conceptos que acabamos de explicar, en donde se agrupan RDFS y OWL como ONTOLOGÍAS, y se aprecian las relaciones existentes entre todos los conceptos.

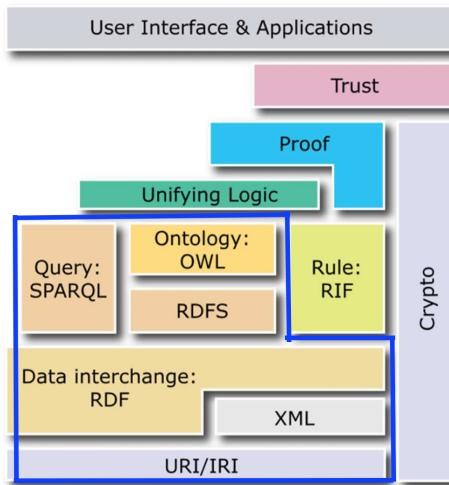


Figura 3.24: Tecnologías de la Web Semántica explicadas [35]

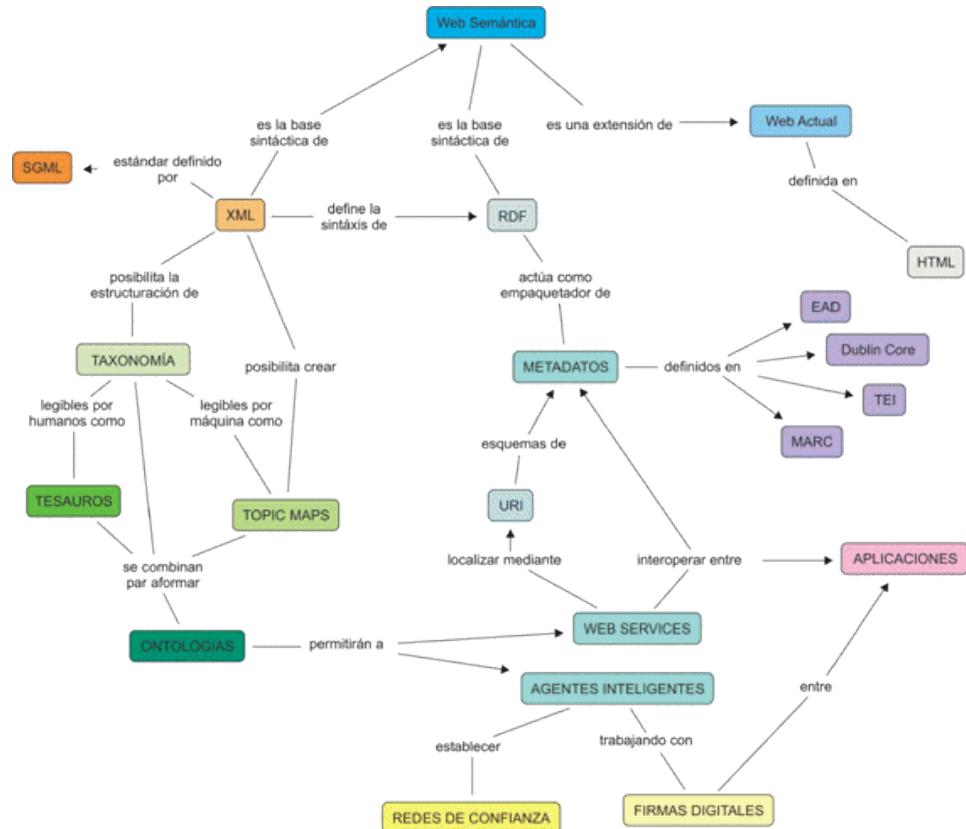


Figura 3.25: Mapa conceptual de la Web Semántica [24]

En otro orden de ideas y haciendo referencia a las capas de la Web Semántica que acabamos de comentar, *¿de qué herramientas disponemos para manejar las tecnologías que acabamos de exponer?* En el mercado, nos podemos encontrar diversas herramientas de la Web Semántica. De entre todas las posibles, nosotros nos vamos a centrar en dos:

1. **Protegé**: editor libre y marco de ontología de código abierto para construir un sistema de adquisición de conocimiento, apoyado por una fuerte comunidad académica [33].
2. **GraphDB**: familia de bases de datos RDF altamente eficientes, robustas y escalables que agiliza la carga y el uso de conjuntos de datos en la nube vinculados, así como sus propios recursos [15].

Como primera opción se consideró usar solo Protegé, pero debido a diversos problemas que comentaremos más adelante, se buscó una alternativa, en este caso, GraphDB.

### 3.3. ¿Qué no es la Web Semántica?

Una vez expuestos los contenidos que comprenden la Web Semántica, es necesario enfatizar qué no es la Web Semántica, con el objeto de evitar posibles confusiones futuras. Durante el presente trabajo, hemos hablado de que la Web Semántica es una extensión de la actual Web, por tanto diremos que una Web no es Semántica cuando sea diferente y tenga estándares incompatibles con la Web actual [16]. No todos los buscadores hacen uso la semántica, ya que la mayoría se basan en similitudes con las palabras usadas para la búsqueda sin hacer uso de la interpretación de la información.

Adicionalmente, la evolución de la Web 2.0 conocida como Web 3.0, es confundida erróneamente con la Web Semántica, pero con la que no tiene nada que ver. Puesto que la Web 3.0 es lo que se experimenta en el dispositivo móvil cuando, por ejemplo, se abre la aplicación de Twitter y se consulta la información de la red social sin necesidad de abrir el navegador [1].

### 3.4. Aplicaciones de la Web Semántica

Los conceptos que acabamos de ver han repercutido en la actualidad, ya que existen diversas aplicaciones que, de una forma u otra, se basan en tecnologías semánticas para la Web. A continuación, mostramos algunas [20]:

- **Producción científica**: en el campo de la ciencia, la publicación de resultados de la experimentación es fundamental para la comunidad

científica. Un ejemplo de ello es el portal **GoPubMed**<sup>4</sup>, que ofrece un buscador semántico de publicaciones científicas en el área de la biomedicina, a través de la ontología *Gene Ontology* que unifica y estructura la terminología sobre genes y productos génicos de un amplio número de organismos. Permite localizar textos relevantes no solo por la ocurrencia de determinadas palabras clave sino por la relación semántica existente entre conceptos biomédicos.

- **Gobiernos abiertos:** muchos gobiernos han impulsando iniciativas de *gobierno abierto*, con el fin de hacer públicos conjuntos de datos para ayudar a comprender la ciudad y desarrollar soluciones a los problemas. **London Datastore** (<https://data.london.gov.uk>) es un portal gratuito y abierto para compartir datos donde cualquiera puede acceder a datos relacionados con Londres. Por ejemplo, en el apartado *transporte* de esta iniciativa de datos gubernamentales del Reino Unido, nos permite consultar la información sobre estaciones de tren, aeropuertos y paradas de autobús entre otros, mediante consultas SPARQL.
- **Colaboración popular masiva:** la exposición, edición, compartición e interconexión de datos estructurados en la Web es muy común. **LinkedGeoData** es una iniciativa para añadir una dimensión espacial a los datos publicados en la Web Semántica y se basa en la información recogida por el proyecto *OpenStreetMap*<sup>5</sup>. Un ejemplo de ello, fue cuando a finales del 2009 muy pocas áreas de la ciudad de Port-au-Prince en Haití estaban etiquetadas. Pero justo después del terremoto de enero de 2010, cuando se hicieron públicas imágenes de satélite del país, miles de personas estudiaron estas imágenes y comenzaron a anotar en el *OpenStreetMap* información detallada sobre las zonas devastadas: carreteras bloqueadas, edificios dañados, hospitales de campaña o muelles en los que atracaban los barcos con ayuda humanitaria. Todos estos datos fueron de gran utilidad para los equipos de rescate que sobre el terreno que consultaban esta información con sus dispositivos móviles.

### 3.5. Conclusiones

Este capítulo ha introducido que el éxito de la Web se debe a la indefinida cantidad de documentos y recursos a los que podemos acceder desde cualquier parte del mundo, sin embargo, esto ha llevado a una situación de desinformación y diferencias de formatos. Para solucionar estos problemas se ha recurrido a la Web Semántica, una red de datos cuya semántica puede ser procesada por máquinas, lo que facilita que ordenadores y personas

<sup>4</sup>Actualmente no disponible (<https://www.gopubmed.org/>).

<sup>5</sup>Mapa mundial abierto al que cualquiera puede añadir datos, parecido al funcionamiento de Wikipedia.

trabajen en cooperación. Lo que hace la Web Semántica es ayudarnos en la búsqueda de un tema, es decir, utiliza métodos de representación del conocimiento mediante una red de nodos conectados entre sí, para no confundir temas parecidos. Para ello, la Web se encarga de enseñar a las máquinas a saber lo que queremos buscar, a partir de algoritmos que permiten marcar semánticamente los contenidos en cualquier documento textual. Para hacer esto real se necesitan tecnologías que conforman la arquitectura de la Web Semántica. El primer elemento necesario para el acceso a los recursos de la Web, es la posibilidad de que puedan ser identificados en cualquier idioma, con lo que se precisa el uso de Unicode e identificadores URI. Para la descripción sintáctica de los recursos se utiliza XML, pero es necesario utilizar lenguajes que permitan imponer restricciones semánticas para descripciones completas, siendo necesario hacer uso de RDF, lenguaje basado en XML que permite expresar triplets que indican el sujeto, predicado y objeto de una sentencia. Estas sentencias indican qué recursos tienen, qué propiedades y con qué valores, identificando cada objeto con una URI. Sin embargo, esta aportación de información no es suficiente, por ejemplo, en caso de que dos entornos utilicen diferentes identificadores para referirse al mismo objeto. Es por eso que se necesitan lenguajes como OWL, que tienen mayor expresividad y capacidad de razonamiento para representar los conocimientos y definir las ontologías, documentos o ficheros que definen formalmente las relaciones entre clases dentro de un dominio de la realidad: cardinalidad, igualdad, topologías de propiedades, caracterización de propiedades o clases enumeradas. Para el desarrollo de ontologías, existen herramientas como Protegé que nos facilitan su creación de manera más gráfica. Disponiendo de las ontologías para describir la información y las relaciones, se necesitan agentes inteligentes para rastrear la Web de forma automática y localizar exclusivamente, los recursos buscados, con el significado y concepto precisos con el que se interpreta el término buscado. Para ello, se dispone de dos lenguajes de consulta semántica: SPARQL, estándar ampliamente utilizado para consultar datos RDF.



## Capítulo 4

# Estándares de Consulta de Información Geográfica y GeoSPARQL

**RESUMEN:** Este capítulo define la arquitectura común que presentan las geometrías de características simples, estándar definido por OGC, y en el cual se basa el lenguaje de consulta geoespacial GeoSPARQL que vamos a usar en la prueba de concepto. Los conocimientos expuestos en este capítulo hacen referencia al nexo de unión existente entre las áreas de SIG y Web Semántica.

### 4.1. Introducción

Uno de los objetivos del presente trabajo es estudiar las herramientas de la Web Semántica que se pueden usar para representar e incorporar Información Geográfica. Para ello es necesario conocer el nexo de unión que guardan ambos ámbitos aquí expuestos para su adecuada utilización conjunta.

### 4.2. Estándar OGC para la Información Geográfica

El estándar expuesto en este apartado establece una arquitectura común y define los términos a usar dentro de dicha arquitectura para la representación geográfica y nivel lógico. La representación geográfica que vamos a usar se basa en el estándar muy parecido a *ANSI/ISO SQL* y *SQL/MM Part 3: Spatial*, definido por el organismo OGC, que contiene: jerarquía de clases y funciones estándar para datos espaciales en SQL [34].

En la figura 4.1 se presenta la jerarquía de clases para geometrías de características simples y como se puede observar es la misma que la mostrada

en la figura 2.6 para la codificación WKT, presentada en el capítulo 2, sección 2.1.2 Representación de datos geoespaciales.

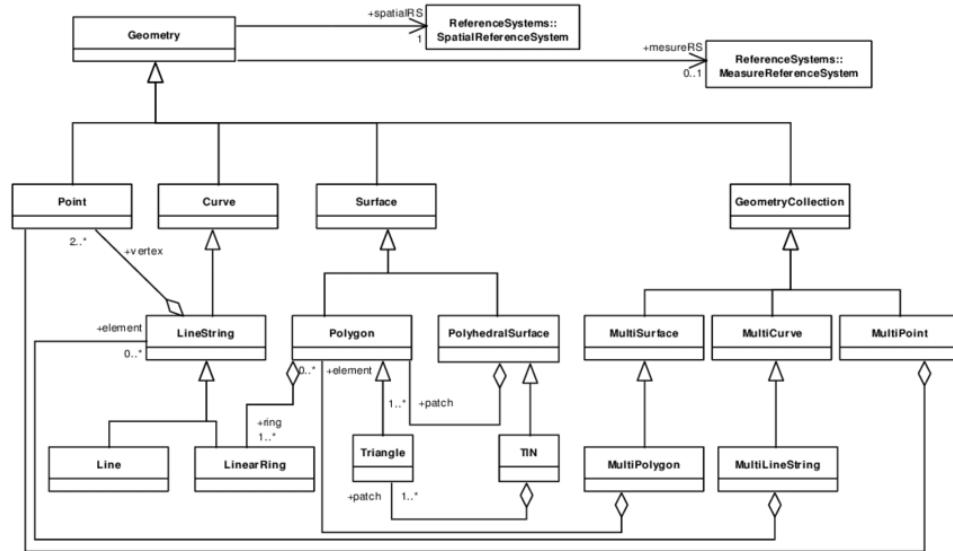


Figura 4.1: Jerarquía de clases [28]

Entonces, la figura 4.1 muestra el modelo de objetos para la geometría de entidad simple usando la notación UML. La clase de *Geometry* tiene subclases para *Point*, *Curve*, *Surface* y *GeometryCollection*, en donde cada objeto geométrico está asociado a un sistema de referencia espacial, que describe el espacio de coordenadas en el que se define el objeto geométrico [28]. Para comprender mejor las clases representadas, vamos a ver una pequeña definición de cada una de ellas [23] (figura 4.2):

- ***Point***: representa una ubicación única en el espacio de coordenadas, tiene valores de coordenadas *x* e *y*.
- ***Curve***: es una geometría unidimensional.
- ***LineString***: es un subtipo de la clase *Curve*, no tiene intersecciones propias y es cerrada si su punto de inicio es igual a su punto final.
- ***Line***: es un subtipo de *LineString* con exactamente dos puntos.
- ***LinearRing***: es un subtipo de *LineString* cerrada y simple.
- ***Surface***: es una geometría bidimensional (p.e. polígono con agujeros).
- ***Polygon***: es un subtipo de *Surface* simple plana que tiene exactamente un límite exterior y puede tener varios límites interiores que no se cruzan.

- **Triangle**: es un subtipo de *Polygon* con 3 vértices distintos no colineales y sin límite interior.
- **Polyhedral Surface**: es una colección contigua de *Polygon* que comparte segmentos en los límites comunes.
- **Triangulated Irregular Network**: un conjunto de puntos de triangulación para representar superficies en 3D.
- **Geometry Collection**: es un conjunto de geometrías distintas.
- **MultiPoint**: es un subtipo de *Geometry Collection* cuyos elementos son *Point* que no están conectados.
- **MultiCurve**: es un subtipo de *Geometry Collection* cuyos elementos son *Curve*.
- **MultiLineString**: es un subtipo de *Geometry Collection* cuyos elementos son *LineString*.
- **MultiSurface**: es un subtipo de *Geometry Collection* bidimensional cuyos elementos son *Surface*.
- **MultiPolygon**: es una colección de *MultiSurface* cuyos elementos son *Polygon*. Los límites de cada polígono pueden no cruzarse.

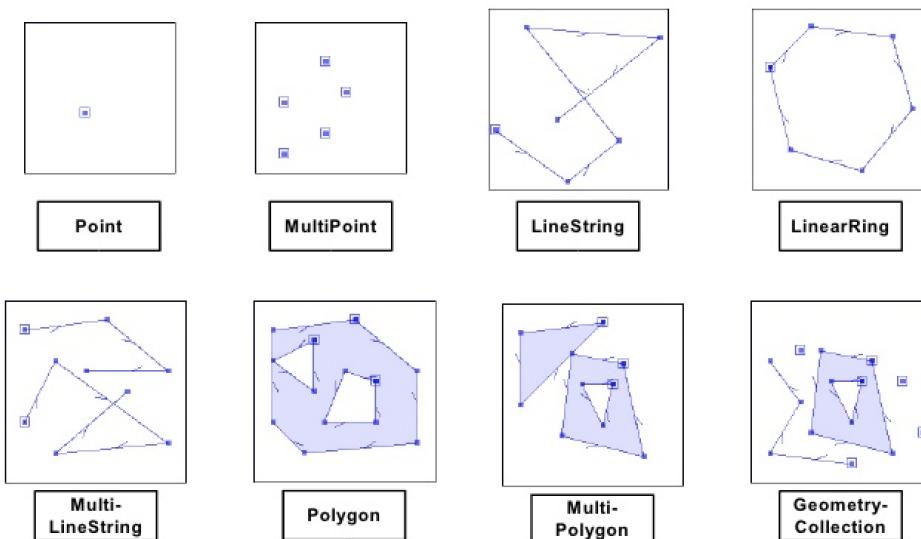


Figura 4.2: *OpenGIS Simple Features Access* [4]

Una vez que hemos definido las clases que componen el estándar, es necesario conocer las operaciones para las relaciones espaciales de la clase *Geometry* (figura 4.3).

