**Trabajo de datos enlazados**

Alexander Gutiérrez Saavedra.

Marzo 2017.

Universidad Internacional Menéndez Pelayo

Máster Universitario en Investigación en Inteligencia Artificial

Web semántica y datos enlazados



**Tabla de Contenidos**

[Introducción 1](#_Toc511335070)

[Capítulo 1 Proceso de transformación 2](#_Toc511335071)

[Especificación 2](#_Toc511335072)

[Modelado y Generación 5](#_Toc511335073)

[Enlazado de datos 10](#_Toc511335074)

[Capítulo 2 Aplicación y explotación 17](#_Toc511335075)

[Consulta de recursos 17](#_Toc511335076)

[Consulta de información para un tipo de recurso 19](#_Toc511335077)

[Conclusiones 21](#_Toc511335078)

[Lista de referencias 22](#_Toc511335079)

# Introducción

Promover una fuente de datos enlazados, requiere conocer con más detalle el origen de los datos, desde su estructura hasta su dominio. Cuando tenemos la claridad suficiente de este detalle, es posible entender y diseñar la forma como se debe abordar un proyecto de datos enlazados, determinando si es necesario o no la construcción o reúso de ontologías que permitan exponer los datos a quien los requiera. Teniendo claridad en estos puntos se hace necesario explorar con profundidad la data que se desea enlazar, depurando las inconsistencias que no van de acuerdo con los requisitos definidos y mapeando las reglas allí contempladas, para finalmente generar nuestros datos enlazados.

A continuación, se resume en dos capítulos el proceso a través del cual se construyeron los datos enlazados, para finalmente ser consumidos desde una aplicación Java.

# Capítulo 1 Proceso de transformación

En este capítulo se detalla el proceso de transformación realizado para garantizar la correcta transformación de los datos; desde la selección, diseño; y publicación de los datos transformados. Siguiendo el aprendizaje del curso es necesario contar con los pasos del ciclo de vida de los datos enlazados: Especificación, Modelado, Generación, Enlazado, Publicación y Explotación. A continuación, se describe detalladamente cada uno de los pasos que aplica, ya que en este trabajo no llegaremos a la publicación

## Especificación

Para realizar el paso de especificación fue necesario partir esta tarea en dos: identificación y análisis de la fuente de datos y diseño de la URI.

Para la identificación de la información se determinó usar la fuente de datos expuesta por el Open data de la alcaldía de Medellín, la cual es una entidad que expone datos abiertos de manera asequible y gratuita sin restricción alguna. Estos datos cumplen las características de ser un escenario real, estar disponibles y accesibles sin restricción, son datos que pueden ser manipulados y procesados de manera automática y además el dominio de los datos, permite enlazarlos con otras entidades genéricas.

Los datos manejados bajo esta entidad contienen información de diferentes ámbitos enmarcados en la ciudad de Medellín Colombia tales como: hábitat y medio ambiente, infraestructura, ordenamiento territorial, movilidad, educación, cultura, educación, salud, seguridad, desarrollo económico y población.

Para este trabajo se tomaron los datos asociados a la movilidad vehicular con la subcategoría de accidentes de tránsito, registrados por la Secretaría de Movilidad de la Alcaldía de Medellín, desde enero 1 del año 2017 hasta el 31 de julio del mismo año.

Para entender la información de la data seleccionada, es necesario tener claridad en el concepto de accidente de tránsito como un evento, generalmente involuntario, causado mínimo por un vehículo en movimiento, el cual genera daños a personas y/o bienes, que afectan la circulación normal de los vehículos que se movilizan en la vía o vías cercanas a la generación del evento.

A nivel de D:\Personal\Master\Materias\Web semántica\GIT\AlexanderGutierrezSaavedra\4. Documentacion\imagenes\ccby.png se identificó que no tiene un tipo de licencia de uso definida, pero si se declara en los términos de uso la usabilidad sin restricción[[1]](#footnote-2). También se identificó que el titular de los datos es Grupo SIG de la Alcaldía Medellín

Los datos fueron obtenidos en un archivo csv separado por comas, con la información de los eventos generados alrededor de un accidente (coordenadas de ubicación X, Y, fecha, hora, día, periodo, clase, dirección, tipo, gravedad, barrio, comuna y diseño).

