ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN MATEMÁTICAS

Curso Académico 2022/2023

Trabajo Fin de Grado

ANOTACIÓN SEMÁNTICA DE DATOS MEDIANTE EL MÉTODO CSV@RDF

Autora: Marta Gámez Valero

Directora: Paloma Cáceres García de Marina

RESUMEN

La Web Semántica es un proyecto creado como evolución de la World Wide Web a fin de complementar y expandir el uso de esta, aumentando también sus capacidades. A través de este proyecto, la web es capaz de entender la información que se le aporta, facilitando así al usuario unos resultados más exactos de su búsqueda. Este proyecto se desarrolla empleando ontologías, que son las responsables de la comprensión de datos e información por parte de la web.

Para poder usar la Web Semántica, se deben crear diferentes ontologías para añadir metadatos a los datos y así permitir a la web la comprensión de estos y mejorar la búsqueda y la información obtenida a partir de esta. En este TFG se realiza un proceso de anotación semántica de datos a partir del método CSV@RDF definido por el grupo de investigación VORTIC3 de la Escuela Técnica Superior de Ingeniería Informática. En este trabajo se siguen los pasos de dicho método, con el objetivo de identificar sus fortalezas y debilidades para añadir modificaciones y/o mejoras.

PALABRAS CLAVE:

Web Semántica, MDA, DSL, ontología, metadatos.

ABSTRACT

The Semantic Web is a project created as a World Wide Web's development to complement and expand its use, increasing its capabilities. Through this project, the web can understand the information provided and to give the user more accurate search results. This project has been developed using ontologies, which are the responsible for the understanding of data and information by the web.

To use the Semantic Web, different ontologies need to be created to add metadata to the data, allowing the web to understand that data and improve the search and, consequently, the information obtained. In this work, a semantic data annotation process has been carried out using the CSV@RDF method defined by the VORTIC3 research group at the Higher Technical School of Computer Engineering. The steps of this method have been followed, with the aim of identifying its strengths and weaknesses to add modifications and/or improvements.

KEYWORDS:

Semantic Web, MDA, DSL, ontology, metadata.

Índice:

Índice de figuras:

1. Introducción.

La plataforma web *DataReportal* ofrece cientos de datos e información acerca de las acciones online realizadas por las personas, ayudando así a empresas y organizaciones en la toma de decisiones.

El reciente informe "Digital 2022 April Global Statshot[1]" de dicha plataforma muestra que más del 60% de la población mundial tiene acceso a Internet y hace uso de ella y los que tienen acceso, además, pasan más de 3 horas al día conectados. Estos datos agrupan a más de 5 mil millones de usuarios.

De entre todos los medios digitales que nos rodean en la actualidad, Internet es el medio con un crecimiento más veloz y cuya evolución destaca sobre todos los demás medios. Sobre todo, la red informática mundial, la World Wide Web (WWW), ha evolucionado y se ha modernizado a pasos extraordinarios desde su creación, hace apenas dos décadas, en 1990.

La WWW fue creada con el objetivo de intercambiar información entre científicos de una misma o distinta institución. Tras el crecimiento experimentado desde su aparición, se ha convertido en una red de uso cotidiano para la sociedad, en la cual se puede encontrar todo tipo de documentos, imágenes, archivos e información.

A pesar de tratarse del medio más potente y amplio, aún queda mucho camino que recorrer ya que se trata de una novedad de la cual se puede encontrar nuevas funcionalidades. Este camino está en constante desarrollo, puesto que la red busca mejoras e innovaciones constantemente. Entre los últimos mecanismos que buscan mejorar la red actual y tener una gran repercusión en el futuro de esta, sobre todo en lo relacionado con la automatización de procesos, se encuentra la web semántica.

La WWW trabaja conectando información a través de diferentes algoritmos que analizan palabras clave y consultas, proporcionando unos resultados más o menos exactos. La Web Semántica va un paso más allá: optimizar el intercambio de información en la web. Así, la propia tecnología puede distinguir y procesar la información de búsqueda de una manera más eficaz e inteligente, puesto que entiende el significado de la información que contiene.

[1] https://datareportal.com/reports/digital-2022-april-global-statshot

De esta manera, la Web Semántica, siendo una extensión de la World Wide Web, se puede encargar de mecanismos ahorrando trabajo a las personas. Para ello, además de conectar información como hace la red actual, debe procesar el contenido semántico, es decir, aquellos significados legibles por máquinas, permitiendo a estas comprender y distinguir significados de los diferentes contenidos que son capaces de encontrar.

Se trata de un mecanismo muy eficiente ya que proporcionan resultados más adecuados y exactos para la búsqueda realizada, y no solamente analiza textos, sino todos los objetos dotados de significado (imágenes, sonido, números, símbolos, …).

Por ejemplo, si se desea buscar información sobre una persona reconocida mundialmente, la web semántica no solo nos ofrece información básica como su edad y su profesión, sino que también ofrece datos de su biografía, su trayectoria profesional u otros aspectos que puedan interesar al usuario de esta persona. Además, la web semántica, al contrario que la web actual, no ofrece resultados irrelevantes como puede ser noticias de otros famosos que comparten profesión con nuestra búsqueda, anuncios de usuarios apodados con el mismo nombre que nuestra búsqueda, biografía e información acerca de otras celebridades que han podido compartir momentos puntuales con nuestra búsqueda, …

Sin embargo, para poder lograr estos mecanismos es necesario dotar de significado los medios necesarios para comprender la búsqueda solicitada y poder ofrecer la mejor respuesta. Para hacer esto posible se realiza un etiquetado de los contenidos (o metadatos), los cuales deben tener una semántica específica, la ontología.

Por este motivo, los responsables de la información en la web deben incluir los metadatos y la ontología de los datos concretos con el objetivo de que la máquina pueda comprender los datos y establecer relaciones entre ellos.

En este trabajo se lleva a cabo un conjunto de mejoras sobre un método ya definido de anotación semántica de datos, con el objetivo de acercar este método a toda persona interesada en emplearlo, sin necesidad de ser experto en el tema.

En el desarrollo de este documento se encuentra descrito el proceso de selección de datos sobre los cuales se trabaja y el modo de llevar a cabo el método. Además, se exponen las dificultades encontradas durante su desarrollo y posibles mejoras que se pueden aplicar para una mejor comprensión.

2. Objetivos del trabajo.

El objetivo principal de este trabajo es la adaptación y análisis de un método de anotación semántica de datos, denominado CSV@RDF, a partir de un conjunto de datos en formato Excel.

Este objetivo principal, a su vez, se divide en varios objetivos secundarios:

- OS_A. Analizar los datos seleccionados.
- OS_B. Verificar la validez del método.
- OS_C. Encontrar posibles dificultades del método.
- OS_D. Proponer mejoras para la evolución del método.

El objetivo secundario OS_A consiste en llevar a cabo un análisis de los datos originales sobre los que se quiere realizar la anotación semántica. Dicho análisis consistirá en un estudio de sus atributos (estructura, formato, contenido, significado) además de verificar que el conjunto total de dichos datos sea consistente en relación con dichos atributos.

El objetivo secundario OS_B trata de probar el método pudiendo verificar, o no, su validez, verificando la ontología generada.

El objetivo secundario OS_C y OS_D deriva del objetivo OS_B, puesto que al utilizar el método se podrán encontrar una serie de dificultades a las que se deberá hacer frente. Debido a dichas dificultades expuestas, se propondrá diferentes formas de ser solucionadas a fin de ser resueltas para una mejor versión futura de dicho método.

3. Estudios previos.

Este trabajo se inicia realizando diversos estudios previos acerca de todo lo que rodea a la web semántica y al método empleado. A pesar de conocer algunos conceptos, se tiene una idea muy ligera de ellos, por lo que se estudia en mayor profundidad previamente a la realización de este trabajo.

3.1. Arquitectura dirigida por modelos.

La arquitectura dirigida por modelos (MDA – Model Driven Architecture), junto a las ontologías que serán tratadas en próximos apartados, son recursos muy populares en el desarrollo de sistema de información (Sánchez, Cavero y Marcos, 2005).

MDA se trata de una propuesta desarrollada por el Object Management Group (OMG) basada en el uso de modelos para la especificación, diseño y construcción de sistemas de software. Esta arquitectura promueve el uso de modelos y transformaciones de modelos para el desarrollo de sistemas de software (Martínez, 2008).

Los modelos se utilizan como representaciones formales (es decir, cuenta con una semántica y sintaxis bien definida) del funcionamiento, comportamiento o estructura del sistema a construir (Raistrick, Francis, Wright, Carter y Wilkie, 2004).

Donoso (2013) hace referencia a la distinción de cuatro clases de modelos dentro de la MDA definiendo así su ciclo de vida. Para poder realizar el paso de un modelo a otro se realizan diferentes transformaciones los modelos deben usar un lenguaje bien definido para asegurar así una interpretación automática por parte de la máquina. El lenguaje recomendado por MDA es el Lenguaje de Modelado Unificado (UML).

A continuación se detallan las diferentes clases de modelos del ciclo de vida de la MDA:

- Modelo Independiente de la Computación (CIM): se describe el ámbito de uso del sistema a fin de contextualizar el problema. Hace referencia a la etapa de captura de requisitos y se conoce también como el modelo del dominio.
- Modelo Independiente de Plataforma (PIM): representa la lógica del sistema con gran nivel de abstracción a fin de poder ser usado independientemente de la plataforma utilizada para su implementación.