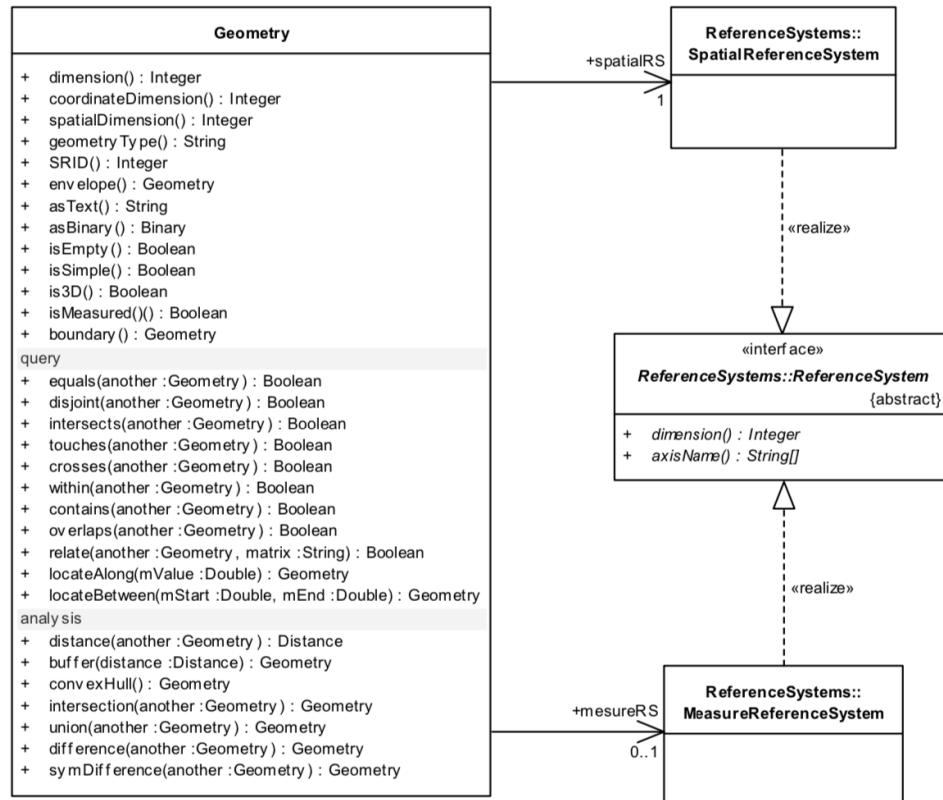


Figura 4.3: Operaciones de la clase *Geometry* [28]

#### 4.2.1. Operadores booleanos de relaciones espaciales

Comecemos por los **métodos para probar las relaciones espaciales entre objetos geométricos** de la clase *Geometry*. Para cada uno de los métodos de la tabla 4.1, el tipo de retorno es entero, pero se interpreta como booleano, VERDADERO = 1, FALSO = 0. Corresponde a la parte de *query* de la figura 4.3 [28, 23]. En la tabla 4.1 se representan los operadores booleanos de relaciones espaciales, y en las figuras 4.5 y 4.4 se presenta de forma gráfica el resultado de realizar la operación *Touches* y *Within*.

Cuadro 4.1: Operadores Booleanos de relaciones espaciales

OPERADOR	DESCRIPCIÓN
Equals	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico es “espacialmente igual” a otra <i>Geometry</i>
Disjoint	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico es “espacialmente disjunto” de otra <i>Geometry</i>
Intersects	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico “se cruza espacialmente” con otra <i>Geometry</i>
Touches	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico “toca espacialmente” otra <i>Geometry</i> (figura 4.5)
Within	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico está “espacialmente dentro” de otra <i>Geometry</i> (figura 4.4)
Contains	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico “contiene espacialmente” otra <i>Geometry</i>
Overlaps	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico “se superpone espacialmente” a otra <i>Geometry</i>
Crosses	(anotherGeometry: Geometry): Integer - Devuelve 1 (VERDADERO) si este objeto geométrico “cruza espacialmente” otra <i>Geometry</i>

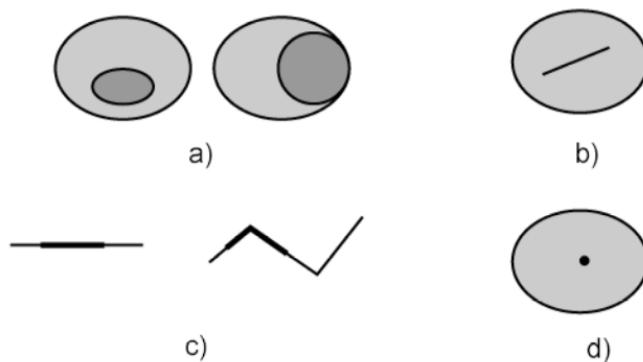
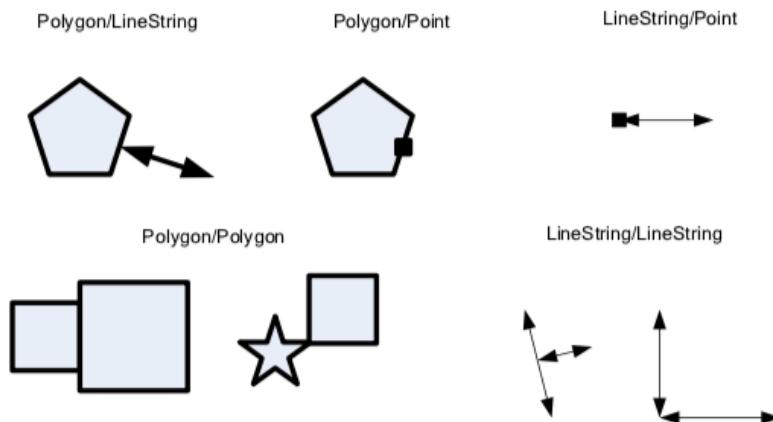


Figura 4.4: Ejemplos del operador *Within* [28]

Figura 4.5: Ejemplos del operador *Touches* [28]

#### 4.2.2. Métodos de apoyo al análisis espacial

Por otro lado los **métodos que apoyan el análisis espacial** de la clase *Geometry* son análisis geométricos que dependen de la precisión de las representaciones de coordenadas y las limitaciones de la interpolación lineal en este estándar. La precisión del resultado a un buen nivel estará limitada por estos y otros problemas relacionados (tabla 4.2). Corresponde a la parte de *analysis* de la figura 4.3 [28]. En la tabla 4.2 se representan los operadores booleanos de relaciones espaciales, y en la figura 4.6 se presenta de forma gráfica la función *Buffer* y *ConvexHull*.

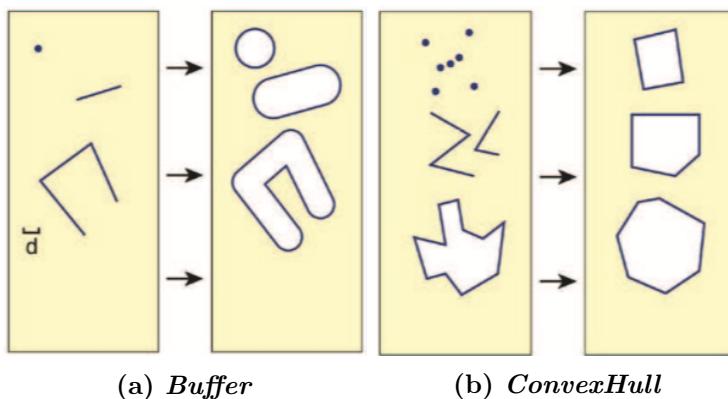


Figura 4.6: Ejemplos de los métodos [34]

Cuadro 4.2: Métodos de apoyo al análisis espacial

FUNCIÓN	DESCRIPCIÓN
Distance	(anotherGeometry: Geometry) : Double - Devuelve la distancia más corta entre dos puntos de dos objetos geométricos
Buffer	(distance: Double) : Geometry - Devuelve un objeto geométrico que representa todos los puntos cuya distancia desde este objeto geométrico es menor o igual que la distancia (figura 4.6 a)
ConvexHull	( ) : Geometry - Devuelve un objeto geométrico que representa el casco convexo de este objeto geométrico (figura 4.6 b)
Intersection	(anotherGeometry: Geometry) : Geometry - Devuelve un objeto geométrico que representa la intersección del conjunto de puntos de este objeto geométrico con otra Geometry
Union	(anotherGeometry: Geometry) : Geometry - Devuelve un objeto geométrico que representa la unión del conjunto de puntos de este objeto geométrico con otra Geometry
Difference	(anotherGeometry: Geometry) : Geometry - Devuelve un objeto geométrico que representa la diferencia de conjunto de puntos de este objeto geométrico con otra Geometry
SymDifference	(anotherGeometry: Geometry) : Geometry - Devuelve un objeto geométrico que representa la diferencia simétrica del conjunto de puntos de este objeto geométrico con otra Geometry

Con esto acabamos de mostrar una pincelada de los métodos y las funciones para manejar las subclases de las clases de *Geometry*. Asimismo, estos elementos están incluidos en el lenguaje de consulta SQL que soportan SGBD como PostGIS. A continuación, vamos a definir GeoSPARQL, lenguaje de consulta geoespacial muy relacionado con el estándar aquí expuesto.

### 4.3. GeoSPARQL

GeoSPARQL es un estándar establecido por *Open Geospatial Consortium* (OGC). Es un lenguaje de consulta basado en SPARQL para recuperar información geoespacial de conjuntos de datos RDF en la Web Semántica [45]. GeoSPARQL define gran parte de lo que se requiere para un lenguaje de consulta de este tipo al proporcionar vocabulario (clases, propiedades y

funciones) que se pueden utilizar en gráficos RDF y consultas SPARQL para representar y consultar datos geoespaciales (figura 4.7). GeoSPARQL sigue el diseño modular típico de los estándares OGC [27, 23].

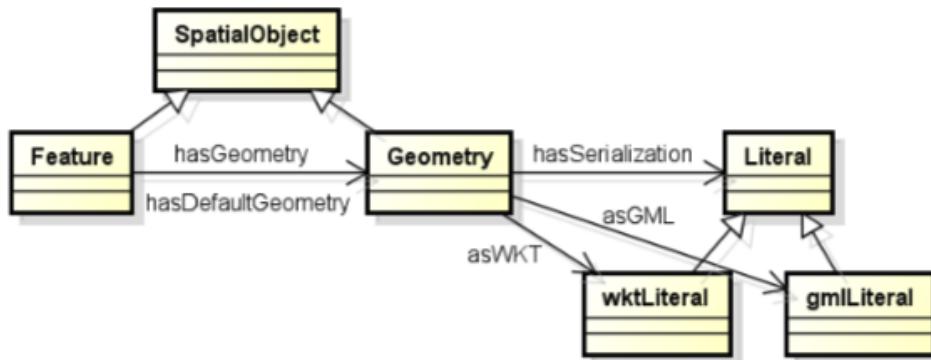


Figura 4.7: Clases y propiedades básicas de GeoSPARQL [38]

GeoSPARQL no define un vocabulario completo para representar información espacial, sino que define un conjunto central de clases, propiedades y tipos de datos que se pueden utilizar para construir patrones de consulta (figura 4.7) [27]. Es un lenguaje muy enfocado para la comunidad SIG. Por otro lado, para poder hacer uso de este tipo de consultas es necesario aplicar las especificaciones del estándar que vienen descritas en la documentación oficial<sup>1</sup>. En la tabla 4.3 podemos encontrar sus relaciones topológicas, mientras que en la tabla 4.4 podemos encontrar las funciones de GeoSPARQL que incluyen alternativas de todas las propiedades de relación topológica, aplicadas como funciones en literales de geometría [38].

Cuadro 4.3: Relaciones topológicas con sus significados

OBJETO	DESCRIPCIÓN
<b>sfEquals</b>	Espacialmente igual
<b>sfDisjoint</b>	Disjunto (no puede tocar)
<b>sfIntersects</b>	Comparten al menos un punto
<b>sfTouches</b>	Se tocan externamente
<b>sfWithin</b>	Está dentro (puede tocar el límite)
<b>sfContains</b>	El inverso de <b>sfWithin</b>
<b>sfOverlaps</b>	Tienen algunos puntos comunes, misma dimensión
<b>sfCrosses</b>	Por ejemplo, área cruza la línea

<sup>1</sup><https://www.opengeospatial.org/standards/geosparql>

Cuadro 4.4: Funciones para comparar y manipular geometrías

FUNCIÓN	DESCRIPCIÓN
<code>distance</code>	La distancia de dos literales geométricos medidos en unidades dadas
<code>buffer</code>	Literal de geometría como un literal de entrada con un buffer agregado, dado el radio y las unidades del buffer
<code>convexHull</code>	El casco convexo de un literal de geometría
<code>intersection</code>	La intersección de dos literales de geometría
<code>union</code>	Unión de dos literales de geometría
<code>difference</code>	La diferencia de dos literales de geometría
<code>symDifference</code>	Establecer la diferencia simétrica de dos literales de geometría
<code>envelope</code>	El cuadro delimitador de un literal de geometría
<code>boundary</code>	El límite de un literal de geometría
<code>getSrid</code>	URI del sistema de referencia espacial de un literal de geometría

Una decisión de diseño crucial de GeoSPARQL es usar valores de literales para codificar geometrías como una sola unidad (puntos, líneas y polígonos) e introducir dos tipos de datos RDF `geo:wktLiteral` y `geo:gmlLiteral` para estos literales. Dicha extensión está parametrizada por el estándar de serialización de OGC para codificar literales de geometría (WKT o GML). En nuestro caso, como ya hemos ido comentando vamos hacer uso del literal `geo:wktLiteral` que consiste en un URI opcional que identifica el sistema de referencia de coordenadas seguido de la codificación WKT de una geometría, por ejemplo [23]:

```
1 # Primera manera para representar el literal wktLiteral
2 "POINT(-83.38 33.95)"^^geo:wktLiteral
3
4 # Segunda manera para representar el literal wktLiteral
5 "<http://www.opengis.net/def/crs/EPSG/0/4326>
6 POINT(33.95 -83.38)"^^geo:wktLiteral
```

También se definen propiedades para representar metadatos de geometrías (por ejemplo, `geo:dimension` que captura la dimensión topológica) o para asociar geometrías con sus literales (`geo:asWKT`). Además, esta extensión define funciones para realizar operaciones no topológicas en geometrías (por ejemplo, `geof:distance` o `geof:ConvexHull`), las cuales son las mismas definidas anteriormente por el estándar [23].

En el apartado siguiente (capítulo 5), haremos uso de este lenguaje

para la obtención de la información geoespacial deseada, centrándonos en varios ejemplos prácticos, y su uso de cara al futuro.

#### **4.4. Conclusiones**

Este capítulo ha introducido el estándar usado en el lenguaje de consulta geoespacial GeoSPARQL, como nexo de unión de las áreas de los SIG y la Web Semántica. Se ha apreciado, que las funciones usadas en el GeoSPARQL son las mismas que OGC ha definido en su estándar. Con esto acabamos de comprobar como ha sido posible unir dos ámbitos que previamente parecían ser independientes el uno respecto del otro.

## Capítulo 5

# Web Semántica Geoespacial

**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. Para realizar esta integración, se desarrolla la ontología GEOARES.

### 5.1. Introducción a la prueba de concepto

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 se utilizarán los conceptos presentados en los capítulos dedicados a los sistemas SIG y a la Web Semántica, y así comprender que el principal nexo de unión entre ambas tecnologías es la estandarización de las operaciones y funciones usadas en las consultas de GeoSPARQL; tal y como hemos visto en el anterior capítulo. Conociendo todos estos conceptos es posible comenzar con el desarrollo de la ontología GEOARES, 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 y usaremos las herramientas destinadas a la generación de información geoespacial con QGIS, para permitir exportar la información de los *Shapefile* a hojas de cálculo; generación de documentos *RDF* con Protegé, para permitir llevar la información de los *Shapefile* hacia documentos en formato *RDF* de manera gráfica; consumo con GraphBD, para permitir la visualización de archivos en formato *RDF* y la realización de las consultas con el lenguaje estándar de consulta geoespacial *GeoSPARQL*; y visualización de la información geoespacial obtenida

de las consultas con 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, la versión de R es la 3.6.1, la versión de Protegé es la 5.5.0 y la versión de GraphDB Free es la 8.10.1, softwares usado en un macOS 10.14

## 5.2. 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/>.

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* [22]. A continuación, se muestran los pasos para su obtención:

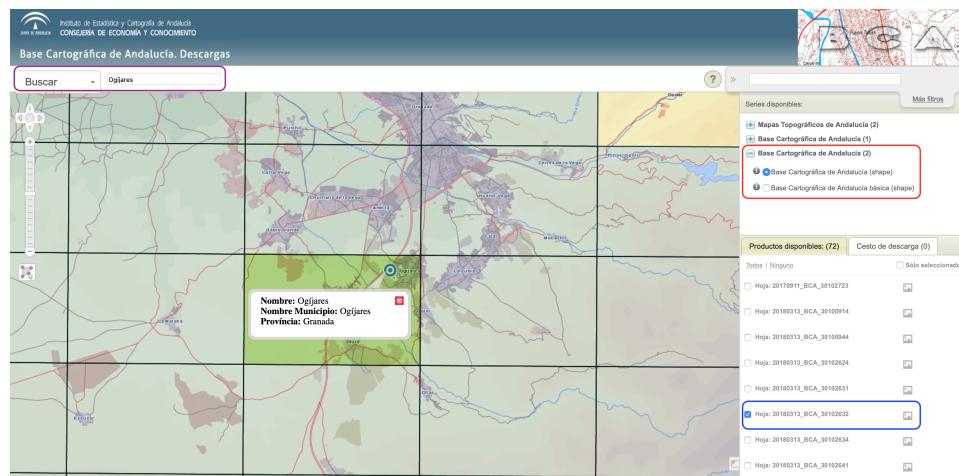


Figura 5.1: Portal del Instituto de Estadística y Cartografía [22]

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 5.1). Nos aparecerán varios cuadrantes, escogemos aquel con el que queramos trabajar; yo me he descargado el cuadrante que aparece seleccionado y remarcado en color, asociado a mi pueblo Ogíjares.
2. La carpeta descargada contiene mapas de áreas muy diversas, las cuales hacen uso de geometría de punto, de línea o de polígono. Entre las capas que se nos proporcionan nos podemos encontrar diversos modelos de datos (tabla 5.1). Si queremos saber más debemos acceder a <https://www.juntadeandalucia.es/institutodeestadisticaycartografia/prodCartografia/bc/modelo.htm> y “Nuevo modelo de datos”.

