

Ontología de monumentos de la villa de Madrid

Autor: Víctor Iglesias Castán

Tutora: Claudia Reina Fajardo

Profesor: Joan Arnedo Moreno

Grado de Ingeniería Informática

Computación

Enero del 2025

Créditos/Copyright



Esta obra está sujeta a una licencia de [Atribución / Reconocimiento – NoComercial - SinDerivados 4.0 Internacional](#)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Ontología de monumentos de la villa de Madrid
Nombre del autor:	Víctor Iglesias Castán
Nombre del colaborador/a docente:	Claudia Reina Fajardo
Nombre del PRA:	Joan Arnedo Moreno
Fecha de entrega (mm/aaaa):	01/2025
Titulación o programa:	Grado de Ingeniería Informática
Área del Trabajo Final:	Web semántica
Idioma del trabajo:	Español
Palabras clave:	Ontología Monumentos

Resumen del Trabajo:

En todas las poblaciones los monumentos forma parte esencial del espacio público, pero la información disponible para transeúntes y turistas suele ser limitada, reduciéndose a inscripciones conmemorativas en el mejor de los casos. Este trabajo aborda esta carencia mediante la creación de una ontología de monumentos que aprovecha las posibilidades de la web semántica y la filosofía de los datos abiertos. Aunque ha sido concebida para incluir monumentos de cualquier ciudad del mundo, inicialmente se ha poblado con los datos abiertos del Ayuntamiento de Madrid.

Se presentan ejemplos de consultas SPARQL para enriquecer la ontología con información proveniente de Wikidata acerca de aspectos como datos biográficos el autor, el estilo arquitectónico o el personaje homenajeado. Por la naturaleza de las ontologías, tiene el potencial de ser integrada en ontologías de dominio más amplio, como turismo o urbanismo. Además, puede servir a modo de base de datos en una aplicación cliente / servidor, en la que mediante GeoSPARQL se proporciona a usuarios de dispositivos móviles la información de un monumento a partir de su geolocalización.

En definitiva, el proyecto facilita el acceso a información cultural local enriquecida con recursos globales mediante las tecnologías de la web semántica. Asimismo, busca fomentar el interés del ciudadano por su patrimonio local, promoviéndose así también la importancia de su conservación.

Abstract:

In all cities, monuments are an essential part of the public space, but the information available to passers-by and tourists is often limited, typically reduced to commemorative inscriptions at best. This work addresses this shortcoming by creating a monument ontology that leverages the possibilities of the semantic web and the open data philosophy. Although it has been conceived to include monuments from any city in the world, it has initially been populated with open data from the Madrid City Council.

Examples of SPARQL queries are presented to enrich the ontology with information from Wikidata, covering aspects such as the author's biographical data, architectural style or the depicted character. Due to the nature of the ontologies, it has the potential to be integrated in ontologies of a wider domain, such as tourism or urbanism. Furthermore, it can serve as a database for a client / server application, where GeoSPARQL is used to provide mobile device users with information about a monument based on their geolocation.

In conclusion, this project facilitates access to local cultural information enriched with global resources through semantic web technologies. Additionally, it aims to spark citizens' interest in their local heritage, thus promoting the importance of its conservation.

Resumen

En todas las poblaciones los monumentos forma parte esencial del espacio público, pero la información disponible para transeúntes y turistas suele ser limitada, reduciéndose a inscripciones conmemorativas en el mejor de los casos. Este trabajo aborda esta carencia mediante la creación de una ontología de monumentos que aprovecha las posibilidades de la web semántica y la filosofía de los datos abiertos. Aunque ha sido concebida para incluir monumentos de cualquier ciudad del mundo, inicialmente se ha poblado con los datos abiertos del Ayuntamiento de Madrid.

Se presentan ejemplos de consultas SPARQL para enriquecer la ontología con información proveniente de Wikidata acerca de aspectos como datos biográficos el autor, el estilo arquitectónico o el personaje homenajeado. Por la naturaleza de las ontologías, tiene el potencial de ser integrada en ontologías de dominio más amplio, como turismo o urbanismo. Además, puede servir a modo de base de datos en una aplicación cliente / servidor, en la que mediante GeoSPARQL se proporciona a usuarios de dispositivos móviles la información de un monumento a partir de su geolocalización.

En definitiva, el proyecto facilita el acceso a información cultural local enriquecida con recursos globales mediante las tecnologías de la web semántica. Asimismo, busca fomentar el interés del ciudadano por su patrimonio local, promoviéndose así también la importancia de su conservación.

Palabras clave

Ontología, Web semántica, Monumentos, Datos abiertos, Linked data, Trabajo de Final de Grado

Sumario

1 Introducción.....	10
1.1 Prefacio.....	10
1.2 Descripción.....	11
1.3 Objetivos generales.....	12
1.3.1 Objetivos principales.....	12
1.3.2 Objetivos secundarios.....	13
1.4 Metodología y proceso de trabajo.....	13
1.4.1 Requisitos.....	14
1.4.2 Dominio, alcance, usuarios y usos.....	15
1.5 Planificación.....	16
1.5.1 Tareas.....	16
1.5.2 Hitos.....	17
1.5.3 Diagrama de Gantt.....	17
1.6 Presupuesto.....	18
1.6.1 Equipamiento técnico.....	19
1.7 Estructura del resto del documento.....	20
2 Análisis de mercado.....	20
2.1 Web semántica.....	20
2.2 Ontología.....	21
2.3 XML.....	22
2.4 RDF y RDF Schema.....	23
2.5 OWL.....	24
2.6 SPARQL.....	24
2.7 Datos enlazados.....	25
2.8 Datos abiertos y conocimiento abierto.....	25
3 Propuesta.....	25
4 Diseño.....	26
4.1 Integración.....	26
4.2 Especificación.....	27
4.3 Conceptualización.....	29
4.4 Formalización.....	31
4.5 Implementación.....	33
4.5.1 Propiedades de datos.....	33
4.5.2 Propiedades de objetos.....	36
4.5.3 Entidades.....	38
4.6 Evaluación.....	45
5 Implementación.....	45
5.1 Requisitos de instalación.....	46
5.2 Instrucciones de instalación.....	46
5.2.1 Protégé.....	46
5.2.2 Apache Jena Fuseki.....	46
5.2.3 Virtuoso Open-Source Edition.....	47

6 Demostración.....	47
6.1 Consultas a la ontología.....	47
6.2 Consultas federadas.....	51
6.3 Consultas geoespaciales.....	53
7 Conclusiones y líneas de futuro.....	55
7.1 Conclusiones.....	55
7.2 Líneas de futuro.....	56
Bibliografía.....	58

Figuras y tablas

Índice de figuras

Figura 1: Diagrama de Gantt.....	17
Figura 2: Pila tecnológica de la web semántica.....	21
Figura 3: Tripleta como grafo dirigido.....	23
Figura 4: Diseño UML de la ontología.....	32
Figura 5: Grafo de las clases y sus relaciones.....	33
Figura 6: Propiedades de datos.....	34
Figura 7: Propiedad de datos para las coordenadas.....	35
Figura 8: Propiedad de datos para el nombre.....	35
Figura 9: Propiedades de objetos.....	36
Figura 10: Propiedad de datos hasAdministrativeDivision.....	37
Figura 11: Propiedad de datos isAdministrativeDivisionOf.....	38
Figura 12: Script de importación inicial.....	39
Figura 13: Métricas de la ontología después de cargar las entidades.....	40
Figura 14: Entidad Escultura Abstracta.....	40
Figura 15: Entidad Autor.....	41
Figura 16: Entidad Distrito.....	41
Figura 17: Entidad Fuente.....	42
Figura 18: Entidad Puerta.....	42
Figura 19: Entidad Elemento Conmemorativo.....	43
Figura 20: Entidad Ornamentación.....	43
Figura 21: Entidad Grupo Escultórico.....	43
Figura 22: Entidad Estatua.....	44
Figura 23: Entidad Edificación Singular.....	44
Figura 24: Entidad Barrio.....	44
Figura 25: Resultado del razonador Pellet.....	45
Figura 26: Monumento por su nombre.....	48
Figura 27: Monumento por nombre aproximado.....	48
Figura 28: Todos los monumentos de un distrito.....	49
Figura 29: Todos los monumentos de un barrio.....	50

Figure 30: Monumentos construidos en un año aproximado.....	50
Figura 31: Todos los monumentos de un/a autor/a.....	51
Figura 32: Consulta que complementa la información de la ontología con la de Wikidata.....	52
Figura 33: Resultado consulta federada.....	53
Figura 34: Monumento por ubicación mediante st_distance.....	54
Figura 35: Monumento por ubicación mediante st_distance con radio ampliado.....	55

Índice de tablas

Tabla 1: Tareas por hito.....	16
Tabla 2: Cronología de los hitos.....	17
Table 3: Horas por tarea.....	19
Tabla 4: Campos del fichero CSV.....	28
Tabla 5: Campos de Wikidata interesantes para incorporar.....	29

1 Introducción

1.1 Prefacio

El presente documento es la memoria de mi trabajo de final de grado de ingeniería informática. Éste trabajo pretende explorar las posibilidades que ofrece actualmente la web semántica, sus herramientas y los datos abiertos. Para ello, se propone crear una ontología que pueda tener su interés y utilidad dentro del mundo de las ontologías, así como ser una pieza clave dentro de un entorno de aplicación cliente / servidor que empodere a las personas.

A partir de los datos abiertos acerca de los monumentos localizados en la villa de Madrid, se definirán las clases, jerarquías, propiedades y restricciones de una ontología. Este modelo no se diseñará de forma que quede circunscrito a los datos de Madrid, sino que será generalizado para que pueda contener monumentos de todas las ciudades del mundo. A continuación, la ontología se poblará con los datos de la capital.

Finalmente, se pondrán a prueba las posibilidades que ofrece la web semántica mediante la integración de la información contenida en la Wikidata acerca de monumentos. La Wikidata contiene información estructurada que mediante consultas semánticas puede relacionarse con los contenidos de la ontología, de tal manera que se enriquece la información disponible.

1.2 Descripción

En los desplazamientos por las ciudades, ya sea en la localidad de residencia del transeúnte o en una visita debida a diferentes motivos como turismo o trabajo, es frecuente encontrarse con monumentos. El espacio público no es un museo, por lo que toda la información acerca del monumento está contenida en una placa conmemorativa o inscripción, por lo que el observador carece de información más detallada, desde su contexto histórico a una breve biografía de su autor, o incluso a quién o qué se está homenajeando. De hecho, no es infrecuente que tan solo se tenga un conocimiento superficial incluso de monumentos por los que se pasa casi a diario.

Actualmente, cada vez más ayuntamientos se suman a la filosofía de datos abiertos, y comparten de forma abierta toda clase de datos acerca de su localidad, como monumentos, jardines y demás puntos de interés. Estos datos permitirían poblar una ontología de monumentos. Gracias a la omnipresencia del móvil y otros dispositivos portátiles, un transeúnte podría acceder a la información contenida en la ontología acerca del monumento que está viendo en ese momento mediante una aplicación móvil, ya sea a partir de las coordenadas de su posición actual o buscándolo por su nombre.

Además, la información contenida en dicha ontología podría ampliarse más allá de los datos suministrados por los entes públicos. Por un lado, la Wikipedia es el recurso colaborativo más grande construido por la humanidad y en ella se encuentra información detallada acerca de muchos monumentos de ciudades alrededor del mundo. Por otro lado, tenemos la Wikidata, un proyecto que sigue el paradigma de los datos enlazados. En su creación se realizó una gran importación de datos desde la Wikipedia y a partir de ahí siguió incorporándose más información. La estructura de la información que contiene está pensada para que pueda ser consumida por aplicaciones y servicios de la web semántica, así como sistemas de inteligencia artificial.

Finalmente, el hecho de tener toda esta información contenida en una ontología, la hace accesible a otras ontologías, programas y sistemas informáticos, por ejemplo relacionados con el urbanismo, el turismo, etc. Esto no es casual, pues las ontologías son la principal herramienta de la web semántica para conseguir su objetivo: que las aplicaciones puedan interpretar, compartir y reutilizar la información.

El presente trabajo no se centra en la realización de la aplicación móvil (lado cliente) ni de la aplicación del lado servidor, pero sí incluye la ontología a la que deberá consultar la aplicación servidor para satisfacer las peticiones del cliente, el poblado de esta con datos provenientes del Portal de datos abiertos del Ayuntamiento de Madrid, así como demostraciones de consultas SPARQL, tanto a la ontología como su integración con la Wikidata para ampliar la información que contiene acerca de un monumento.

1.3 Objetivos generales

1.3.1 Objetivos principales

Los objetivos principales son dos. En primer lugar, crear una ontología de monumentos. Esta se diseñará para poder contener información de monumentos de cualquier ciudad del mundo, aunque por las restricciones de tiempo, para este proyecto se poblará sólo con los datos abiertos que proporciona Madrid. En segundo lugar, enriquecer la información contenida en la ontología con la información de la Wikidata mediante consultas semánticas en SPARQL.

El objetivo desde la perspectiva del usuario será poder obtener información acerca del monumento que está viendo: historia, descripción de los elementos, biografía de su autor, etc. Este objetivo se conseguirá mediante una aplicación para dispositivos móviles cuyo desarrollo no está en el alcance del presente trabajo. Esta aplicación del lado cliente consultará a una aplicación del lado servidor que a su vez obtendrá la información de la ontología.

En cuanto a los objetivos personales, por un lado y a pesar de llevar muchos años trabajando como programador, la mayoría dedicados al desarrollo web, me llama la atención la web semántica. En consecuencia, con este trabajo puedo mejorar mis conocimientos acerca de la web semántica y las tecnologías relacionadas: las ontologías, el estándar de notación RDF con sus triplets y su representación en forma de grafo, el estándar OWL, el lenguaje SPARQL para hacer consultas sobre información basada en RDF y el concepto de datos abiertos.

Por otro lado, me llamó mucho la atención el proyecto de la Wikipedia desde que lo conocí hace muchos años, pues me pareció una forma excelente de hacer llegar conocimientos a cualquier ser humano con acceso a Internet, pudiendo así vencer las limitaciones económicas y físicas que padezca la persona en su entorno. Lamentablemente, en la última década los usuarios cada vez pasan más tiempo en plataformas cuyos contenidos mayoritarios están pensados sólo para entretenimiento, por ejemplo TikTok, Instagram o Facebook, de manera que parte de Internet se ha convertido en una nueva “caja tonta”. Frente a esta realidad, pueden explorarse los proyectos DBpedia y Wikidata como herramientas para enriquecer otras aplicaciones, colaborando así a que Internet mantenga el potencial que se le vio desde sus inicios para mejorar la cultura y la formación de las personas.

1.3.2 Objetivos secundarios

Esta ontología, a su vez, puede ser integrada en ontologías de dominio más general, por ejemplo de turismo, urbanismo o geografía. Esta posibilidad no es para nada algo secundario sino que la interoperabilidad es una de las aplicaciones de la web semántica. La única razón por la que se ha listado como objetivo secundario es porque no será abordado en este trabajo, si bien será posible pues la ontología será públicamente accesible y con una licencia que lo permite.

Finalmente, hay dos objetivos más que podrían lograrse si este proyecto avanza más allá del presente trabajo y llega a los usuarios. En primer lugar, el hecho de que las personas tengan un mayor conocimiento del arte y la cultura de su entorno favorecerá que haya mas atención en el cuidado y protección del patrimonio de nuestras ciudades, más allá de las localizaciones que atraen el turismo de masas. En segundo lugar, este sistema es más eficiente desde el punto de vista energético y de consumo de materiales que si los usuarios hacen las consultas en Google, no se diga si las hacen en una IA entrenada con cantidades ingentes de datos sin clasificar provenientes de Internet.

1.4 Metodología y proceso de trabajo

A la hora de crear una ontología existen diversas metodologías. Algunas de ellas resultado de la experiencia obtenida en el desarrollo de un proyecto concreto y que pueden adaptarse mejor a proyectos de características similares; otras con un enfoque más generalista. Algunas metodologías son: Ontology Development 101, Methontology, On-To-Knowledge, Text2Onto, SENSUS-Based y Grüninger y Fox. Algo frecuente en estas metodologías es que son un proceso iterativo, característica que no por casualidad es compartida con el desarrollo de software.