- Modelo Específico de Plataforma (PSM): deriva del anterior con especificaciones y detalles de una plataforma específica, aquella en la que se implementará.
- Código: se trata del código generado tras la codificación y las pruebas pertinentes para un correcto desarrollo

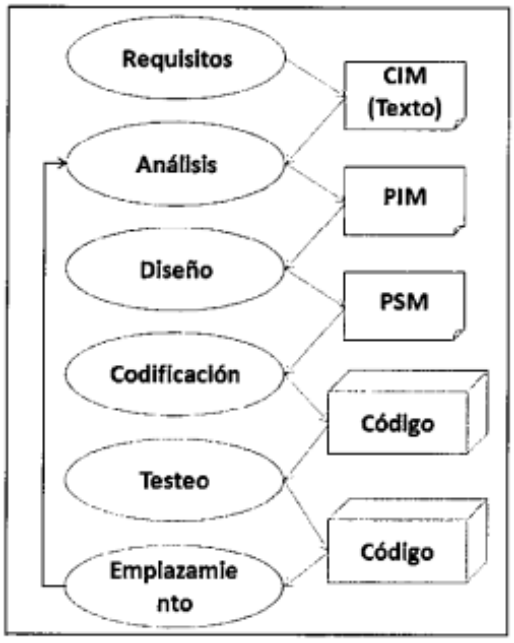


Figura 1. Ciclo de vida de desarrollo de software en MDA
Fuente: "Utilizando un enfoque dirigido por modelos, realizar un análisis y diseño de un sistema de gestión basado en BarBocca Pub & Restaurant", por Gonzalo Nícolas Donoso (2013)

3.2. Web semántica.

En primer lugar, es necesario conocer qué es la web semántica. Como se ha adelantado en la introducción, se trata de una mejora de la web actual permitiendo a las máquinas comprender todo lo que se les plantea. Para ello, los recursos se encuentran organizados empleando diferentes conceptos y lenguajes que se comentan a continuación. Cabe mencionar que este entendimiento por parte de las máquinas no es igual que el de las personas debido a la falta de razonamiento propia de los seres humanos; se trata de un entendimiento que les permite obtener conclusiones e ideas gracias a procesos lógicos matemáticos.

Esta extensión de la web busca incluir el procesado automático para facilitar el manejo de la gran cantidad de información y resolver los problemas existentes de intercambio de información y conocimiento entre varios sistemas (Márquez, 2007).

A pesar de haber ofrecido el fundamento principal de la web semántica, no se ha tratado una definición de ella. Su creador, Tim Berners-Lee la define como: "*La Web Semántica es una extensión de la Web en la cual la información se da mediante un significado bien definido, lo que facilita que los ordenadores y la gente trabajen en cooperación*." (Berners-Lee, Hendler y Lassila, 2001).

Se puede decir que la web actual es un grafo dirigido, en el cual se encuentran todos los recursos enlazados entre sí y enlazados con diferentes buscadores que permiten a los usuarios acceder al recurso de interés. Sin embargo, cuando el usuario realiza una búsqueda, el sistema no ofrece un resultado preciso y totalmente relacionado puesto que, al contrario que la web semántica, no puede entender la consulta.

Con el objetivo de entender de una mejora manera el modo de relacionar conceptos en la web semántica se presenta la siguiente imagen. Se puede comprobar cómo la web actual enlaza los recursos entre sí, mientras que la web semántica dota de características propias a cada uno de los elementos enlazados, permitiendo crear una estructura semántica con todos ellos y concluir una información determinada:

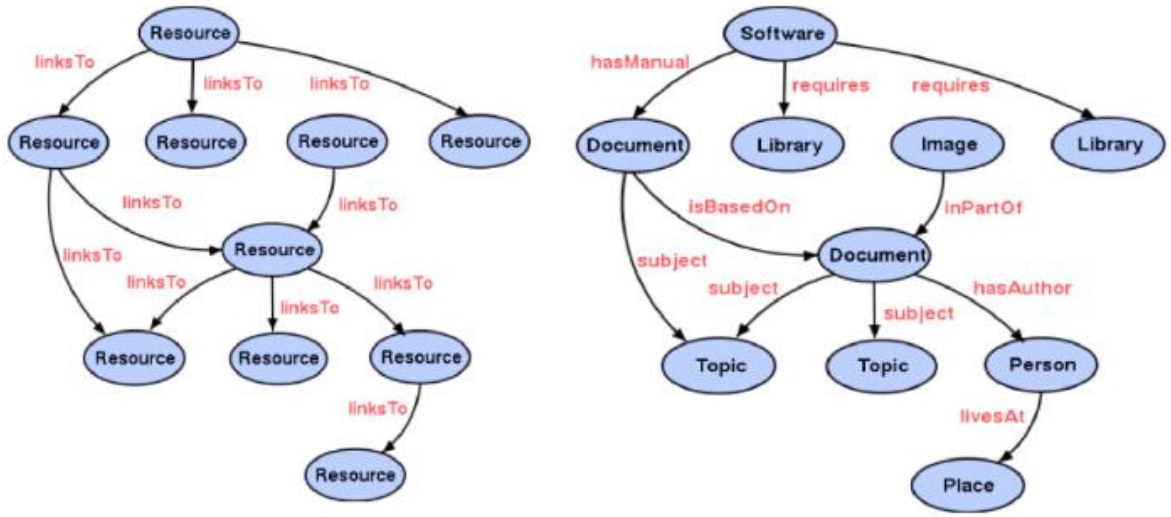


Figura 2. Web Actual vs Web Semántica
Fuente: "La Web Semántica", por Santiago Márquez Solís (2007).

Se puede observar cómo los datos constan de un mayor contexto y significado; por ello se utilizan los metadatos.

3.3. Metadatos.

Los metadatos consisten en información de la información, puesto que ofrecen datos sobre las características de un recurso de información, además de poder ofrecer su localización y recuperación.

Los metadatos pueden ser de tipo:

- Metadatos descriptivos: describen e identifican las características y la información de los recursos de información.
- Metadatos administrativos: facilita el registro y el manejo de aspectos administrativos de un recurso de información, como derechos de autor o permisos, entre otros.
- Metadatos estructurales: ayudan en la navegación y visualización de los recursos de información proporcionando información acerca de su estructura interna.
- Metadatos semánticos: ofrecen un significado o contexto a la información sobre los atributos de los recursos.

3.4. Ontología.

Dotar de significado a cada uno de los recursos, de forma que los diferentes sistemas puedan leerlos, se realiza gracias a la ontología, la cual se refiere al vocabulario empleado en un aspecto determinado de forma que un sistema puede utilizarlo. Una ontología no solo define un concepto, sino que también sus relaciones y propiedades.

Las ontologías es el componente principal de la web semántica, debido a su utilidad en la organización y representación del conocimiento en la web. Presentan una estructura formalizada para describir conceptos, relaciones y propiedades en un dominio específico (Berners-Lee, Hendler y Lassila, 2001).

3.5. Lenguaje XML.

Además de la ontología, para un correcto desarrollo de la web semántica es necesario el lenguaje XML (Extensible Markup Language). Se trata del lenguaje en el cual se basa la web

semántica. XML permite definir otros lenguajes, por ejemplo los lenguajes RDF y OWL que serán descritos más adelante.

Una característica de este lenguaje es su capacidad de etiquetar documentos de todas las clases, no solamente hipertexto, permitiendo así intercambiar cualquier tipo de datos de una manera más universal. También, al tratarse de un fichero de texto cualquier programa puede leerlo.

Este lenguaje se trabaja incluyendo entre < > el nombre de la etiqueta, la cual podrá tener diferentes valores. Por este motivo se describe como un metalenguaje ya que almacena datos y su estructura.

Márquez (2007) remarca que las adaptaciones del lenguaje XML en los OWL (Online Writing Lab) y RDF (Resource Description Framework) son las más utilizadas y se tratan de los estándares adoptados para ser utilizado en las ontologías.

3.5.1. Lenguaje RDF.

El lenguaje RDF es un lenguaje para representar información sobre recursos en la World Wide Web (Cáceres, 2006). Es un lenguaje basado en la descripción sobre un objeto: este posee una propiedad, la cual a su vez tiene un valor.

Este lenguaje está basado en la descripción sobre un objeto, siendo este representado de manera explícita a partir de tres componentes principales: recursos, propiedades, sentencias.

Los recursos son cualquier objeto de información dentro de la web que es identificado por un URI. Las propiedades son las características, relaciones o atributos que describen un recurso. Las sentencias son la unión de recurso, propiedad y valor de dicha propiedad consistiendo así en la estructura sujeto, predicado y objeto.

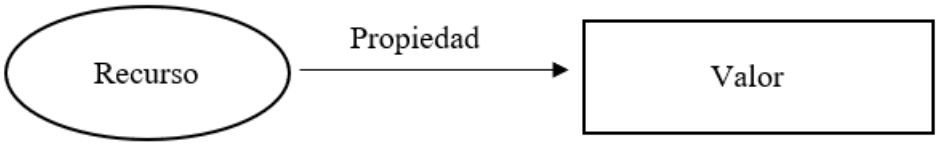


Figura 3. Estructura de los objetos en el lenguaje RDF.
Fuente: Elaboración propia, basada en "La Web Semántica y el lenguaje RDF", por Jesús Cáceres Tello (2006).

Estas tripletas (Figura 2) se combinan formando grafos RDF representando información de una manera estructura y enlazada.

3.5.2. Lenguaje OWL.

El lenguaje OWL, al tratarse de una extensión del lenguaje RDF, emplea las tripletas ya comentadas. Se trata de un lenguaje de etiquetado semántico que procesa la información mediante máquinas y como extensión, puede definir ontologías más complejas, añadiendo más vocabulario en la descripción de propiedades incluyendo, por ejemplo, cardinalidad, y especificando restricciones facilitando la comprensión y el razonamiento automático de los datos.

La diferencia entre estos dos lenguajes radica en que RDF representa e intercambia datos sin definir las ontologías mediante un nivel profundo de expresividad, mientras que el lenguaje OWL, basado en el anterior, ofrece mayor capacidad para la definición de ontologías formales y semánticamente más ricas.

Con el objetivo de identificar las propiedades en RDF de una forma universal, unificando así el lenguaje, se utilizan URIs (Uniform Resources Identifier). Se ha visto cómo las tripletas RDF consiste en la relación semántica de dos conceptos, los cuales están identificados mediante URIs. Una URI, a pesar de tener el mismo formato que una URL, no tiene por qué estar vinculada con un recurso localizado. Una URI se utiliza para identificar conceptos.

Así, cuando un sistema busca obtener información sobre un dato identificado por una URI, al hacer una llamada http se desreferencia el recurso, obteniendo información fácil de procesar en formato RDF (Caro, 2012).

4. Fuentes de datos.

El conjunto de datos “Universidades, colegios mayores, residencias universitarias y otros”, sobre el cual se realiza este trabajo, es un conjunto de datos abiertos accesible en el Portal de datos abiertos del Ayuntamiento de Madrid[2], siendo un documento disponible en varios formatos para su descarga.

Este conjunto de datos fue incorporado en el Portal de datos abiertos del Ayuntamiento de Madrid en marzo de 2014, siendo continua y frecuente su actualización ya que se actualizan anualmente.