**Cuadro 5.1: Esquema del modelo de datos descargado**

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



Edificaciones.cpg



Edificaciones.dbf



Edificaciones.prj



Edificaciones.shp



Edificaciones.shx

**Figura 5.2: 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 soporta geometrías como puntos, líneas y polígonos [5]. Es originario de *Environmental Systems Research Institute (ESRI)*<sup>1</sup> y consta principalmente de un archivo principal, un archivo de índice y una tabla dBase; las demás extensiones son ficheros genéricos, como se puede ver en la figura 5.2. Estos archivos suelen requerir poco espacio de almacenamiento en disco y se pueden leer y escribir con facilidad. Los diferentes archivos que componen fundamentalmente el formato Shapefile tienen el mismo nombre cada uno con su respectiva extensión, como se aprecia en la figura 5.2:
  - **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 SIG, 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 puntos de cota de Ogíjares (Granada) [19], basados en el sistema de referencia ETRS89 (figura 5.3).

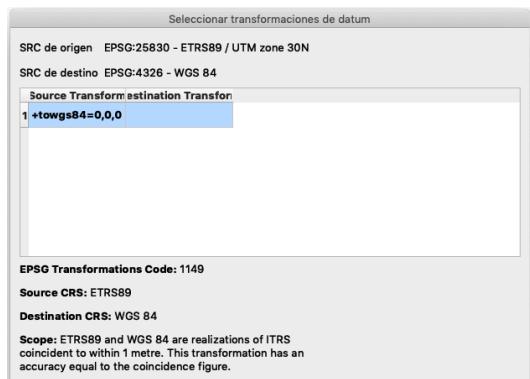


Figura 5.3: Sistema de referencia usado en los Shapefiles

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

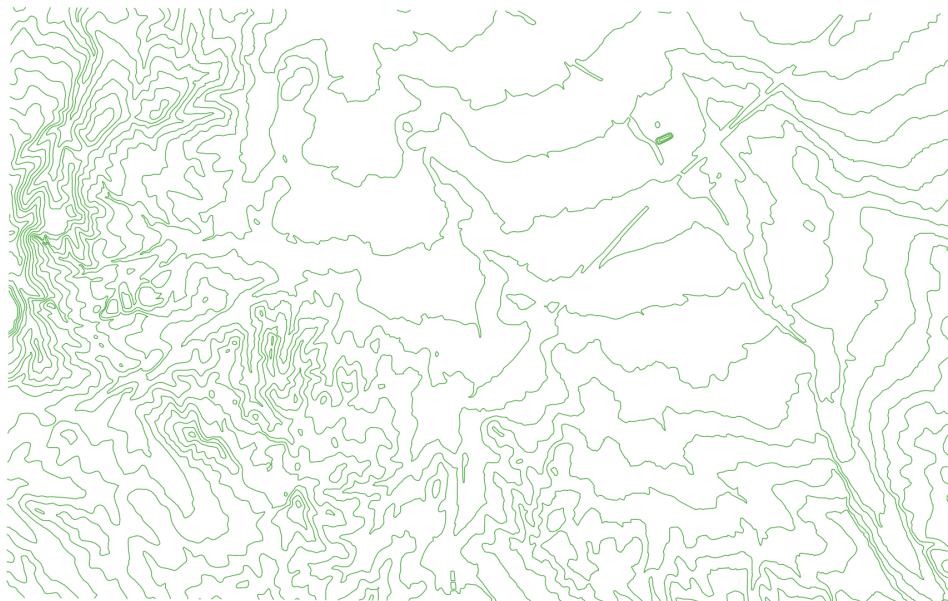
- **Edificaciones:** se agrupan en esta capa varios elementos que conforman edificaciones con geometría de polígono (figura 5.4). Dispone de la información:
  - GID.
  - ID DE LA HOJA.
  - ESTADO: estado de uso de la entidad o el tramo de la entidad.
    - En construcción (CON), en ruinas (RUI), sin clasificar (SCL) o en uso (USO).
  - TIPO: tipo de elemento según su función.
    - Caseta o cobertizo (CAS), caso genérico (CGN), chabola (CHA), chimenea (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 5.4: Visualización del elemento *Edificaciones* 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 5.5). 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 5.5:** Visualización del elemento *Curvas de Nivel* 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 5.6). 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).

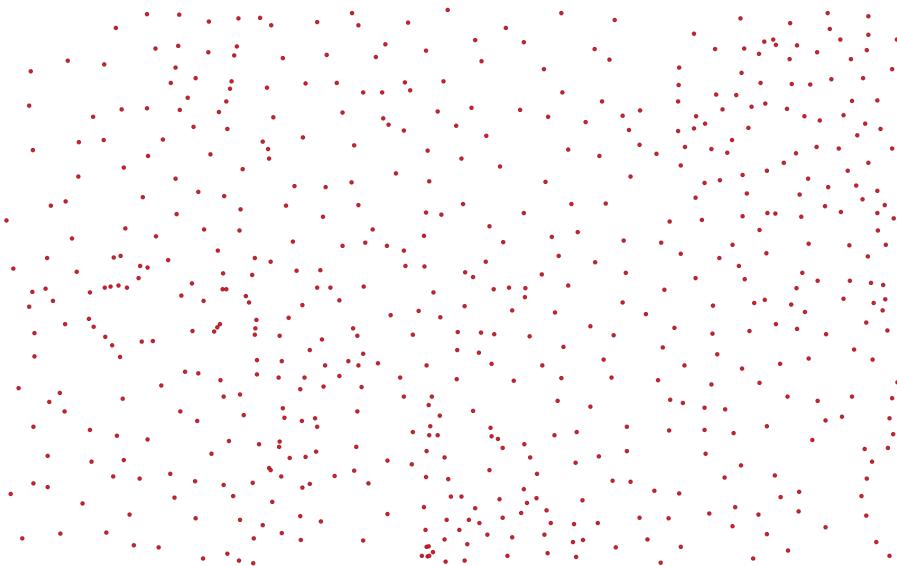


Figura 5.6: Visualización del elemento *Punto Cota* en QGIS

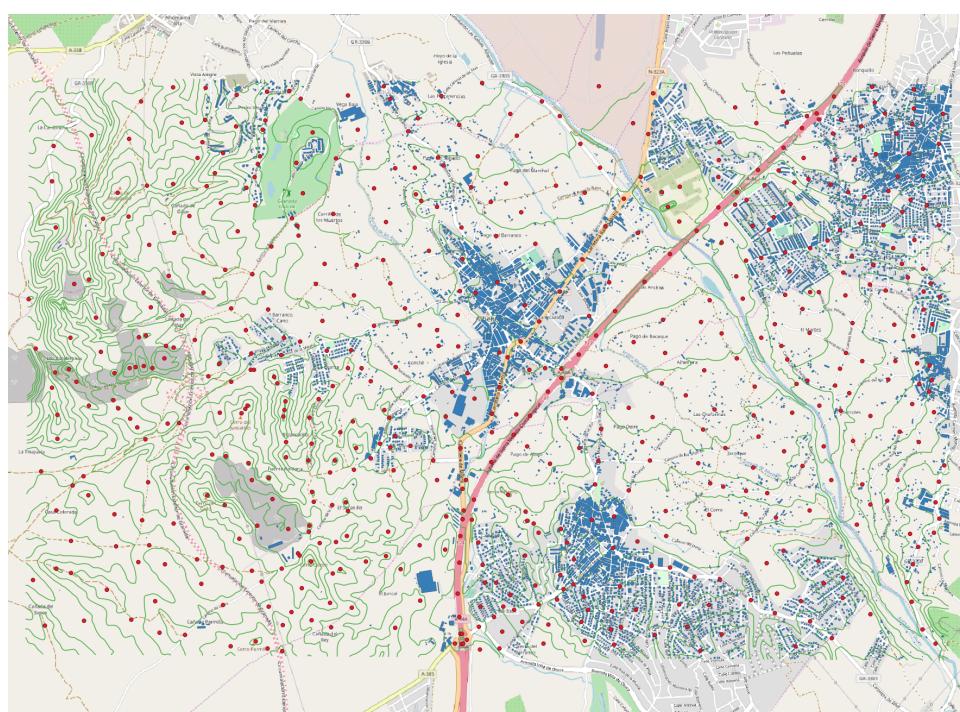


Figura 5.7: Visualización de las tres geometrías en QGIS

*Nota:* Estos ficheros se encuentran en el directorio `datos<shapefile`.

Con esto se han presentado los datos que vamos hacer accesibles a través de la Web Semántica. En la figura 5.7 es posible visualizar las tres geometrías juntas y localizadas en QGIS a través de la capa *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. Por tanto, al disponer de los datos listos para su uso, es necesario saber que para poblar una ontología con Protegé, se necesitan los datos geoespaciales en formato para CSV. Adicionalmente, como hemos visto, la representación de la ubicación de cada geometría la vamos a usar mediante la codificación WKT. No obstante, QGIS nos ofrece diversas funcionalidades, entre las que se encuentran las anteriores mencionadas y que nos permiten realizar el trabajo de una manera más sencilla.

Para obtener el formato CSV en QGIS se realiza el mismo proceso en las tres capas: *click derecho sobre la capa en cuestión*, **Exportar>Guardar objetos como** y nos aparecerá una ventana como la de la figura 5.9. Además, para obtener la codificación WKT y poder hacer consultas a partir de la ubicación de las geometrías, debemos seleccionar en GEOMETRY la opción AS\_WKT (figura 5.8) y ya podemos guardar los CSV (figura 5.10). En las tablas 5.2, 5.3, 5.4 y 5.5 se muestran fragmentos de dichos CSV.

**Cuadro 5.2: Parte del CSV para Curvas de Nivel I**

WKT	GID	ID	TIPO	CATEGORIA
LINESTRING (440675.4 4106319.93, ....)	122602	102632	CGN	NOR
LINESTRING (440972.5 4106237.28, ...)	172684	102632	CGN	NOR
LINESTRING (440976.2 4106311.24, ...)	186391	102632	CGN	NOR
...	...	...	...	...
LINESTRING (442507.6 4105386.75, ...)	471150	102632	DEP	MAE

**Cuadro 5.3: Parte del CSV para Curvas de Nivel II**

WKT	PROCEDENCIA	COTA
LINESTRING (....)	CMB	770
LINESTRING (...)	CMB	780
LINESTRING (...)	CMB	780
...	...	...
LINESTRING (...)	CMB	750

Cuadro 5.4: Parte del CSV para Puntos de Cota

WKT	GID	ID	TIPO	CONTEXTO	COTA
MULTIPOINT ((446228.92 4108432.53))	890947	102632	TER	CGN	721.77
MULTIPOINT ((446265.60 4108675.83))	890948	102632	TER	CGN	715.40
...	...	...	...	...	...
MULTIPOINT ((442401.99 4104501.95))	891529	102632	TER	CGN	771.75

Cuadro 5.5: Parte del CSV para Edificaciones

WKT	GID	ID	TIPO	ESTADO
POLYGON ((446020.74 4107035.28, ...))	181062	102632	EDI	USO
POLYGON ((446050.16 4107127.71, ...))	181064	102632	EDI	USO
POLYGON ((441430.69 4106827.08, ...))	758271	102632	EDI	USO
...	...	...	...	...
POLYGON ((446168.83 4108720.87, ... ))	1330488	102632	PAT	CGN

*Nota:* Los CSV se encuentran en el directorio `datos<csv`.

En las figuras 5.8, 5.9 y 5.10 se muestra el proceso para obtener la información que acabamos de comentar.



Figura 5.8: Seleccionar opciones para guardar el Shapefile

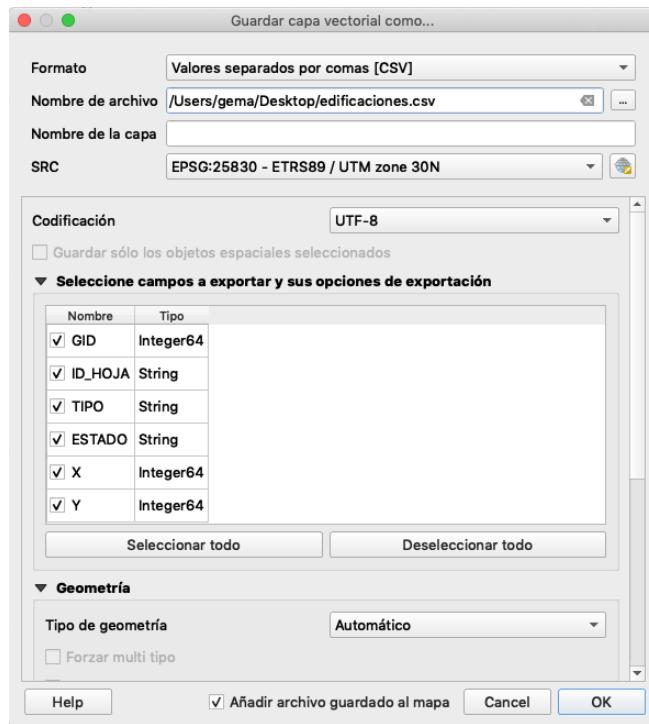


Figura 5.9: Obtener la información de los Shapefiles en CSV

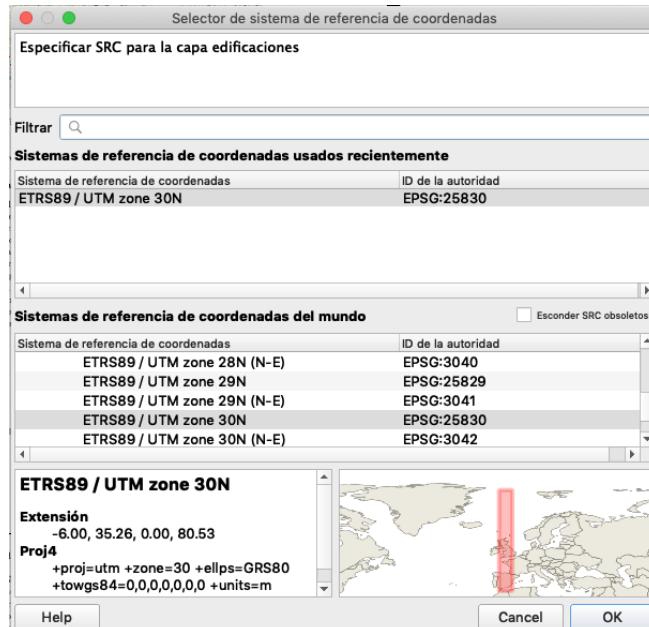


Figura 5.10: Guardando el fichero CSV de los Shapefiles

Por tanto, la información que se muestra en las anteriores tablas, es la que se utilizará para poblar nuestra ontología, siendo necesario disponer de ellas en formato Excel.

### 5.3. Creación de la ontología GEOARES

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. En este caso, crearemos una nueva ontología a medida llamada GEOARES. 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. Los datos geoespaciales interoperables hacen posible la distribución rápida de información y el intercambio entre departamentos [45].

No obstante, escribir directamente en lenguajes como RDF y OWL para la creación de la ontología resultan sumamente difícil y son propensos a errores. Afortunadamente, existen en el mercado entornos gráficos para visualizar y construir ontologías de forma más fácil, como **Protegé** (figura 5.11). Es por eso que vamos hacer uso de este software para la creación de nuestra ontología.