Probablemente, la que mejor encaja con este proyecto es Methontology [1], propuesta por el Laboratorio de Inteligencia Artificial de la Universidad Politécnica de Madrid. Esta metodología propone un ciclo de vida basado en iteraciones, donde cada iteración consiste en el mismo conjunto de actividades o pasos. Las actividades están inspiradas en la norma IEEE 1074 para el proceso de desarrollo de software. El fruto de cada iteración es un prototipo de la ontología.

Las razones por las que encaja mejor son dos:

1. El desarrollo de esta ontología se ve favorecido por un proceso iterativo. Por ejemplo, al crear las instancias mediante la carga de datos del fichero CSV, podrían descubrirse deficiencias en la ontología que deberán subsanarse.
2. Esta ontología es ampliable mediante la incorporación de nuevas fuentes de datos abiertos. Nuevas propiedades de los mismos podrían requerir cambios en la ontología, por lo que es correcto verla como un prototipo que evoluciona a medida que se enriquece con nuevas

fuentes de datos. En este sentido, la ontología resultante de este proyecto hay que verla como un prototipo.

Las actividades o pasos son:

- **Especificación:** Define el alcance de la ontología. En esta fase se debe decidir y delimitar:
 - Los objetivos de su creación.
 - El dominio de actuación.
 - Por quién y para qué será utilizada.
- **Conceptualización:** Se debe organizar y estructurar el conocimiento adquirido mediante la creación de un glosario de términos que pertenezcan al dominio, definirlos y estructurarlos.
- **Formalización:** Del modelo conceptual de la actividad anterior se pasa a un modelo formal, más riguroso. De las diferentes herramientas para lograr este objetivo, para este trabajo se empleará el diseño UML.
- **Integración:** Debe contemplarse si existen y pueden integrarse otras ontologías para reducir costes y evitar duplicidades.
- **Implementación:** Se convierte el modelo formal en un modelo computable. En este trabajo se empleará *Protégé* para lograrlo. Este paso y el de formalización se retroalimentan en un proceso iterativo, pues durante la implementación se van descubriendo matices que implican cambios en la formalización.
- **Evaluación:** Debe verificarse que la ontología cumpla con los objetivos planteados en la especificación.
- **Documentación:** Para que la ontología se pueda compartir y reutilizar, se debe documentar. Esta memoria será la documentación de la ontología.
- **Mantenimiento:** Consiste en establecer cuáles líneas de mantenimiento y modificación se llevarán a cabo. Esta actividad está fuera del alcance del presente trabajo, corresponderá a quienes reutilicen o amplíen la ontología en el futuro.

1.4.1 Requisitos

El producto final consta de la ontología como parte fundamental. En segundo lugar, consta de las consultas en SPARQL que sirven como prueba de concepto del potencial de las ontologías y los datos enlazados (*linked data*).

Los requisitos de la ontología son los siguientes:

-
- Contener información acerca de los monumentos de Madrid provenientes de datos abiertos: nombre, descripción, año, dirección, barrio, distrito y enlace a la ficha del monumento en el sitio web “Patrimonio cultural y paisaje de Madrid”.
 - Ser ampliable a monumentos de otras ciudades.
 - Los monumentos estarán clasificados por tipos.

Los requisitos que deben cumplir las consultas son los siguientes:

- Localizar el monumento cercano a una geolocalización y proporcionar la información disponible en la ontología de ese monumento.
- Encontrar monumentos por nombre, barrio, distrito, año, autor y ciudad.
- Ampliar la información disponible mediante consultas cruzadas con el punto de acceso SPARQL de la Wikidata, de manera que se pueda incluir su historia, descripción de sus elementos, biografía del autor o autores, etc.

1.4.2 Dominio, alcance, usuarios y usos

El dominio del proyecto son los monumentos localizados en la villa de Madrid. El alcance del proyecto es la creación de la ontología de monumentos de Madrid a partir de los datos abiertos disponibles. Este alcance implica una primera iteración del producto, aunque tal vez requiera retornar a una actividad anterior cuando se descubra que son necesarios cambios. El alcance no incluye las nuevas iteraciones que requieran la incorporación de más datos de otras ciudades: se limita estrictamente a la capital.

El perfil de los usuarios finales del producto desarrollado será técnico, ya sea para ampliar la ontología con más instancias o integrarla en otra ontología, o ya sea para crear las aplicaciones lado cliente y servidor, en ambos casos se requieren personas de suficiente perfil técnico.

Los posibles usos del producto final son tres:

- Ampliar la ontología con monumentos de más ciudades.
- Integrarla en ontologías de propósito más general.
- Crear una aplicación que permita a usuarios finales obtener información desde su dispositivo móvil del monumento que están contemplando.

1.5 Planificación

La planificación del proyecto se estructura en hitos, los cuales derivan de las entregas de las PEC. Por cada hito se han definido sus tareas. La visualización en forma gráfica del espacio temporal de las tareas se realiza mediante el diagrama de Gantt.

1.5.1 Tareas

A continuación, se muestran las tareas requeridas de cada PEC o hito:

PEC 1
Lectura de los materiales de las asignaturas TFG y Web semántica
Informarse acerca de la web semántica, los datos abiertos, las ontologías y tecnologías relacionadas
Elaboración de una propuesta inicial
PEC 2
Análisis de la información acerca de monumentos disponible en ficheros open data
Análisis de la información acerca de monumentos contenida en Wikidata y en DBpedia
Elaboración del estado del arte
Crear proyecto en Github
PEC 3
Describir mediante UML el modelo que contiene toda la información necesaria
Creación de la ontología con Protégé a partir del diseño UML
Programación del script de importación de datos
Creación de consultas que demuestren las posibilidades de la ontología y su integración con Wikidata
PEC 4
Pulir la versión definitiva de la ontología
Versión definitiva en Github: fichero de la ontología y fichero README.md
Pulir la versión definitiva de la memoria
Creación del vídeo de defensa del TFG
Creación del vídeo tráiler del proyecto
PEC 5
Preparación de la defensa del trabajo
Defensa síncrona del trabajo

Tabla 1: Tareas por hito

1.5.2 Hitos

Los hitos coinciden con las entregas parciales de las PEC:

Tarea	Fecha inicio	Fecha fin
PEC 1	25/09/2024	20/10/2024
PEC 2	21/10/2024	17/11/2024
PEC 3	18/11/2024	08/12/2024
PEC 4	09/12/2024	12/01/2025
PEC 5	13/01/2025	?

Tabla 2: Cronología de los hitos

1.5.3 Diagrama de Gantt

A partir de las tareas e hitos definidos previamente, se ha elaborado el siguiente diagrama de Gantt:

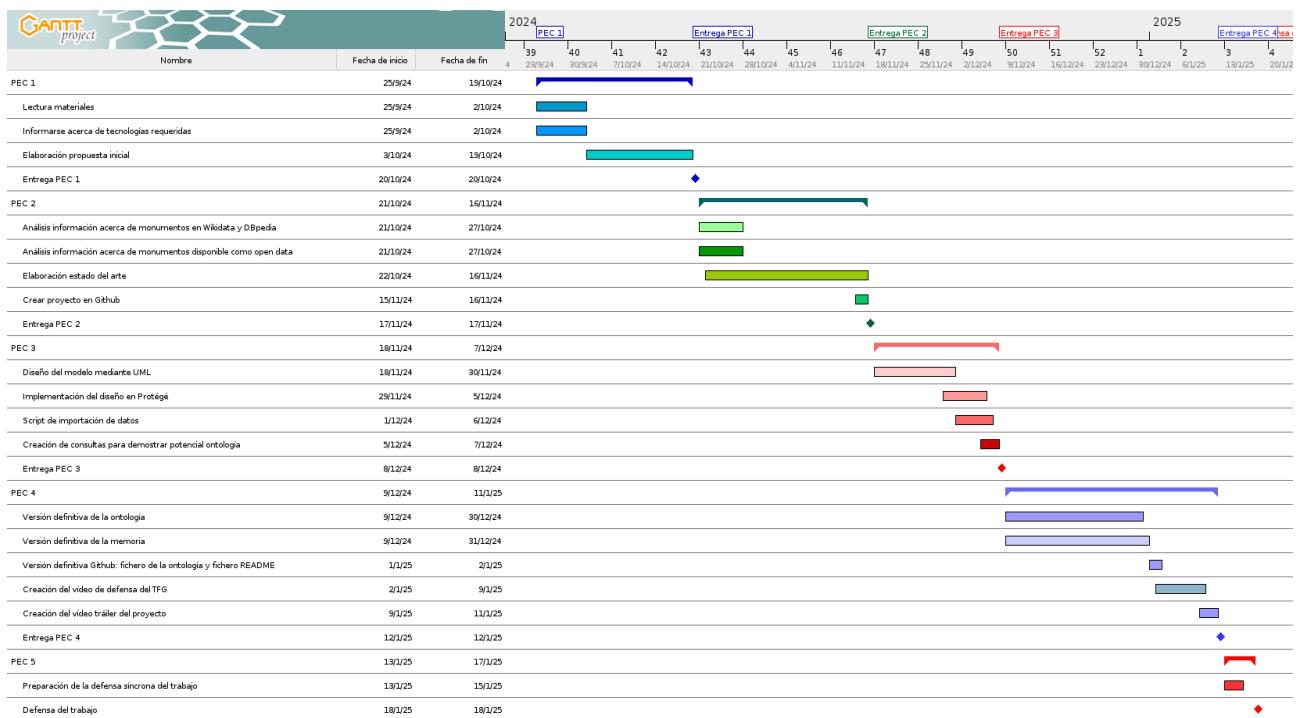


Figura 1: Diagrama de Gantt

Como todavía se desconoce la fecha exacta de la defensa del trabajo, se ha planteado la fecha del 18 de enero de 2025.

La planificación (tareas e hitos) se ha hecho acorde con los enunciados de cada PEC: estos especifican qué partes del proyecto se deben realizar. Ahora bien, existe cierta coincidencia entre la planificación dada y la metodología Methontology:

- La PEC 1 coincide con la actividad de especificación, así como las 2 primeras tareas de la PEC 2.
- La tarea del estado del arte coincide con la actividad de integración, pues tal vez se encuentre una ontología que pueda reutilizarse.
- En la PEC 3 se llevan a cabo las actividades de formalización e implementación. Las consultas que se crearán también servirán para realizar la actividad de evaluación de la ontología.
- En la PEC 4 aparecen las tareas de creación de las versiones definitivas de la ontología y la memoria, las cuales pueden verse como otra iteración después de la evaluación, pues según se haya visto en las consultas, podría ser necesario regresar a la actividad de conceptualización y avanzar hasta la de evaluación.

Un posible riesgo está en cumplir con los plazos de la PEC 3, pues a efectos prácticos concentra prácticamente todo el desarrollo del producto y tan sólo se dispone de 20 días.

1.6 Presupuesto

La limitación en este proyecto es de tiempo, éste viene dado por el total de créditos ECTS de la asignatura: 12. En la UOC, un crédito ECTS equivale a 25 horas de trabajo del estudiante [2], por lo que se dispone de 300 horas para este proyecto. Además, las fechas de entrega de los hitos también vienen estipuladas.

El periodo desde el inicio del semestre hasta el hito de la PEC 4 es de 106 días. De ahí a la PEC 5 se han puesto 5 días, si bien se desconoce la fecha exacta de la defensa del trabajo, es un tiempo razonable para la defensa del trabajo y la preparación de esta. Por lo tanto, puede considerarse que se disponen de 111 días.

El equipo humano del proyecto está compuesto por una única persona, por lo que las tareas del proyecto se desarrollarán en función de mi disponibilidad. Actualmente trabajo a jornada completa 40 horas semanales, además de cursar otras 2 asignaturas, pero dispongo de días de vacaciones para avanzar en el proyecto. Aproximadamente son 2,7 horas por día, unas 19 a la semana.

La siguiente tabla desglosa las tareas del proyecto y sus horas. A partir de aquí se podría realizar un presupuesto asignando a cada tipo de tarea un precio por hora.

Hito	Actividad	Horas
PEC 1		
	Lectura de los materiales	10
	Informarse acerca de tecnologías requeridas	11
	Elaboración propuesta inicial	46
PEC 2		
	Análisis información acerca de monumentos en Wikidata y DBpedia	9
	Análisis información acerca de monumentos disponible como open data	10
	Elaboración estado del arte	56
PEC 3		
	Diseño del modelo mediante UML	30
	Implementación del diseño en Protégé	16
	Script de importación de datos	6
	Creación de consultas para demostrar potencial ontología	10
PEC 4		
	Versión definitiva de la ontología	28
	Versión definitiva de la memoria	33
	Versión definitiva Github: fichero de la ontología y fichero README	2
	Creación del vídeo de defensa del TFG	21
	Creación del vídeo tráiler del proyecto	8
PEC 5		
	Preparación de la defensa síncrona del trabajo	5

Table 3: Horas por tarea

1.6.1 Equipamiento técnico

El equipamiento técnico se divide en software y hardware. En lo que respecta al último, basta con una estación de trabajo sin especiales prestaciones. Concretamente, se dispone de un equipo con CPU Intel Core i5-4460 a 3,2 GHz, 16 GB de RAM y 1TB de disco duro.

Respecto al software, el sistema operativo es Linux, si bien el software requerido es agnóstico del sistema operativo. El diseño UML de la ontología se realizará con la herramienta en línea “draw.io” y se implementará con el programa Protégé, posiblemente se deba añadir una extensión para poder trabajar con GeoSPARQL. No obstante, Protégé no tiene la capacidad de acceder a servicios remotos para realizar consultas distribuidas que integren la información contenida en la ontología con la de DBpedia o Wikidata, por lo que se empleará Apache Jena Fuseki como servidor SPARQL. GeoSPARQL también puede ser integrado en Fuseki, se analizará esta opción. Si bien Virtuoso podría funcionar mejor cuando hay un gran volumen de datos, Fuseki será suficiente para la prueba de concepto del presente trabajo.

1.7 Estructura del resto del documento

Después de este capítulo introductorio, la memoria se divide en los siguientes capítulos:

- Análisis de mercado: Se explica en qué consiste la web semántica y las diferentes tecnologías y estándares relacionados.
- Propuesta: Se listan los elementos que se esperan obtener del presente trabajo.
- Diseño: Se expone el proceso de diseño de la ontología. A continuación se documenta su implementación en Protégé.
- Implementación: En primer lugar se exponen las razones y dificultades que han llevado a la decisión de emplear un software u otro y los requisitos para su instalación. A continuación, se explica cómo instalar los diferentes programas requeridos.
- Demostración: Se demuestra cómo se han alcanzado los requisitos formulados para este proyecto mediante consultas SPARQL.
- Conclusiones y líneas de futuro: La memoria finaliza con las conclusiones extraídas de este proyecto y cómo podría ser ampliado en el futuro desde diferentes perspectivas.

2 Análisis de mercado

En este apartado se explica el concepto de web semántica, así como las tecnologías y estándares relacionados en los que se basa el presente trabajo.

2.1 Web semántica

La web actual es la 2.0. En sus páginas se presenta la información de forma comprensible para los humanos. La web semántica no es una nueva web, una especie de versión 3.0, sino una extensión de la actual que la complementa, cuyo objetivo final es la interrelación de los recursos disponibles en la web, es decir, debe convertir la web en una red de interconexión de datos que puedan ser comprendidos por aplicaciones.

Actualmente, la información de la web no está estructurada, Además, contiene gran cantidad de texto en lenguaje natural, siempre ambiguo. En consecuencia, para que la web semántica sea una realidad, debe lograrse describir el significado, es decir, la semántica.

Por otro lado, además de haberse logrado dotar de significado a la web, también se debe disponer de aplicaciones y tecnologías que empleen la información semántica para ofrecer nuevos servicios. Este software es lo que el W3C denomina servicios web y agentes.

Si se logran ambos objetivos, sería realidad el siguiente ejemplo de una aplicación: el buscador inteligente. Mediante el mismo, un usuario formula que desea irse de vacaciones a una localidad el próximo fin de semana. El buscador sabrá cuáles son las fechas del próximo fin de semana y sugerirá unos vuelos. Una vez el usuario selecciona los que considera mejores, el buscador se encarga automáticamente de las reservas. A continuación, el buscador sugiere diferentes alojamientos y se encarga también de reservar los seleccionados.