El portal, además de ofrecer el archivo de datos, ofrece la representación de los datos en un mapa, permitiendo así localizar donde se encuentran los centros tratados, comprobando, así como la mayoría de los centros se encuentran en la ciudad de Madrid, aunque podemos encontrar algunos centros en ciudades de su alrededor:

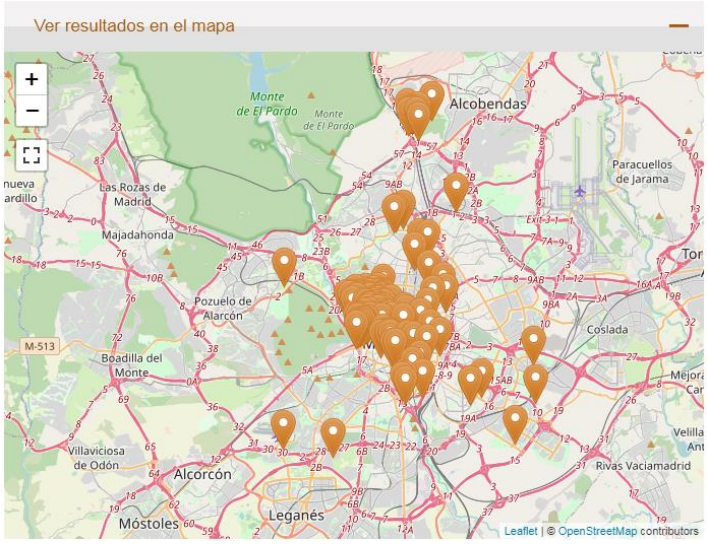


Figura 4. Mapa representativo de los centros tratados en los datos seleccionados. Fuente: Portal de datos abiertos del Ayuntamiento de Madrid.

Los datos fuente de este trabajo son datos pertenecientes al sector de Educación que reflejan los centros universitarios, tanto privados como públicos, de la ciudad de Madrid, pudiendo encontrar centros ubicados fuera de la ciudad cuya sede central sí se encuentra en Madrid.

[2] Portal de datos abiertos del Ayuntamiento de Madrid. https://datos.madrid.es/portal/site/egob

Además de las universidades, este fichero reúne los colegios mayores, residencias y escuelas universitarias.

Para todos estos centros se recogen sus datos más significativos de localización y contacto (dirección postal, teléfono, fax, email), además del transporte más cercano a ellos y el nivel de accesibilidad para personas con dificultades físicas o visuales. A parte de esta información, para las residencias universitarias y colegios mayores se recoge además el tipo de centro del cual se trata (mixto, masculino o femenino). En cambio, los centros de enseñanza superior y escuelas universitarias reúnen las titulaciones impartidas en cada una de ellas.

Además del fichero correspondiente en el formato deseado, se puede encontrar una documentación asociada que comenta la estructura del conjunto de datos a fin de acercar los datos a toda persona interesada, facilitándole la lectura de dicho documento.

5. Aplicación del método.

A continuación, se desarrolla el proceso de conversión de los datos seleccionados en datos semánticos. El proceso seguido es el definido en la Arquitectura Dirigida por Modelos (MDA) [6], el cual consiste en la realización de una serie de transformaciones hasta lograr nuestro objetivo.

- **Fase 1: CIM – Esquema gráfico y listado de requisitos de datos.**

En primer lugar, se seleccionan los datos sobre los cuales se desea trabajar. En este caso, los datos son "Universidades, colegios mayores, residencias universitarias y otros" obtenidos del portal de datos abiertos del ayuntamiento de Madrid, como ya se ha comentado en el capítulo anterior. Este archivo de datos se encuentra en diferentes formatos disponibles; sin embargo, como el método sobre el que nos basamos utiliza los datos en un archivo CSV, empleamos los datos en dicho formato.

Seguidamente, se debe definir las especificaciones del dominio: el conjunto de datos seleccionado incluye información acerca de los diferentes centros universitarios de enseñanza y vivienda para los estudiantes en la ciudad de Madrid. Respecto a los campos incluidos, cabe mencionar que, aunque la mayoría de los campos están completos y guardan una lógica, en algunos casos no se encuentran datos referidos a ese campo, se encuentran pocos o los datos que contienen no hace referencia al campo en el que se hallan.

Una vez analizados los datos seleccionados, se debe generar un esquema gráfico, el cual representa cómo se distribuyen los datos en los archivos fuente de los cuales se parte. En nuestro caso, el fichero de datos seleccionado contiene un único archivo CSV, generando así, a partir de él, un esquema gráfico formado por una única caja, que comparte título con el archivo y contiene todos los campos del conjunto de datos seleccionados (a excepción del campo *PK*, el cual es un campo clave de uso interno de la plataforma de datos):

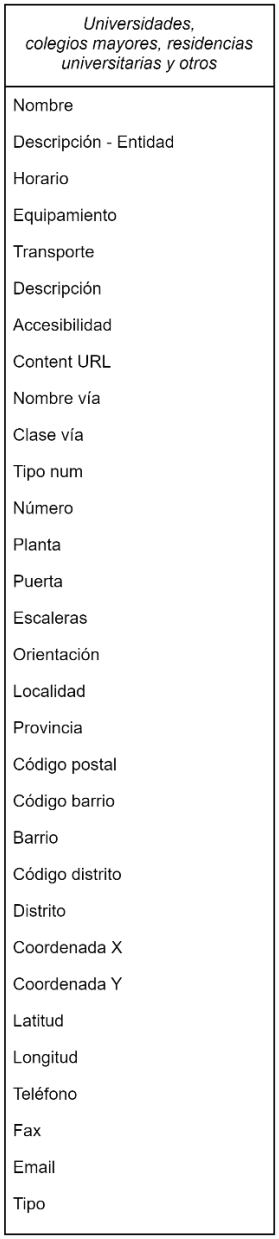


Figura 5. Esquema gráfico.
Fuente: elaboración propia.

Dado que los datos fuente están en un único archivo, se simplifica la tarea de generar el esquema gráfico porque no existen posibles relaciones entre diversos ficheros fuente.

El esquema gráfico obtenido es traducido al inglés con el objetivo de emplear un idioma más universal a la hora de crear la ontología, resultando el siguiente:

Figura 6. Esquema gráfico en inglés.
Fuente: elaboración propia.

Gracias al esquema gráfico elaborado se tiene una base sobre la cual se creará los requisitos de datos, posteriormente validados.

Disponer del archivo de datos en formato CSV es una gran ventaja puesto que favorece la aplicación del método y permite generar de manera rápida y sencilla datos semiestructurados.

En el esquema gráfico elaborado se observa los diferentes campos del archivo. A continuación, se expone una descripción detallada de cada uno de estos campos a fin de lograr una correcta comprensión de ellos:

NAME: apelativo del centro sobre el que hace referencia el conjunto de datos.

DESCRIPTION_ORGANIZATION: información complementaria sobre el centro (centro adscrito o similar de otro centro principal).

SCHEDULE: horario de apertura, de cierre y/o de actividades del centro.

EQUIPMENT: información complementaria sobre el centro (titulaciones disponibles en los centros universitarios y tipología masculina, femenina o mixta en los centros residenciales).

TRANSPORT: estaciones de metro y cercanías, paradas de autobuses y sus líneas correspondientes en las proximidades del centro.

DESCRIPTION: información adicional sobre el centro (titulaciones disponibles en los centros universitarios).

ACCESSIBILITY: nivel de accesibilidad al centro, siendo 0 no accesible, 1 accesible, 2 instalación parcialmente accesible para personas con movilidad reducida, 3 si no se tiene información sobre accesibilidad para personas con movilidad reducida, 4 si se cuenta con ayuda de lengua de signos, 5 si existe señalización podotáctil, 6 si existe bucle de inducción magnético.

CONTENT_URL: dirección del recurso en Internet para una rápida y fácil localización.

ROAD_NAME: denominación de la vía.

ROAD_TYPE: tipo de vía: calle, avenida, carretera, glorieta, paseo, plaza, ronda, autovía.

TYPE OF NUMBER: V (número), S/N (sin número), KM (kilómetro).

NUMBER: número de la dirección postal del centro.

FLOOR: nivel de un edificio en el que se encuentra el centro.

DOOR: puerta del edificio en el que se encuentra el centro.

STAIRS: escalera de acceso al centro del edificio en el que se encuentra.

POSITION: información adicional sobre la dirección del centro.

LOCALITY: localidad de ubicación del centro: Madrid.

PROVINCE: provincia de ubicación del centro: Madrid.

POST_CODE: código postal de la dirección donde se ubica el centro.

NEIGHBORHOOD_CODE: código referencial del barrio.

NEIGHBORHOOD: apelativo del barrio.

AREA_CODE: código referencial del distrito.

AREA: apelativo del distrito.

COORDINATE_X: conjunto de 6 números que hace referencia a la coordenada X sobre la que se encuentra el centro en el plano de la ciudad. Dicha coordenada está proyectada en el sistema de referencia ETR89.

COORDINATE_Y: conjunto de 7 números que hace referencia a la coordenada Y sobre la que se encuentra el centro en el plano de la ciudad. Dicha coordenada está proyectada en el sistema de referencia ETR89.

LATITUDE: coordenada de latitud proyectada en el sistema de referencia WGS84.

LONGITUDE: coordenada de longitud proyectada en el sistema de referencia WGS84.

TELEPHONE: teléfono de contacto del centro.

FAX: número de fax del centro.

EMAIL: dirección de correo de contacto del centro.

TYPE: descripción de la tipología de la instalación (colegio mayor, escuelas universitarias, centros de formación profesional, universidad, facultades, clínicas centros veterinarios, otros).

En este punto se puede generar la lista de requisitos de datos, a través de la cual se describe con detalle el dominio de los datos. Esta lista será una referencia útil para las siguientes fases del proceso y, por ello, la descripción realizada debe ser completa, clara y coherente.

La lista de requisitos de datos de nuestro fichero es:

R1. Todo centro dispone de un nombre propio.

R2. En caso de tratarse de un centro adscrito a otro, esta información es recogida en la descripción de la entidad.

R3. Si el centro es un centro universitario, equipamiento indicará las diferentes titulaciones que se ofertan. Por el contrario, si el centro es una residencia universitaria, se indicará si es un centro femenino, masculino o mixto.

R4. Para todo centro se indica un transporte, recogiendo el tipo (bus, metro, bicimad, aparcamiento, cercanías Renfe) y sus correspondientes líneas y/o estaciones.