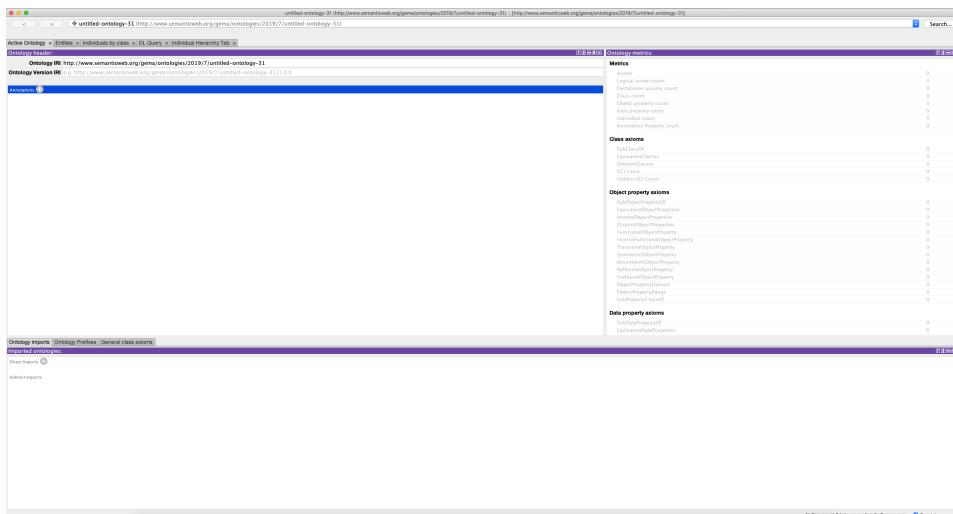
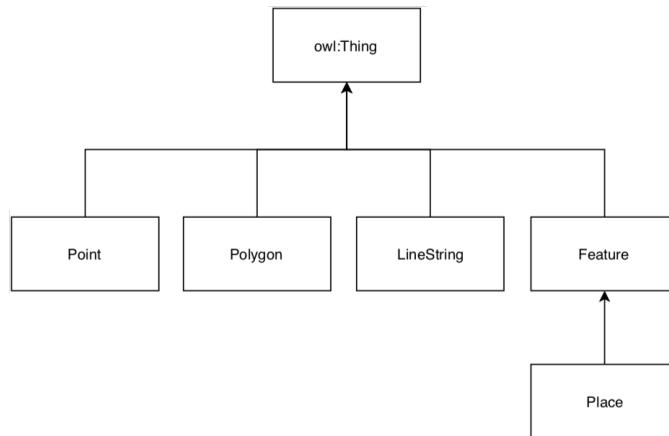


Figura 5.11: Inicio de Protegé

Hay que tener en cuenta que la ontología está enfocada para ser manejada en España y los datos obtenidos son procedentes de Andalucía, por lo cual las clases aparecerán en Español, a excepción de las que se nos especifique en GeoSPARQL según el estándar especificado. Básicamente una ontología geoespacial se basa en las clases mostradas en la figura 5.12.



**Figura 5.12:** Estructura para una ontología geoespacial

No obstante, aparte de hacer uso de las especificaciones de una ontología geoespacial, es interesante añadir información específica de nuestros datos geográficos, con el fin de enriquecer más nuestra ontología y hacerla más personalizable. Entonces, lo primero que tenemos que hacer es definir los URIs con los que vamos a trabajar, como vamos hacer uso tanto de SPARQL como GeoSPARQL, necesitaremos los que se marcan (figura 5.13).

Ontology imports		Ontology Prefixes	General class axioms
<b>Ontology prefixes:</b>			
Prefix		Value	
geo		#	
my		http://www.opengis.net/ont/geosparql#	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 5.13:** Definición de URIs para la ontología

A continuación, creamos las clases y subclases por las que estará formada nuestra ontología. En la figura 5.14 se puede ver el resultado final.

- La clase *Polygon* con IRI <http://www.opengis.net/ont/sf>

- La clase *Point* con IRI <http://www.opengis.net/ont/sf>
- La clase *LineString* con IRI <http://www.opengis.net/ont/sf>
- La clase *Feature* con IRI <http://www.opengis.net/ont/geosparql>
  - La subclase *GID* con IRI <http://example.org/ApplicationSchema>
- La clase *Edificaciones* con IRI <http://example.org/ApplicationSchema>
  - La subclase *TipoEdificaciones* con IRI <http://example.org/ApplicationSchema>
  - La subclase *EstadoEdificaciones* con IRI <http://example.org/ApplicationSchema>
- La clase *PuntoCota* con IRI <http://example.org/ApplicationSchema>
  - La subclase *TipoPuntoCota* con IRI <http://example.org/ApplicationSchema>
  - La subclase *ContextoPuntoCota* con IRI <http://example.org/ApplicationSchema>
- La clase *CurvaNivel* con IRI <http://example.org/ApplicationSchema>
  - La subclase *CategoríaCurvaNivel* con IRI <http://example.org/ApplicationSchema>
  - La subclase *ProcedenciaCurvaNivel* con IRI <http://example.org/ApplicationSchema>
  - La subclase *TipoCurvaNivel* con IRI <http://example.org/ApplicationSchema>

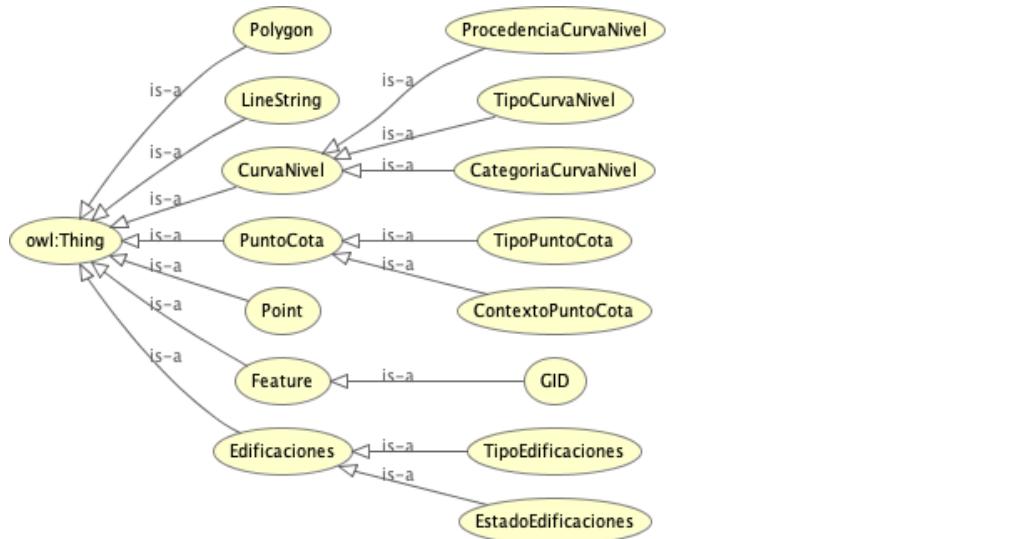


Figura 5.14: Ontología de prototipo

Al usar el formato WKT, tenemos que crear el tipo de dato `geo:wktLiteral` en *Datatype* con IRI <http://www.opengis.net/ont/geosparql> (figura 5.15).

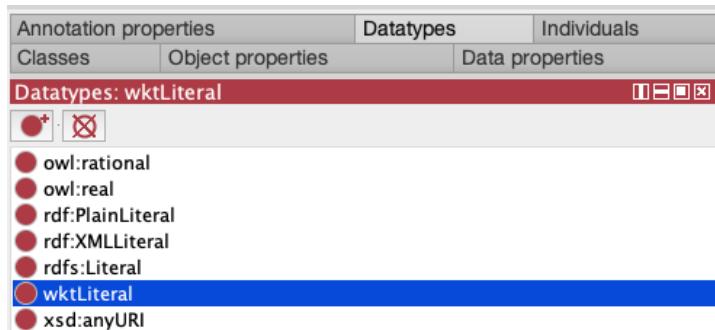


Figura 5.15: Crear el tipo de dato `geo:wktLiteral`

Por otro lado, para asociar geometrías con sus literales es necesario crear `geo:asWKT`. Además de crear la propiedad `tieneCota` con IRI <http://www.opengis.net/ont/geosparql> para obtener el valor de la cota en las geometrías de puntos y de líneas, atributo elemental en los elementos de curva de nivel y puntos de cota (figura 5.16).

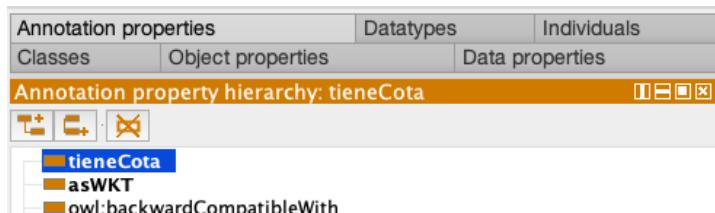


Figura 5.16: Crear las propiedades

Con esto ya hemos terminado de definir nuestra ontología. A continuación, pasamos a poblarla, en donde deberemos identificar las instancias de los datos geográficos escogidos.

## 5.4. Poblar la ontología

Protegé nos permite poblar una ontología haciendo uso de las herramientas que ofrece, es por eso que la elección de dicho software ha sido motivada principalmente por esta característica. Crear 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 escogida para poblar es

que debemos aprender unas reglas, sin embargo, en cualquier caso esta forma es más eficiente que hacerlo a mano, lo que conlleva una línea de aprendizaje menor. Para guardar las instancias de cada una de las capas y asociarlas a las clases ya creadas, vamos hacer uso del GID, y no del ID HOJA, puesto que el GID es único para cada valor.

*Nota:* Las reglas se encuentran en el directorio `datos<reglas`.

#### 5.4.1. Edificaciones

Empezamos poblando la ontología con las Edificaciones cuya geometría es de polígonos. En la figura 5.17 se nos abre una ventana que nos permite generar reglas, como veremos seguidamente, y así crear las instancias y clases necesarias correspondientes a esta clase (figura 5.19).

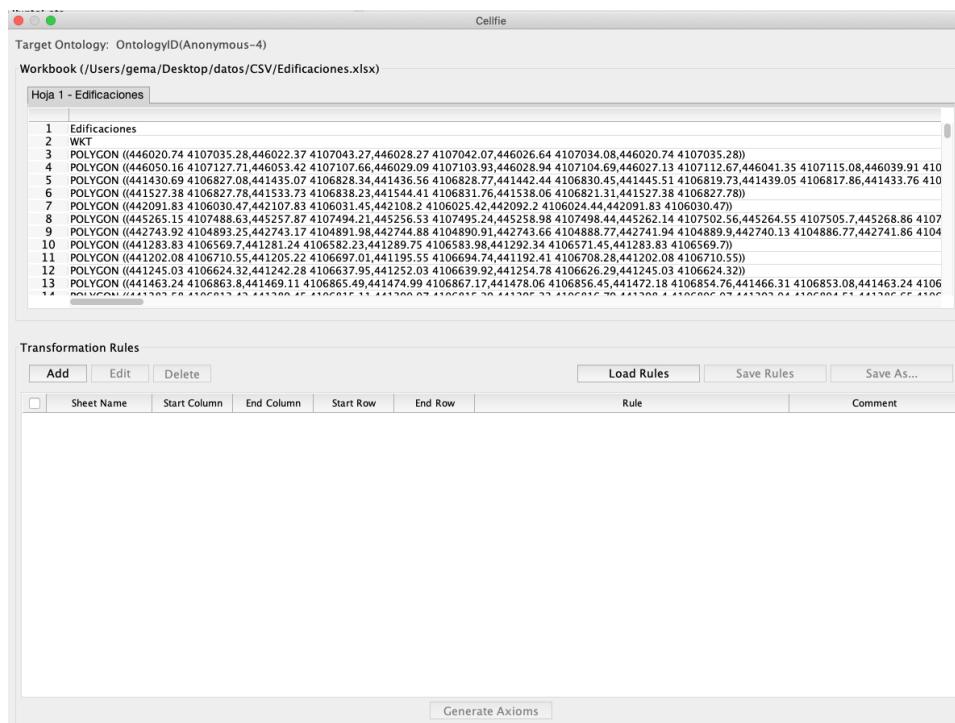


Figura 5.17: Poblar ontología para los datos de Edificaciones

```

1 # Regla para agregar las instancias de la clase Polygon
2 # a la propiedad geo:asWKT
3 Individual: @B*
4 Annotations: asWKT @A* (wktLiteral)

```

```

1 # Regla para agregar las instancias GID
2 # a la clase GID y Polygon
3 Individual: @B*
4 Types: GID, Polygon

```

```

1 # Regla para agregar las subclases que tiene
2 # la clase TipoEdificaciones
3 Class: @D*
4 SubClassOf: TipoEdificaciones
5
6 # Asignar los GID a las subclases de TipoEdificaciones
7 Individual: @B*
8 Types: @D*

```

```

1 # Regla para agregar las subclases que tiene
2 # la clase EstadoEdificaciones
3 Class: @E*
4 SubClassOf: EstadoEdificaciones
5
6 # Asignar los GID a las subclases de EstadoEdificaciones
7 Individual: @B*
8 Types: @E*

```

Después de haber creado las reglas con éxito (figura 5.19) obtenemos un diagrama como el de la figura 5.18.



Figura 5.18: Creación de las subclases para TipoEdificaciones y EstadoEdificaciones de Edificaciones

	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: @D*	
✓	Hoja 1 - Edificacion/A	E	3	7303		Individual: @B* Annotations: asWKT @A*	
✓	Hoja 1 - Edificacion/A	E	3	7303		Class: @D* SubClassOf: TipoEdificaciones	
✓	Hoja 1 - Edificacion/A	E	3	7303		Class: @E* SubClassOf: EstadoEdificaciones	
✓	Hoja 1 - Edificacion/A	E	3	7303		Individual: @B* Types: @E*	

Figura 5.19: Reglas para poblar la ontología con Edificaciones

#### 5.4.2. Curvas de Nivel

Seguimos poblando la ontología con las Curvas de Nivel cuya geometría es de líneas y generamos de igual forma las reglas necesarias así crear las instancias y clases necesarias correspondientes a esta clase (figura 5.21).

```

1 # Regla para agregar las instancias GID
2 # a la clase GID y LineString
3 Individual: @B*
4 Types: GID, LineString

```

```

1 # Regla para agregar las subclases que tiene
2 # la clase TipoCurvaNivel
3 Class: @D*
4 SubClassOf: TipoCurvaNivel
5
6 # Asignar los GID a las subclases de TipoCurvaNivel
7 Individual: @B*
8 Types: @D*

```

```

1 # Regla para agregar las subclases que tiene
2 # la clase ProcedenciaCurvaNivel
3 Class: @F*
4 SubClassOf: ProcedenciaCurvaNivel
5
6 # Asignar los GID a las subclases de
7 # ProcedenciaCurvaNivel
8 Individual: @B*
9 Types: @F*

```

```

1 # Regla para agregar las subclases que tiene
2 # la clase CategoriaCurvaNivel
3 Class: @E*
4 SubClassOf: CategoriaCurvaNivel
5
6 # Asignar los GID a las subclases de CategoriaCurvaNivel
7 Individual: @B*
8 Types: @E*

```

```

1 # Regla para agregar las instancias de la clase
2 # LineString a la propiedad geo:asWKT
3 Individual: @B*
4 Annotations: asWKT @A* (wktLiteral)

```

```

1 # Regla para agregar las instancias de la clase
2 # CurvaNivel a la propiedad tieneCota
3 Individual: @B*
4 Annotations: tieneCota @G* (xsd:double)

```

Después de haber creado las reglas con éxito (figura 5.21) obtenemos un diagrama como el de la figura 5.20.

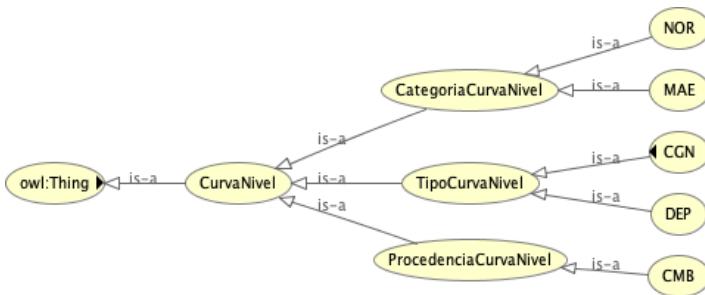


Figura 5.20: Creación de las subclases para TipoCurvaNivel, ProcedenciaCurvaNivel y CategoriaCurvaNivel de CurvaNivel

	Sheet Name	Start Column	End Column	Start Row	End Row	Rule	Comment
✓	Hoja 1 - curvanivel	A	G	3	155	Individual: @B*	
✓	Hoja 1 - curvanivel	A	G	3	155	Types: @D*	
✓	Hoja 1 - curvanivel	A	G	3	155	Individual: @B*	
✓	Hoja 1 - curvanivel	A	G	3	155	Types: CID, LineString	
✓	Hoja 1 - curvanivel	A	G	3	155	Individual: @B*	
✓	Hoja 1 - curvanivel	A	G	3	155	Types: @F*	
✓	Hoja 1 - curvanivel	A	G	3	155	Individual: @B*	
✓	Hoja 1 - curvanivel	A	G	3	155	Annotations: asWKT @A*	
✓	Hoja 1 - curvanivel	A	G	3	155	Class: @D*	
✓	Hoja 1 - curvanivel	A	G	3	155	SubClassOf: TipoCurvaNivel	
✓	Hoja 1 - curvanivel	A	G	3	155	Class: @E*	
✓	Hoja 1 - curvanivel	A	G	3	155	SubClassOf: CategoriaCurvaNivel	
✓	Hoja 1 - curvanivel	A	G	3	155	Individual: @B*	
✓	Hoja 1 - curvanivel	A	G	3	155	Annotations: tieneCota @G* (xsd:double)	
✓	Hoja 1 - curvanivel	A	G	3	155	Class: @F*	
						SubClassOf: ProcedenciaCurvaNivel	

Figura 5.21: Reglas para poblar la ontología con CurvaNivel

### 5.4.3. Puntos de Cota

Por último, poblamos la ontología con los Puntos de cota cuya geometría es de puntos y generamos reglas necesarias así creas las instancias y clases necesarias correspondientes a esta clase (figura 5.23).

```

1 # Regla para agregar las instancias GID
2 # a la clase GID y Point
3 Individual: @B*
4     Types: GID, Point

```

```

1 # Regla para agregar las subclases que tiene
2 # la clase TipoPuntoCota
3 Class: @E*
4     SubClassOf: TipoPuntoCota
5
6 # Asignar los GID a las subclases de TipoPuntoCota
7 Individual: @B*
8     Types: @D*

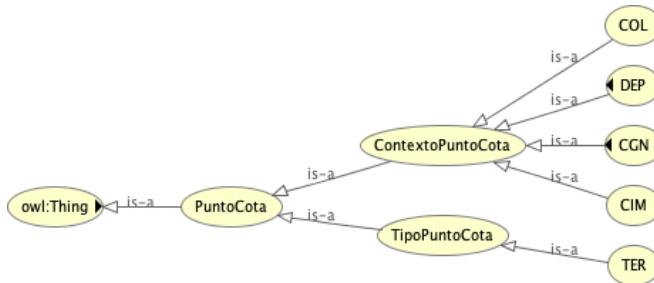
```

```

1 # Regla para agregar las subclases que tiene
2 # la clase ContextoPuntoCota
3 Class: @E*
4     SubClassOf: ContextoPuntoCota
5
6 # Asignar los GID a las subclases de ContextoPuntoCota
7 Individual: @B*
8     Types: @E*

```

Después de haber creado las reglas con éxito (figura 5.23) obtenemos un diagrama como el de la figura 5.22.