El archivo contiene las siguientes características:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Tamaño** | **Filas** | **Tema** |
| Accidentalidad\_2017 | 4.7 MB | 24.353 | Accidentes de tránsito registrados por la Secretaría de Movilidad de la Alcaldía de Medellín |

Analizando la fuente de datos se pudo detectar la información de la tabla 1:

Tabla 1. Estructura de la data de accidentalidad.

|  |  |  |  |
| --- | --- | --- | --- |
| Columna | Tipo | Comentario/rango | Problemas |
| OBJECTID | Long | Identificador único | Ninguno |
| X | Double | Coordenada x | Ninguno |
| Y | Double | Coordenada y | Ninguno |
| RADICADO | String (50) | 12 - 1577959 | Ninguno |
| FECHA | Date(8) | 2017/01/01-2017/07/31 | Ninguno |
| HORA | String(8) | Único | Ninguno |
| DIA | String(36) | Único | Ninguno |
| PERIODO | String(4) | Solo 2017 | Ninguno |
| CLASE | String(50) | Único | Ninguno |
| DIRECCION | String(100) | Único | Ninguno |
| DIRECCION\_ENC | String(500) | Único | Ninguno |
| TIPO\_GEOCOD | String(50) | Único | Ninguno |
| GRAVEDAD | String(50) | Único | Ninguno |
| BARRIO | String(50) | Único | Ninguno |
| COMUNA | String(50) | Único | Ninguno |
| DISENO | String(50) | Único | Ninguno |

Para la creación de la URI se constituyó a partir de:

1. Raíz de la URI: http://antioquia.opendata.co
2. Nombramiento de recursos: http://antioquia.opendata.co/<nombre\_recurso>/

## Modelado y Generación

Como definición de las ontologías a usar, se determinó crear una propia basada en la fuente de datos, dado que los datos que allí se maneja son muy propios de este problema y no se encontró un vocabulario adecuado para las necesidades requeridas. Para la creación de la propia ontología, se ejecutaron los siguientes pasos:

1. Definir el dominio y el alcance

El dominio de la ontología está enmarcado con la información contenida en un accidente de tránsito, el cual se entiende como un evento, generalmente involuntario, causado mínimo por un vehículo en movimiento, el cual genera daños a personas y/o bienes, que afectan la circulación normal de los vehículos que se movilizan en la vía o vías cercanas a la generación del evento. Esta ontología nos ayudará a catalogar un accidente según el tipo, ubicación, gravedad del accidente y fecha de ocurrencia. La ontología nos ayudará a responder preguntas asociadas a los accidentes de tránsito exclusivamente, bajo la información delimitada explicada anteriormente. Esta ontología será usada sólo para temas de práctica, no tendrá un uso comercial ni de otro tipo.

1. Términos importantes

Algunos de los términos importantes que se usarán en esta ontología son: accidente, ubicación, radicado, periodo, clase, dirección, gravedad, comuna, diseño, vehículo.

1. Definición de clases y jerarquías

Se definieron algunas clases: Accidente, Ubicación, Dirección, Ocurrencia.

1. Definición de propiedades

Para cada clase se definieron algunas propiedades. Para la clase Accidente se tiene: identificador, radicado, clase, gravedad

Para la clase Ubicación se tiene: coordenadaX, CoordenadaY

Para la clase Dirección se tiene: nomenclatura, barrio, comuna, diseño.

Para la clase Momento se tiene: hora, día, fecha, periodo.

1. Implementacion

Se procede con la creación de la ontología, para lo cual se usó la herramienta Protégé sobre Windows. Allí se realizaron los siguientes pasos:

1. Creación de la URI, según lo definido en el estándar de esta ontología.
2. Se crearon las clases, como se ve en la figura 1.

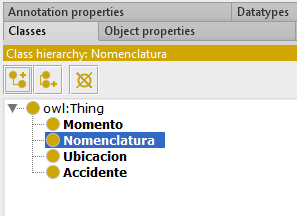


Figura 1. Clases de la ontología.