R5. La descripción indica otras titulaciones encontradas en los centros.

R6. Para todo centro se indica una accesibilidad, indicando el nivel de accesibilidad del centro, pudiendo tomar valores del 0 al 6. Los datos fuente, en este caso, toman los valores: 0 no accesible, 1 accesible, 2 instalación parcialmente accesible para personas con movilidad reducida, 3 si no se tiene información sobre accesibilidad para personas con movilidad reducida.

R7. Para todo centro se recoge la url dedicada a dicho centro dentro del portal web del Ayuntamiento de Madrid, donde se recoge toda su información.

R8. Para todo centro se recoge su dirección, indicando el tipo de vía en el que se encuentra (calle, avenida, carretera, glorieta, paseo, plaza, ronda, autovía), el nombre de la vía, el tipo de número que caracteriza su dirección (V, S/N, KM) y el número.

R9. Si un centro cuenta con información adicional en su dirección que no ha sido recogida, lo hace orientación, la cual explica cómo encontrar el centro añadiendo información sobre su dirección.

R10. Todo centro se ubica en la localidad y provincia de Madrid, lo cual es recogido en dichos campos.

R11. Todo centro, debido a su ubicación, cuenta con un código postal.

R12. Todo centro se encuentra ubicado en un barrio, el cual es identificado por un código y un nombre.

R13. Todo centro se encuentra ubicado en un distrito, el cual es identificado por un código y un nombre.

R14. Todo centro cuenta con dos coordenadas (x representada por seis números e y representada por siete números), proyectadas en el sistema de referencia ETR89, que indican el lugar donde se halla el centro en el plano de la ciudad.

R15. Todo centro está identificado por sus coordenadas de latitud y longitud, indicadas en el sistema de referencia WGS84.

R16. Todo centro dispone de un número de teléfono de contacto.

R17. Si el centro cuenta con fax, se indica su número.

R18. Si el centro dispone de dirección de correo de contacto, esta se indica.

R19. Todo centro es clasificado según la tipología de la instalación, pudiendo ser: colegio mayor, escuelas universitarias, centros de formación profesional, universidad, facultades, clínicas centros veterinarios, otros. Esto se indica siguiendo la estructura */contenido/entidadesYorganismos/Universidades/*. Esta indicación se realiza para todas las universidades, y para el resto de los centros se completa la estructura incluyendo al final de esta *Facultades*, *OtrosCentrosUniversitarios*, *EscuelasUniversitarias*, *ColegiosMayores* o *CentrosFormacionProfesionPrivadosConcertados*, en función del centro al que haga referencia.

Como se ha comentado, las especificaciones, el esquema gráfico y los requisitos de datos serán útiles en las siguientes fases, sobre todo como metadatos para ser incluidos en la ontología.

- **Fase 2: PIM – Modelo de dominio objetivo.**

Esta fase hace uso del esquema gráfico y el listado de requisitos de datos obtenidos en la fase anterior (fase CIM). De manera manual, se realiza una serie de transformaciones a fin de mostrar en el esquema de dominio objetivo solamente aquellos datos de interés. Sin embargo, cabe señalar que los cambios realizados no son aleatorios, sino que están basados en Larman[1], como el MDA, siguiendo su guía a la hora de generar clases y asociaciones.

Las transformaciones y consideraciones llevadas a cabo son:

- El esquema gráfico es representado por una única caja. Por tanto, se cuenta con una única clase cuyo nombre es *Universities, university residences and others.*
- La existencia de una caja única provoca la ausencia de asociaciones entre clases.
- Los campos que no son interesantes son eliminados. En nuestro caso, se descartan los campos *Schedule, Floor, Door, Stairs.*
- A cada campo se le asigna un tipo de dato teniendo en cuenta los valores que recoge y el listado de requisitos de datos. Esto se realiza con el objetivo de transformar los campos en atributos. Así, la asignación de tipos de datos en los atributos de nuestro caso es:
    - Name: String
    - Description_Organization: String
    - Equipment: String
    - Transport: String
    - Description: String
    - Accessibility: Enum
    - Content_URL: String
    - Road_Name: String
    - Road_Type: String
    - Type of number: String
    - Number: Int
    - Position: String
    - Locality: String
    - Province: String
    - Post_Code: Int
    - Neighborhood_Code: Int

  - Neighborhood: String
  - Area_Code: Int
  - Area: String
  - Coordinate_X: Int
  - Coordinate_Y: Int
  - Latitude: Int
  - Longitude: Int
  - Telephone: Int
  - Fax: Int
  - Email: String
  - Type: String

- Se inserta una nota por cada atributo de tipo *enum*, la cual recoge la lista de posibles valores que puede tomar. Es el caso del atributo *Accessibility*.

Una vez realizadas estas transformaciones se obtiene el esquema de dominio objetivo, el cual es representado en el siguiente diagrama de clases UML:

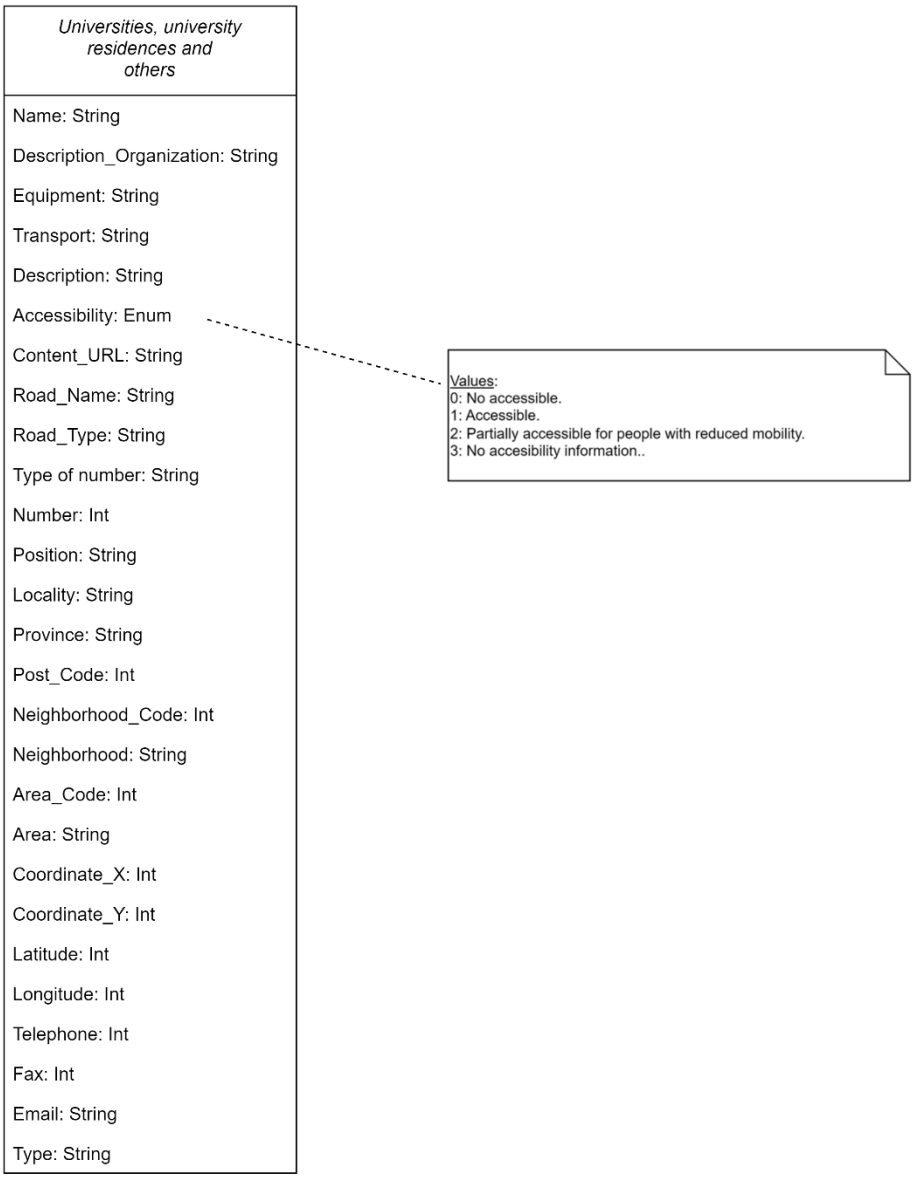


Figura 7. Esquema de dominio objetivo.
Fuente: elaboración propia.

- **Fase 3: PSM – Esquema ontológico y ontología.**

En esta fase se desarrolla tanto el esquema ontológico como la ontología de dominio.

Hay que aplicar una serie de nuevas transformaciones al modelo de dominio objetivo recientemente obtenido, con el fin de obtener el esquema ontológico.

Se tiene presente que la fuente de transformación es un diagrama de clases UML y se busca crear un modelo semántico, una ontología, mediante RDFS y OWL.

Las transformaciones realizadas emplean el lenguaje de programación QVT, siguiendo el modelo base definido por Cáceres y Garrido (2021):

- Toda clase UML se transforma en clase OWL.
- Un atributo UML definido por un tipo de dato básico (string, char, integer, float, boolean o date) se transforma en un OWL Datatype Property con dominio de clase OWL y su rango de clase RDFS asociada al tipo de dato básico correspondiente. En el caso de este trabajo, todos los atributos, a excepción de *Accesibilidad*, pertenecen a este tipo, ya que se tratan de datos de tipo básico: integer, string.
- Un atributo de UML definido por un tipo de dato correspondiente a otra clase se transforma en un OWL Object Property cuyo dominio es la clase OWL correspondiente y su rango es la clase OWL a la que hace referencia. En nuestro caso, al contar solamente con una clase, no hacemos uso de esta transformación.
- Un atributo UML definido por un dato de tipo enum se tranforma en un OWL Object Property cuyo dominio es la clase OWL correspondiente y su rango es la clase RDF Alt. En este trabajo, el atributo Accesibilidad es de este tipo.

A estas transformaciones se pueden sumar más, referentes a atributos definidos por tipos de datos de colección. Sin embargo, no se trabaja al no encontrar atributos de UML definidos así en este trabajo.

Por tanto, con el objetivo de obtener las clases OWL del esquema ontológico se realizan las transformaciones oportunas utilizando como base el modelo de dominio objetivo obtenido en la fase anterior (ilustración 4).