En la siguiente figura se muestra la pila de la web semántica, tal y como la presentó Tim Berners-Lee en la presentación “Semantic Web – XML2000” [3]:

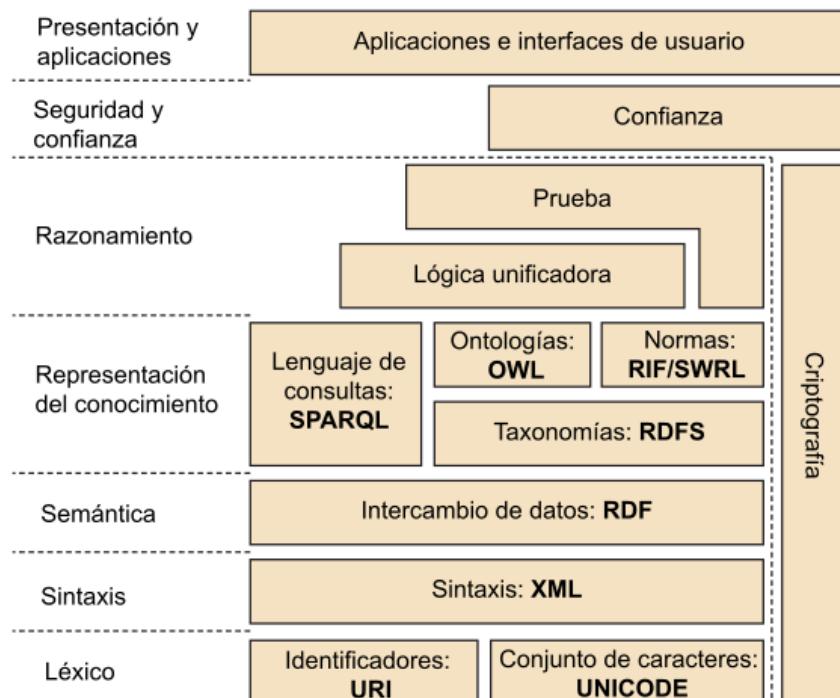


Figura 2: Pila tecnológica de la web semántica

En los siguientes subapartados se explican estas tecnologías y estándares.

2.2 Ontología

En filosofía, la ontología es la parte de la metafísica que estudia el ser. El ser abarca toda la realidad y todas las entidades que lo componen. La ontología examina qué tienen en común todas las cosas para así poder articular la estructura básica del ser. También investiga cómo pueden agruparse las

cosas en tipos básicos, como las categorías de particulares y universales. Las particulares son entidades únicas, no repetibles, por ejemplo la persona Sócrates, mientras que las universales son generalizaciones, entidades repetibles, como el color azul. Otro contraste que realiza es entre objetos concretos que existen en el espacio y el tiempo, como un geranio, y objetos abstractos que existen fuera del espacio y del tiempo, como el número 3. Estos sistemas de categorías pretenden ofrecer un inventario exhaustivo de la realidad empleando categorías como sustancia, propiedad, relación, estado de cosas y acontecimiento.

Esta introducción no es baladí, pues la ontología en informática guarda cierta similitud con su homónima en la metafísica. Aquí, una ontología consiste en una definición formal de tipos, propiedades y relaciones entre entidades que existen para un dominio en particular [4]. Las tres principales propiedades de una ontología son:

- Utiliza una representación explícita, por lo que suele estar escrita en un lenguaje formal y en un soporte digital.
- Su conceptualización es compartida: representa la información que un conjunto de personas tienen respecto a un dominio de discurso, por lo que no puede representar el punto de vista de un único individuo.
- Representa el dominio de discurso relevante para un problema concreto.

La herramienta principal para conseguir el objetivo de la web semántica es la ontología, pues permite la definición formal de los conceptos del dominio, su jerarquía y las relaciones que los unen entre sí.

2.3 XML

Este acrónimo proviene de las siglas en inglés de *eXtensible Markup Language* y consiste en un lenguaje de marcas para almacenar, transmitir y reconstruir datos de cualquier índole, lo cual lo hace especialmente idóneo para el intercambio de información estructurada entre diferentes plataformas. Un lenguaje de marcas es aquel en el que se especifica la estructura y el formato del documento, y potencialmente las relaciones entre sus partes.

Un documento XML está formado por dos secciones principales:

- El prólogo: Aquí es donde se define que se trata de un documento XML, su versión, así como entradas opcionales con información acerca de su procesamiento. En el siguiente ejemplo se muestra un prólogo con las partes opcionales entre corchetes:

```
<?xml version="1.0" [encoding="UTF-8"] [standalone="yes"] ?>
```

- La estructura de datos o cuerpo del documento: contiene los datos y su jerarquía, es decir, contiene la información del documento. En el siguiente ejemplo se muestra una parte del cuerpo de un documento en el que dentro de la etiqueta “Personal” hay varias marcas “Empleado”, cada una con la misma estructura: NSS, Nombre, etc. Asimismo, la marca “Personal” podría estar formando parte de una estructura mayor:

```

<Personal>
  ...
  <Empleado>
    <NSS>111-11-1111</NSS>
      <Nombre>Ana</Nombre>
      <Apellidos>López Gómez</LastName>
      <Salario>65000.00</Salario>
      <Departamento>Contabilidad</Departmento>
    </Empleado>
    <Empleado>
      ...
    </Empleado>
  ...
</Personal>
  
```

Nótese que cada marca señala un segmento de información con un sentido claro y específico.

2.4 RDF y RDF Schema

Este acrónimo proviene de las siglas en inglés *Resource Description Framework* y es una notación estándar para la representación e intercambio de información sobre recursos, particularmente recursos web. Ésta notación representa un grafo dirigido compuesto de triplets sujeto – predicado – objeto¹, en las que el sujeto y el objeto son nodos y el predicado es el arco que los une. Cada una de estas tres partes se identifica de forma inequívoca mediante un identificador de recursos uniforme, URI por sus siglas en inglés, para garantizar que no hay ambigüedades en la estructura semántica.



Figura 3: Triplete como grafo dirigido

Por otro lado, *RDF Schema* (abreviado como RDFS) es una extensión de RDF para poder describir vocabularios de términos y relaciones sencillas entre ellos, por ejemplo jerárquicas. Para ello, proporciona los conceptos de clases y de propiedades, así como el de jerarquía entre ellas mediante las anotaciones rdfs:Class, rdfs:subClassOf, rdf:property y rdfs:subPropertyOf. Asimismo, se indica

¹ Curiosamente, en los lenguajes orientados a objetos, la llamada al método de un objeto tiende a seguir la estructura sujeto.predicado(objeto)

que un recurso es instancia de una clase mediante `rdf:type`. Nótese que tanto las instancias como las propiedades ya están incluidas en el estándar RDF.

2.5 OWL

Para poder representar definiciones y restricciones más complejas que las posibles con RDFS, existe una extensión del mismo: el estándar *Web Ontology Language* (OWL por sus siglas en inglés). Este estándar enriquece la definición de clases mediante las operaciones de unión, intersección y complementación de clases. Las propiedades son ampliadas con restricciones de cardinalidad y valores específicos, además de la existencia de las propiedades inversas, simétricas y transitivas. Asimismo, aporta el concepto de equivalencia tanto a las clases como a las propiedades. Finalmente, permite aplicar a la ontología herramientas de razonamiento inferencial, pues OWL está basado en la lógica descriptiva.

Gracias a estas características con las que amplía RDFS, la extensión OWL permite la definición de ontologías estructuradas basadas en la web, a diferencia de los lenguajes anteriores, que fueron pensados para comunidades específicas de usuarios, por ejemplo en el campo de las ciencias.

En los tres estándares descritos (RDF, RDFS y OWL), es necesario automatizar su interpretación, para lo cual deben almacenarse en un fichero. Los tres estándares definen el modelo de datos pero no su *serialización*, para la cual existen diferentes sintaxis como Notation3, XML y Turtle, siendo esta última la que se empleará en este trabajo por la legibilidad de su sintaxis. No obstante, en vez de guardarse en un documento, también podría hacerse en un fichero con una *serialización* en formato binario.

2.6 SPARQL

SPARQL es un lenguaje para la consulta (*query*) de la información almacenada en el formato RDF, lo cual también incluye sus extensiones RDFS y OWL. Además de ser un lenguaje de consultas, también es el protocolo para acceder a datos RDF, tal y como indican sus siglas en inglés: *SPARQL Protocol and RDF Query Language*. Este lenguaje guarda ciertas similitudes con SQL, el lenguaje para hacer consultas en bases de datos relacionales, si bien en este caso los datos almacenados son triplets.

Una de sus extensiones es GeoSPARQL, que permite representar y hacer consultas geoespaciales en el formato RDF, así como su integración en el contexto de los datos enlazados.

2.7 Datos enlazados

Los datos enlazados frecuentemente son referidos como datos vinculados o también por su nombre en inglés: *linked data*. Este término fue acuñado por Tim Berners-Lee en una nota de diseño acerca del proyecto de la web semántica. Su idea es que para conseguir que la web semántica sea una realidad, debe disponerse de un gran volumen de datos interconectados entre ellos y en un formato estándar, de manera que puedan ser tratados por ordenadores.

En la nota mencionada previamente, Tim Berners-Lee señaló estos cuatro principios:

1. Los recursos públicos de la web deben ser identificados mediante URIs.
2. Las URIs deben ser accesibles mediante el protocolo HTTP.
3. Cuando se accede a una URI, esta devuelve la información por medio de estándares como RDF o SPARQL.
4. Los datos deben incluir enlaces a otras URIs, es decir, a otros recursos externos, así se potencia el descubrimiento de datos relacionados en diferentes fuentes.

DBpedia y Wikidata son dos destacados ejemplos de proyectos que siguen el paradigma de los datos enlazados.

2.8 Datos abiertos y conocimiento abierto

Los datos abiertos (*open data* en inglés) son una filosofía y una práctica que persigue que ciertos datos estén disponibles para todo el mundo, sin restricciones como los derechos de autor o las patentes, mediante el empleo de licencias abiertas. Esta filosofía es parecida a la de otros movimientos “abiertos”, como por ejemplo el código abierto (*open source*), la educación abierta o el conocimiento abierto. Éste último amplía el concepto de datos abiertos: El conocimiento abierto consiste en una serie de principios y metodologías en la producción y difusión del conocimiento para garantizar el acceso, uso, modificación y redistribución de la información sin restricciones legales, sociales o tecnológicas.

3 Propuesta

Una vez explicadas las tecnologías y estándares en los que se basa el siguiente trabajo, se listan los elementos que se esperan obtener:

- Una ontología con los monumentos de la villa de Madrid.

-
- Un *script* de importación para cargar en la ontología los datos del fichero CSV del portal de datos abiertos del Ayuntamiento de Madrid [5].
 - Un fichero con ejemplos de consultas SPARQL: unas extraen información exclusivamente de la ontología, mientras que otras la amplían integrando información procedente de Wikidata, de manera que sirven de ejemplo de las posibilidades de la web semántica.

Sobre este último punto debe decirse que se hizo un pequeño estudio acerca de si era mejor hacer la integración de Wikidata o DBpedia. Dados los campos del fichero CSV (ver [punto 4](#)) y los campos disponibles en la DBpedia, se comprobó que sólo los campos de latitud y longitud podrían emplearse para relacionar ambas fuentes de datos, requiriendo además modificaciones, pues las coordenadas son mucho más precisas en el CSV. Mediante una consulta SPARQL se comprobó que de todos los monumentos ubicados en Madrid, la DBpedia sólo tiene las coordenadas de tres de ellos. Por otro lado, mediante otra consulta se comprobó que en la Wikidata muchos de los monumentos sí las tienen, aunque no todos.

Además, la Wikidata tiene estas otras ventajas:

- Su campo *catalogCode* coincide con el campo *PK* del fichero CSV.
- Se actualiza con mayor frecuencia y cuenta con una comunidad activa.
- De cada dato relacionado se muestra su identificador, con lo cual es fácil acceder a los datos del mismo. Por ejemplo, se puede mostrar la información del autor, además de otros datos de los que carezca la ontología.
- La documentación para hacer consultas es algo mejor.

Su principal desventaja es que hay más campos diferentes entre entidades (monumentos) que en la DBpedia, lo cual puede dificultar la integración de su información.

4 Diseño

En este apartado se explica en primer lugar el proceso de diseño de la ontología a partir de la información existente. A continuación, se implementa el diseño en Protégé.

4.1 Integración

Tal y como se recuerda en el documento de Methontology, las ontologías se crean para ser reutilizadas, por lo que se han buscado proyectos existentes. Entre ellos destacan estos tres:

-
- HERACLES Ontology. Esta ontología se centra en el impacto potencialmente dañino del cambio climático sobre el patrimonio cultural, principalmente monumentos y edificios. Para ello, se centra en los datos que expertos de diferentes ámbitos necesitan para averiguar cómo el cambio climático afecta al patrimonio cultural y cómo mitigar sus efectos nocivos. Forma parte del proyecto HERACLES (*HEritage Resilience Against CLimate Events on Site*) de la Unión Europea.
 - CityGML CHADE (*Cultural Heritage Application Domain Extension*).[5] Se trata de un proyecto que extiende el módulo Building[6] mediante la extensión ADE de CityGML para contener información acerca de la historia, condición y estado de la protección del monumento. ADE, *Application Domain Extension* por sus siglas en inglés, es un mecanismo de CityGML para dotarlo de más conceptos según necesidades específicas. Por su parte, CityGML[7] es un modelo de datos estandarizado abierto y un formato de intercambio para almacenar modelos digitales en 3D de ciudades y paisajes.
 - BCH Ontology. Consiste en la integración y expansión de tres ontologías que ya existían previamente: Geneva CityGML, MONDIS (Monument Damage) y CIDOC-CRM (*International Council of Museums*). Su énfasis está en la conservación preventiva del patrimonio y en las iniciativas de datos abiertos. Un caso de uso práctico se realizó con el centro histórico de la ciudad de Cuenca, Ecuador, declarado Patrimonio de la Humanidad por la UNESCO [6].

Los objetivos de estos ambiciosos proyectos no coinciden con los de este. Además, el presente proyecto también incluye obras más humildes que un monumento, como puede ser una fuente o una pequeña estatua. Adicionalmente, no se requiere almacenar modelos 3D, por lo que el estándar CityGML es innecesario. Así pues, no se reutilizarán estas ontologías para este proyecto.

4.2 Especificación

Se continua con la fase de especificación de la metodología Methontology. Estos son los campos en el fichero CSV:

Campo	Comentarios	Tipo
PK	Se puede relacionar con el campo <i>catalogCode</i> de Wikidata	Entero
NOMBRE		Texto
SUBTIPO	Posibles valores: Escultura conceptual o abstracta Grupo Escultórico Elemento conmemorativo, Lápida Estructura Puente, Construcción Civil Fuente, Estanque, Lámina de agua Elemento de ornamentación Edificación Singular Puerta, Arco triunfal	Texto
DESCRIPCIÓN		Texto
FECHA	Año o descripción tipo “Cercano a 1808”	Texto
AUTORES	Primero los apellidos, después el nombre separado por una coma. Si hay más de uno se separan por un guión	Texto / array
PDF	URL del PDF que describe el monumento en la web de Patrimonio Cultural y Paisaje Urbano de Madrid	Texto
CONTENT-URL	URL a la web que describe el monumento en la misma web	Texto
NOMBRE-VIA		Texto
CLASE-VIAL	Posibles valores: Avenida Calle ...	Texto
TIPO-NUM	Posibles valores: V F	Texto
NUM		Entero
PLANTA		Vacío
PUERTA		Vacío
ESCALERAS		Vacío
ORIENTACIÓN		Vacío
LOCALIDAD	Siempre es “Madrid”	Texto
PROVINCIA	Siempre es “Madrid”	Texto
CODIGO-POSTAL		Entero
CODIGO-BARRIO		Entero
BARRIO		Texto
COD-DISTRITO		Entero
DISTRITO		Texto
COORDENADA-X		Entero
COORDENADA-Y		Entero
LATITUD		Decimal
LONGITUD		Decimal
TELÉFONO		Vacío
FAX		Vacío
EMAIL		Vacío
TIPO		Vacío

Tabla 4: Campos del fichero CSV

La combinación de la información de la ontología con la Wikidata se hará mediante la intersección de los campos *PK* y *catalogCode*. Incluso sería posible mediante *describedAtURL*, pues se trata de la URL a la página web del monumento en el sitio de Patrimonio y Paisaje de Madrid. No obstante, es más óptimo hacerlo mediante el identificador o clave primaria *PK*. Si en una futura ampliación de esta ontología se añadieran monumentos de otras ciudades que no pudieran relacionarse mediante el *catalogCode*, quedarían las coordenadas como única alternativa.