1. Se definieron las propiedades a cada una de las clases. Ver figura 2.

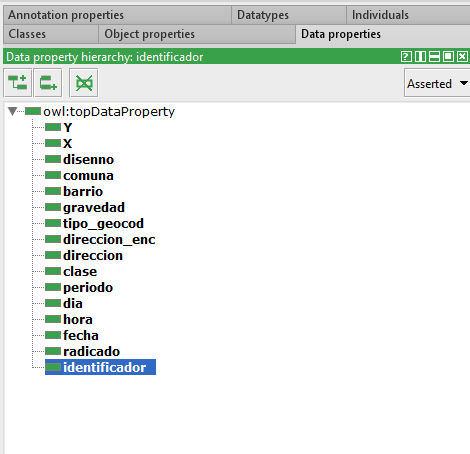


Figura 2. Propiedades de las clases de la ontología

1. Se asignaron las propiedades y tipo de dato a cada una de las clases. Ver figura 3.

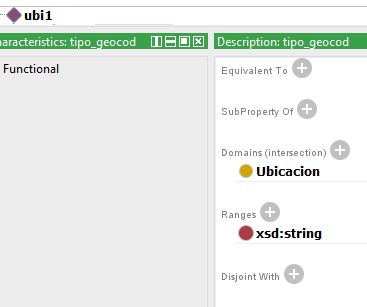


Figura 3. Propiedades de las clases de la ontología.

1. Se crearon las relaciones y se asignaron las clases correspondientes a cada relación, ver figura 4.

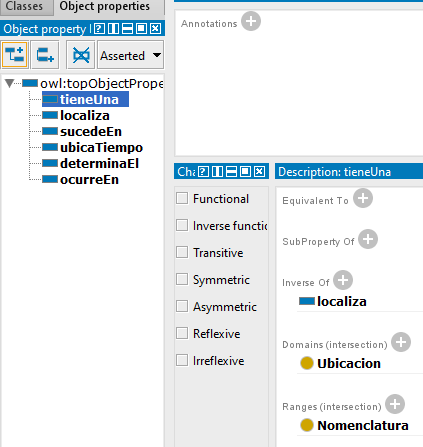


Figura 4. Relaciones y clases asociadas.

1. Se crearon instancias de clases con data de ejemplo y tipos de datos, para ver la forma de las relaciones, ver figura 5 y 6.

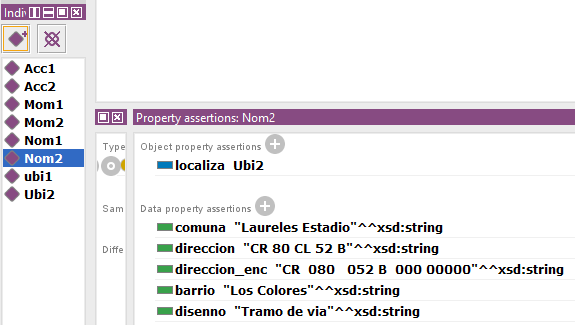


Figura 5. Ejemplos de clases dentro de la ontología.

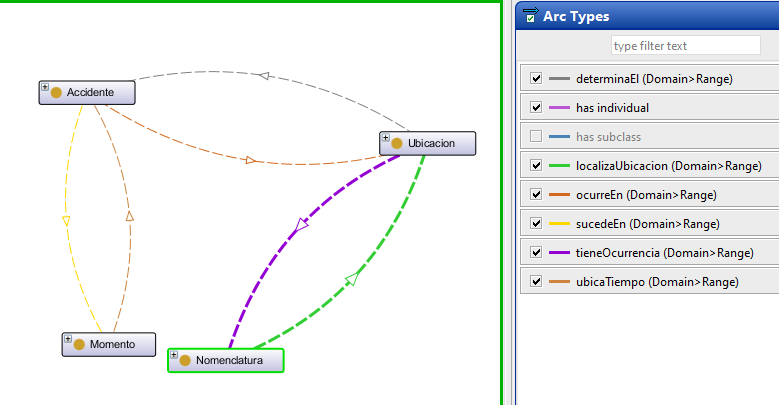


Figura 6 Grafo de relaciones entre las clases de la ontología.