Las clases UML se transforman en clases OWL, manteniendo el nombre que ya tenían y son representadas mediante un recuadro y una línea discontinua que sigue un recorrido desde ellas hasta el recurso semántico OWL: Class. Esta línea discontinua es etiquetada como rdf: type.

Como se ha comentado, en nuestro caso, casi todos los atributos son de tipo básico y, por ello, se transforman en OWL Datatype Properties para el esquema ontológico. A excepción del dato Accessibility, que es de tipo enum, y se transforma en OWL Object Property.

Además, en nuestro caso solamente se cuenta con una clase, la cual recibe el nombre de "Center", la cual hace referencia a universidades, colegios mayores, residencias universitarias y otros. Cada uno de estos centros es identificado un valor *PK*, el cual es único y de tipo centro, añadiendo a cada uno de ellos sus propiedades.

Esta clase contiene el nombre de todos los atributos de datos de tipo básico. Cada uno de los nombres de estos atributos será un OWL Datatype Property y será representado como un cuadro, del cual sale una línea discontinua etiquetada como rdf: type y está unida a otro cuadro llamado OWL:DataTypeProperty.

Aquellos cuadros que recogen atributos cuyo nombre es de tipo string, es decir, es una cadena de datos, contienen, además, una línea continua, etiquetada como rdf: range, que lo une al cuadro xsd: string.

Los cuadros que hacen referencia a atributos de tipo int se unen al cuatro xsd: integer a través de una línea continua etiquetada como rdf: range.

Por último, el atributo de tipo enum, *Accesibilidad*, es recogido en una elipse por tratarse de OWL Object Property y es unida por una línea continua etiquetada como rdf: range al cuadro rdf: Alt.

Se plantea una mejora dentro de las transformaciones realizadas: adición de la propiedad *País*, la cual complementará a las propiedad de *Localidad* y *Provincia* de cada uno de los centros recogidos.

En nuestro trabajo esta nueva propiedad será común a todos los centros, puesto que todos ellos se encuentran en España, en concreto en la ciudad de Madrid.

El esquema ontológico resulta, por tanto, de la siguiente manera:

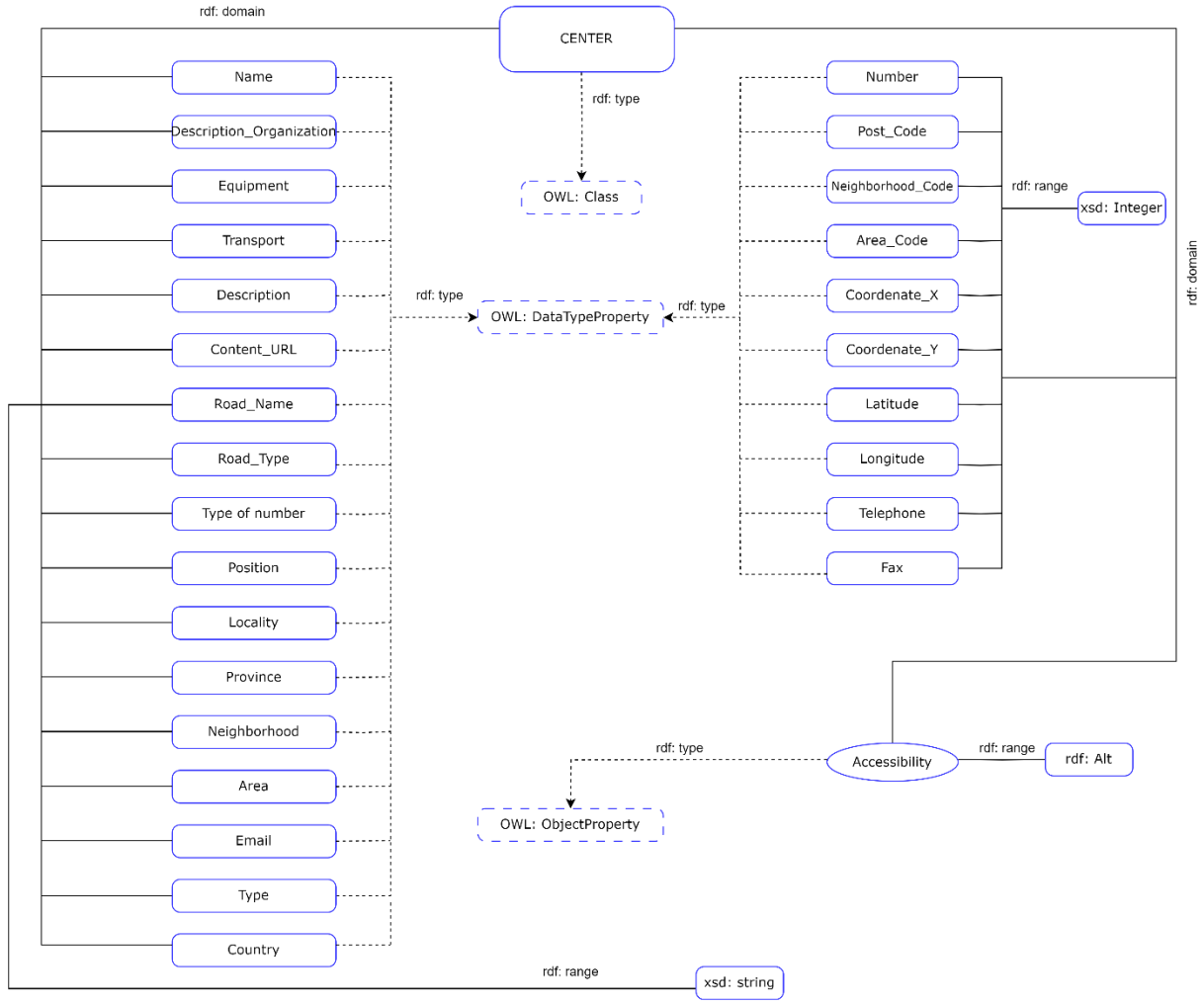


Figura 8. Esquema ontológico.
Fuente: elaboración propia.

De esta forma, a modo de resumen, se puede comentar que Center en una clase OWL, que supone además el origen (rdf:domain) de todas las relaciones (DataProperty). La única diferente es la relación de accesibilidad (ObjectProperty), puesto que son valores concretos donde cada uno de ellos tienen un significado diferente y por ello se representan como rdf:Alt (o un valor u otro). Los valores concretos que toman el resto de campos son de dos tipos: string o integer.

Una vez obtenido el esquema ontológico, utilizando RDFS y OWL, se genera la ontología. Dicha ontología se crea teniendo en cuenta las especificaciones obtenidas en la Fase 1, puesto que es de utilidad al ser información incorporada a la definición de la ontología.

Se recomienda usar términos de otras ontologías, reutilizando así vocabularios existentes, con el objetivo de realizar una ontología más completa. Por este motivo, en nuestro caso, reutilizaremos términos ya definidos en ontologías como DBPedia.

- **Fase 4: Código – DSL código.**

A partir del esquema ontológico generado en la fase anterior, se crea el código DSL, el cual al ejecutarse generará el conjunto de datos semánticos.

Este es creado al aplicar transformaciones empleando la técnica MOFM2T (Model to Text), la cual se utiliza para crear un código fuente a partir de modelos de objetos definidos con un lenguaje de modelado MOF, el cual es basado en el lenguaje de modelado UML, como es nuestro caso. La aplicación de estas transformaciones resulta el código DSL, el cual es específico para el conjunto de datos que se busca anotar semánticamente.

Esta técnica, además, garantiza coherencia del código generado y reduce la cantidad de código que los ingenieros hubieran escrito sino manualmente.

Este lenguaje es el elegido para realizar las transformaciones puesto que es el recomendado por la OMG (Object Management Group), una organización encargada de establecer estándares dentro del ámbito de las tecnologías orientadas a objetos.

Para facilitar este proceso de tranformación de datos CSV en datos RDF se sigue el lenguaje CSV@RDF, el cual es un lenguaje específico de dominio (DSL), al igual que Cáceres y Garrido (2021) hacen en su trabajo.

Este lenguaje CSV@RDF estructura las sentencias del código en encabezado y cuerpo, como se sigue:

En primer lugar, este lenguaje define un encabezado, donde se indica la ubicación de las fuentes de datos y las ontologías utilizadas de la siguiente manera:

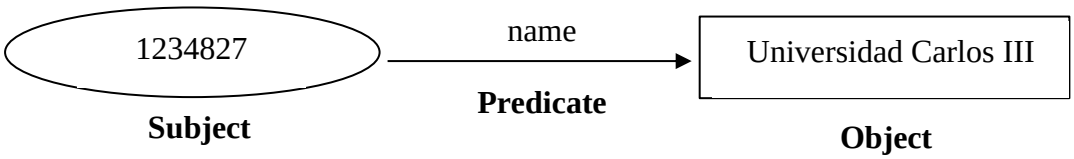
El comando FILE indica la ruta del fichero que contiene los datos y el alias que tendrá para ser nombrado a lo largo del código:

```
FILE(#datos,listaCentros.csv)
```

El comando PREFIX especifica las ontologías a utilizar, indicando el prefijo y el espacio de nombres de los vocabularios semánticos de la ontología:

```
PREFIX(stu, http://www.vortic3.com/studentplace/)
```

En segundo lugar, se elabora el cuerpo del código siguiendo el modelo de tripleta (sujeto, predicado y objeto). A continuación se muestra un ejemplo visual para facilitar su comprensión:



El campo PK es el establecido para el sujeto (SUBJECT), el cual es declarado clave en cada una de las filas del archivo CSV utilizado puesto que es único, y se relaciona con los demás campos.

```
SUBJECT(#datos.PK,stu:centro)
```