A continuación, se listan campos de Wikidata que son interesantes para incorporar:

Campo	Comentarios
Autor / Arquitecto / Creador	Puede contener más de uno
Imagen	
Estilo arquitectónico	
Inicio (<i>Inception</i>)	Fecha de construcción
Evento significativo	Puede contener varios sucesos como elementos de la Wikidata (incendio, reconstrucción...) y el año del mismo
<i>Depicts</i>	Personaje histórico homenajeado

Tabla 5: Campos de Wikidata interesantes para incorporar

En general, la estructura de datos no es muy consistente en los diferentes monumentos, pues no todos tienen estos campos. Además, para unos se emplea el campo Autor y para otros Creador, sin que parezca que haya un criterio para usar uno u otro.

4.3 Conceptualización

A partir de toda la información recopilada, se crea un glosario de términos que ya puede agruparse en clases y propiedades. Las primeras son:

- **Monumento:** Es el foco de la ontología. Sus tipos son:
 - Escultura abstracta.
 - Grupo escultórico.
 - Elemento conmemorativo.
 - Estatua

- Construcción civil
 - Fuente
 - Ornamentación
 - Edificación singular
 - Puerta
- **Autor/a:** El creador de la obra, pudiendo haber más de uno.
 - **País** donde está ubicado el monumento.
 - **División administrativa** donde está ubicado el monumento. No se ha empleado el término “provincia” sino que se ha generalizado para poder englobar las divisiones administrativas de diferentes países, como departamento, estado, prefectura, etc. Un país tiene varias divisiones administrativas.
 - **Ciudad** donde está ubicado el monumento. Una división administrativa puede tener varias ciudades.
 - **Distrito** donde está ubicado el monumento. Una ciudad puede tener varios distritos.
 - **Barrio** donde está ubicado el monumento. No siempre está informado en el CSV. Un distrito puede tener varios barrios.

Las propiedades de los monumentos son:

- **Clave primaria** o *PK* por las siglas en inglés de *Primary Key*.
- **Nombre.** Esta propiedad es común a todas las clases.
- **Descripción:** texto descriptivo acerca de diferentes aspectos del monumento.
- **Dirección** donde está ubicado el monumento. Está compuesta por la concatenación de los campos CLASE-VIAL, NOMBRE-VIA y NUM del CSV.
- **Código postal** del lugar donde está ubicado el monumento.
- **Coordenadas: latitud y longitud** del punto donde está ubicado el monumento.
- **Fecha:** Año o muy breve texto acerca de cuándo fue erguido el monumento.

- **URL de información:** Una o varias URL a una página web o documento PDF con información acerca del monumento.

4.4 Formalización

La formalización de la conceptualización anterior puede lograrse mediante la visualización de la ontología y su especificación, siendo el diagrama de clases UML una buena herramienta para ello, si bien esta herramienta no permite la representación exacta de ontologías. En una ontología, las propiedades de datos expresan una relación entre clases y tipos de datos, mientras que en la programación orientada a objetos cada propiedad es de una clase. No obstante, en el diagrama UML se muestran las propiedades como si pertenecieran a la clase en aras de facilitar su comprensión.

Mediante el empleo de la flecha en ambos extremos de la relación se indica que ésta tiene su inversa, aunque no se muestre el nombre de la inversa para no complicar la legibilidad. La inversa de la relación *hasGeometry* no se ha creado, por ser poco útil. No se han creado relaciones entre el monumento y su país, así como tampoco con su división administrativa, pues ambas entidades se pueden deducir en una consulta mediante la ciudad a la que pertenece el monumento. Sin embargo, se han creado relaciones específicas como *cityHasMonument*, *districtHasMonument*, etc. en vez de la genérica *hasMonument* para evitar tener que inferir demasiada información a partir de la jerarquía de clases, pues esto haría algunas consultas lentas si la ontología crece.

Finalmente, después de documentarse acerca de cómo funcionan las consultas geoespaciales, se ha visto la necesidad de crear la clase *Geometry* y que contenga las propiedades longitud y latitud. Tal y como se explicó en el subapartado [1.4 Metodología y proceso de trabajo](#), la integración y la formalización se realizan en un proceso iterativo.

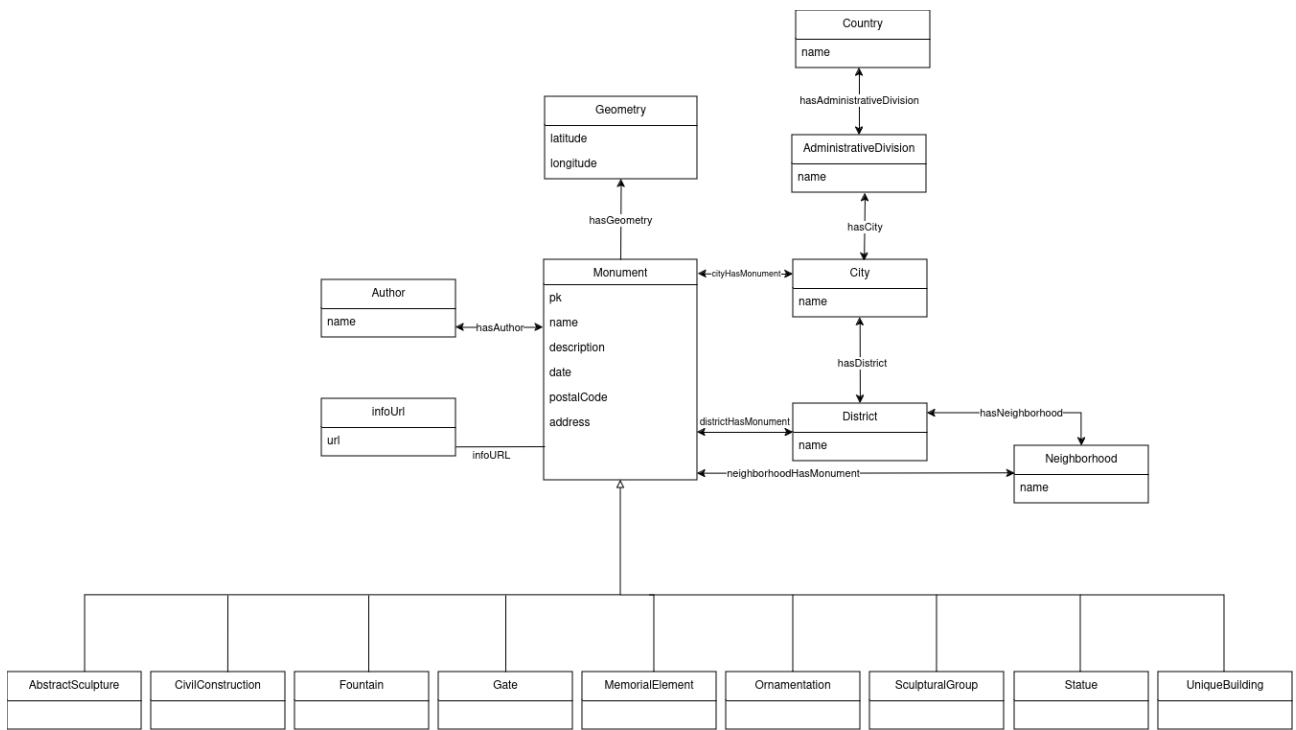


Figura 4: Diseño UML de la ontología

4.5 Implementación

En la siguiente figura se muestra el grafo de la implementación final en Protégé:

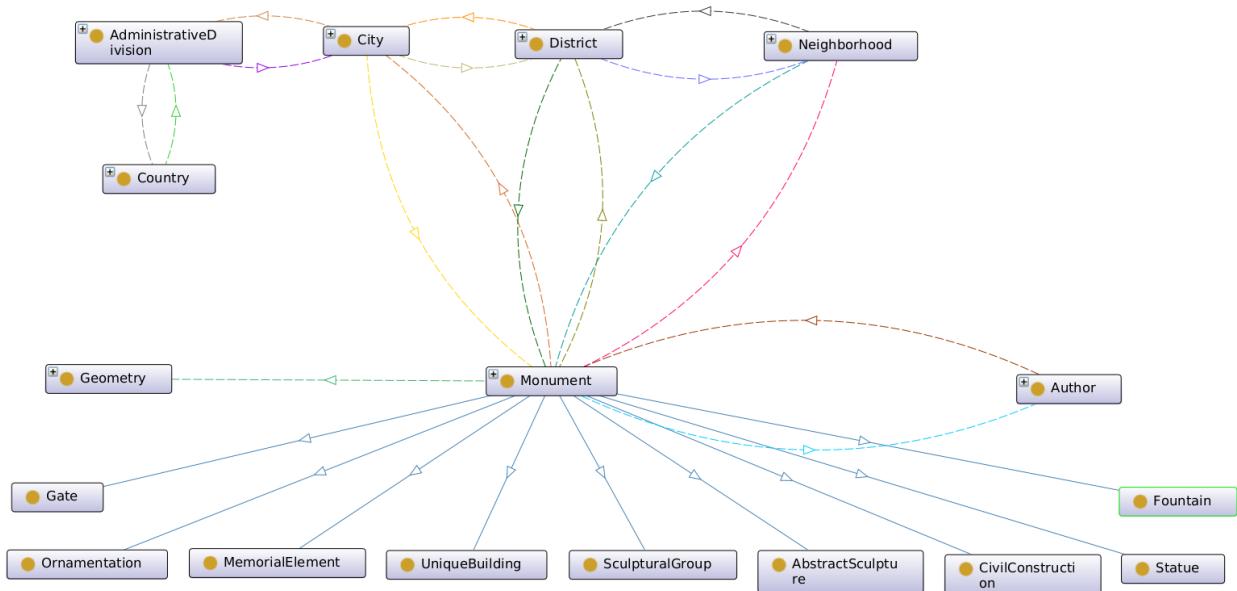


Figura 5: Grafo de las clases y sus relaciones

El lenguaje OWL define las categorías *owl:ObjectProperty* y *owl:DatatypeProperty*, mientras que en UML existen las relaciones para clases y los atributos para datos, pero es importante destacar que no hay una correspondencia entre ambos. La clase *infoURL* es un buen ejemplo de ello: puede observarse que ha desaparecido. Desde la perspectiva del diseño mediante un diagrama de clases UML, tiene sentido que la URL de información se represente en una clase. Sin embargo, en el lenguaje OWL una propiedad puede tener más de un valor; más aun, es el comportamiento por defecto si no se especifica una cardinalidad. Como además no se va a guardar ninguna información específica acerca de la URL, tiene más sentido que sea un propiedad de datos, como se verá en el siguiente subapartado.

4.5.1 Propiedades de datos

En el lenguaje OWL, una propiedad de datos relaciona una clase con valores literales o datos primitivos como una cadena de texto o una fecha. En la siguiente figura se muestra la propiedad *infoURL*, nótese que no está definida como funcional, para así poder aceptar múltiples valores.

The screenshot shows the OntoGraf interface with the following details:

- Data property hierarchy:** InfoURL (selected)
- Annotations:**
 - rdfs:label: InfoURL
 - rdfs:comment: URL to the website with information about the monument
- Description:**
 - Functional
 - Equivalent To: Monument
 - Ranges: xsd:anyURI

Figura 6: Propiedades de datos

Las propiedades de datos coinciden con las listadas en el apartado de conceptualización y sólo queda destacar dos de ellas por su particularidad. En primer lugar, las coordenadas se guardan en la propiedad *geo:asWKT*, cuyo rango se ha definido como *geo:wktLiteral*, que es el estándar para datos espaciales en GeoSPARQL. Los literales siguen este formato:

```
geo:asWKT "POINT(longitud latitud)"^^geo:wktLiteral
```

Por ejemplo:

```
geo:asWKT "POINT(-3.680873418964599 40.42004278090423)"^^geo:wktLiteral
```

Annotations: geo:asWKT

Characteristics:

- Functional
- Equivalent To
- SubProperty Of
- Domains (intersection): Geometry
- Ranges: geo:wktLiteral
- Disjoint With

Figura 7: Propiedad de datos para las coordenadas

En segundo lugar, la propiedad *name* es común a todas las clases, excepto a la de geometría. Como puede observarse en su correspondiente figura, en el dominio están todas las clases que tienen esa propiedad. Nótese que no hace falta incluir todas las subclases de monumento sino que basta con añadir la superclase *Monument*.

Annotations: name

Characteristics:

- Functional
- Equivalent To
- SubProperty Of
- Domains (intersection): Monument, District, Author, City, Country, AdministrativeDivision, Neighborhood
- Ranges: rdfs:Literal

Figura 8: Propiedad de datos para el nombre

4.5.2 Propiedades de objetos

Una propiedad de objetos relaciona individuos (instancias de clase) entre sí. Cabe destacar que muchas de las relaciones tienen su inversa. Si hay una relación de los individuos de la clase A con los individuos de la clase B, su relación inversa relacionaría cada individuo de B con su correspondiente individuo de A. En el grafo de la [figura 5](#), cada par de nodos unidos por dos arcos representan una relación y su inversa. Aquellos pares que están unidos por un único arco carecen de relación inversa.

Figura 9: Propiedades de objetos

En general, se ha seguido la convención de que el nombre de las relaciones empieza por “has” y el de sus inversas por “is”, si bien se ha hecho alguna excepción cuando no ha sido posible crear un nombre gramaticalmente correcto en inglés, como por ejemplo *cityHasMonument*. A continuación se listan las propiedades de objetos:

- **cityHasMonument:** Relaciona la ciudad con el monumento. Su inversa es **isLocatedInCity**.
- **districtHasMonument:** Relaciona el distrito con el monumento. Su inversa es **isLocatedInDistrict**.
- **hasAdministrativeDivision:** Relaciona el país con la división administrativa. Su inversa es **isAdministrativeDivisionOf**.
- **hasAuthor:** Relaciona el monumento con su autor/a o autores/as. Su inversa es **isAuthorOf**.

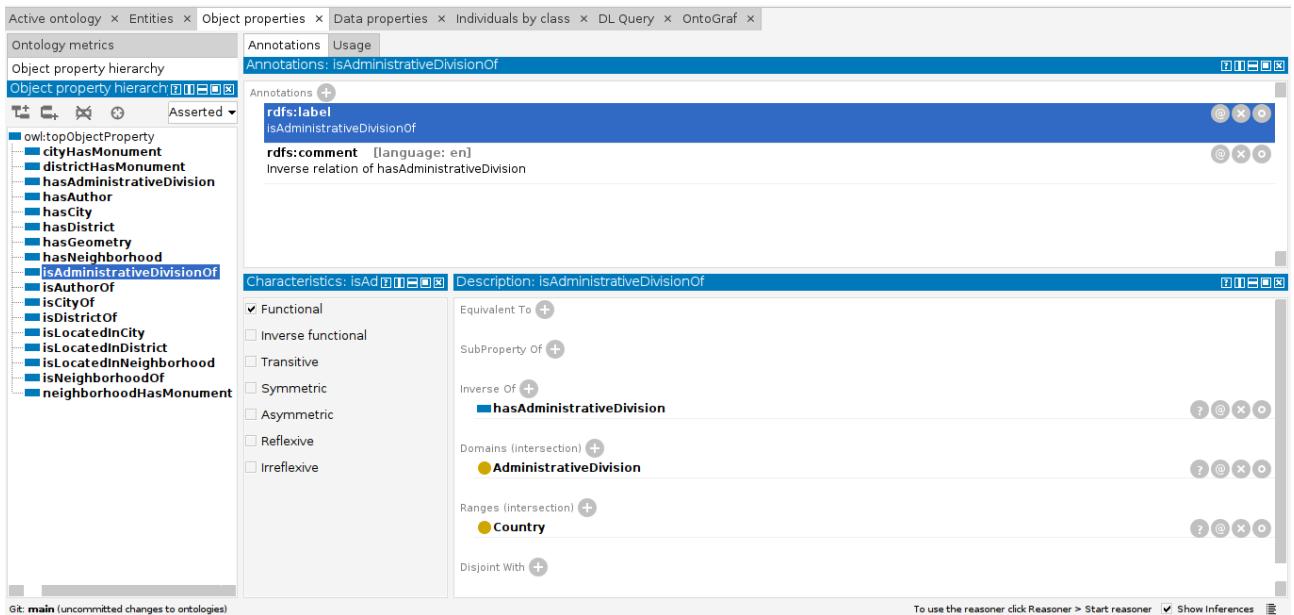
- **hasCity:** Relaciona la división administrativa con su ciudad. Su inversa es **isCityOf**.
- **hasDistrict:** Relaciona la ciudad con su distrito. Su inversa es **isDistrictOf**.
- **hasGeometry:** Relaciona el monumento con una geometría, i.e., con sus coordenadas. No tiene inversa.
- **hasNeighborhood:** Relaciona el distrito con el barrio. Su inversa es **isNeighborhoodOf**.
- **neighborhoodHasMonument:** Relaciona el barrio con el monumento. Su inversa es **isLocatedInNeighborhood**.