## Enlazado de datos

Para esta labor se utilizó el aplicativo LodRefine de la siguiente manera:

1. Se crea un proyecto nuevo en la herramienta, con los datos del open data de la alcaldía de Medellín en formato csv. Ver figura 7 y 8.

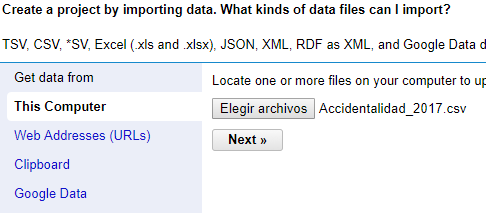


Figura 7. Creación del proyecto en LodRefine.

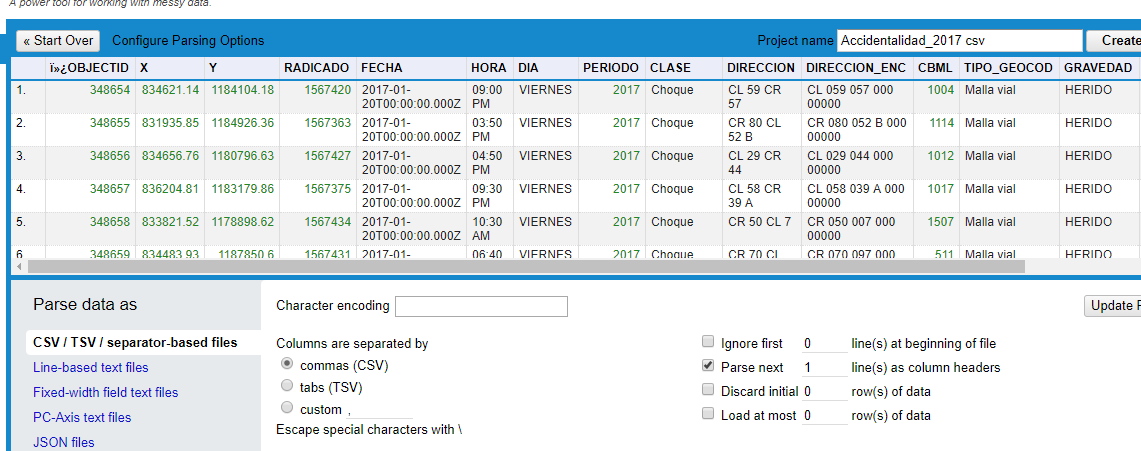


Figura 8. Proyecto creado en LodRefine.

1. Se definen las propiedades de importación de los datos (24.352 registros) y se procede a finalizar la creación del proyecto. Figura 9.

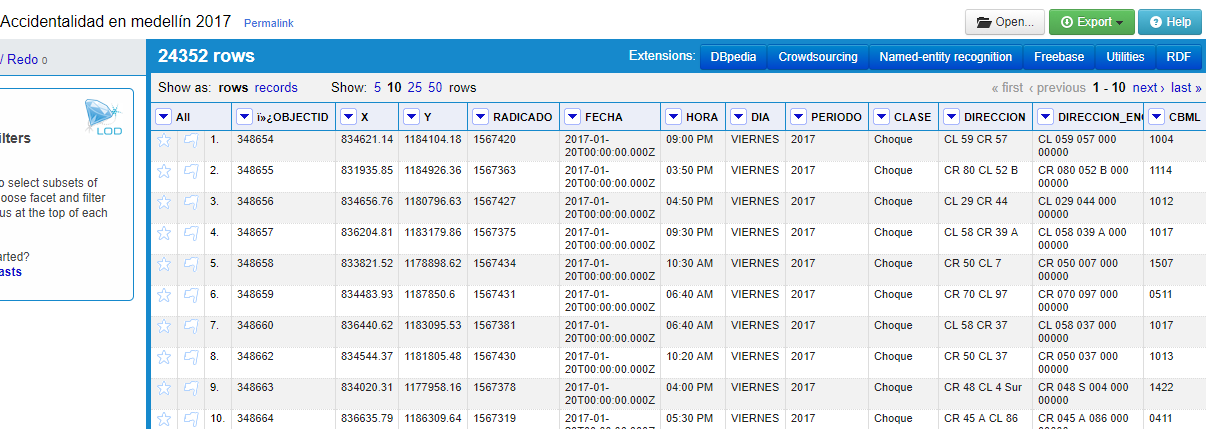


Figura 9. Archivo csv importado correctamente en LodRefine..