**Figura 5.22:** Creación de las subclases para TipoPuntoCota y ContextoPuntoCota de PuntoCota

```

1 # Regla para agregar las instancias de la clase Point
2 # a la propiedad geo:asWKT
3 Individual: @B*
4     Annotations: asWKT @A* (wktLiteral)

```

```

1 # Regla para agregar las instancias de la clase PuntoCota
2 # a la propiedad tieneCota

```

```

3 Individual: @B*
4 Annotations: tieneCota @F* (xsd:double)

```

	Sheet Name	Start Column	End Column	Start Row	End Row	Rule	Comment
✓	Hoja 1 - PuntoCota	A	F	3	585	Individual: @B*	
✓	Hoja 1 - PuntoCota	A	F	3	585	Annotations: tieneCota @F* (xsd:double)	
✓	Hoja 1 - PuntoCota	A	F	3	585	Individual: @B*	
✓	Hoja 1 - PuntoCota	A	F	3	585	Types: @D*	
✓	Hoja 1 - PuntoCota	A	F	3	585	Individual: @B*	
✓	Hoja 1 - PuntoCota	A	F	3	585	Types: GID, Point	
✓	Hoja 1 - PuntoCota	A	F	3	585	Class: @D*	
✓	Hoja 1 - PuntoCota	A	F	3	585	SubClassOf: TipoPuntoCota	
✓	Hoja 1 - PuntoCota	A	F	3	585	Individual: @B*	
✓	Hoja 1 - PuntoCota	A	F	3	585	Annotations: asWKT @A*	
✓	Hoja 1 - PuntoCota	A	F	3	585	Class: @E*	
✓	Hoja 1 - PuntoCota	A	F	3	585	SubClassOf: ContextoPuntoCota	
✓	Hoja 1 - PuntoCota	A	F	3	585	Individual: @B*	
✓	Hoja 1 - PuntoCota	A	F	3	585	Types: @E*	

Figura 5.23: Reglas para poblar la ontología con CurvaNivel

Y con esto hemos terminado de poblar nuestra ontología (figuras 5.25 y 5.24), para ver la ontología es posible ver el fichero `ontologia-geoares.owl`.

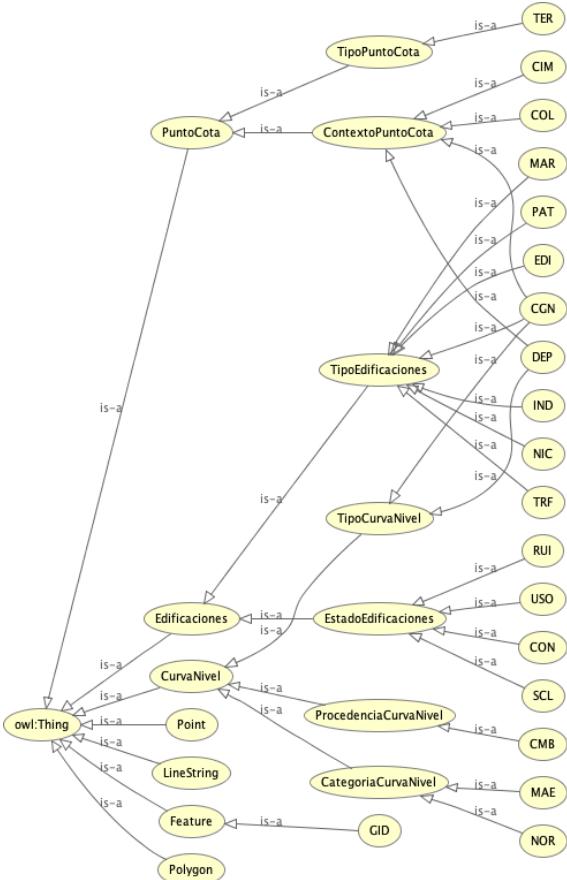


Figura 5.24: Ontología GEOARES

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 5.25: Información sobre la ontología GEOARES

## 5.5. Realización de consultas GeoSPARQL

Una vez creada nuestra ontología podemos pasar a hacer las consultas. Como GeoSPARQL es una extensión de SPARQL, para hacer uso de dicho lenguaje es necesario instalar un plugin. Sin embargo, debido a la imposibilidad de usarlo en Protegé, ya que no detecta los prefijos que tiene GeoSPARQL, he encontrado otras alternativas (en el Apéndice B: Problemas y Soluciones abarco todos los problemas encontrados durante la prueba de concepto y su posible solución). Así que he estado buscando si hay algún plugin con dicha librería y he hallado que GraphDB permite hacer uso de GeoSPARQL. En el Apéndice C: Instalación de GraphDB se presenta cómo instalar e inicializar GraphDB.

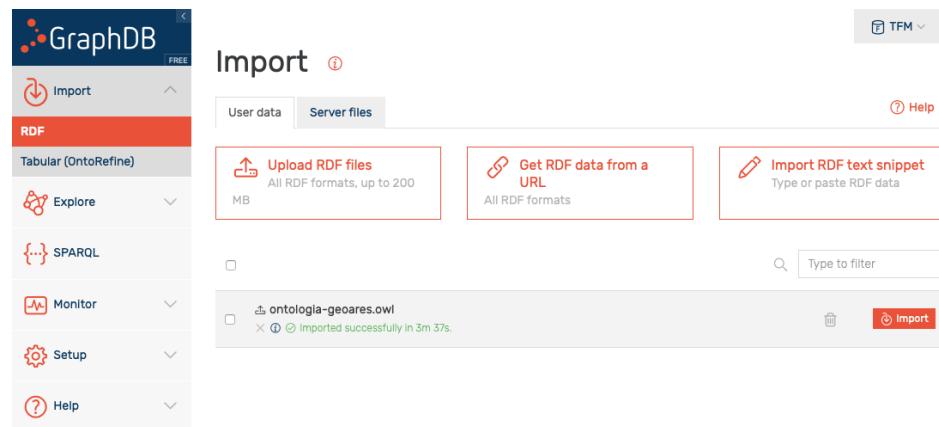
### 5.5.1. Activar los plugin necesarios para manejar GeoSPARQL

Para manejar GeoSPARQL es necesario habilitar el complemento GeoSPARQL. Cuando el complemento está habilitado, indexa todos los datos GeoSPARQL existentes en el repositorio y reindexa automáticamente cualquier actualización.

```

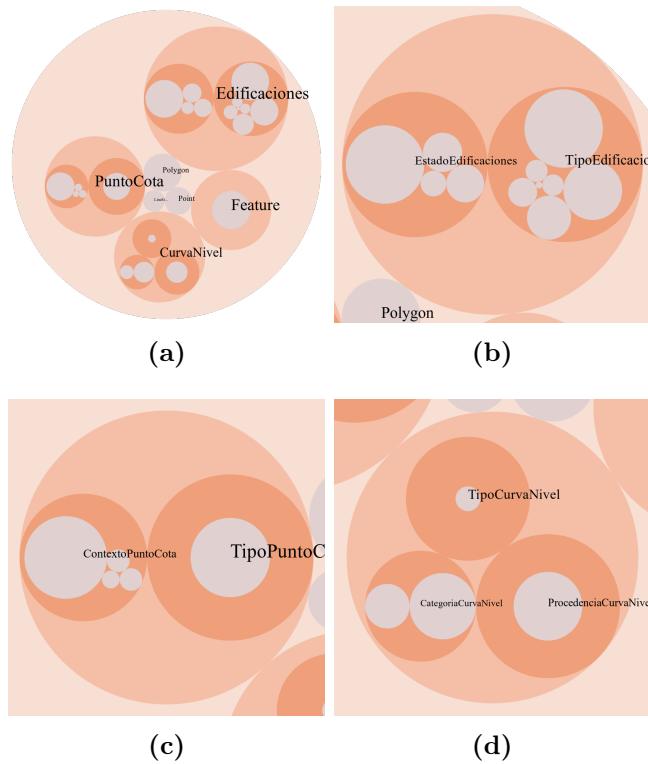
1 PREFIX : <http://www.ontotext.com/plugins/geosparql#>
2
3 INSERT DATA {
4   _:s :enabled "true" .
5 }
```

Con esto, ya podemos pasar a importar la ontología y hacer pruebas con ella (figura 5.26). Además, este software nos permite movernos gráficamente por la información contenida en la ontología GEOARES y así inspeccionar el grafo creado (figura 5.27).



**Figura 5.26:** Cómo importar una ontología en GraphDB

Asimismo, una buena funcionalidad que tiene esta herramienta es la posibilidad de inspeccionar la ontología por sus diferentes clases e instancias (figura 5.27), lo que nos permite buscar o ver el tamaño de cada clase.



**Figura 5.27:** Visualizar la ontología completa en GraphDB

Seguidamente, comenzamos la realización de las consultas de SPARQL y GeoSPARQL, clasificando las consultas de acuerdo a la geometría que contiene. Al principio comenzaremos con consultas más sencillas, hasta la obtención de la consulta que nos permita saber las **edificaciones que se encuentran a una cierta altura o cercanas a una cota** con el fin de obtener un fin, para ello, hemos considerado dos geometrías que nos pueden dar la misma aproximación de altura pero con geometrías distintas, una con puntos y otra con líneas.

### 5.5.2. Consultas donde intervienen solo polígonos

*Ejemplo 1.* Consulta que obtiene los tres primeros polígonos de Edificaciones que pertenecen a la clase TipoEdificaciones y son EDI (Edificación). Con esta consulta queremos obtener las edificaciones que están edificadas.

```

1 SELECT ?x ?p
2 WHERE {
3   ?x rdf:type geo:EDI .
4   ?polygon geo:asWKT ?p
5 }
```

En la figura 5.28 podemos observar los GID correspondientes y sus coordenadas asociadas.

	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 4107127.71,446053.42 4107107.66,446029.09 4107103.93,446028.94 4107104.69,46027.13 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 5.28: Salida de la consulta para el Ejemplo 1

*Ejemplo 2.* Consulta que obtiene la geometría específica de WKT mediante el operador `geof:sfEquals`. Con esta consulta queremos obtener las Edificaciones correspondientes a una determinada ubicación.

```

1 SELECT ?f
2 WHERE {
3   ?f geo:asWKT ?fWKT .
4
5   FILTER (geof:sfEquals(?fWKT , '' ,
```

```

6   POLYGON ((446050.16 4107127.71, 446053.42 4107107.66,
7     446029.09 4107103.93, 446028.94 4107104.69, 446027.13
      4107112.67, 446041.35 4107115.08, 446039.91
      4107125.24, 446050.16 4107127.71))
8 } ,'^^geo:wktLiteral))

```

En la figura 5.28 podemos observar cómo una ubicación exacta solo referencia a una geometría. Además, si pinchamos sobre ella nos devuelve información relevante al respecto (figura 5.30).

		f	
1	geosparql:181064		

Figura 5.29: Salida de la consulta para el Ejemplo 2

	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))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
2	geos-parql:181064	rdf:type	geosparql:ED1	<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 5.30: Información de la consulta para el Ejemplo 2

Para comprobar que corresponde dicha salida con la información de la ontología, veamos el fragmento correspondiente al GID 181064.

```

1 <!-- http://www.opengis.net/ont/geosparql#181064 -->
2
3 <geo:asWKT rdf:datatype="http://www.opengis.net/ont/
  geosparql#wktLiteral"> 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))

```

```
4 | </geo:asWKT>
```

### 5.5.3. Consultas donde intervienen solo puntos

*Ejemplo 3.* Consulta que obtiene los cinco primeros puntos de PuntoCota con una cota de 700 metros de altitud, y que pertenecen a la clase Contexto-PuntoCota y son DEP(Depresión). Con esta consulta queremos obtener los Puntos de Cota que tiene más de una cierta altura y son depresiones.

```
1 | SELECT DISTINCT ?point ?pointCota
2 | WHERE {
3 |   ?point rdf:type geo:DEP ;
4 |   geo:tieneCota ?pointCota ;
5 |
6 |   FILTER (?pointCota > 700)
7 | }
```

En la figura 5.31 podemos observar los cinco primeros puntos obtenidos y sus correspondientes cotas.

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

Figura 5.31: Salida de la consulta para el Ejemplo 3

*Ejemplo 4.* Consulta que obtiene la geometría específica de WKT mediante el operador `geof:sfWithin`. Con esta consulta queremos obtener los Puntos de Cota que están dentro de una determinada ubicación, pudiendo obtener un único valor o varios.

```
1 | SELECT ?f
2 | WHERE {
3 |   ?f geo:asWKT ?fWKT .
4 |
5 |   FILTER (geof:sfWithin(?fWKT, ''',
6 |           MULTIPOINT ((446228.92 4108432.53))
7 |           ''''^^geo:wktLiteral))
8 | }
```

En la figura 5.32 podemos observar que solo se encuentra dentro de esa ubicación una cota.

	f
1	geosparql:890947

Figura 5.32: Salida de la consulta para el Ejemplo 4

#### 5.5.4. Consultas donde intervienen solo líneas

*Ejemplo 5.* Consulta que obtiene la geometría específica de WKT mediante el operador `geof:sfEquals`. Con esta consulta obtener las Curvas de Nivel correspondientes a una ubicación exacta.

```

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

En la figura 5.33 podemos observar cómo una ubicación exacta solo referencia a una geometría.

	f
1	geosparql:172684

Figura 5.33: Salida de la consulta para el Ejemplo 5

Para comprobar que corresponde dicha salida con la información de la ontología, veamos el fragmento correspondiente al GID 172684.

```

1 <!-- http://www.opengis.net/ont/geosparql#172684 -->
2
```

```

3 <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)
4 </geo:asWKT>

```

### 5.5.5. Consultas donde intervienen líneas y puntos

*Ejemplo 6.* Consulta que obtiene las cinco primeras geometrías (líneas y puntos) que tienen una cota superior a 700 metros de altitud. Con esta consulta estamos quedándonos con las alturas correspondientes a un valor.

```

1 SELECT ?f
2 WHERE {
3   ?f geo:tieneCota ?fCota
4   FILTER (?fCota > 700)
5 }

```

En la figura 5.34 podemos observar los GID obtenidos.

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

Figura 5.34: Salida de la consulta para el Ejemplo 6

*Ejemplo 7.* Consulta que obtiene las cinco primeras geometrías (líneas y puntos) que tienen una cota superior a 720 metros de altitud y muestra tanto su ubicación con WKT como su cota. Con esta consulta estamos quedándonos con las alturas correspondientes a un valor, y mostrando el valor de su cota y su ubicación. En la figura 5.35 se observa el resultado.

```

1 SELECT DISTINCT ?point ?pointWKT ?pointCOTA
2 WHERE {
3   ?point geo:asWKT ?pointWKT ;
4   geo:tieneCota ?pointCOTA .
5   FILTER (xsd:double(?pointCOTA) > 720)
6 }

```

	point	pointWKT	pointCOTA
1	geosparql:122602	"LINESTRING (440675.4 4106319.93,440670.97 4106322.69,440666.96 4106326.35,4406 65.15 4106330.44,440664.95 4106335.46,440666.77 4106340.7,440669.25 4106342.74, 40674.38 4106344.42,440679.8 4106344.52,440685.13 4106343.42,440690.32 4106341.3 2,440694.75 4106338.61,440698.49 4106335.0,440700.62 4106330.14,440700.19 410632 6.38,440698.02 4106323.2,440694.1 4106320.67,440689.16 4106319.53,440675.4 410631 9.93)"^^< <a href="http://www.opengis.net/ont/geosparql#wktLiteral">http://www.opengis.net/ont/geosparql#wktLiteral</a> >	"770.0"^^xsd:double
2	geosparql:172684	"LINESTRING (440972.52 4106237.28,440972.4 4106232.04,440972.15 4106226.75,44096 7.68 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 410623 4.81,440958.11 4106240.01,440958.05 4106245.3,440958.04 4106250.86,440958.16 410 6255.93,440958.41 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)"^^< <a href="http://www.opengis.net/ont/geosparql#wktLiteral">http://www.opengis.net/ont/geosparql#wktLiteral</a> >	"780.0"^^xsd:double
3	geosparql:186391	"LINESTRING (440976.2 4106311.24,440973.32 4106318.03,440973.13 4106323.23,44097 3.29 4106328.26,440973.5 4106333.37,440973.76 4106339.73,440974.28 4106344.93,44 0974.76 4106350.18,440975.15 4106355.29,440975.27 4106360.94,440975.57 4106366.27 ,440977.22 4106383.16,440981.36 4106384.03,440984.26 4106382.85,440986.76 41063 79.78,440987.63 4106375.83,440987.91 4106370.67,440987.97 4106365.25,440987.85 41 06360.0,440987.11 4106354.98,440985.73 4106349.88,440984.53 4106344.41,440984.1 4106339.26,440983.98 4106333.75,440983.14 4106328.19,440982.12 4106323.22,44098 1.29 4106318.02,440976.2 4106311.24)"^^< <a href="http://www.opengis.net/ont/geosparql#wktLiteral">http://www.opengis.net/ont/geosparql#wktLiteral</a> >	"780.0"^^xsd:double

Figura 5.35: Salida de la consulta para el Ejemplo 7

Hay que tener en cuenta que hay muchas maneras de hacer las consultas, por lo que para realizar una consulta, no es necesario realizarla de la misma forma que estamos haciéndola nosotros.

### 5.5.6. Consultas donde intervienen polígonos y puntos

*Ejemplo 8.* Consulta que obtiene las cinco primeras geometrías de puntos cercanas a un metro del GID de Edificaciones 889549. Con esta consulta nos estamos quedando los Puntos de Cota que están cercanos al GID 889549, con esto es posible obtener la altura asociada a dicha edificación.