En cuanto a la cardinalidad, sucede que una relación puede aceptar varios valores y su inversa no, como se ilustra en las siguientes figuras:

The screenshot shows the OntoGraf interface with the following details:

- Annotations:** hasAdministrativeDivision
- rdfs:label:** hasAdministrativeDivision
- Asserted in:** <http://www.semanticweb.org/vic/ontologies/2024/10/monuments>
- Characteristics:** has/Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive.
- Description:** hasAdministrativeDivision
- Inverse Of:** isAdministrativeDivisionOf
- Domains (intersection):** Country
- Ranges (intersection):** AdministrativeDivision
- Disjoint With:** (empty)

Figura 10: Propiedad de datos hasAdministrativeDivision

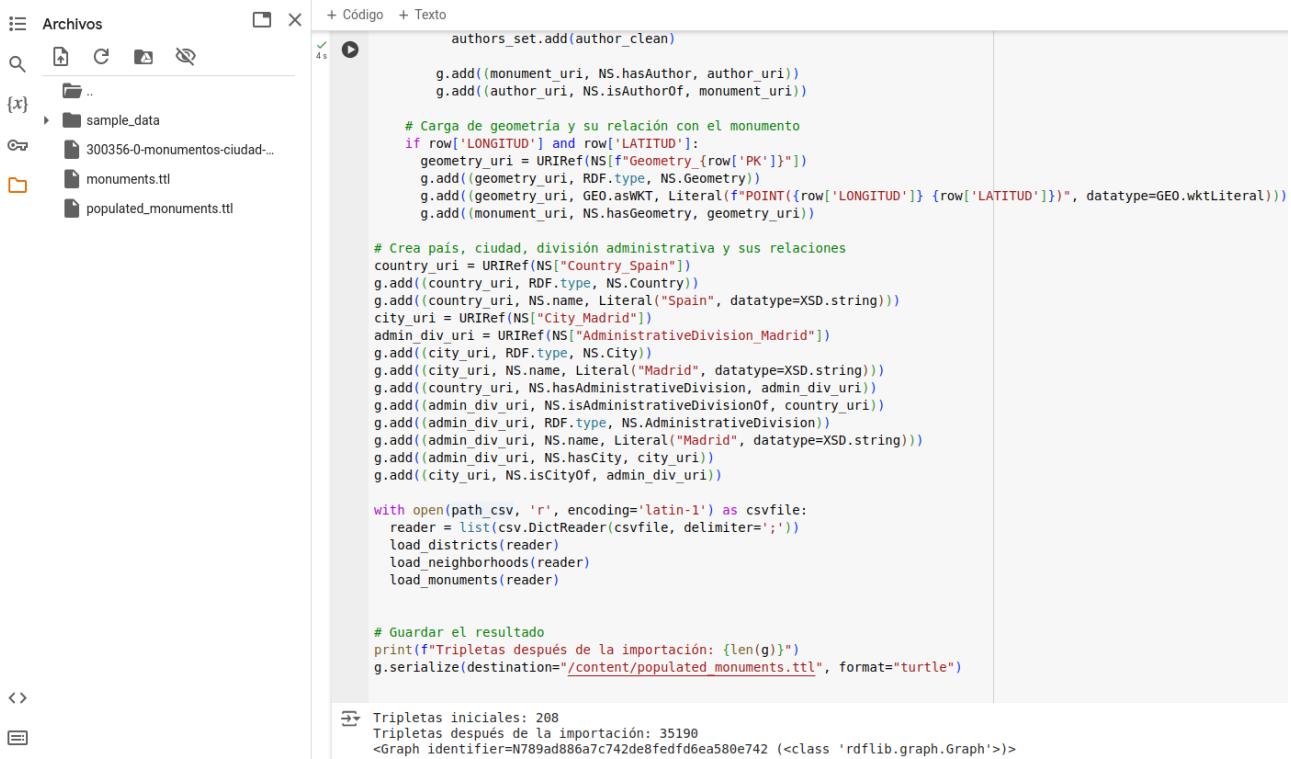
Figura 11: Propiedad de datos *isAdministrativeDivisionOf*

Un país puede tener más de una división administrativa, pero una división administrativa sólo puede pertenecer a un único país.

4.5.3 Entidades

Una vez implementada la ontología, debe poblarla con las entidades. En el mundo de las ontologías, el término “entidad” es mejor que “instancia”, un término característico de la programación orientada a objetos, pues también son entidades las relaciones (propiedades de objetos) entre los individuos, así como sus propiedades de datos. Además, una ontología busca modelar conocimiento del mundo real, por lo que el término “entidad” resulta más adecuado. Incluso desde la perspectiva de la filosofía, su rama de la ontología o metafísica general habla de entes y entidades [7].

La carga de las entidades se ha realizado a partir del fichero CSV del portal de datos abiertos del Ayuntamiento de Madrid, mediante un *script* programado en Python 3 que hace uso de la librería *rdflib* [8] para poder trabajar con RDF. Este *script* se ha programado y ejecutado en el entorno *Colab* de Google [9], si bien puede también ejecutarse en un entorno local. A partir de dos variables que son las rutas al fichero CSV y al fichero que contiene la ontología recién diseñada (sin entidades), el *script* produce como salida la misma ontología pero cargada con todos los monumentos incluidos en el CSV.



```

Archivos + Código + Texto
sample_data
300356-0-monumentos-ciudad...
monuments.ttl
populated_monuments.ttl

authors_set.add(author_clean)

g.add((monument_uri, NS.hasAuthor, author_uri))
g.add((author_uri, NS.isAuthorOf, monument_uri))

# Carga de geometria y su relación con el monumento
if row['LONGITUD'] and row['LATITUD']:
    geometry_uri = URIRef(NS["Geometry_{row['PK']}"])
    g.add((geometry_uri, RDF.type, NS.Geometry))
    g.add((geometry_uri, GEO.asWKT, Literal(f"POINT({row['LONGITUD']}) {row['LATITUD']}")))
    g.add((monument_uri, NS.hasGeometry, geometry_uri))

# Crea país, ciudad, división administrativa y sus relaciones
country_uri = URIRef(NS["Country_Spain"])
g.add((country_uri, RDF.type, NS.Country))
g.add((country_uri, NS.name, Literal("Spain", datatype=XSD.string)))
city_uri = URIRef(NS["City_Madrid"])
admin_div_uri = URIRef(NS["AdministrativeDivision_Madrid"])
g.add((city_uri, RDF.type, NS.City))
g.add((city_uri, NS.name, Literal("Madrid", datatype=XSD.string)))
g.add((country_uri, NS.hasAdministrativeDivision, admin_div_uri))
g.add((admin_div_uri, NS.isAdministrativeDivisionOf, country_uri))
g.add((admin_div_uri, RDF.type, NS.AdministrativeDivision))
g.add((admin_div_uri, NS.name, Literal("Madrid", datatype=XSD.string)))
g.add((admin_div_uri, NS.hasCity, city_uri))
g.add((city_uri, NS.isCityOf, admin_div_uri))

with open(path_csv, 'r', encoding='latin-1') as csvfile:
    reader = list(csv.DictReader(csvfile, delimiter=';'))
    load_districts(reader)
    load_neighborhoods(reader)
    load_monuments(reader)

# Guardar el resultado
print(f"Tripletas después de la importación: {len(g)}")
g.serialize(destination=_content/populated_monuments.ttl, format="turtle")

Tripletas iniciales: 208
Tripletas después de la importación: 35190
<Graph identifier=N789ad886a7c742de8fedf6ea580e742 (<class 'rdflib.graph.Graph'>)>

```

Figura 12: Script de importación inicial

Básicamente, el funcionamiento del *script* se puede resumir en los siguientes pasos:

1. Crear un grafo a partir del fichero con la estructura de la ontología. En este grafo se irán creando todas las entidades a partir de los datos del fichero CSV.
2. Crear el país, la ciudad, la división administrativa y sus relaciones.
3. Cada línea del CSV contiene toda la información de un monumento y por cada línea sigue estos pasos:
 - 1) Añade el distrito del monumento si no existe ya junto con sus relaciones con la ciudad.
 - 2) Añade el barrio del monumento si no existe ya junto con sus relaciones con el distrito.
 - 3) Añade el monumento, sus propiedades de datos, sus relaciones con la ciudad, con el distrito y con el barrio. A continuación, añade el autor o autores si no existen ya y su relación con el monumento. Finalmente, añade su geometría (coordenadas) y su relación con el monumento.

Ontology metrics:	
Metrics	
Axiom	35,595
Logical axiom count	33,692
Declaration axioms count	44
Class count	17
Object property count	17
Data property count	8
Individual count	4,375
Annotation Property count	3

Figura 13: Métricas de la ontología después de cargar las entidades

Si bien este *script* está adaptado a las particularidades del fichero de datos que proporciona el portal de datos abiertos del ayuntamiento de Madrid, los pasos a seguir para incorporar en el futuro otras fuentes de datos abiertos deberían seguir siendo los mismos.

A continuación, se muestran de ejemplo algunas de las entidades cargadas:

Active ontology × Entities × Object properties × Data properties × Individuals by class × DL Query × OntoGraf ×

Individuals: AbstractSculpture_11064118 — AbstractSculpture_11064118 — http://www.semanticweb.org/vic/ontologies/2024/10/monuments/AbstractSculpture_11064118

Annotations Usage

Annotations: AbstractSculpture_11064118

Annotations

Ontology metrics Rules

Ontology metrics:

Description: AbstractSculpture_11064118 | Property assertions: AbstractSculpture_11064118

Types AbstractSculpture

Same Individual As

Different Individuals

Object property assertions

- hasAuthor Author_VÁZQUEZ_GARCÍA_Higinio'
- isLocatedInDistrict District_DISTRITO
- hasGeometry Geometry_11064118
- isLocatedInCity City_Madrid

Data property assertions

- infoURL '<https://patrimonioypaisaje.madrid.es/FrameWork/generacionPDFMonumenta/11064118.pdf?idMonumento=d334cd1fc86e610vgvCM200000f4a900aRCRD&PK=11064118>
- infoURL '<https://patrimonioypaisaje.madrid.es/sites/v/index.jsp?vgnextchannel=8fac3cb702aa4510vgvCM1000008a4a900aRCRD&vgnextoid=d334cd1fc86e610vgvCM1000008a4a900aRCRD&PK=11064118>
- date '1971'@es
- name 'Escultura a la Universidad en Cantoblanco'
- PK '11064118'
- description 'El monumento fue inaugurado en los jardines de la Universidad Autónoma de Madrid en 1971, el mismo año en que se fundó dicho campus; la escultura, así como otra pieza más compleja que es Higinio Vázquez García, situada en los mismos jardines, le fueron encargadas a su autor para conmemorar el nacimiento del campus, que en ese entonces era un centro de investigación y ocupaba Alrededor de 100000 pies cuadrados de gran tamaño y estética minimalista, la cual por su diseño cercano a las esculturas contiene algunos de los materiales constructivos de estos, piedras y mármol. La consecuencia ligados formalmente entre si. Higinio Vázquez García es un escultor zamorano, nacido en 1930, especializado fundamentalmente en escultura sacra, y fue llamado, como otras jóvenes figuras del moderno panorama artístico, para dignificar y enriquecer el espacio urbano, la imagen externa y las dependencias de las primeras arquitecturas de la Universidad Autónoma en el momento de su creación. Su escultura civil se caracteriza por la estrecha relación que guarda con la arquitectura en la que se inserta'@es
- address 'CUIDAD UNIVERSITARIA AUTONOMA'

Negative object property assertions

Negative data property assertions

owl:Thing

- AdministrativeDivision
- Author
- City
- Country
- District
- Geometry
- Monument
 - AbstractSculpture
 - CivilConstruction
 - Fountain
 - Gate
 - MemorialElement
 - Ornamentation
 - SculpturalGroup
 - Statue
 - UniqueBuilding
- Neighborhood

Asserted

Figura 14: Entidad Escultura Abstracta

Figura 15: Entidad Autor

Figura 16: Entidad Distrito

Figura 17: Entidad Fuente

Figura 18: Entidad Puerta

Active ontology | Entities | Object properties | Data properties | Individuals by class | DL Query | OntoGraf | Individuals: MemorialElement_400436 ≡ MemorialElement_400436 — MemorialElement_400436 — http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_400436

Description: Mem | Property assertions: MemorialElement_400436

Object property assertions:

- islocatedInDistrict District_CHAMBERI
- hasGeometry Geometry_400436
- hasAuthor Author_VEGA_SAMPER_José_Maria_de_la'
- islocatedInNeighborhood Neighborhood_VALLEHERMOSO
- islocatedInCity City_Madrid

Data property assertions:

- PK "400436"
- postalCode "28040"
- infoURL "<https://patrimonioypaisaje.madrid.es/sites/v/index.jsp?vgnextchannel=8fac3cb702aa4510VgnVCM1000008a4a900aRCRD&vgnextoid=2ad8091d1b9c45102e085a0>"
- description "Se trata de un monumento dedicado a la localización en el barrio de las Comillas, se construyó el Colegio Mayor San Pablo, perteneciente a la Asociación Católica Nacional de Propagandistas, cuya primera piedra se colocó en 1945. En esa fecha, el edificio quedaba próximo a otros colegios mayores situados en los bordes de la Ciudad Universitaria, pero fuera del recinto de sus terrenos. Se construyó en un estilo entre moderno y neoclásico, con una monumental entrada de rampas y escalinatas que conducen a un portón tetrástilo con triple puerta, a cuyos lados se situan de modo simétrico dos lápidas conmemorativas de las obras, encuadradas en la arquitectura del edificio, que se inauguró el 7 de marzo de 1951."@es
- infoURL "<http://www.semanticweb.org/vic/ontologies/2024/10/monuments>"
- date "1951" @es
- address "CALLE SAN PABLO, 30"
- name "Colegio Mayor San Pablo"

Negative object property assertions:

Negative data property assertions:

Figura 19: Entidad Elemento Conmemorativo

Active ontology | Entities | Object properties | Data properties | Individuals by class | DL Query | OntoGraf | Individuals: Ornamentation_400132 ≡ Ornamentation_400132 — Ornamentation_400132 — http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Ornamentation_400132

Description: Ora | Property assertions: Ornamentation_400132

Object property assertions:

- islocatedInCity City_Madrid
- islocatedInNeighborhood Neighborhood_BERRUGUETE
- islocatedInDistrict District_TETUAN
- hasGeometry Geometry_400132
- hasAuthor Author_FERNÁNDEZ_BALBUENA_Gustavo'

Data property assertions:

- PK "400132"
- infoURL "<https://patrimonioypaisaje.madrid.es/sites/v/index.jsp?vgnextchannel=8fac3cb702aa4510VgnVCM1000008a4a900aRCRD&vgnextoid=a258091d1b9c4510091d1b9c45102e085a0>"
- date "1923" @es
- description "En este barrio cuyo origen data del primer cuarto del siglo XX, se encuentran varias vaquerías y alfarerías, que quizás justificaba su proximidad con la Dehesa de la Villa y zonas de huertos como Valdeconejos o la Huerta del Obispo. El edificio en que se emplaza esta obra es una antigua vaquería, obra muy significativa de la arquitectura prerrenacentista de Gustavo Fernández Balbuena, y en su fachada, muy escuetamente decorada con enmarcados de ladrillo aplastillado, aparecen dos figuras de cerámica en alto relieve representando dos lecheras, sin que se conozca su autoría, aunque son atribuidas por algunos autores a la Escuela de Cerámica de la Manzanilla, entonces dirigida por Francisco Alcántara y su hijo Jacinto Alcántara @es"
- address "CALLE FRANCOS RODRÍGUEZ 42"
- infoURL "<https://patrimonioypaisaje.madrid.es/FrameWork/generacionPDFMonumenta/400132.pdf?idMonumento=a258091d1b9c4510091d1b9c45102e085a0aRCRD&tipoMon=M>" @xs
- name "Lecheras"
- postalCode "28039"

Figure 20: Entidad Ornamentación

Active ontology | Entities | Object properties | Data properties | Individuals by class | DL Query | OntoGraf | Individuals: SculpturalGroup_400121 ≡ SculpturalGroup_400121 — SculpturalGroup_400121 — http://www.semanticweb.org/vic/ontologies/2024/10/monuments/SculpturalGroup_400121

Description: Scul | Property assertions: SculpturalGroup_400121

Object property assertions:

- islocatedInNeighborhood Neighborhood_GUINDALERA
- hasAuthor Author_LAIZ_CAMPOS_Emilio'
- islocatedInDistrict District_SALAMANCA
- islocatedInCity City_Madrid

Data property assertions:

- PK "400121"
- infoURL "<https://patrimonioypaisaje.madrid.es/FrameWork/generacionPDFMonumenta/400121.pdf?idMonumento=af28091d1b9c4510091d1b9c45102e085a0aRCRD&tipoMon=M>" @~
- description "Este monumento, realizado por el escultor madrileño Emilio Laiz Campos (1917 - 1982) e instalado en el patio del Museo Taurino de la Plaza Monumental de las Ventas, fue promovido por la Diputación Provincial como reconocimiento a los toreros que participan desinteresadamente en las corridas benéficas organizadas por aquella a favor del Hospital Provincial; inaugurándose el 15 de junio de 1967 con la asistencia del propio presidente de la Diputación, González Bueno, y la totalidad de los diputados miembros, incluido el Sr. Mateos, visitador del Hospital, así como D. Llino Stuyck y el señor Escanciano en representación de la Empresa que gestionaba la plaza en aquel momento." @es
- address "CALLE AL CALA 23"
- name "Toreros de la Beneficencia"
- date "1967" @es
- postalCode "28028"

Figura 21: Entidad Grupo Escultórico

The screenshot shows the OntoGraf interface with the following details:

- Individuals:** Statue_400000 — Statue:400000 — http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Statue_400000
- Description:** Statue_400000
- Property assertions:**
 - hasAuthor: Author_PINTO_MALDONADO_Luis'
 - islocatedInNeighborhood: Neighborhood_CIUDAD_UNIVERSITARIA
 - islocatedInDistrict: District_MONCLOA-ARAVACA
 - hasGeometry: Geometry_300000
 - islocatedInCity: City_Madrid
- Data property assertions:**
 - infURI: "https://patrimonioypaisaje.madrid.es/sites/v/index.jsp?vgnextchannel=8fac3cb702aa4510VgnVC1M000008a4900aRCRD&vgnextoid=ea18091d1b9c4510091d1b9c45102e08"
 - date: '1973 @es'
 - description: "En el jardín anterior del Colegio Mayor Colombiano Miguel Antonio Caro, se sitúa este busto con pedestal, en un lateral del acceso, enfrentado a otro de forma y ejecución similar que, en el lateral derecho, representa al que fuera presidente de Colombia y quien da nombre al centro. Fue erigido a instancias del Instituto Colombiano de Educación Técnica en el Exterior, como el de Miguel Antonio Caro, y en la década de los setenta, y, aunque lleva la firma del escultor, L. Pinto Maldonado, acompañada de la fecha, el último dígito se borradó, si bien parece corresponder a un tres, lo que sería lógico dado que 1873 fue el año de nacimiento de este erudito, escritor y político colombiano, oriundo de Popayán y fallecido en Bogotá en 1943. El centenario de su muerte, en 1973, originó la iniciativa de la administración pública para su instalación en el jardín anterior del Colegio Mayor. Fue elaborado por el escultor Luis Pinto Maldonado, que realizó una serie de bustos de personajes ilustres de la cultura y la política colombiana, entre ellos su nombramiento como académico de la Lengua y de la Historia, pero también representaría a Colombia en diversos acontecimientos, rechazando el cargo de ministro de Gobierno. El autor de la obra, el artista bogotano Luis Pinto Maldonado, fue profesor en la Escuela de Bellas Artes de Bogotá entre las décadas de 1930 y 1950 y se especializó en la realización de esculturas conmemorativas, ubicadas en espacios públicos de Sudamérica, EE.UU., Japón y Europa. Recibió en 1957 el premio 'Guillermo Valencia' por el busto del maestro que se encuentra en el patio de la Universidad del Cauca, en Popayán."
 - infoURI: "https://patrimonioypaisaje.madrid.es/sites/v/index.jsp?vgnextchannel=8fac3cb702aa4510VgnVC1M000008a4900aRCRD&idMonumento=ea18091d1b9c4510091d1b9c45102e085a0aRCRD&ipoMon=M^~"
 - address: 'Avda Señora 6'
 - PK: '400000'
 - name: 'Maestro Guillermo Valencia'
- Annotation properties:**
- Datatypes:**
- Classes:** owl:Thing, AdministrativeDivision, Author, City, Country, District, GeometricObject
- Object properties:** asserted

Figura 22: Entidad Estatua

The screenshot shows the OntoGraf interface with the following details:

- Individuals:** UniqueBuilding_408912 — UniqueBuilding_408912 — http://www.semanticweb.org/vic/ontologies/2024/10/monuments/UniqueBuilding_408912
- Description:** UniqueBuilding_408912
- Property assertions:**
 - hasAuthor: Author_RODRIGUEZ_DE_PARTEARROYO_CONDE_Francisco'
 - islocatedInNeighborhood: Neighborhood_CANILLAS
 - hasGeometry: Geometry_408912
 - islocatedInDistrict: District_HORTALEZA
 - islocatedInCity: City_Madrid
- Data property assertions:**
 - date: '1986 @es'
 - address: 'C Aconcagua'
 - name: 'Templo Villarrosa'
 - infoURI: "https://patrimonioypaisaje.madrid.es/sites/v/index.jsp?vgnextchannel=8fac3cb702aa4510VgnVC1M000008a4900aRCRD&vgnextoid=c1e8091d1b9c4510091d1b9c45102e08"
 - description: "Elemento ornamental del parque de Villarrosa, ubicado aproximadamente en su centro, que corresponde a las obras realizadas durante la segunda fase de ejecución del recinto, llevada a cabo entre 1986 y 1987, aunque el anteproyecto original data de 1981. Esta relacionada por composición y uso de elementos propios de los, vigentes en la época de su implantación, con la Fuente en cascada, reloj de sol, río y monumento a Tierno Galván pudiendo considerarse un rico conjunto debido a la idea unitaria del mismo arquitecto." @es
 - infoURI: "https://patrimonioypaisaje.madrid.es/sites/v/index.jsp?vgnextchannel=8fac3cb702aa4510VgnVC1M000008a4900aRCRD&idMonumento=c1e8091d1b9c4510091d1b9c45102e085a0aRCRD&ipoMon=M^~"
 - PK: '408912'
- Annotation properties:**
- Datatypes:**
- Classes:** owl:Thing, Object properties
- Object properties:** asserted

Figura 23: Entidad Edificación Singular

The screenshot shows the OntoGraf interface with the following details:

- Individuals:** Neighborhood_ACACIAS — Neighborhood_ACACIAS — http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Neighborhood_ACACIAS
- Description:** Neighborhood_ACACIAS
- Property assertions:**
 - neighborhoodHasMonument: Ornamentation_409166
 - neighborhoodHasMonument: Fountain_409911
 - neighborhoodHasMonument: Fountain_409912
 - neighborhoodHasMonument: Fountain_409910
 - neighborhoodHasMonument: AbstractSculpture_409165
 - neighborhoodHasMonument: AbstractSculpture_409168
 - neighborhoodHasMonument: MemorialElement_409169
 - neighborhoodHasMonument: Statue_408917
 - neighborhoodHasMonument: Fountain_409279
 - neighborhoodHasMonument: Fountain_409158
 - neighborhoodHasMonument: Fountain_409035
 - isNeighborhoodOf: District_ARQUINZUELA
 - neighborhoodHasMonument: MemorialElement_408245
 - neighborhoodHasMonument: Fountain_409919
 - neighborhoodHasMonument: Ornamentation_409333
 - neighborhoodHasMonument: MemorialElement_409249
 - neighborhoodHasMonument: MemorialElement_408246
 - neighborhoodHasMonument: Fountain_409280
- Data property assertions:**
 - name: 'ACACIAS'
- Negative object property assertions:**
- Negative data property assertions:**
- Annotation properties:**
- Datatypes:**
- Classes:** owl:Thing, AdministrativeDivision, Author, City, Country, District, GeometricObject, Monument, Neighborhood
- Object properties:** asserted

Figura 24: Entidad Barrio

4.6 Evaluación

Una vez poblada la ontología mediante el *script* de importación, el razonador Pellet muestra que es coherente:

```
----- Running Reasoner -----
Pre-computing inferences:
  - class hierarchy
  - object property hierarchy
  - data property hierarchy
  - class assertions
  - object property assertions
  - same individuals
Ontologies processed in 606 ms by Pellet
```

Figura 25: Resultado del razonador Pellet

La ontología cumple con los requisitos especificados para el producto final. También las consultas creadas satisfacen los requisitos que se esperaba que cumplieran las consultas, como puede verse en detalle en el [apartado 6](#).

5 Implementación

El *plugin* de SPARQL tiene un conocido *bug* para la versión 5.6 de Protégé que impide escribir en la caja de consultas, por lo que se ha intentado con el *plugin* alternativo Snap SPARQL Query, pero funciona bastante mal, por lo que finalmente se ha optado por importar la ontología en Apache Jena Fuseki y ejecutar ahí las consultas. Ciertamente, Jena Fuseki era la herramienta que finalmente debía emplearse para las consultadas federadas y las geoespaciales, pero hubiera ahorrado tiempo poder ejecutar directamente en Protégé las consultas que sólo dependen de la información contenida en la ontología.

Cuando fue el momento de abordar las consultas geoespaciales surgió un grave imprevisto: no fue posible habilitar completamente GeoSPARQL en Jena Fuseki, de manera que las consultas con funciones de distancias no daban error pero tampoco mostraban resultados. Después de indagar, resultó que el problema radicaba en que Jena Fuseki no calculaba las distancias entre puntos, es decir, la función `geof:distance` no devolvía nada. Esto es debido a que Jena Fuseki no soporta de forma nativa las funciones de GeoSPARQL, por lo que se debe instalar y configurar un módulo. Lamentablemente, no fue posible completar la configuración, incluso dedicándole bastante más tiempo del planificado, debido a la escasa documentación para una tarea más compleja de lo que cabía esperar.

En consecuencia, se optó por Virtuoso Universal Server, que además de ser un *middleware*, en la parte que interesa para este trabajo es también un servidor de diferentes tipos de bases de datos, entre ellos relacionales, orientadas a objetos, XML y RDF. Aunque inicialmente Virtuoso se

consideró fuera del alcance del presente trabajo, tal y como se comentó, pues su capacidad de trabajar a un elevado rendimiento con un gran volumen de datos almacenados en diferentes tipos de bases de datos conlleva una mayor complejidad para su instalación y configuración. La existencia de una versión de código abierto del mismo inclinó la balanza a su favor frente a otras opciones comerciales como Stardog, siguiendo así la filosofía de todo este trabajo. Sin embargo, se descubrió posteriormente que implementa sus propias funciones para las distancias en vez de las definidas en el estándar GeoSPARQL, lo cual es un hecho un tanto inesperado en un software con versión de código abierto. No obstante, dichas funciones propietarias, desde un punto de vista externo, tienen un uso muy similar, ya que únicamente cambia su nombre, tal y como se muestra en el [subapartado 6.3.](#)

5.1 Requisitos de instalación

Como se ha dicho previamente, los requerimientos de hardware son bajos. A nivel de software, el requisito indispensable es tener instalado Java SDK. El sistema sobre el que se ha ejecutado el software para el presente trabajo emplea OpenJDK y ha funcionado perfectamente. No existe ningún requerimiento respecto al sistema operativo, pues tanto Protégé como Apache Jena Fuseki emplean Java, que es prácticamente omnipresente en todos los sistemas. Virtuoso no lo emplea, pero puede instalarse en Windows, MAC OS y existen también paquetes para las diferentes distribuciones de Linux.

5.2 Instrucciones de instalación

5.2.1 Protégé

Su instalación es muy sencilla, pues desde el sitio web de Protégé se puede [descargar](#) el fichero tar.gz que lo contiene. Si bien hay una versión que incluye el Java Runtime Environment (JRE), que hace innecesario que Java esté instalado en el sistema, Jena Fuseki sí lo requiere, por lo que mejor no instalar esta versión.

5.2.2 Apache Jena Fuseki

La instalación también es muy sencilla, se puede [descargar](#) el servidor SPARQL en un fichero tar.gz. Una vez descomprimido, en el directorio de destino debe ejecutarse un *script* que lanzará el servidor:

```
./fuseki-server
```

Una vez el servicio está activo, se puede acceder al servidor mediante un navegador web en la URL <http://localhost:3030/>

5.2.3 Virtuoso Open-Source Edition

La instalación de Virtuoso depende del sistema operativo, el procedimiento para cada sistema está explicado en la documentación [10]. Aquí se van a explicar sólo los puntos clave del proceso. En este caso, al instalarse en un sistema operativo Ubuntu, se empleó *apt*, el gestor de paquetes típico de todas las distribuciones basadas en Debian GNU/Linux. Los paquetes de Virtuoso ya están en los repositorios, por lo que se deben instalar los paquetes *virtuoso-opensource*, *virtuoso-vad-isparql*, *virtuoso-vad-ods*, *virtuoso-vad-cartridges* y opcionalmente *virtuoso-vad-tutorial* mediante la orden “*apt install*”. La instalación del primer paquete requiere que se introduzca una contraseña para los usuarios *dba* y *dav*, si se deja en blanco el servicio de Virtuoso no se iniciará automáticamente al finalizar la instalación.

Para poder importar la ontología, debe modificarse el fichero de configuración de Virtuoso para conceder permisos de lectura al directorio donde está ubicado el fichero de la ontología. A continuación, ya se puede ejecutar el binario *isql*, pero debe tenerse en cuenta que Ubuntu lo ha renombrado a *isql-vt* para evitar un conflicto con un binario de unixODBC del mismo nombre. El acceso con identificación es necesario para luego poder realizar la importación:

```
isql-vt 1111 dba laContraseñaDeEsteUsurio
```

Una vez dentro, sólo queda importar la ontología. Nótese la peculiaridad de que no es necesario crear previamente un grafo para la ontología:

```
DB.DBA.TTLP_MT(file_to_string_output('/home/vic/Documents/UOC/TFG/ontologias/populated_monuments.ttl'), '', 'monuments');
```

Para poder realizar consultas es necesario acceder a la URL <http://localhost:8890/conductor> y validarse con el usuario *dba* y su contraseña.

6 Demostración

En el [subapartado 1.4.1](#) se especificaron los requisitos que deben cumplir las consultas para demostrar el potencial de esta ontología y de los datos enlazados. A continuación, se mostrarán las consultas que satisfacen los requisitos, así como sus resultados.