A continuación, se elabora el predicado-objeto. Esta sentencia incluye todas las propiedades y está compuesta por dos subsentencias: el predicado y el objeto. Mientras que el predicado indica el término de la ontología usado como propiedad, el objeto representa el valor del campo dentro del fichero de datos que lo contiene.

```
PREDICATE-OBJECT(PREDICATE(stu:name), OBJECT(#datos.NOMBRE))

PREDICATE-OBJECT(PREDICATE(stu:description_organization),
OBJECT(#datos.DESCRIPCION_ENTIDAD))

PREDICATE-OBJECT(PREDICATE(stu:equipment),
OBJECT(#datos.EQUIPAMIENTO))

PREDICATE-OBJECT(PREDICATE(stu:transport), OBJECT(#datos.TRANSPORTE))

PREDICATE-OBJECT(PREDICATE(stu:description),
OBJECT(#datos.DESCRIPCION))

PREDICATE-OBJECT(PREDICATE(stu:content_URL),
OBJECT(#datos.CONTENT_URL))

PREDICATE-OBJECT(PREDICATE(stu:roadName), OBJECT(#datos.NOMBRE_VIA))

PREDICATE-OBJECT(PREDICATE(stu:roadType), OBJECT(#datos.CLASE_VIAL))

PREDICATE-OBJECT(PREDICATE(stu:typeNumber), OBJECT(#datos.TIPO_NUM))

PREDICATE-OBJECT(PREDICATE(stu:number), OBJECT(#datos.NUM))

PREDICATE-OBJECT(PREDICATE(stu:position), OBJECT(#datos.ORIENTACION))
```

```
  PREDICATE-OBJECT(PREDICATE(stu:locality), OBJECT(#datos.LOCALIDAD))

  PREDICATE-OBJECT(PREDICATE(stu:postCode),
OBJECT(#datos.CODIGO_POSTAL))

  PREDICATE-OBJECT(PREDICATE(stu:neighborhoodCode),
OBJECT(#datos.COD_BARRIO))

  PREDICATE-OBJECT(PREDICATE(stu:neighborhood), OBJECT(#datos.BARRIO))

  PREDICATE-OBJECT(PREDICATE(stu:areaCode),
OBJECT(#datos.COD_DISTRITO))

  PREDICATE-OBJECT(PREDICATE(stu:area), OBJECT(#datos.DISTRITO))

  PREDICATE-OBJECT(PREDICATE(stu:coordinateX),
OBJECT(#datos.COORDENADA_X))

  PREDICATE-OBJECT(PREDICATE(stu:coordinateY),
OBJECT(#datos.COORDENADA_Y))

  PREDICATE-OBJECT(PREDICATE(stu:latitude),OBJECT(#datos.LATITUD))

  PREDICATE-OBJECT(PREDICATE(stu:longitude),OBJECT(#datos.LONGITUD))

  PREDICATE-OBJECT(PREDICATE(stu:telephone), OBJECT(#datos.TELEFONO))

  PREDICATE-OBJECT(PREDICATE(stu:fax), OBJECT(#datos.FAX))

  PREDICATE-OBJECT(PREDICATE(stu:type), OBJECT(#datos.TIPO))
```

Sin embargo, como se ha comentado con anterioridad, se reutilizará vocabularios de ontologías ya existentes. En este caso, el predicado-objeto se construye con la siguiente estructura, indicando dicha ontología reutilizada:

```
  PREDICATE-OBJECT(PREDICATE(dbo:country), OBJECT(España))

  PREDICATE-OBJECT(PREDICATE(dbo:email), OBJECT(stu:#datos.EMAIL))

  PREDICATE-OBJECT(PREDICATE(dbo:province),
OBJECT(stu:#datos.PROVINCIA))
```

En nuestro caso, se cuenta con datos de tipo enum cuyas sentencias serían:

```
  PREDICATE-OBJECT(PREDICATE(stu:accessibility),
OBJECT(#datos.ACCESIBILIDAD,VALUES(0,1,2,3), ENUMS(none, accessible,
partially, no info)))
```

En nuestro caso, al tratar solamente con un fichero de datos, no se emplean los comandos *QUERY* y *MATCH*, los cuales permiten obtener valores siguiendo rutas definidas de navegación entre los diferentes archivos fuente.

Tras haber comentado las diferentes sentencias, se describen las diferentes reglas utilizadas para transformar el lenguaje usando la técnica MOFM2T (*Model to Text)*.

Las reglas de transformación llevadas a cabo son:

- Por cada una de las clases OWL se genera una sentencia FILE, cuyo primer parámetro contiene el símbolo # seguido del nombre de esa clase. Este parámetro será el identificador de clase. El segundo parámetro, al cual llamaremos (a) se inicia vacío y se hará referencia a él posteriormente.
  Un ejemplo en nuestro caso: FILE(#datos,).

- Se genera una sentencia PREFIX con parámetros vacíos a los que se hará referencia posteriormente (b).

- Por cada clase OWL también se genera una sentencia SUBJECT, cuyo primer parámetro vuelve a ser el identificador de clase (#...) seguido de "." y del atributo que identifica esa clase. El segundo parámetro es el alias asignado a la ontología (en nuestro caso, stu) seguido de ":" y el nombre de la clase.

- Por cada una de las propiedades de tipo de datos OWL se genera una sentencia PREDICATE-OBJECT, el cual contiene un parámetro para PREDICATE y otro parámetro para OBJECT. El parámetro del PREDICATE es el alias de la ontología seguido de ":" y el nombre de la propiedad, y el parámetro del OBJECT es el identificador de clase seguido de "." y el nombre de la propiedad.

- Por cada una de las propiedades de objeto OWL se genera una sentencia PREDICATE-OBJECT. De nuevo, el PREDICATE contiene como parámetro al alias de la ontología seguido de ":" y el nombre de la propiedad. El parámetro del OBJECT es el alias seguido de ":" y el identificador de la clase acompañado de "." y el nombre de la propiedad.

- Por cada una de las propiedades de objeto OWL cuyo rango se define como RDF Alt se genera una sentencia PREDICATE-OBJECT. En esta sentencia, el parámetro

del PREDICATE es el alias de la ontología seguido de ":" y el nombre de la propiedad. El OBJETO varía respecto a otras sentencias, puesto que su parámetro se forma con el identificador de la clase seguido de "." y el nombre de la propiedad, seguido del comando VALUES y la lista de valores a sustituir separados por comas y el comando ENUMS seguido de los diferentes valores literales por los que se sustituirán los valores.

- Por cada propiedad de objeto OWL cuyo rango se define como RDF Bag o RDF Seq se genera una sentencia PREDICATE-OBJECT. Sin embargo, en nuestro caso no contamos con ninguna propiedad de este tipo.

A esta serie de transformaciones realizadas a partir del esquema ontológico y la ontología, se le suma una serie de transformaciones manuales con el objetivo de completar el código DSL obtenido como resultado. Estas transformaciones son aplicadas directamente sobre el código DSL de la siguiente manera:

- El segundo parámetro de la declaración FILE, el cual habíamos definido como vacío, se completa indicando la ruta en la que se ubica el archivo fuente.
- La sentencia PREFIX se completa indicando en el primer parámetro el alias de la ontología y en el segundo parámetro se indica el espacio de nombres asociado a esa ontología.

Por tanto, en esta última fase, a partir del esquema ontológico y la ontología, se ha obtenido el código DSL. Este código será ejecutado, obteniendo así los datos semánticamente anotados y servirá como validación del método aplicado al conjunto de datos fuente seleccionado en este TFG.

6. Validación

Una vez desarrollado todo el proceso indicado por Cáceres y Garrido (2021) con el objetivo de obtener datos semánticamente anotados, se procede a poner a prueba el código DSL obtenido pudiendo así validar dicho trabajo.

La validación se ha llevado a cabo siguiendo el método CSV@RDF empleando el archivo CSV donde se recogen los datos y el código DSL elaborado durante todo el proceso descrito en apartados anteriores.

Previamente, a ejecutar en el comando de Windows nada, se debe tener en un archivo txt el código DSL elaborado. En nuestro caso recibe el nombre codigoDSL.txt, el cual debe ser leído por el programa para poder obtener los resultados esperados. Este archivo puede ser consultado en el repositorio Zenodo[3].

Además, otro aspecto que se debe tener en cuenta de manera previa es tener configurada en el ordenador la variable de entorno JAVA_HOME, la cual permite hacer funcionar a todo programa que necesite contenido de Java, como es nuestro caso. Esta variable de entorno a veces está incorporada por defecto. En caso contrario, se debe configurar instalando antes (si no se cuenta con ello) el kit de desarrollo de Java (Java Development Kit, JDK), gracias al cual el ordenador podrá contar con todas las herramientas de java.

En este momento, tras abrir la consola de Windows, se inserta la sentencia `cd Desktop`, la cual permite entrar al escritorio del ordenador donde se hallan los archivos implicados en el código. Tras ello, se ejecuta el siguiente comando, indicando el nombre del fichero de texto que contiene el código DSL a ejecutar:

```
java –jar DSLengine2RDF.jar codeDSL.txt
```

Tras hacer frente a los errores que pueden surgir durante la ejecución, y los cuales son tratados en detalle en el siguiente apartado, se genera un fichero formato rdf con nombre *output*, en el cual se encuentran anotados semánticamente los datos de interés de nuestro fichero de datos CSV original. Debido a resultar un fichero con numerosas líneas de código, se muestra, a continuación, solamente una parte de lo obtenido en el fichero RDF. En concreto, se muestra

[3] https://zenodo.org/record/8110848

la salida correspondiente a los datos del centro Colegio Mayor Diego de Covarrubias, el cual se idéntica por el PK 10297714:

```
</rdf:Description>

      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

      xmlns:stu="http://www.vortic3.com/studentsplace/"

      xmlns:dbo="https://dbpedia.org/page/"

      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >

    <rdf:Description rdf:about="http://www.vortic3.com/studentsplace/10297714">

      <dbo:province rdf:resource="http://www.vortic3.com/studentsplace/MADRID"/>

      <stu:number>10</stu:number>

      <stu:accessibility>none</stu:accessibility>

      <stu:equipment>Tipo mixta.</stu:equipment>

      <stu:postCode>28040</stu:postCode>

      <stu:transport>Bus: U  160  161  A.</stu:transport>

     <stu:content_URL>

        http://www.madrid.es/sites/v/index.jsp?vgnextchannel=bfa48ab43d6bb410VgnVCM100000171f5
        a0aRCRD&amp;vgnextoid=8f15849d706fa510VgnVCM2000001f4a900aRCRD</stu:content_URL>

      <stu:neighborhood>CIUDAD UNIVERSITARIA</stu:neighborhood>

      <stu:area>MONCLOA-ARAVACA</stu:area>

      <stu:fax>913 941 158</stu:fax>

      <stu:coordinateX>438214</stu:coordinateX>

      <stu:roadType>AVENIDA</stu:roadType>

      <stu:coordinateY>4476373</stu:coordinateY>

      <stu:typeNumber>V</stu:typeNumber>

      <stu:locality>MADRID</stu:locality>

      <stu:position></stu:position>

      <stu:description_organization></stu:description_organization>

      <dbo:country>España</dbo:country>

      <stu:neighborhoodCode>3</stu:neighborhoodCode>

      <stu:areaCode>9</stu:areaCode>

      <dbo:email rdf:resource="http://www.vortic3.com/studentsplace/cmm@pas.ucm.es"/>
```

```
    <stu:roadName>SENECA</stu:roadName>
    <stu:longitude>-37.284.790.953.706.000</stu:longitude>
    <stu:description></stu:description>
    <rdf:type rdf:resource="http://www.vortic3.com/studentsplace/centro"/>
    <stu:type>/contenido/entidadesYorganismos/Universidades/ColegiosMayores</stu:type>
    <stu:name>Colegio Mayor Diego de Covarrubias</stu:name>
    <stu:latitude>4.043.570.784.001.130</stu:latitude>
    <stu:telephone>913 941 030</stu:telephone>
</rdf:Description>
```

El conjunto completo de datos semánticamente anotados se puede encontrar y consultar en el repositorio web Zenodo[4].