```

1 SELECT ?f
2 WHERE {
3   geo:889549 geo:asWKT ?WKT .
4   ?f rdf:type sf:Point ;
5   geo:asWKT ?fWKT .
6 }

```

```

7   FILTER  (?fGeom != ?WKT)
8 }
9 ORDER BY ASC(geof:distance(?cWKT, ?fWKT, uom:metre))

```

En la figura 5.36 podemos apreciar los 5 puntos más cercanos al 889549.

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

Figura 5.36: Salida de la consulta para el Ejemplo 8

*Ejemplo 9.* Consulta que obtiene las cinco primeras geometrías de polígonos cercanas a un metro de las geometrías de puntos. Con esta consulta nos quedamos con las Edificaciones que están a más de 700 metros de altitud.

```

1 SELECT ?p
2 WHERE {
3   ?p rdf:type sf:Polygon ;
4   geo:asWKT ?pWKT .
5   ?f rdf:type sf:Point ;
6   geo:tieneCota ?pointCota ;
7   geo:asWKT ?fWKT .
8
9   FILTER (?pointCota > 700)
10 }
11 ORDER BY ASC(geof:distance(?pWKT, ?fWKT, uom:metre))

```

En la figura 5.36 podemos apreciar los 5 edificios más cercanos a una cota superior a 700 metros de altura.

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

Figura 5.37: Salida de la consulta para el Ejemplo 9

### 5.5.7. Consultas donde intervienen polígonos y líneas

*Ejemplo 10.* Consulta que obtiene las cinco primeras geometrías de líneas cercanas a un metro del GID de Edificaciones 889549. Con esta consulta nos estamos quedando las Curvas de Nivel que están cercanos al GID 889549, con esto es posible obtener la altura asociada a dicha edificación.

```

1 SELECT ?f
2 WHERE {
3   geo:889549 geo:asWKT ?WKT .
4   ?f rdf:type sf:LineString ;
5   geo:asWKT ?fWKT .
6
7   FILTER (?fGeom != ?WKT)
8 }
9 ORDER BY ASC(geof:distance(?cWKT, ?fWKT, uom:metre))

```

En la figura 5.36 podemos apreciar las 5 líneas más cercanos al edificio 889549.

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

Figura 5.38: Salida de la consulta para el Ejemplo 10

### 5.5.8. Consultas donde intervienen polígonos, líneas y puntos

*Ejemplo 11.* Consulta que obtiene las cinco primeras geometrías más cercanas a un metro del GID de Edificaciones 889549. Con esta consulta nos estamos quedando con las características más cercanas al GID 889549.