1. Se procede con la transformación de los datos para que concuerden los tipos de datos con los ya identificados anteriormente. En la figura 10 se pueden ver las columnas trasformadas en color verde.

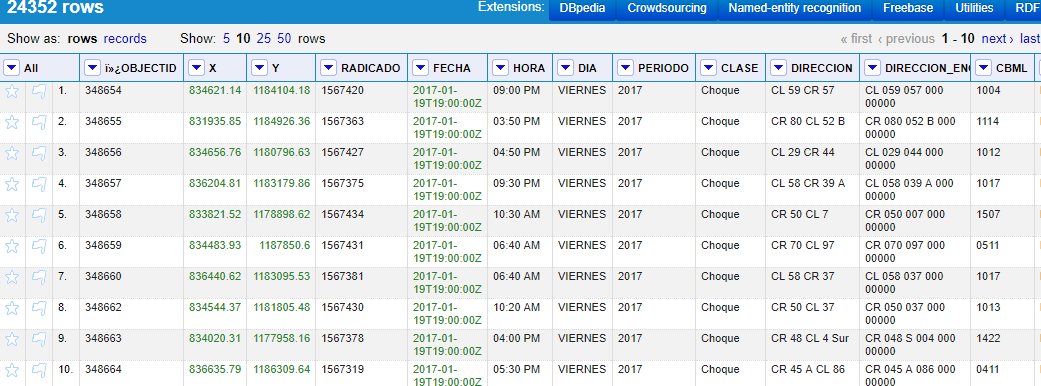


Figura 10. Transformación de columnas en LodRefine.

1. Realizamos algunos cambios en la data para corregir valores con la ayuda de los Facet, que fueron cargados con error debido al uso de tildes y ñ. En la figura 11 se puede observar este cambio.

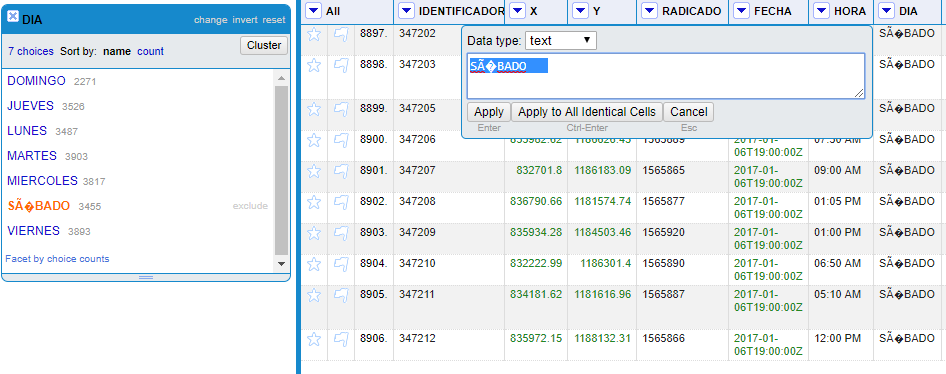


Figura 11. Corrección de data en LodRefine..