Por último, para realizar la validación, se utiliza el recurso *Validator* [5], el cual verifica que la semántica esté correctamente formulada con sus partes correspondientes (sujeto, predicado, objeto). Además, esta herramienta ofrece las tripletas y se muestra el grafo, donde se pueden diferenciar los valores literales de aquellos que son recursos. En este trabajo se muestra la validación realizada con la parte de código correspondiente al centro Colegio Mayor Diego de Covarrubias, cuyo PK es 10297714. Se muestra el grafo del modelo obtenido a través de dicho validador correspondiente al centro residencial universitario citado, puesto que el archivo completo es demasiado grande y se empeora su visualización, ya que el esquema completo contaría con 29 campos para cada uno de los centros.

[4] https://zenodo.org/record/8110848
[5] https://www.w3.org/RDF/Validator/

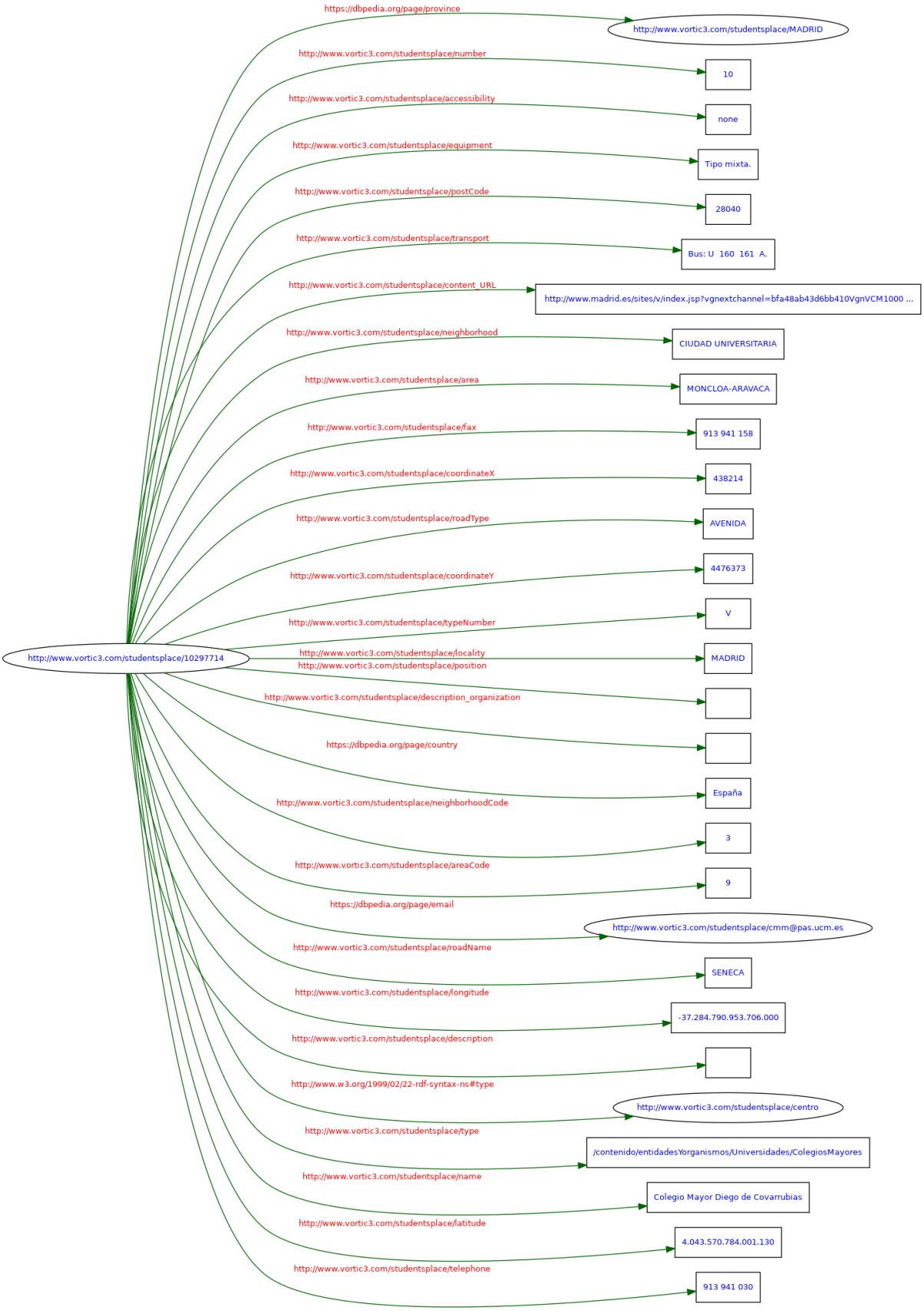


Figura 9. Grafo correspondiente al Colegio Mayor Diego de Covarrubias.
Fuente: W3C Validation Service [Fotografía], por W3C (2023). https://www.w3.org/RDF/Validator/rdfval

6.1. Incidencias detectadas y mejoras añadidas.

Una vez seguido el método propuesto se puede comentar los aspectos positivos y negativos de este, al igual que proponer posibles mejoras para facilitar el uso del método para personas menos expertas, cumpliendo así los objetivos restantes propuestos.

Cabe mencionar que la elección del fichero de datos elegido, el cual trata un tema y contiene vocabulario común y cercano, ha permitido que el proceso desarrollado no suponga mayor dificultad, siendo además unos datos útiles puesto que permite el acceso a datos de contacto e información de los diferentes centros para los estudiantes interesados. Sin embargo, el tratarse de un tema más abstracto y lejano para el usuario puede dificultar el entendimiento del proceso y, por tanto, su desarrollo.

Dentro del proceso de anotación semántica, se reconoce como parte más compleja el final de la fase 1 y el comienzo de la fase 2, es decir, la construcción de la lista de requisitos y del modelo de dominio objetivo.

Sin embargo, merece hacer referencia al desarrollo del código DSL como la parte del método que más tiempo ha llevado. No por complejidad, si no por los errores que han surgido durante su ejecución y pueden ser evitados, ahorrando así tiempo en futuros trabajos:

- Toda URI tiene como cabeza en su dirección http. En caso de olvidar su escritura, la dirección no es reconocida como una URI.
- Evitar el uso del guion, sustituyendo este, por ejemplo, por un guion bajo en el nombre de los campos del fichero de datos.
- Los datos, dentro de un CSV, son separados por comas y, el método probado trabaja con archivos CSV. Esto es, reconoce aquellos datos que son separados por comas. Luego, si el archivo no está separado por comas, tal y como es un CSV original, surgen errores provocando así la no ejecución del código.

Como se observa, son errores de fácil y rápida solución siempre y cuando se detecten sin dificultad. Sin embargo, un inconveniente que veo a este proceso es el no saber si lo estás haciendo bien hasta este punto, en el que se ejecuta el código DSL elaborado y se observan posibles errores.

Durante el desarrollo del trabajo han surgido algunas ideas de mejora de este trabajo, las cuales han sido propuestas y llevadas a cabo.

En este trabajo se añade como mejora la terminología *Country*, el cual no tiene mayor importancia con los datos utilizados, puesto que todos los centros universitarios se encuentran en España. Sin embargo, esta inclusión es útil en el futuro uso de la anotación semántica creada con datos del mismo tipo, pero de diferentes países. Esto puede permitir un análisis de los centros universitarios y sus diferencias comparando entre diversos países.

Igualmente, todos los centros recogidos en el fichero de datos trabajado se encuentran en Madrid, por lo que ampliar a datos de otras provincias y localidades de territorio español puede ser igual de interesante, permitiendo así recoger los datos de cada uno de los centros y, permitiendo a los estudiantes, buscar información sobre estos conociendo, entre otros aspectos, por ejemplo, las titulaciones impartidas en cada uno de ellos o la tipología de centro.

7. Conclusiones y futuro trabajo

Llegados a este punto se puede confirmar la consecución los objetivos propuestos inicialmente. Como se ha comprobado durante el desarrollo del trabajo se ha verificado la validez del método para anotar semánticamente un conjunto de datos, tras ser analizados y conocer mejor los datos a tratar. También se han encontrado una serie de limitaciones ya mencionadas.

La mayor dificultad encontrada se halla en el escaso conocimiento inicial acerca del tema de dicho trabajo. Sin embargo, realizando un estudio previo y estudiando conceptos clave se ha podido desarrollar sin problemas mayores el método propuesto.

Dentro de este método la dificultad destacada se ha encontrado en la ejecución del código DSL y la obtención del grafo correspondiente, debido a la existencia de pequeños errores no apreciables a primera vista. Es por ello por lo que se le debe dar importancia, a la hora de corregir posibles errores en este paso, a todos los caracteres especiales empleados: puede tratarse de un error a la hora de su lectura.

Asimismo, durante el desarrollo de este trabajo han surgido ideas de proyectos complementarios que se pueden llevar a cabo y son comentadas a continuación.

Como trabajo futuro, puede ser interesante la ampliación comentada de datos de otras ubicaciones, la cual puede ser trabajada como un proyecto de universidad. Considero que, trabajar este proceso en grupos facilitaría su desarrollo y, además, acercaría la realidad a clase, enseñando a los alumnos contenidos igualmente pero de una manera más práctica y menos abstracta, viendo la utilidad de sus conocimientos durante este proceso. Así, además, cada grupo de estudiantes podría utilizar datos diferentes, pudiendo ser de la misma temática, permitiendo así un aumento de datos semánticos.