```

1 SELECT ?f
2 WHERE {
3   geo:889549 geo:asWKT ?WKT .
4   ?f geo:asWKT ?fWKT .
5
6   FILTER (?fGeom != ?WKT)
7 }
8 ORDER BY ASC(geof:distance(?cWKT, ?fWKT, uom:metre))

```

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

Figura 5.39: Salida de la consulta para el Ejemplo 11

Acabamos de comprobar como es posible obtener de manera eficaz los GID ubicados en el mapa, sin embargo *¿dónde se encuentran exactamente ubicados?* A continuación, contestamos esta pregunta.

*Nota:* Las consultas se encuentran en el directorio *consultas*.

## 5.6. Visualización de la Información Geográfica

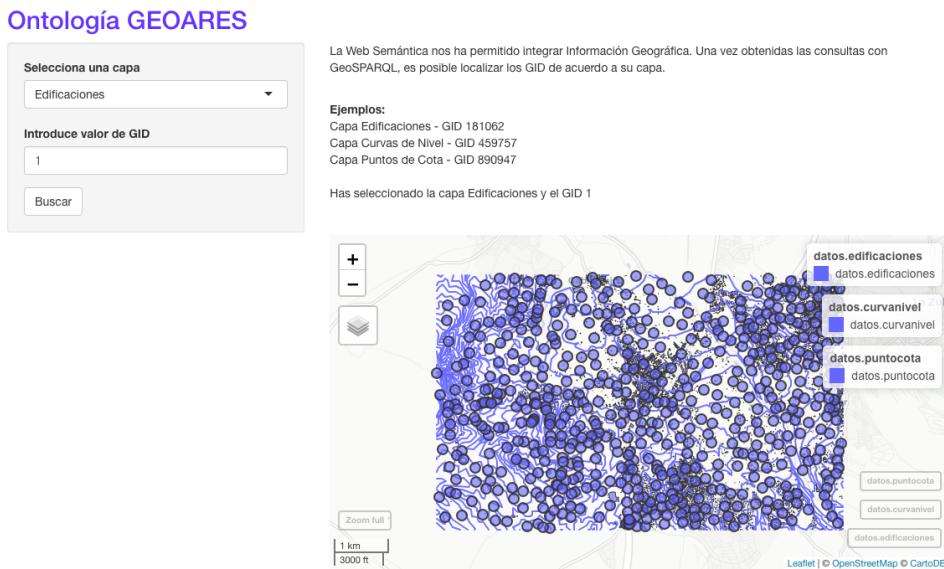


Figura 5.40: Página principal del mapa Web

Con los GID que antes hemos mostrado, no conocemos información acerca de a que GID está haciendo referencia *¿entonces como podemos visualizarlo y ubicarlos en un mapa?* La respuesta es muy simple con Leaflet

que es una biblioteca JavaScript de código abierto ampliamente utilizada para crear aplicaciones de mapeo Web. Para hacer uso de ella se suele usar principalmente Java, sin embargo, nosotros hemos hecho uso de R al haber encontrado un tutorial que explica muy bien su utilización con Shiny<sup>2</sup> (<https://rstudio.github.io/leaflet/shiny.html>). Para ejecutarla hace falta abrir el fichero en R Studio y pulsar el botón “Run App”.

En las figuras 5.40, 5.41, 5.42 y 5.43 podemos ver una visualización del mapeo Web realizado para encontrar los GID de nuestros datos en el mapa. En la figura 5.40 se observa la página inicial de la aplicación. Con esto, se extiende el uso que se le puede dar a la prueba de concepto, ya que esta aplicación ha permitido que podamos obtener y ubicar en un mapa información.

En las figuras 5.41, 5.42 y 5.43 se observan las geometrías ubicadas en el mapa para las distintas geometrías que tenemos, para ello se ha hecho un ejemplo con cada una de ellas. Por esa misma razón, a partir de la realización de la consulta para obtener los edificios que están a una determinada altura es posible saber de manera gráfica donde se encuentra su ubicación exacta mediante el uso de esta aplicación. No obstante, hemos puesto que se ubique en el mapa las tres geometrías que hemos venido comentando. A continuación, se muestran tres ejemplos, en donde se debe pulsar el botón “Buscar” para que los ubique en el mapa.

#### Ontología GEOARES

The screenshot shows a Shiny application interface. On the left, a sidebar titled "Ontología GEOARES" contains a dropdown menu set to "Edificaciones", a text input field with "181062", and a "Buscar" button. Above the input field is a note: "La Web Semántica nos ha permitido integrar Información Geográfica. Una vez obtenidas las consultas con GeoSPARQL, es posible localizar los GID de acuerdo a su capa." Below the input field is a message: "Has seleccionado la capa Edificaciones y el GID 181062". To the right is a map of a residential area with several buildings highlighted in blue. A callout box labeled "shape1" points to one of the buildings. A tooltip provides detailed information about the selected feature:

Feature ID	0
1GID	181062
2ID_HOJA	102632
3TIPO	EDI
4ESTADO	USO
5X	NA
6Y	NA
7geometry	sfc_POLYGON

At the bottom left of the map are zoom controls and a scale bar indicating "30 m" and "100 ft". At the bottom right, there are credits: "Leaflet | © OpenStreetMap © CartoDB".

**Figura 5.41:** Obtención de la información del GID para Edificaciones

<sup>2</sup>Shiny es un framework Web para R.

### Ontología GEOARES



Figura 5.42: Obtención de la información del GID para Curvas de Nivel

### Ontología GEOARES

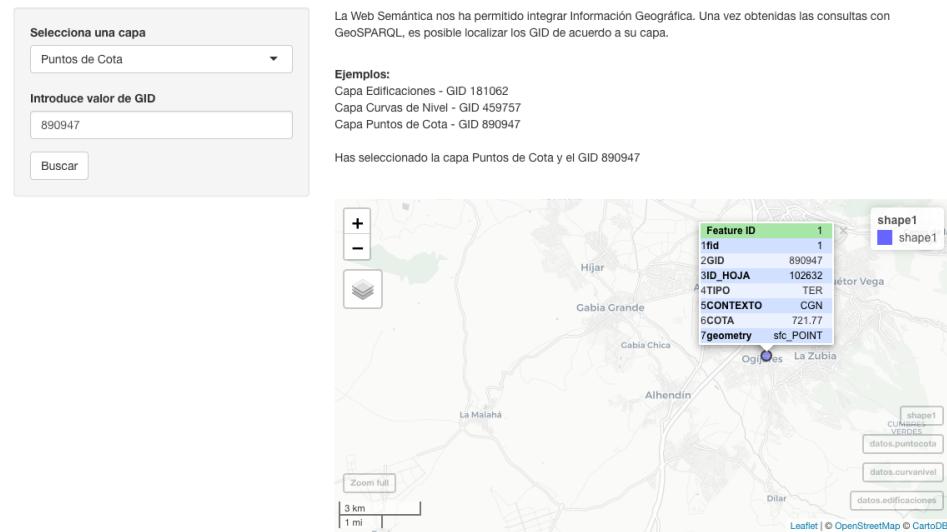


Figura 5.43: Obtención de la información del GID para Puntos de Cota

**Nota:** El código se encuentra en el directorio *visualizacion*.

### **5.7. Conclusiones**

En este capítulo, se ha presentado la prueba de concepto realizada para la Web Semántica Geoespacial en donde se han obtenido los datos del Instituto de Estadística y Cartografía de Andalucía, en concreto la información geoespacial de edificaciones, curvas de nivel y puntos de cota de Ogíjares (Granada) en formato Shapefile. A partir de esa información se ha creado la ontología GEOARES en RDF/OWL y poblado mediante la herramienta Protegé, haciendo uso de unas determinadas reglas. Seguidamente, debido a diversos problemas, se ha hecho uso de la alternativa GraphDB para aplicar las búsquedas mediante los lenguajes de consulta GeoSPARQL y SPARQL. Finalmente, se ha creado una pequeña aplicación de mapa Web para obtener las salidas de las consultas en un mapa y así observar su visualización. Esta prueba de concepto nos ha permitido comprobar cómo la Web Semántica permite hacer uso de diversas herramientas esenciales que facilitan la búsqueda de la información, en este caso de Información Geográfica.

## Capítulo 6

# Conclusiones y Trabajo Futuro

**RESUMEN:** Este capítulo recoge los principales aspectos y contribuciones obtenidos para la integración de Información Geográfica en la Web Semántica. De igual manera, comprobaremos el cumplimiento de los objetivos que introducimos inicialmente en el [capítulo 1](#). Asimismo, se comentarán posibles propuestas de mejora de cara a posibles iteraciones del proyecto, además de comentar otras posibles aplicaciones relacionadas, que caen fuera del ámbito de este trabajo.

### 6.1. Conclusiones

En este TFM se ha profundizado en el uso de la Web Semántica Geoespacial, en concreto en la posibilidad de integrar Información Geográfica en la Web Semántica mediante el uso de herramientas específicas, a través de la creación de la ontología GEOARES. Se han estudiado algunas de las tecnologías que se pueden utilizar para representar e integrar Información Geográfica, como los lenguajes de ontologías (RDF, OWL) y de consultas (SPARQL y GeoSPARQL). Se han valorado las herramientas de la Web Semántica para permitir crear la ontología, como GraphDB y Protegé, esta última no ha permitido incluir el *plugin* de GeoSPARQL, no obstante, la alternativa encontrada, GraphDB, ha permitido solventar los problemas ocasionados con Protegé.

Así, se ha desarrollado una prueba de concepto trabajando con información geoespacial, concretamente con archivos en formato *Shapefile* de la provincia de Granada, en particular de mi pueblo, Ogíjares, y centrándonos en la información geoespacial obtenida de edificaciones, puntos de cota y curvas de nivel, para su integración en la ontología y su posterior visualización en un mapa a través de R. Dentro de todas las posibilidades de análisis que ofrece la Web Semántica, se ha prestado especial atención a la relaciona-

da con Información Geográfica, en donde, gracias a los estándares definidos, es decir, las funciones usadas en GeoSPARQL, las cuales son las mismas que OGC ha definido en su estándar para el tratamiento de datos SIG, es posible combinar dos ámbitos que previamente parecían ser independientes el uno respecto del otro. Entonces, se han utilizado tecnologías tanto de la Web Semántica como de los SIG que nos allanan el camino para ambas áreas, permitiendo la heterogeneidad e interoperabilidad de los datos geográficos gracias a la capa semántica que estos presentan.

A la vista de los resultados obtenidos, considero que haber realizado esta prueba de concepto me ha abierto nuevos puntos de vista hacia otra perspectiva de la Web. Parte de mi contribución ha sido enseñar las herramientas que facilitan el trabajo para la puesta en marcha de una ontología. Por tanto, mediante el desarrollo de este TFM, valoro que se han conseguido los objetivos señalados en la subsección 1.5.2 del capítulo 1, en concreto:

1. Se ha mostrado mediante la utilización de la Web Semántica Geoespacial, cómo los estándares que se usan en el lenguaje de consulta geoespacial corresponden a los mismos definidos por el organismo OGC para los datos geográficos (ver capítulo 4).
2. Se han definido dos fuentes principales de información con las que obtener datos públicos geográficos de buena calidad y se han usado los datos de una de ellas, en concreto los del Instituto de Estadística y Cartografía de Andalucía (ver sección 5.2).
3. Se han considerado herramientas de la Web Semántica de gran alcance para representar e integrar Información Geográfica, entre las que destacamos Protegé y GraphDB (ver capítulo 5).
4. Se ha mostrado mediante Protegé cómo es posible agregar conocimiento de dominio geoespacial sobre la estructura de los datos de manera sencilla (ver sección 5.3).
5. Se ha mostrado mediante Protegé cómo es posible poblar una ontología de manera eficaz haciendo uso de diversas reglas (ver sección 5.4).
6. Se ha realizado un estudio acerca de las carencias o mejoras que supone usar Protegé (ver Apéndice B).
7. Se han mostrado la estructura de los datos geoespaciales y sus especificaciones para realizar consultas con el lenguaje GeoSPARQL (ver sección 4.3 y sección 5.5).
8. Se ha aprendido a crear un mapa Web en R mediante el uso de la biblioteca *Leaflet* y el framework *Shiny* para ubicar información geoespacial en un mapa (ver sección 5.6).

9. Se ha mostrado, mediante ejemplos, formas de trabajo sobre la Web Semántica Geoespacial (ver capítulo 5).
10. Se ha prestado especial atención a desarrollar un trabajo autocontenido, tratando que sea accesible a cualquier persona con un perfil tecnológico básico que esté interesado en el tema.
11. He adquirido experiencia y conocimientos en un tema nuevo para mí, la Web Semántica, y ampliado otros como el caso de los Sistemas de Información Geográfica.

## 6.2. Proyectos relacionados

Dentro de los proyectos relacionados con la Web Semántica y los SIG, destacan los documentados en el libro ***Geospatial Semantic Web*** de *Chuanrong Zhang, Tian Zhao y Weidong Li*. Este libro cubre temas clave relacionados con la Web Semántica Geoespacial; ontología geoespacial para la interoperabilidad semántica; creación, intercambio e integración de ontologías; desafíos de la Web Semántica Geoespacial; desarrollo de aplicaciones Web Semánticas Geoespaciales, entre otros. Asimismo, se describen tecnologías, como WFS, WMS, RDF, OWL y GeoSPARQL, y muestra cómo usar las tecnologías de la Web Semántica Geoespacial para resolver problemas prácticos del mundo real, como los datos espaciales [45].

También son remarcables los trabajos siguientes: “La Web semántica y las tecnologías del lenguaje humano” en [42], “Geography 2.0—A mash-up perspective” [8] y “Data Models and Query Languages for Linked Geospatial Data” en [23]; en donde se cubren las áreas de la Web Semántica, SIG y la combinación de ambas, entre otros.

Por otro lado, destaca un proyecto de la Universidad de Almería **XOSM**, *XQuery for OpenStreetMap* (<http://xosm.ual.es/XOSM/query.php>), que hace uso de la Información Geográfica Voluntaria mediante OpenStreetMap (OSM) para ofrecer datos de mapas urbanos y rurales. XOSM es una herramienta para el procesamiento de consultas geoespaciales en OSM equipada con un lenguaje de consulta basado en un conjunto de operadores definidos para OSM implementados como una biblioteca del lenguaje de consulta *XML XQuery*. Esta biblioteca proporciona un repertorio de funciones espaciales que actúan sobre la representación XML de una capa OSM y permite la definición de consultas que combinan capas OSM y capas creadas a partir de recursos de datos abiertos vinculados (KML, GeoJSON, CSV y RDF).

En la figura 6.1 se puede observar la página de inicio de esta herramienta, en donde, explica en qué consiste.

The screenshot shows the XOSM (XQuery for OpenStreetMap) homepage. At the top, there are navigation links for 'XOSM', 'Home', and 'Query'. On the right, there is a 'Contact us' link. The main title 'XOSM' is prominently displayed, followed by the subtitle 'XQuery for OpenStreetMap'. Below this, a section titled 'The Aim of the Project' describes the tool's purpose in processing geospatial queries on OSM. To the right of this text is a graphic featuring a magnifying glass over a map, surrounded by various terms related to geospatial data and XOSM.

**XOSM Team**

Jesús Manuel Almendros-Jiménez, Antonio Becerra-Terón, Manuel Torres

**XOSM Publications**

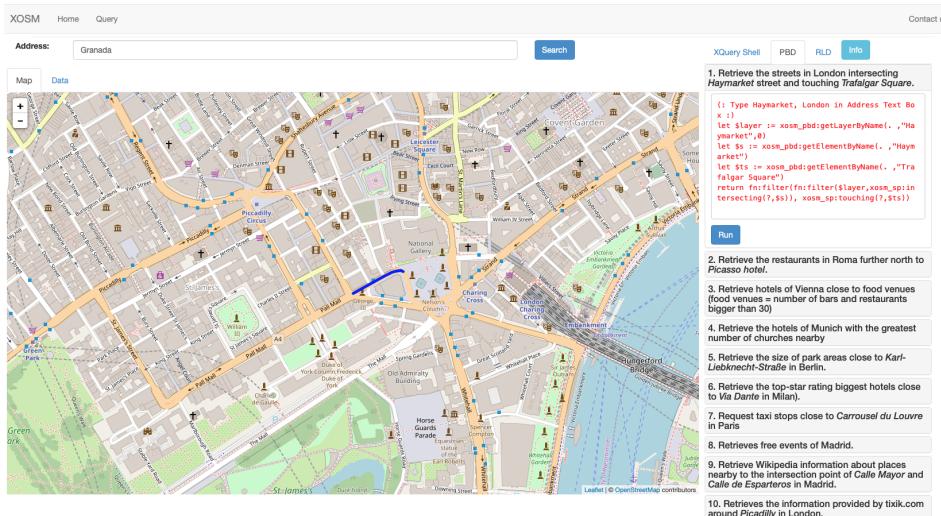
- Jesús Manuel Almendros-Jiménez, Antonio Becerra-Terón. XQuery-based Query Processing in Open Street Map. Geographical Information Systems Theory, Applications and Management. GISTAM 2015, Revised Selected Papers: 50-68, Springer, 2016.
- Jesús Manuel Almendros-Jiménez, Antonio Becerra-Terón, Manuel Torres. Aggregation Operators in Geospatial Queries for Open Street Map. OTM Conferences 2015: 501-518, Springer, 2015.
- Jesús Manuel Almendros-Jiménez, Antonio Becerra-Terón. Distance Based Queries in Open Street Map. DEXA Workshops 2015: 235-239, IEEE Computer Society Press, 2015.
- Jesús Manuel Almendros-Jiménez, Antonio Becerra-Terón. Querying Open Street Map with XQuery. GISTAM 2015: 61-71, SciTePress, 2015.

Figura 6.1: Página de inicio de XOSM

Por otro lado, las figuras 6.2 y 6.3 nos muestran un ejemplo del funcionamiento de XOSM en donde se representan en un mapa las calles de Londres que cruzan la calle *Haymarket* y tocan *Trafalgar Square*.

The screenshot shows the XOSM interface. At the top, there are navigation links for 'XOSM', 'Home', and 'Query'. On the right, there is a 'Contact us' link. The main area features a map of London with several orange lines highlighting specific street segments. A search bar at the top left contains the address 'Le. Calzada de Castro, Almería, Spain'. To the right of the map, there is a 'XQuery Shell' tab, which is currently active, showing the number '1'. Below the shell are buttons for 'Clear Map', 'Clear Query', and 'Run'.

Figura 6.2: Hacer consulta en XOSM



**Figura 6.3: Salida de una consulta de ejemplo en XOSM**

Y cómo se observa, este TFM se mantiene en la línea de estos proyectos.

### 6.3. Trabajo futuro

Durante la realización de este TFM se han encontrado o han podido surgir temas, ejemplos o aplicaciones bastante relacionados pero que no se han estudiado en el mismo, por este motivo, comentaremos algunos aspectos a profundizar en el futuro:

1. Crear una plataforma parecida a la mostrada en las figuras ?? y **6.3**, en donde, a partir de la salida de la consulta que se haga, obtener el GID ubicado automáticamente en el mapa, es decir, integrar en una herramienta la prueba de concepto realizada.
2. Generar otro ejemplo con la misma Información Geográfica pero obtenida de otra parte, para comprobar cómo la Web Semántica, gracias a su estructura, permite la heterogeneidad.

Adicionalmente, considero que este tema y el enfoque que le he dado puede tener interés para el público en general por lo que valoro la posibilidad de darle más difusión al trabajo.







## Apéndice A

# Planificación del TFM

TAREAS	TIEMPO			
	Noviembre	Enero-Abril	Junio-Julio	Agosto
Elección del tema	Reuniones con el tutor			
Búsqueda de mapas	Obtención de datos geoespaciales			
Documentar		Lectura papers	Redacción de los capítulos	
SIG		Repaso de conceptos		
Web Semántica			Estudio de los conceptos de la Web Semántica	
Pruela de concepto		Creación de la ontología, consultas y visualización		
Anexos			Agregación de documentación a temas poco tratados	
Conclusiones			Realización de las conclusiones	
Repaso y entrega			Repaso para su entrega	
Reuniones	Durante todo el proceso he estado en contacto con el tutor del TFM o bien a través de tutorías o bien a través del correo electrónico			



## Apéndice B

# Problemas y Soluciones

*Nota:* La prueba se ha realizado en un macOS 10.14

Como hemos venido diciendo a lo largo de los capítulos, no ha sido posible hacer uso del lenguaje estándar de consulta para información geoespacial GeoSPARQL en Protegé 5.5.0. El problema ha surgido cuando en la realización de las consultas geoespaciales no se detectaban las URIs ni lo estándares de GeoSPARQL. No obstante, previamente a haber elegido como alternativa GraphDB, se han revisado algunas vías para intentar su instalación y uso en Protegé. Por tanto, se ha estado buscando si había algún plugin con dicha librería que funcionara.

Lo primero que se ha encontrado ha sido “*Protegé 4 Plugin for Oracle Database*”<sup>1</sup> ([https://protegewiki.stanford.edu/wiki/Protege\\_4\\_Plugin\\_for\\_Oracle\\_Database](https://protegewiki.stanford.edu/wiki/Protege_4_Plugin_for_Oracle_Database)), plugin que supuestamente trae consigo la posibilidad de hacer consultas con GeoSPARQL, aunque sólo permite la versión 5.2 de Protegé y no la 5.5, como hemos probado de primeras. Así que nos hemos descargado la nueva versión de Protegé y el plugin de la siguiente página<sup>2</sup> en donde se ha seleccionado la opción *Download Oracle Database 19c, 18c, and 12c Support for Apache Jena 3.1, Apache Jena Fuseki 2.4, and Protégé Desktop 5.2*. Al descomprimir la descarga obtenemos un directorio (*Oracle19c\_Jena-3.1.0\_Build\_20190711*) que contiene dentro una carpeta llamada *protege\_plugin*, en la cual te indica los pasos para instalar dicho plugin. Sin embargo, no ha habido éxito al seguir dichos pasos y por si acaso, se ha probado tanto para Protegé 5.5, Protegé 5.2 y Protegé 4. Por otro lado, se han buscado en Internet las maneras de instalar un plugin en Protegé, y

---

<sup>1</sup>The Support for Apache Jena is an adapter that provides a feature rich Java-based interface to RDF Semantic Graph that implements the well-known Apache Jena Graph, Model, BulkUpdateHandler, and DatasetGraph APIs. It supports SPARQL 1.1 and Open GeoSpatial Consortium (OGC) GeoSPARQL queries.

<sup>2</sup><https://www.oracle.com/technetwork/database/options/spatialandgraph/downloads/index-156999.html>

todos indican la misma forma que viene en dicha documentación. De todas maneras se ha verificado que se haya instalado correctamente (figura B.1), no obstante sin éxito en su funcionamiento ya que seguía sin detectar los prefijos de GeoSPARQL (figura B.2).

Installed Plugins	Name/ID	Version	Qualifier
OBO Annotations Editor	0.4.0		
OntoGraf	2.0.3		
Ontology Debugger	0.2.1		
Ontop OBDA Protégé plugin	1.18.1		
Oracle Plugin	3.1.0	20190711	
OWL Code Generation Plug-in	2.0.0		
OWL Difference	6.0.2		

Figura B.1: Comprobación del Plugin Oracle Database instalado

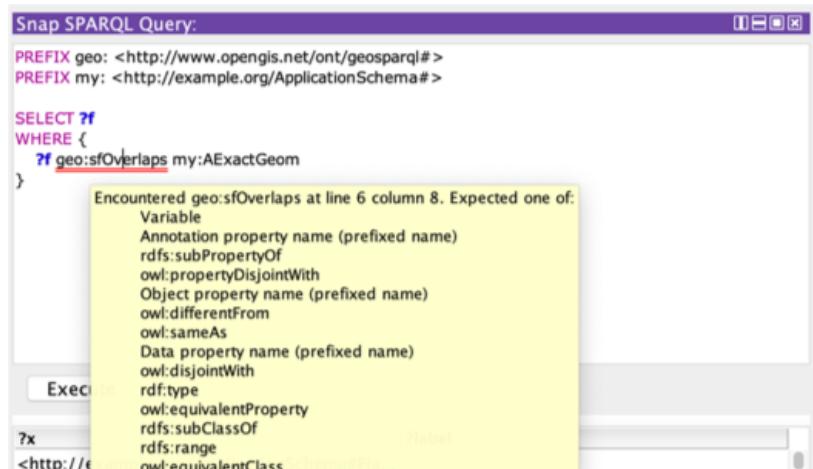
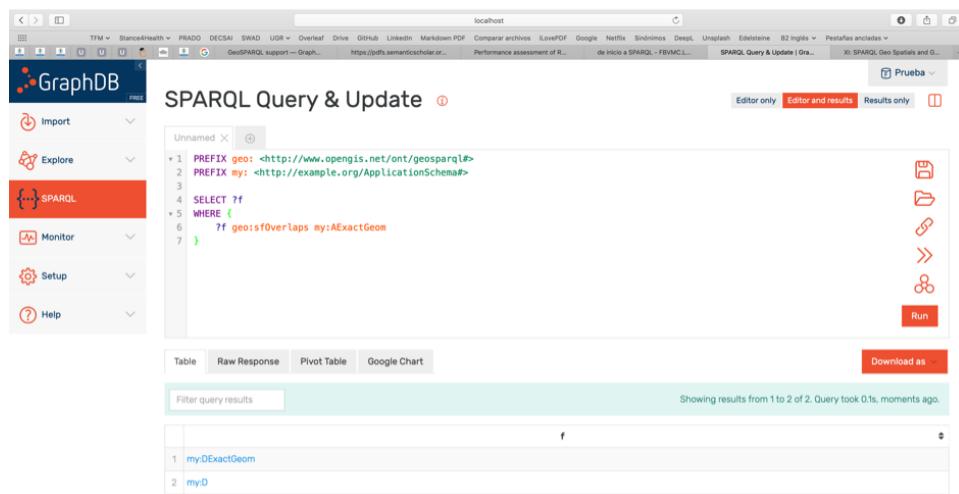


Figura B.2: Error de GeoSPARQL en Protegé

Por otra parte, para asegurar de que la consulta de GeoSPARQL en Protegé que se estaba realizando era correcta se ha usado un ejemplo proporcionado por el organismo de estándares OGC (<https://www.opengeospatial.org/standards/geosparql>) y al probarlo en otra herramienta se ha podido verificar que funcionaba. Pero al no conseguir éxito en Protegé, se han buscado otros softwares que permitieran hacer uso del lenguaje GeoSPARQL. De primeras se han buscado herramientas recomendadas para hacer uso de GeoSPARQL, y la que aparecía con más frecuencia era **GraphDB** (<https://www.ontotext.com/products/graphdb/#Try-GraphDB>). Además, es una de las que aconseja la página de Wikipedia ([https://en.wikipedia.org/wiki/OGC\\_GeoSPARQL](https://en.wikipedia.org/wiki/OGC_GeoSPARQL)). Por eso mismo nos hemos descargado la versión gra-

tuita del programa e instalado (se necesita tener Java instalado con una determinada versión y requisitos especiales), al instalarse se abre en un servidor, por lo que todo se maneja desde el navegador y se ha vuelto a probar el ejemplo de antes (<http://graphdb.ontotext.com/documentation/standard/geosparql-support.html#plugin-control-predicates>) y con GraphDB sí que ha funcionado (figura B.3). En el Apéndice C es posible ver la instalación de este software.



The screenshot shows the GraphDB interface with the SPARQL Query & Update tab selected. A query is being run:

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX my: <http://example.org/ApplicationSchema#>

SELECT ?f
WHERE {
  ?f geo:sfOverlaps my:AExactGeom
}
```

The results table shows two rows:

	f
1	my:DExactGeom
2	my:O

Figura B.3: Comprobación del funcionamiento de GeoSPARQL en GraphDB

Se observa como en la figura B.3, dicho software nos permite hacer uso del lenguaje de consulta GeoSPARQL, así como una interfaz muy intuitiva y sencilla.



## Apéndice C

# Instalación de GraphDB

*Nota: La instalación y uso de GraphDB se ha hecho en macOS 10.14*

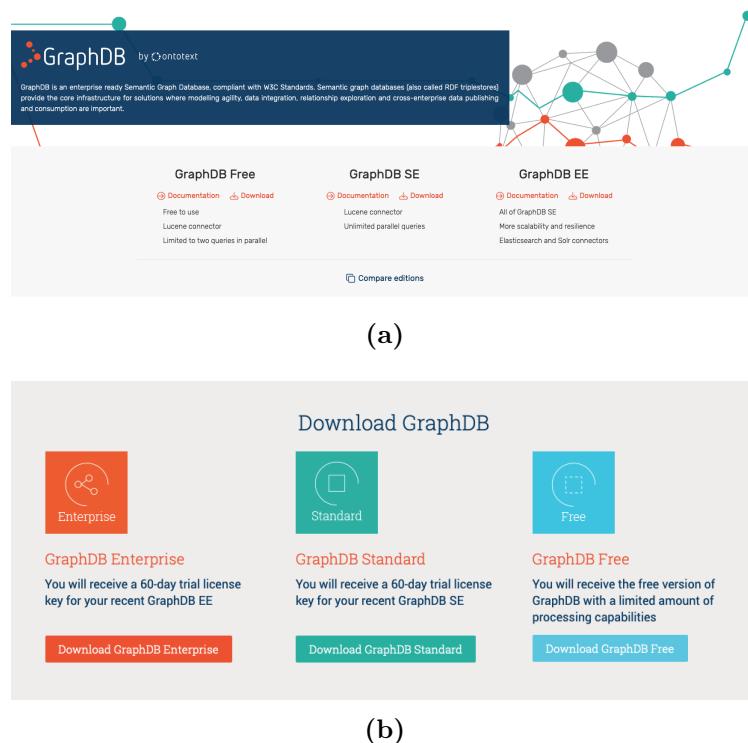


Figura C.1: Página para la descarga de GraphDB Free

**GraphDB** es una base de datos de grafos semánticos, que cumple con los estándares W3C. Proporciona la infraestructura central para soluciones donde la agilidad de modelado, la integración de datos, la exploración de

relaciones y la publicación y consumo de datos son importantes. Además, su uso nos ha permitido solventar los problemas ocasionados con Protegé para la realización de las consultas con información geográfica georreferenciada. Es por eso que es indispensable conocer las requisitos que necesita y los pasos para su correcta instalación. Para instalar GraphDB debemos acceder a la página <http://graphdb.ontotext.com> y descargarnos la última versión de *GraphDB Free*, tal como lo vemos en la figura C.1 (hemos descargado la versión gratuita). Previamente a la descarga de este software es necesario cumplimentar un breve formulario (figura C.2). Además, como GraphDB es una aplicación Java, requiere de Java 8 instalado, para instalarlo ve al siguiente enlace [https://java.com/en/download/help/download\\_options.xml](https://java.com/en/download/help/download_options.xml).

**Fully Featured Semantic Graph Database for Massive Data and Moderate Query Loads**

GraphDB Free edition is a fully functional member of the GraphDB family. Its only limitation is that it can handle no more than two queries in parallel. This edition is a perfect starting point for smart data proofs-of-concept and for projects that require an on-premise or embedded semantic graph database.