6.1 Consultas a la ontología

La primera y más básica de las consultas probablemente sea encontrar un monumento a partir de su nombre:

The screenshot shows a SPARQL query interface with the following details:

- Query URL:** /monuments/query
- Format:** JSON (selected), also available in Turtle.
- Query Content:**

```

1 #Monumento a partir de su nombre:
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 SELECT ?monument ?date ?address ?description
7 WHERE {
8   ?monument :name "Conjunto escultórico parque Arquitecto Antonio Palacios" ;
9   :date ?date ;
10  :address ?address ;
11  :description ?description ;
12  rdf:type ?type .
13  ?type rdfs:label ?typeLabel
14 }

```
- Results:**

monument	date	address	description
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/AbstractSculpture_408984 >	"Cercano a 1990"@es	CALLE FENELON 17	"El conjunto escultórico está dedicado al arquitecto gallego Antonio Palacios, que modernizó la ciudad de ..."
- Page Options:** Simple view, Ellipse, Filter query results, Page size: 50.

Figura 26: Monumento por su nombre

Mediante la palabra clave PREFIX se asocia una etiqueta con un URI [11]. Por ejemplo, la siguiente asociación:

PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

Asocia la etiqueta “rdf” con todo el vocabulario de RDF, es decir, da acceso al espacio de nombres (*namespace*) de RDF. De este modo, el código de la consulta resulta mucho más legible, pues esta tediosa línea:

<<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>> ?type

Se resume en un elegante “rdf:type ?type”.

La siguiente expresión, sin etiqueta, es sintácticamente correcta:

PREFIX : <<http://www.semanticweb.org/vic/ontologies/2024/10/monuments>>

Su función es dar acceso al espacio de nombres de la ontología de monumentos.

Una forma más práctica de buscar por el nombre es hacerlo a partir del nombre aproximado:

The screenshot shows a SPARQL query interface with the following details:

- Query URL:** /monuments/
- Format:** JSON (selected), also available in Turtle.
- Query Content:**

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4
5 SELECT ?monument ?typeLabel ?name ?date ?address ?description
6 WHERE {
7   ?monument :name ?name ;
8   :date ?date ;
9   :address ?address ;
10  :description ?description ;
11  rdf:type ?type .
12  ?type rdfs:label ?typeLabel
13  FILTER(CONTAINS(LCASE(?name), "parque arquitecto antonio palacios")) .
14 }

```
- Results:**

monument	typeLabel	name	date	address	description
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/AbstractSculpture_408984 >	AbstractSculpture	Conjunto escultórico parque Arquitecto Antonio Palacios	"Cercano a 1990"@es	CALLE FENELON 17	"El conjunto escultórico está dedicado al arquitecto gallego Antonio Palacios, que modernizó la ci..."
- Page Options:** Simple view, Ellipse, Filter query results, Page size: 50.

Figura 27: Monumento por nombre aproximado

A continuación, se muestran las consultas que satisfacen los demás requisitos estipulados para las consultas y que no dependen de contenido externo:

The screenshot shows a SPARQL query interface with the following details:

- Query URL:** /monuments/query
- Format:** JSON (selected) and Turtle (available)
- Query Content:**

```

1 #Todos los monumentos de un distrito:
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 SELECT ?monument ?name
7 WHERE {
8   ?district :name "CENTRO" .
9   ?monument :isLocatedInDistrict ?district ;
10  :name ?name .
11 }
12

```
- Results:**

monument	name
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Statue_408033 >	Agustín Lara
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_408129 >	Miguel de Cervantes
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_408509 >	Mateo Inurria
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Fountain_408089 >	Fuentecillas del Prado
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Statue_408022 >	Felipe III
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Fountain_408078 >	Fuente mural en la Cuesta de la Vega
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Statue_408013 >	Luigi Boccherini
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_408123 >	Calderón de la Barca
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_409529 >	José Zorrilla y Moral
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Statue_408002 >	Alonso II
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Statue_408042 >	Emilia Pardo Bazán
- Table Headers:** Table, Response, 364 results in 0.072 seconds.
- Filter Options:** Simple view, Ellipse, Filter query results, Page size: 50.

Figura 28: Todos los monumentos de un distrito

/monuments/query

```

1 # Todos los monumentos del barrio Goya con sus propiedades:
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 SELECT ?monument ?name ?typeLabel ?date ?address ?description ?postalCode
7 WHERE {
8   ?neighborhood :name "GOYA".
9   ?monument :isLocatedInNeighborhood ?neighborhood .
10  ?rdf:type ?type .
11  ?type rdfs:label ?typeLabel .
12  ?monument :name ?name .
13  # Cada propiedad en su propio bloque OPTIONAL. De lo contrario,
14  # si una no está, no se mostrarán las demás propiedades.
15  OPTIONAL { ?monument :postalCode ?postalCode . }
16  OPTIONAL { ?monument :description ?description . }
17  OPTIONAL { ?monument :date ?date . }
18  OPTIONAL { ?monument :address ?address . }
19 }

```

Table Response 10 results in 0.048 seconds

monument	name	typeLabel	date	address	description	postalCode
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_40...	Camillo José Cela	MemorialElement	"1996"@es	C Alcalá 185	"El insigne escritor de Padrón, académico, premio ..."	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_40...	Consuelo Rubio	MemorialElement	"1985"@es	C O'Donnell 11	"Cuatro años después de su temprana muerte de la...	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/AbstractSculpture_40...	Olivio en bronce	AbstractSculpture	"2005"@es	Pza Dali	"La remodelación de la plaza de Dalí y avenida de F...	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Statue_409189>	Francisco de Goya	Statue	"1998"@es	C Alcalá	"Esta escultura de Goya es ya el quinto monumento...	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_40...	Alejandro Busuioceanu	MemorialElement	"1968"@es	C General Pardiñas 3	"En 1968, el Instituto de Cultura Hispánica, represe...	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_40...	Rafael de León	MemorialElement	"Posterior a 1982"@es	CALLE MAIQUEZ 12	"Recordando el lugar en que murió en Madrid el es... 28009"	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Ornamentation_4095...	Losas de bronce sobre p...	Ornamentation	"2005"@es	Pza Dali	"Este conjunto de lápidas incrustadas en el pavime...	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/Fountain_409578>	Fuente de los Libros	Fountain	"2005"@es	Pza Dali	"La remodelación de la plaza de Dalí y avenida de F...	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/MemorialElement_40...	Federico García Lorca	MemorialElement	"1984"@es	C Alcalá 96	"El Teatro Español promovió la instalación de esta l...	
< http://www.semanticweb.org/vic/ontologies/2024/10/monuments/SculpturalGroup_408...	Newton y dolmen de Dalí	SculpturalGroup	"1986"@es	Pza Dali	"La primera idea de esta escultura surgió de una co...	

Showing 1 to 10 of 10 entries

Figura 29: Todos los monumentos de un barrio

/monuments/query

```

1 #Monumentos construidos en un año aproximado:
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 SELECT ?monument ?typeLabel ?name ?date ?address ?description
7 WHERE {
8   ?monument :name ?name ;
9   ?date ?date ;
10  ?address ?address ;
11  ?description ?description ;
12  ?rdf:type ?type .
13  ?type rdfs:label ?typeLabel
14  FILTER(CONTAINS(LCASE(?date), "1990")) .
15 }

```

Table Response 34 results in 0.059 seconds

monument	typeLabel	name	date	address	description
< http://www.semanticweb.org/vic/ontolog...	Fountain	Fuente del paseo Pintor Rosales	"1990"@es	Pº Pintor Rosales	"Se instaló en 1990, por el Departamento de Vías Públicas del Ayuntamiento de Madrid."
< http://www.semanticweb.org/vic/ontolog...	Statue	La Violetera	"1990"@es	Pza Gabriel Miró 1	"El escultor decidió rendir homenaje al compositor José Padilla a través de esta obra."
< http://www.semanticweb.org/vic/ontolog...	Fountain	Fuente en la plaza del Pardo (antigua plaza del Caudillo)	"1990"@es	Pza del Pardo	"Fuente instalada como elemento ornamental de la plaza en la que se encuentra la estatua del Caudillo."
< http://www.semanticweb.org/vic/ontolog...	AbstractSculpture	Cinco obeliscos Centro Comercial Arturo Soria Plaza	"Cercano a 1990"@es	C Arturo Soria 126	"Cuando a finales de los años ochenta se construyó el Centro Comercial Arturo Soria, se instaló esta fuente."
< http://www.semanticweb.org/vic/ontolog...	Fountain	Fuente de la escultura 'Ventana a Madrid'	"Cercano a 1990"@es	Pº Quince de Mayo	"La fuente fue instalada en la década de los 90 como parte del conjunto de esculturas 'Ventana a Madrid'."
< http://www.semanticweb.org/vic/ontolog...	MemorialElement	Eduardo Lozano y Ponce de León	"1990"@es	Pza Doctor Lozano	"Monolito conmemorativo dedicado a la memoria del ilustre doctor, catedrático y político Eduardo Lozano y Ponce de León."
< http://www.semanticweb.org/vic/ontolog...	Fountain	Fuentes en la plaza de Ciudad Lineal	"Cercano a 1990"@es	C Alcalá 416	"En este lugar, hoy conocido como plaza de la Ciudad Lineal, se instaló esta fuente en 1990."
< http://www.semanticweb.org/vic/ontolog...	SculpturalGroup	Cástor y Pólux	"1990"@es	C Villablanca 41	"Elemento ornamental ubicado en un pequeño parque frente a un grupo de esculturas dedicadas a los hermanos Cástor y Pólux."
< http://www.semanticweb.org/vic/ontolog...	AbstractSculpture	Conjunto escultórico parque Arquitecto Antonio Palacios	"Cercano a 1990"@es	CALLE FENELON 17	"El conjunto escultórico está dedicado al arquitecto gallego Antonio Palacios."
< http://www.semanticweb.org/vic/ontolog...	MemorialElement	Centro Básico de Servicios Sociales y 3 ^{er} Edad	"1990"@es	C Villalonso 12	"Lápida conmemorativa de la inauguración del centro por parte del alcalde de Madrid."
< http://www.semanticweb.org/vic/ontolog...	MemorialElement	Muro de Berlín	"1990"@es	PLAZA VIRGEN GUADALUPANA	"El parque de Berlín se realizó en 1967, cuando se enlazaron los tramos de la antigua muralla de Berlín."
< http://www.semanticweb.org/vic/ontolog...	Fountain	Fuente elíptica en la avenida de Logroño	"Cercano a 1990"@es	Avda Logroño 183	"Fuente ornamental, situada en una glorieta frente al antiguo Castillo de la Mota."
< http://www.semanticweb.org/vic/ontolog...	MemorialElement	Aula de Informática Reina Sofía	"1990"@es	C Camino de la Arboleda	"Placa conmemorativa de la inauguración el día 19 de junio de 1990, en el marco de la Exposición Universal de Madrid."
< http://www.semanticweb.org/vic/ontolog...	Fountain	Fuente circular en la glorieta Marques de Vadillo	"1990"@es	Gta Marqués de Vadillo	"La fuente, obra de los canteros municipales del Ayuntamiento de Madrid."

Figure 30: Monumentos construidos en un año aproximado

The screenshot shows a SPARQL query interface with the following details:

- Query URL:** /monuments/query
- Format:** JSON (selected from a dropdown menu)
- Result Format:** Turtle (selected from a dropdown menu)
- Query Content:**

```

1 #Todos los monumentos de un/a autor/a;
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments/>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 SELECT ?monument ?name ?typeLabel ?date ?address ?postalCode ?description
7 WHERE {
8   ?author :name "ANTONINI, Lucía" .
9   ?monument :hasAuthor ?author ;
10  ?monument :type ?type .
11  ?type rdfs:label ?typeLabel .
12  ?monument :name ?name .
13  OPTIONAL { ?monument :postalCode ?postalCode . }
14  OPTIONAL { ?monument :description ?description . }
15  OPTIONAL { ?monument :date ?date . }
16  OPTIONAL { ?monument :address ?address . }
17 }

```
- Results:**
 - Table View:** Shows 1 result in 0.028 seconds.
 - Columns:** monument, name, typeLabel, date, address, postalCode, description.
 - Row Data:**

monument	name	typeLabel	date	address	postalCode	description
<http://www.semanticweb.org/vic/ontologies/2024/10/monum...>	El buzón de las palabras	SculpturalGroup	"2005"@es	Pza Carabanchel		"Este pequeño-gran monumento se vincula a un emblemático lugar situado en...

Figura 31: Todos los monumentos de un/a autor/a

6.2 Consultas federadas

Las consultas más interesantes son aquellas que demuestran las posibilidades de la web semántica para enriquecer la base de conocimiento de la aplicación con datos enlazados de otras fuentes en tiempo real. Como se ha explicado previamente, se empleará Wikidata para este propósito.

En la ontología existe el monumento a Hernán Cortés que servirá de ejemplo. La propiedad PK de este monumento tiene el valor 400070, esta propiedad servirá para enlazar con el mismo monumento en Wikidata y obtener la siguiente información adicional del personaje representado: enlace al recurso de Wikidata del personaje homenajeado, fecha y lugar de su nacimiento, enlace a imagen y nacionalidad. La información que se va a mostrar contenida en la ontología es el nombre, el tipo de monumento, la fecha de construcción, la dirección y la descripción. Esta es la consulta:

```

1 v PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments/>
2 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 SELECT ?name ?typeLabel ?date ?address ?depictedPerson ?depictedPersonImage
7     ?depictedPersonCitizenshipLabel ?dateOfBirth ?placeOfBirth ?description
8 WHERE {
9     BIND("400070" AS ?catalogCode) # catalogCode = "400070"
10    # Desde la ontología de monumentos:
11    ?monument :name ?name ;
12        :PK ?catalogCode ;
13        rdf:type ?type .
14        ?type rdfs:label ?typeLabel .
15        OPTIONAL { ?monument :description ?description . }
16        OPTIONAL { ?monument :date ?date . }
17        OPTIONAL { ?monument :address ?address . }
18
19    # Desde Wikidata
20    SERVICE <https://query.wikidata.org/sparql> {
21        ?wikidataMonument wdt:P528 ?catalogCode ; # Propiedad catalogCode de Wikidata
22            wdt:P180 ?depictedPerson .
23        OPTIONAL {
24            ?depictedPerson wdt:P569 ?dateOfBirth .
25            ?depictedPerson wdt:P19 ?placeOfBirth .
26        }
27        OPTIONAL {
28            ?depictedPerson wdt:P18 ?depictedPersonImage
29        }
30        OPTIONAL {
31            ?depictedPerson wdt:P27 ?depictedPersonCitizenship .
32            ?depictedPersonCitizenship rdfs:label ?depictedPersonCitizenshipLabel .
33            FILTER(LANG(?depictedPersonCitizenshipLabel) = "es")
34        }
35    }
36 }

```

Figura 32: Consulta que complementa la información de la ontología con la de Wikidata

Seguidamente, se explican aspectos importantes de esta consulta. En primer lugar, el acceso a las propiedades de Wikidata mediante la etiqueta wdt se consigue mediante el siguiente prefijo:

PREFIX wdt: <<http://www.wikidata.org/prop/direct/>>

Así, para acceder a la propiedad P180 (representa a), no es necesario escribir toda la URL sino que basta con “wdt:P180”. Sin embargo, el acceso a la Wikidata se consigue mediante el empleo de la cláusula SERVICE:

SERVICE <<https://query.wikidata.org/sparql>>

Todos los campos extraídos de Wikidata están contenidos en esta cláusula. Por otro lado, mediante BIND se asigna 400070 a la variable *catalogCode*. Esta variable se emplea para seleccionar el monumento tanto en la parte de la consulta sobre la ontología local como en la parte sobre Wikidata, es decir, conecta los datos de la ontología con los de Wikidata.

Este es el resultado:

name	typeLabel	date	address	depictedPerson	depictedPersonImage	depictedPersonCitizenshipLabel	dateOfBirth	placeOfBirth	description
1 Hernán Cortés	Statue	"1980"01	Avda Séneca 4	<http://www.wikidata.org/entity/Q7326>	<http://commons.wikimedia.org/wiki/Special:FilePath/Retrato%20de%20Hern%C3%A1n%20Cort%C3%ADo.jpg>	"España"01	"1485-01-01T00:00:00Z"01	<http://www.wikidata.org/entity/Q668384>	"Busto de Hernán Cortés inspirado ...

Figura 33: Resultado consulta federada

En el código de la consulta puede observarse que para algunos campos se acude a la etiqueta o *label* (*rdfs:label*), mientras que en otros se emplea directamente el recurso. En los resultados de la consulta puede observarse la diferencia: Cuando no se emplea la etiqueta se muestra la URL al recurso. Por ejemplo, para *placeOfBirth* se muestra el recurso <https://www.wikidata.org/wiki/Q668384>, el cual es la población de Medellín, de la provincia de Badajoz, España. A partir de ahí, la aplicación podría obtener más información de este recurso. No obstante, si sólo interesa saber el nombre de la población, a la siguiente cláusula OPTIONAL:

```
OPTIONAL {
    ?depictedPerson wdt:P569 ?dateOfBirth .
    ?depictedPerson wdt:P19 ?placeOfBirth .
}
```

Se le deben añadir las siguientes dos líneas:

```
OPTIONAL {
    ?depictedPerson wdt:P569 ?dateOfBirth .
    ?depictedPerson wdt:P19 ?placeOfBirth .
    ?placeOfBirth rdfs:label ?placeOfBirthLabel .
    FILTER(LANG(?placeOfBirthLabel) = "es")}
```

Finalmente, sólo queda añadir *placeOfBirthLabel* a la cláusula SELECT. El filtro por idioma garantiza que la información se muestre en el idioma deseado por el usuario, evitando además que el resultado se duplique tantas veces como etiquetas tiene Wikidata en diferentes idiomas para esta población.

Un tipo de recurso muy interesante de obtener es las imágenes. Por ejemplo, la consulta devuelve una URL para *depictedPersonImage* que contiene una fotografía del personaje representado. Esta imagen podría emplearse en la parte cliente de una aplicación para acompañar los textos y así mejorar la experiencia de usuario.

6.3 Consultas geoespaciales

Estas consultas son especialmente útiles desde la perspectiva de la aplicación lado cliente implementada en un dispositivo móvil. Estos dispositivos disponen de la tecnología GPS integrada, por lo que son capaces de obtener las coordenadas de su ubicación. Generalmente, proporcionan una precisión de 5 decimales tanto para la longitud como para la latitud, lo cual teóricamente implica un margen de error de tan solo $\pm 1,1$ m. Si bien no son infrecuentes las ocasiones en las que

no logran ubicar realmente con tanta exactitud, siempre proporcionarán 5 decimales. En las consultas se van a emplear 4 decimales, lo cual implica un margen de error de unos $\pm 11,1$ m, suficiente para ubicar monumentos y otras obras de arte en la calle.

En el fichero CSV, la precisión de las coordenadas es de 14 decimales, llegando incluso a más en algunos casos. Semejante precisión tiene un margen de error por debajo del milímetro y probablemente sea excesiva para ubicar monumentos. No en vano, cuando se ha realizado la importación de la ontología en Virtuoso, éste ha reducido la precisión de las propiedades geo:wktLiteral al pasarlas a su formato propietario virtrdf:Geometry y las ha dejado en 13 decimales, lo cual sigue siendo más que suficiente para la ubicación de monumentos, pues sigue siendo una precisión superior al mm.

Cuando un usuario accede a la aplicación de su dispositivo móvil, además de las opciones de buscar monumentos por nombre, barrio, etc., sería muy interesante que también tuviera la opción de buscar el monumento donde está ubicado en ese momento. Para ello, la aplicación cliente enviaría las coordenadas de la ubicación a la aplicación servidor y ésta las emplearía para obtener cuál es el monumento ahí ubicado, o los monumentos.

Para saber el monumento o los monumentos cercanos se puede emplear la función geof:distance, la cual devuelve la distancia entre dos geometrías. La función equivalente de Virtuoso se llama bif:st_distance. A partir de una hipotética situación en la que un usuario ubicado en las coordenadas (-3.6042, 40.4076) quiere saber más acerca del monumento que está contemplando, esta es la consulta que proporciona la información:

The screenshot shows the Virtuoso SPARQL Execution interface. The top navigation bar includes links for Home, System Admin, Database, Replication, Web Application Server, XML, Web Services, Linked Data, and NNTP. Below this is a sub-navigation bar with SPARQL, Sponger, Statistics, Graphs, Schemas, Namespaces, Views, and Quad Store Upload. On the left, there's a sidebar with links for Documentation (web), Tutorials (web), Virtuoso Web Site, and OpenLink Software. Version information (0.7.20.3229, Build Jul 4 2024) is also present. The main area is titled "SPARQL Execution". It has tabs for Query and Saved Queries, with "Query" selected. The "Default Graph IRI" dropdown is set to "monuments". The query text is as follows:

```

PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX bif: <bif:if>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX : <http://www.semanticweb.org/vic/ontologies/2024/10/monuments#>

SELECT (bif:st_distance(?wkt, ?refPoint) AS ?distance) ?monument
?name ?typeLabel ?date ?address SUBSTR(?description, 1, 50) AS ?shortDescription ?geometry
FROM <http://www.semanticweb.org/vic/ontologies/2024/10/monuments>
WHERE {
  BIND(bif:st_point(-3.6042, 40.4076) AS ?refPoint)
  ?geometry geo:wkt(?wkt).
  FILTER(bif:st_distance(?wkt, ?refPoint) < 0.01) .
  ?monument :hasGeometry ?geometry ;
    :name ?name ;
    rdf:type ?type .
  OPTIONAL { ?monument :typeLabel . }
  OPTIONAL { ?monument :date ?date . }
  OPTIONAL { ?monument :address ?address . }
  OPTIONAL { ?monument :description ?description . }
}
ORDER BY ?distance
  
```

Below the query text are buttons for Execute, Save, Load, and Clear. The results table has columns: distance, monument, name, typeLabel, date, address, and shortDescription. One result is shown:

distance	monument	name	typeLabel	date	address	shortDescription
0.00761721	http://www.semanticweb.org/vic/ontologies/2024/10/monuments#SculpturalGroup_408967	"Cástor y Pólux"^^ http://www.w3.org/2001/XMLSchema#string	"SculpturalGroup"	"1999"@es	"C Villablanca 41"^^ http://www.w3.org/2001/XMLSchema#string	"Elemento ornamental"

Figura 34: Monumento por ubicación mediante st_distance

En este caso sólo hay un monumento y está a una distancia de 0,01 de la ubicación. A medida que se amplía al radio surgirán más resultados, siempre ordenados por la distancia del más cercano al más lejano. Por ejemplo, estos son los resultados con un radio de 0,05:

distance	monument	name	typeLabel	date	address	shortDescription	geometry
0.09761721	http://www.semanticweb.org/vic/ontologies/2024/10/monuments#SculpturalGroup_408966	"Cástor y Pólux"^^< http://www.w3.org/2001/XMLSchema#string >	"SculpturalGroup"	"1990" ges	"C Villablanca 41"^^< http://www.w3.org/2001/XMLSchema#string >	"Elemento ornamental ubicado en un pequeño parque f"@@es	http://www.semanticweb.org/vic/ontologies/2024/10/monuments#Geometry_408966
0.0346687	http://www.semanticweb.org/vic/ontologies/2024/10/monuments#SculpturalGroup_408966	"Leda y el cisne"^^< http://www.w3.org/2001/XMLSchema#string >	"SculpturalGroup"	"1990" ges	"CALLE VILLABLANCA 41"^^< http://www.w3.org/2001/XMLSchema#string >	"Elemento ornamental ubicado en un pequeño parque f"@@es	http://www.semanticweb.org/vic/ontologies/2024/10/monuments#Geometry_408966

Figura 35: Monumento por ubicación mediante st_distance con radio ampliado

Se han escogido algunas propiedades del monumento para mostrar a modo de ejemplo, pero se podrían escoger muchas más, también la PK y a partir de la misma hacer otra consulta federada para obtener más información desde Wikidata.

En esta ontología es fácil buscar todos los monumentos de un barrio o de un distrito. Sin embargo, no sería muy complejo brindar al usuario la posibilidad de encontrar todos los monumentos dentro del área que dibuje con su dedo en un mapa. En el lado del servidor, esta consulta se resolvería mediante la función geof:sfWithin, cuyo equivalente en Virtuoso es bif:st_within, pues permite evaluar si una geometría está dentro de un área definida por un polígono o por un círculo.

7 Conclusiones y líneas de futuro

7.1 Conclusiones

La temática de este proyecto no está en absoluto relacionada con el área en la que me desempeño profesionalmente desde hace años, el desarrollo de software, por lo que ha sido una experiencia que me ha permitido explorar las ontologías y los conceptos de la web semántica. Además, he podido conocer el nivel de desarrollo alcanzado en la actualidad, pues ha pasado mucho tiempo desde que oí hablar de la web semántica por primera vez, pero pocas veces han trascendido noticias acerca de sus avances.

Las lecciones aprendidas son varias. El diseño y desarrollo de una ontología requiere de conocimientos técnicos, conocimiento de su dominio y una planificación rigurosa. Además de adquirir conocimientos técnicos acerca de las ontologías (tendía a construir en mi mente analogías con el mundo de la programación orientada a objetos que aquí no aplican), he aprendido acerca de cómo construir las consultas SPARQL a partir de entender cómo se almacena la información en forma de triplets. Aunque hay ciertas similitudes entre SPARQL y SQL, una vez más, no se puede caer en analogías con conocimientos previos, pues poco tienen que ver las triplets con las bases de datos relacionales.

Acerca de los objetivos desarrollados, se puede afirmar que se han alcanzado todos los planteados inicialmente: la creación de la ontología con una perspectiva amplía para ser fácilmente extensible más allá de los datos de la villa de Madrid, la importación inicial de datos desde un fichero CSV, la ejecución de consultas para extraer información de la ontología, las consultas federadas para

combinar esta información con información externa proveniente de Wikidata y la realización de consultas geoespaciales.

En cuanto a la planificación, el enfoque iterativo de la metodología Methontology demostró ser la elección adecuada para el desarrollo de la ontología, pues la iteración entre varias fases facilitó la realización de ajustes y mejoras. Sin embargo, la carga de trabajo durante la implementación (PEC 3) fue significativa, lo que planteó dificultades para cumplir los plazos inicialmente establecidos, tal y como se había previsto en los riesgos. Un imprevisto importante fueron las dificultades técnicas con la integración de GeoSPARQL en Jena Fuseki para poder realizar consultas geoespaciales, ya que resultó más compleja de lo esperado y no pudo completarse satisfactoriamente. Este desafío condujo a introducir un cambio en el trabajo: el empleo de Virtuoso como alternativa para ejecutar consultas geoespaciales. Este software se descartó inicialmente a favor de Jena Fuseki debido al tamaño moderado de datos de este proyecto.

Si bien las consultas geoespaciales no están entre los objetivos principales de este proyecto, se ha trabajado arduamente hasta poder lograrlas, mostrando así la capacidad de flexibilizar la planificación inicial y la capacidad de adaptación ante dificultades técnicas. En definitiva, el proyecto ha alcanzado sus metas, constituye una base sólida para futuras ampliaciones y un excelente ejercicio práctico para integrar conceptos y tecnologías de la web semántica.

7.2 Líneas de futuro

Este proyecto presenta un amplio potencial de desarrollo y ampliación. A continuación, se describen diferentes líneas de futuro para ampliar el trabajo realizado:

- Ampliar la ontología con los datos abiertos de otras ciudades. La ontología resultante de este proyecto debe verse como un prototipo inicial que debe evolucionar incorporando datos de otras ciudades. Este proceso podría requerir ajustes y mejoras en la estructura de la ontología para adaptarla a nuevos tipos de datos o características específicas de ciudades de diferentes países. Estos cambios deberían hacerse iterando nuevamente por algunos de los pasos de la metodología Methontology.
- Desarrollar la aplicación, tanto la parte cliente como la parte servidor. Esta línea de futuro es evidente. La parte cliente de la aplicación permitirá a los usuarios acceder de manera sencilla a la información de la ontología mediante dispositivos móviles como teléfonos o tabletas. La parte servidor se encargará de devolver la información requerida en las solicitudes de los clientes mediante la construcción consultas SPARQL que obtendrán dicha información de la ontología, así como de servicios externos como Wikidata e incluso de más fuentes de datos enlazados.

- Integración en ontologías de dominio más general y en sistemas informáticos. Un paso significativo sería evaluar la integración de esta ontología en ontologías más amplias, como las relacionadas con turismo, urbanismo o geografía, así como evaluar si puede incorporarse en sistemas informáticos destinados a la gestión del patrimonio cultural, la planificación urbana, plataformas educativas o en aplicaciones turísticas. Cada una de estas evaluaciones que resulte positiva abriría una nueva línea de ampliación para el proyecto.

Bibliografía

- [1] «METHONTOLOGY: From Ontological Art Towards Ontological Engineering». [En línea]. Disponible en: https://oa.upm.es/5484/1/METHONTOLOGY_.pdf
- [2] «FAQ - Estructura de las enseñanzas - Calidad - (UOC)». Accedido: 15 de octubre de 2024. [En línea]. Disponible en: <https://www.uoc.edu/portal/es/qualitat/qualitat-titulacions/estructura-ensenyaments/faqs/faq-generals.html>
- [3] Robert Clarisó Viladrosa, Jordi Conesa i Caralt, y Robert Clarisó Viladrosa, «Ontologías y web semántica». Universitat Oberta de Catalunya.
- [4] «Ontología (informática)», *Wikipedia, la enciclopedia libre*. 14 de septiembre de 2024. Accedido: 31 de octubre de 2024. [En línea]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Ontolog%C3%ADA_\(inform%C3%A1tica\)&oldid=162454337](https://es.wikipedia.org/w/index.php?title=Ontolog%C3%ADA_(inform%C3%A1tica)&oldid=162454337)
- [5] «Monumentos de la ciudad de Madrid - Portal de datos abiertos del Ayuntamiento de Madrid». Accedido: 20 de octubre de 2024. [En línea]. Disponible en: <https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=eb8e993ae322b610VgnVCM1000001d4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default>
- [6] «Merging and Expanding existing Ontologies to cover the Built Cultural Heritage domain; case study Cuenca-Ecuador - KU Leuven». Accedido: 20 de noviembre de 2024. [En línea]. Disponible en: https://kuleuven.limo.libis.be/discovery/fulldisplay/lirias4086112/32KUL_KUL:Lirias
- [7] «Ontología», *Wikipedia, la enciclopedia libre*. 12 de noviembre de 2024. Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://es.wikipedia.org/w/index.php?title=Ontolog%C3%ADA&oldid=163548008>
- [8] «rdflib 7.1.1 — rdflib 7.1.1 documentation». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://rdflib.readthedocs.io/en/stable/>
- [9] «Google Colab». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://colab.research.google.com/>
- [10] «Virtuoso Open-Source Edition». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://vos.openlinksw.com/owiki/wiki/VOS>
- [11] «SPARQL 1.1 Query Language». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.w3.org/TR/sparql11-query/>
- [12] R. M. Kim Hamilton, *Learning UML 2.0*. [En línea]. Disponible en: <https://learning.oreilly.com/library/view/learning-uml-2-0/0596009828>
- [13] «Ontology Development 101: A Guide to Creating Your First Ontology». [En línea]. Disponible en: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf
- [14] *Methontology*. [En línea Video]. Disponible en: <https://tv.unir.net/videos/13955/47/106/2587/0/Methontology>
- [15] «MOD2-Ontologias-y-web-semantica».
- [16] Robert Clarisó Viladrosa, Jordi Conesa i Caralt, y Jordi Duran Cals, «Introducción a la representación del conocimiento». Universitat Oberta de Catalunya.
- [17] Robert Clarisó Viladrosa, «Introducción al trabajo final». Universitat Oberta de Catalunya.
- [18] José Ramón Rodríguez, «El trabajo final como proyecto». Universitat Oberta de Catalunya.

- [19] José Ramón Rodríguez, «La gestión del proyecto a lo largo del trabajo final». Universitat Oberta de Catalunya.