Además, los datos trabajados no son fijos, puesto que un centro puede desaparecer o modificar algún dato de los recogidos, o incluso pueden surgir nuevos centros dentro de la ciudad Madrid. Estos cambios se podrían añadir ajustando el archivo de datos adaptándose así a las modificaciones y recogiendo los datos actualizados.

7 Conclusiones y trabajo futuro

Con respecto al trabajo futuro, es necesario comentar que podría desarrollarse una herramienta que guiara en el proceso a seguir y que plantea el método CSV@RDF, facilitando así el trabajo a personas con menor conocimiento de la técnica y consiguiendo así llegar a un mayor número de personas.

Además, de esa ayuda, la herramienta podría ayudar a generar los datos semánticamente anotados partiendo de los ficheros fuente. Podría permitir leer los ficheros y los campos de los ficheros e indicar si todos los datos son válidos o no. De esta manera, el trabajador puede buscar una solución para los datos señalados como no válidos y así continuar con el proceso.

Una vez leídos los ficheros y los campos, siendo todos ellos válidos, esta herramienta podría también, a partir de los mismos, establecer qué tipos de elementos del esquema ontológico se le asigna a cada campo. Lo mismo podría hacerse con el código DSL, que podría generarse de una forma semiautomática, a partir del conjunto de instrucciones existente.

Luego, todo trabajo futuro propuesto consiste en la extensión del método CSV@RDF tanto a personas expertas en el tema como personas que pueden incluso no tener conocimientos sobre él, facilitando su comprensión y desarrollo.

8. Bibliografía.

[1]. C. Larman. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development.

[2]. Berners-Lee, T., Hender, J. y Lassila, O. (2001). The Semantic Web. *Scientific-American, 284* (5), 34-43. https://redirect.cs.umbc.edu/courses/graduate/691/spring13/01/papers/semanticWebSciAm.pdf

[3]. Brickley, D., Guha, R.V. (2014). *RDF Schema 1.1 W3C Recommendation 25 February 2014.* W3C. https://www.w3.org/TR/rdf-schema/

[4]. Cáceres, J. (2006). *La Web Semántica y el lenguaje RDF*. https://www.researchgate.net/profile/Jesus-Tello/publication/236658693_La_Web_Semantica_y_el_lenguaje_RDF/links/00463518bc94cead16000000/La-Web-Semantica-y-el-lenguaje-RDF.pdf

[5]. Cáceres, P. (s.f.). *Tema 3. Datos semánticos y Enlazados* [Material no publicado]. Universidad Rey Juan Carlos.

[6]. Cáceres, P. y Garrido M. A. (2021). *An MDA Process with which to Generate Semantic from Data Requirements.* Universidad Rey Juan Carlos, Madrid.

[7]. Caro, C. (2012). *Vocabularios estructurados, Web Semántica y Linked Data: oportunidades y retos para los profesionales de la documentación*. https://gredos.usal.es/bitstream/handle/10366/121953/DBD_UFF_ccaro.pdf?sequence=3&isAllowed=y

[8]. Castells, P. (2003). La web semántica. Sistemas interactivos y colaborativos en la web, 195-212. Disponible en: https://books.google.es/books?hl=es&lr=&id=2V9WB5s9lU4C&oi=fnd&pg=PA195&dq=la+web+semantica&ots=-uLOyyXp4m&sig=lYxhToMGV2jOGiVsUufpFrGEqGo#v=onepage&q=la%20web%20semantica&f=false [Visitado el 24 de noviembre de 2022].

[9]. Codina, L., & Rovira, C. (2006). La web semántica. In *Tendencias en documentación digital*. Trea. Disponible en: http://eprints.rclis.org/8899/ [Visitado el 24 de noviembre de 2022].

[10]. Datosabiertos. (s.f.). *¿Qué son Datos Abiertos?* https://datos.madrid.es/portal/site/egob

[11]. Datosabiertos. (s.f.). Universidades, colegios mayores, residencias universitarias y otros. https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=6ad94560b3104410VgnVCM1000000b205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default

8 Bibliografía

[12]. Donoso Villarroel, G. N. (2013). *Utilizando un enfoque dirigido por modelos, realizar un análisis y diseño de un sistema de gestión basado en BarBocca Pub & Restaurant.* Universidad Andrés Bello. https://repositorio.unab.cl/xmlui/bitstream/handle/ria/21962/a89526_Donoso_G_Utilizando_un_enfoque_dirigido_por_modelos_2013_Tesis_1.pdf?sequence=1&isAllowed=y

[13]. *Guía de Web Semántica*. (2023). Ceweb.br. https://ceweb.br/guias/web-semantica/es/

[14]. Herreros A. y Tórtola, I. (2022). *Proceso de anotación semántica de datos* [Trabajo Fin de Grado no publicado]. Universidad Rey Juan Carlos.

[15]. IONOS. (22 de diciembre de 2021). *¿En qué consiste la web semántica?* https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/web-semantica/

[16]. López Gómez, Y. (s.f.). *¿Qué es la World Wide Web WWW?*. LovTechnology. https://lovtechnology.com/que-es-la-world-wide-web-www/

[17]. Márquez, S. (2007). *La web semántica.* Google Books. https://books.google.es/books?hl=es&lr=&id=afuncWknStoC&oi=fnd&pg=PA55&dq=la+web+semantica&ots=L8vhW_yOOl&sig=_V3VCUezNdm_ftW28gVINIaAQ3k#v=onepage&q=la%20web%20semantica&f=false

[18]. Martín, M. (s.f.). *Búsqueda y Navegación Semántica para el Sistema de Catalogación de Métricas e Indicadores*. Universidad Nacional de La Pampa. https://www.w3.org/2001/sw/Europe/reports/dev_workshop_report_8/Taller-SW.PDF

[19]. Martínez, F. F., y Amaya, M. A. (2017). El papel de los metadatos en la Web Semántica. *Biblioteca universitaria*, *20*(1), 3-10. https://bibliotecauniversitaria.dgb.unam.mx/rbu/article/view/171/166

[20]. Martinez, L. I. (2008). *Componentes MDA para patrones de diseño* (Doctoral dissertation, Universidad Nacional de La Plata). http://sedici.unlp.edu.ar/handle/10915/4147

[21]. Mdn Web Docs. (s.f.). *World Wide Web*. https://developer.mozilla.org/en-US/docs/Glossary/World_Wide_Web

[22]. Méndez Rodríguez, E. M. (1999). *RDF: Un modelo de metadatos flexible para las bibliotecas digitales del próximo milenio*, Barcelona. https://earchivo.uc3m.es/handle/10016/25735

[23]. Ministerio de TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN. (2019, marzo). *30 años de la World Wide Web* | Un día como hoy se publicaba la primera página web de la historia. https://www.mitic.gov.py/noticias/30-anos-de-la-world-wide-web-un-dia-como-hoy-se-publicaba-la-primera-pagina-web-de-la-historia

[24]. Raistrick C., Francis P., Wright J., Carter, C. y Wilkie, I. (2004). *Model driven architecture with executable UML* (Vol. 1). Cambridge University Press. https://books.google.es/books?hl=es&lr=&id=72B8iG7e-NAC&oi=fnd&pg=PR8&dq=Raistrick,+C.+F.+(2004).+Model+Driven+Architectzre+with+Executable+UML.+Cambridge+University.+&ots=dVxCxYfYWS&sig=e-wkGbKbBjE0jwb_QsRr2HDOLRg#v=onepage&q&f=false

[25]. Sánchez D.M., Cavero J.M. y Marcos, E. (2005). *Ontologías y MDA: una revisión de la literatura.* Actas del II Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones, 21. https://www.researchgate.net/profile/Vicente-Pelechano/publication/220776014_MDA_vs_Factorias_de_Software/links/549403d10cf240d1cb4d22fc/MDA-vs-Factorias-de-Software.pdf#page=29

[26]. Saquete, R. (2013). *El impredecible futuro de la Web*. Human Level. https://www.humanlevel.com/blog/seo/el-futuro-de-la-web-semantica.html

9. Anexos.

Anexo I. Permiso de distribución de resultados del TFG.

Permiso de distribución de resultados del TFG

Datos del proyecto:

Título: Anotación semántica de datos mediante el método CSV@RDF

Tutor: Paloma Cáceres García de Marina

Autor/es: Marta Gámez Valero

Titulación: Grado en Matemáticas

Fecha de defensa: Julio 2023

Licencia de distribución:

Licencia del software desarrollado como parte del TFG, entregado a través de la aplicación de TFGs (gestion2.urjc.es/tfg). Marque la opción que corresponda:

- ☑ Licencia MIT (https://opensource.org/licenses/mit-license.php)
- ☐ Licencia Apache v2 (http://www.apache.org/licenses/LICENSE-2.0)
- ☐ Licencia GPLv3 (https://www.gnu.org/licenses/gpl-3.0.en.html)
- ☐ Otra
- ☐ No se concede ningún permiso de distribución.

Licencia de la memoria del TFG entregada a través de la aplicación de TFGs (gestion2.urjc.es/tfg).

Marque la opción que corresponda:

- ☑ Creative Commons Reconocimiento Internacional 4.0 (https://creativecommons.org/licenses/by/4.0/)
- ☐ Creative Commons Reconocimiento-SinObraDerivada 4.0 Internacional (https://creativecommons.org/licenses/by-nd/4.0/)
- ☐ Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional (https://creativecommons.org/licenses/by-sa/4.0/)
- ☐ Otra
- ☐ No se concede ningún permiso de distribución.

Permiso de distribución:

El Trabajo de Fin de Grado arriba especificado, ha sido defendido y calificado en la Escuela Técnica Superior de Ingeniería Informática de la Universidad Rey Juan Carlos. El tutor del trabajo y su autor (abajo firmantes) expresan su deseo de distribuir los elementos especificados más arriba según las licencias que se mencionan, y en su caso, que se incluyen como anexo.

Lo que ponen en conocimiento de la Universidad.

En MÓSTOLES, a 13 de JULIO de 2023

Fdo.: El Tutor

Paloma Cáceres García de Marina

Firmado digitalmente por Paloma Cáceres García de Marina
Fecha: 2023.07.13 17:48:09 +02'00'

Fdo.: Autor/es

Marta