GraphDB Free is an ideal choice for non-critical deployments, but also for laying the developmental foundations of smart data systems at all levels of ambition.

First name\*

Last name\*

Email\*

Company name\*

Job category\*

Industry\*

Country\*

From time to time, we would like to contact you about our products and services, as well as share other relevant content that may be of interest to you. You can adjust or unsubscribe entirely from these communications at any time.

Receive timely updates related to the GraphDB product.

In order to provide the content you requested, we need to store and process your personal data.

I agree to allow Ontotext to store and process my personal data.\*

Ontotext is committed to protecting and respecting your privacy. We will not rent or sell your data. We'll use the information you provide to administer your account and communicate with you relevant information, among other things. For more information on our privacy practices, review our [Privacy Policy](#).

protección de reCAPTCHA

Request GraphDB Free

Figura C.2: Formulario para iniciar la descarga de GraphDB Free

Una vez terminada la descargada (nos hemos descargado la última versión), tendremos un zip descargado con el nombre de *graphdb-free-8.10.1*. Al descomprimir la carpeta, y teniendo Java instalado, es posible iniciar

GraphDB, para ello es necesario ejecutar el script `graphdb` en el directorio `bin` (figura C.3). Para acceder al *Workbench*, abrimos en el navegador la siguiente dirección <http://localhost:7200> (figura C.4).

```

1 # Obtener version de Java
2 $ java -version
3 java version "1.8.0_212"
4 Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
5 Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10)

```

```

1 # Dirigirse al directorio bin
2 $ cd graphdb-free-8.10.1/bin/
3
4 # Ejecutar el script graphdb
5 $ ./graphdb

```

```

MacBook-Pro-de-Gema:bin gema$ ./graphdb
[INFO ] 2019-09-04 15:57:16,841 [main | c.o.g.Config] Using 'file:/Users/gema/Documents/Programas/graphdb-free-8.10.1/conf/logback.xml' as logback's configuration file for graphdb
[INFO ] 2019-09-04 15:57:17,539 [main | c.o.g.s.GraphDB] Starting GraphDB in workbench mode.
[INFO ] 2019-09-04 15:57:24,968 [main | c.o.g.Config] GraphDB Home directory: /Users/gema/Documents/Programas/graphdb-free-8.10.1/conf
[INFO ] 2019-09-04 15:57:24,968 [main | c.o.g.Config] GraphDB Config directory: /Users/gema/Documents/Programas/graphdb-free-8.10.1/conf
[INFO ] 2019-09-04 15:57:24,968 [main | c.o.g.Config] GraphDB Data directory: /Users/gema/Documents/Programas/graphdb-free-8.10.1/data
[INFO ] 2019-09-04 15:57:24,969 [main | c.o.g.Config] GraphDB Work directory: /Users/gema/Documents/Programas/graphdb-free-8.10.1/work
[INFO ] 2019-09-04 15:57:24,969 [main | c.o.g.Config] GraphDB Logs directory: /Users/gema/Documents/Programas/graphdb-free-8.10.1/logs
[INFO ] 2019-09-04 15:57:30,179 [main | c.o.g.s.GraphDB] Started GraphDB in workbench mode at port 7200.

```

Figura C.3: Iniciando GraphDB en el terminal

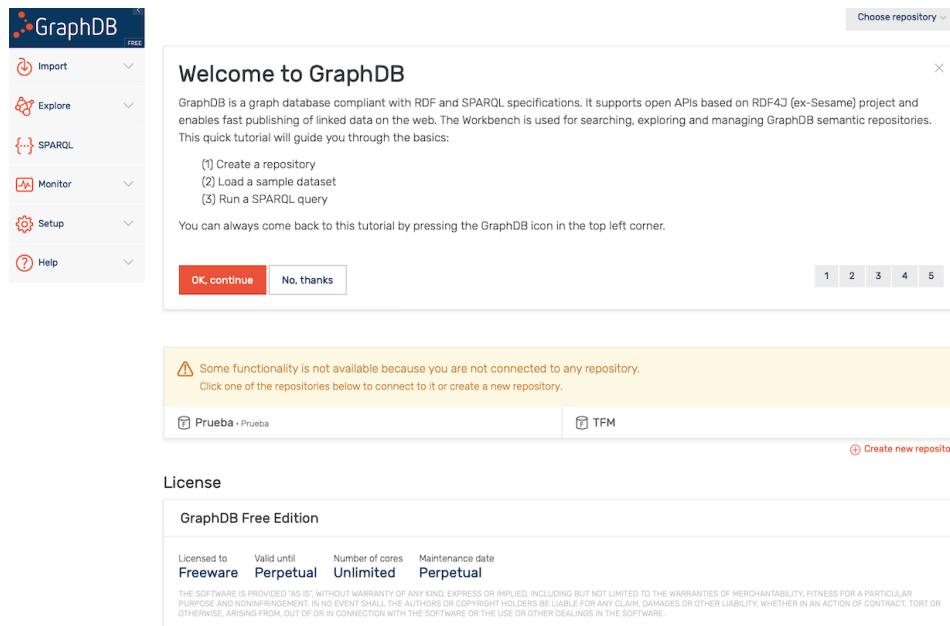


Figura C.4: Página de inicio de GraphDB en *Workbench*

Por último, y no menos importante, es necesario crear un repositorio (**Setup>Repositories**) y conectarse a él, para poder hacer todas las consultas que hemos visto anteriormente.

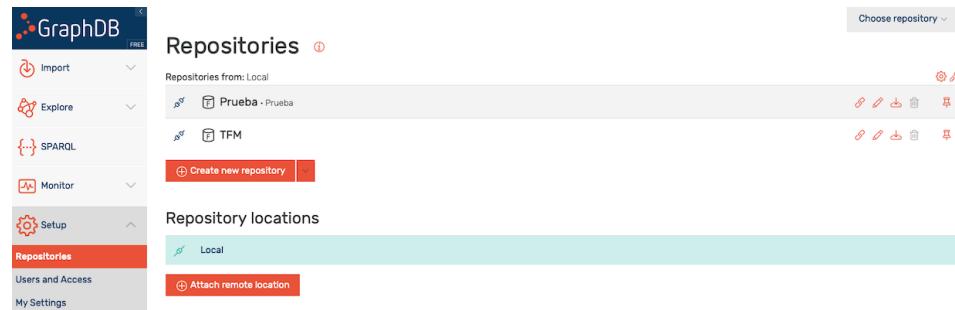


Figura C.5: Página para crear un nuevo repositorio en GraphDB

*Si se quiere saber más acerca de GraphDB, es posible acceder a su documentación en la dirección <http://graphdb.ontotext.com/documentation/free/>, la cual contiene información muy útil y bien explicada.*

# Bibliografía

- [1] S. Agudo. *Web semántica: ¿Qué es y qué relación guarda con Internet?* Disponible en: <https://rootear.com/web/que-es-la-web-semantica>, Acceso en: 2 de Septiembre 2019.
- [2] I. Alonso. *La Proyección UTM*. Disponible en: <http://www.cartesia.org/data/apuntes/cartografia/cartografia-utm.pdf>, Acceso en: 26 de Agosto 2019.
- [3] ArcGIS. *¿Qué son los datos ráster?* Disponible en: <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm>, Acceso en: 27 de Agosto 2019.
- [4] ArcGIS Pro. De contorno de entidad a polígono. How it was published. Disponible en: <https://pro.arcgis.com/es/pro-app/tool-reference/data-management/feature-envelope-to-polygon.htm>, Acceso en: 4 de Septiembre 2019.
- [5] J. F. Baculima and M. X. Cajamarca. *Diseño e implementación de un repositorio ecuatoriano de datos enlazados geoespaciales*. Disponible en: <http://dspace.ucuenca.edu.ec/handle/123456789/19876>, Acceso en: 13 de Agosto 2019.
- [6] T. Berners-Lee. *Uniform Resource Identifiers (URI): Generic Syntax*. Disponible en: <https://www.ietf.org/rfc/rfc3986.txt>, Acceso en: 28 de Agosto 2019.
- [7] Bibliopos. *La ontología y la Web Semántica: recomendaciones del W3C*. Disponible en: <http://www.bibliopos.es/Biblion-A2-Bibliografia-Documentacion/18ontologia-Web-Semantica.pdf>, Acceso en: 14 de Agosto 2019.
- [8] E. Chow. Geography 2.0—a mash-up perspective. pages 15–36, 05 2011.
- [9] Confederación de Empresarios de Andalucía (CEA). *Tipos de SIG*. Disponible en: [http://sig.cea.es/tipos\\_SIG](http://sig.cea.es/tipos_SIG), Acceso en: 27 de Agosto 2019.

- [10] H. Delgado. *Evolución de la Web 1.0, 2.0 y 3.0 - Diferencias y atributos.* Disponible en: <https://disenowebakus.net/etapas-de-transicion-de-la-web.php>, Acceso en: 2 de Septiembre 2019.
- [11] H. Delgado. *Web Semántica, definición, historia y características.* Disponible en: <https://disenowebakus.net/semanatica-web.php>, Acceso en: 13 de Agosto 2019.
- [12] M. Dávila. *SIG: La ciencia que nos permite entender cómo funciona el mundo.* ESRI España. Congreso Estatal RITSI IX 13 de Abril de 2018 en la Universidad Politécnica de Madrid.
- [13] El Mundo en HTML. *Lección 3: Estructura básica en HTML.* Disponible en: <https://fjph32html.wordpress.com/2015/03/08/leccion-3-estructura-basica-en-html/>, Acceso en: 14 de Agosto 2019.
- [14] ESRI España. *Descubre los GIS.* Disponible en: <http://www.esri.es/descubre-los-gis/>, Acceso en: 9 de Mayo 2018.
- [15] GraphDB. *About GraphDB.* Disponible en: <http://graphdb.ontotext.com/documentation/free/about-graphdb.html>, Acceso en: 2 de Septiembre 2019.
- [16] Grupo I+DeaSWeb (Investigación y Desarrollo para la Semántica Web). *Introducción a la Web Semántica: realidades y perspectivas.* Disponible en: [http://cmap.javeriana.edu.co/servlet/SBReadResourceServlet?rid=1264791321546\\_232600024\\_2804](http://cmap.javeriana.edu.co/servlet/SBReadResourceServlet?rid=1264791321546_232600024_2804), Acceso en: 20 de Agosto 2019.
- [17] Grupo TYC GIS. *Crear capas vectoriales a partir de archivos de texto en formato WKT.* Disponible en: <https://www.cursosgis.com/>, Acceso en: 21 de Agosto 2019.
- [18] IGN. *Información Geoespacial.* Disponible en: <http://www.ign.gob.ar/sig>, Acceso en: 2 de Septiembre 2019.
- [19] 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\\_Andalucia.pdf](https://www.juntadeandalucia.es/institutodeestadisticaycartografia/prodCartografia/bc/modelo/00_Modelo_Datos_Base_Cartografica_Andalucia.pdf), Acceso en: 29 de Agosto 2019.
- [20] Instituto de Investigación en Inteligencia Artificial (CSIC). *Diez años construyendo una web semántica.* Disponible en: [http://www.fgcsic.es/lychnos/es\\_es/articulos/construyendo\\_una\\_web\\_semantica](http://www.fgcsic.es/lychnos/es_es/articulos/construyendo_una_web_semantica), Acceso en: 19 de Agosto 2019.

- [21] Instituto Geográfico Nacional. *El sistema ETRS89.* Disponible en: [http://icvficheros.icv.gva.es/ICV/geova/erva/Utilidades/jornada\\_ETRS89/1\\_ANTECEDENTES IGN.pdf](http://icvficheros.icv.gva.es/ICV/geova/erva/Utilidades/jornada_ETRS89/1_ANTECEDENTES IGN.pdf), Acceso en: 2 de Septiembre 2019.
- [22] Junta de Andalucía. *Base Cartográfica de Andalucía. Descargas.* Disponible en: <https://www.juntadeandalucia.es/institutodeestadisticaycartografia/bcadescargas/>, Acceso en: 27 de Agosto 2019.
- [23] M. Koubarakis, M. Karpathiotakis, K. Kyriakos, C. Nikolaou, and M. Sioutis. Data models and query languages for linked geospatial data. pages 290–328, 09 2012.
- [24] M. J. Lamarca. *Hacia la Web Semántica.* Disponible en: [http://www.hipertexto.info/documentos/web\\_semantica.htm](http://www.hipertexto.info/documentos/web_semantica.htm), Acceso en: 2 de Septiembre 2019.
- [25] N. Messieh. *Top 7 Semantic Search Engines as an Alternative to Google.* Disponible en: <https://www.makeuseof.com/tag/top-7-semantic-search-engines-alternative-google-search/>, Acceso en: 20 de Agosto 2019.
- [26] A. Morales. *¿Estás utilizando un CAD cuando deberías estar usando un SIG? 10 diferencias entre SIG y CAD.* Disponible en: <https://mappinggis.com/2018/03/>, Acceso en: 27 de Agosto 2019.
- [27] OGC. *OGC GeoSPARQL – A Geographic Query Language for RDF Data.* Disponible en: <https://www.opengeospatial.org/standards/geosparql>, Acceso en: 29 de Agosto 2019.
- [28] OGC. *OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.* Disponible en: <https://www.opengeospatial.org/standards/sfa>, Acceso en: 4 de Septiembre 2019.
- [29] V. Olaya. *Sistemas de Información Geográfica.* CreateSpace Independent Publishing Platform, 2014. ISBN 1530295947.
- [30] Plataforma Tecnológica Española de la Carretera. *Aplicación de la Web Semántica en Tráfico.* Disponible en: [https://www.ptcarretera.es/wp-content/uploads/2018/07/02\\_2017\\_lisitt\\_def.pdf](https://www.ptcarretera.es/wp-content/uploads/2018/07/02_2017_lisitt_def.pdf), Acceso en: 28 de Agosto 2019.
- [31] Pontificia Universidad Católica de Chile. *La Web Semántica: Herramientas para la publicación y extracción efectiva de información en la Web.* Disponible en: <https://www.coursera.org/learn/web-semantica>, Acceso en: 14 de Agosto 2019.

- [32] Programación web, Drupal, Wordpress. *Función php para leer un archivo XML.* Disponible en: <http://historiasdeunaweb.blogspot.com/2015/01/funcion-php-para-leer-un-archivo-xml.html>, Acceso en: 2 de Septiembre 2019.
- [33] Protegé. *Protegé.* Disponible en: <https://protege.stanford.edu>, Acceso en: 2 de Septiembre 2019.
- [34] J. Samos. *Sistemas de Información Geográficos (SIG).* Asignatura impartida por el Dpto. de Lenguajes y Sistemas Informáticos en la ETSIIT (UGR). Curso 2017-2018.
- [35] J. Samos. *La Web Semántica.* Enero 2019. Apuntes de la asignatura de Web Semántica en el Máster en Desarrollo de Software, Curso 2018/2019.
- [36] Science for a changing world (USGS). Gis data layers visualization. How it was published, 7 1993. An optional note.
- [37] SELFHTML. *URN, URL, URI, IRI.* Disponible en: <https://wiki.selfhtml.org/wiki/Urn-url-uri-iri>, Acceso en: 2 de Septiembre 2019.
- [38] M. Smíd. *OnGIS: Ontology-Based Geospatial Data Integration and Retrieval.* Disponible en: <https://www.semanticscholar.org/paper/b9df99faafabe68b81c93dfa63ebffa9c8821e9d>, Acceso en: 28 de Agosto 2019.
- [39] J. Subirós and D. Varga. *Los Sistemas de Información Geográfica,* pages 357–376. 01 2008.
- [40] N. Turrado. *SEO Semántico: Entiende a los buscadores para que ellos te entiendan a ti.* Disponible en: <https://www.analistaseo.es/posicionamiento-buscadores/seo-semantico/>, Acceso en: 2 de Septiembre 2019.
- [41] Universidad Pedagógica y Tecnológica de Colombia. *Aprovechamiento de recursos mineros - energéticos y generación de un modelo de planeación para la prospección y explotación de minerales estratégicos del departamento de Boyacá .* Disponible en: [http://www.uptc.edu.co/export/sites/default/admon\\_grupo\\_bienes/contratacion/2014/inv\\_publicas/INVITACION\\_PUBLICA\\_No.\\_016\\_DE\\_2014/9.\\_ANEXO\\_8.\\_PROYECTO\\_DE\\_INVESTIGACIxN.pdf](http://www.uptc.edu.co/export/sites/default/admon_grupo_bienes/contratacion/2014/inv_publicas/INVITACION_PUBLICA_No._016_DE_2014/9._ANEXO_8._PROYECTO_DE_INVESTIGACIxN.pdf), Acceso en: 2 de Septiembre 2019.
- [42] M. Vallez. *La Web semántica y las tecnologías del lenguaje humano,* pages 155–180. Enero 2009.

- [43] W3C. *Lenguaje de Ontologías Web (OWL)*. Disponible en: <https://www.w3.org/2007/09/owl-Overview-es.html#s1.3>, Acceso en: 29 de Agosto 2019.
- [44] Wikimedia Commons. *WGS84 reference frame*. Disponible en: [https://commons.wikimedia.org/wiki/File:WGS\\_84\\_reference\\_frame.png](https://commons.wikimedia.org/wiki/File:WGS_84_reference_frame.png), Acceso en: 26 de Junio 2019.
- [45] C. Zhang, T. Zhao, and W. Li. *Geospatial Semantic Web*. Enero 2015. Springer.