1. Se crearon las columnas CARRERA y CALLE a partir de la columna DIRECCIÓN para dar mayor claridad. Ver figura 12.

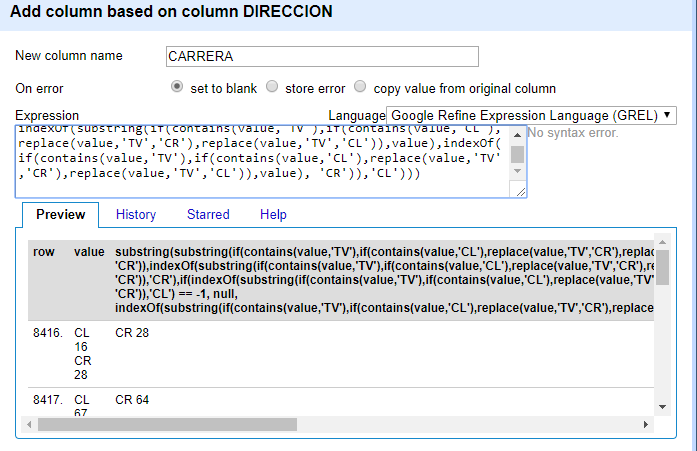


Figura 12. Adición de columnas nuevas en LodRefine..

1. Se eliminan las columnas DIRECCIÓN, DIRECCION\_ENC, ya que estas fueron reemplazadas por CARRERA y CALLE. Ver figura 13.

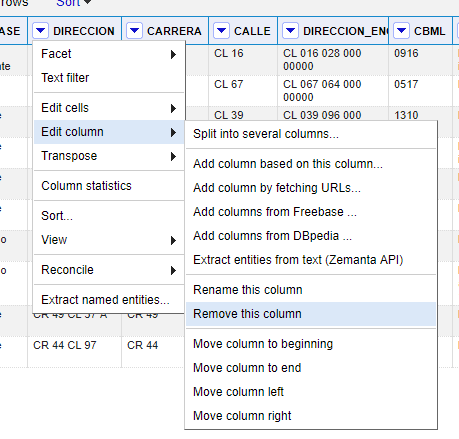


Figura 13. Eliminación de columnas en LodRefine.

Una vez tenemos la data depura, procedemos a la definición del mapping entre el esquema de datos definido y la ontología creada en la primera parte. Para ello procedemos de la siguiente forma:

1. Una vez se transformaron y corrigieron los datos, se procede a la definición del mapping. Para ello se carga la ontología creada en LodRefine como se ilustra en la figura 14.

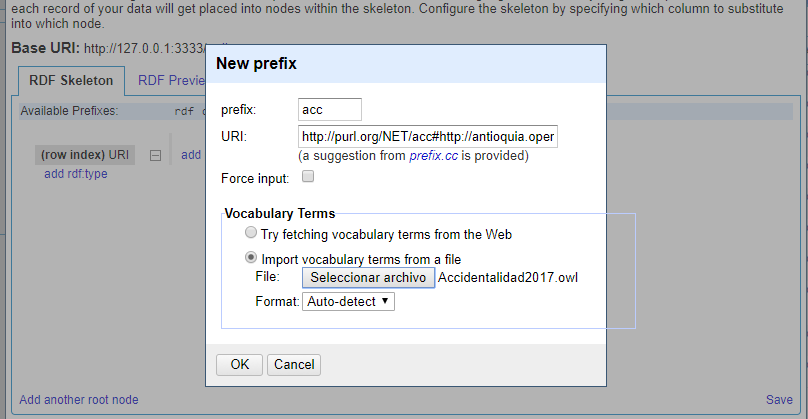


Figura 14. Carga de ontologías en LodRefine..

1. Se define la URI base para los individuos como se ilustra en la figura 15.

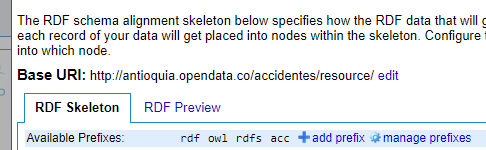


Figura 15. Modificación de la URI en LodRefine.

1. Se crean las propiedades para cada tipo RDF añadido. Ver figura 16.



Figura 16. Adición de propiedades en LodRefine.

1. Finalmente se exportan los datos en formato turtle. Ver figura 17.

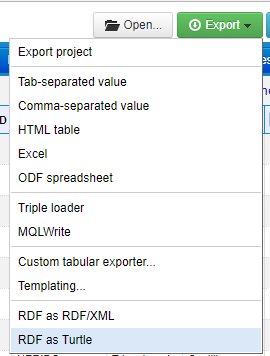
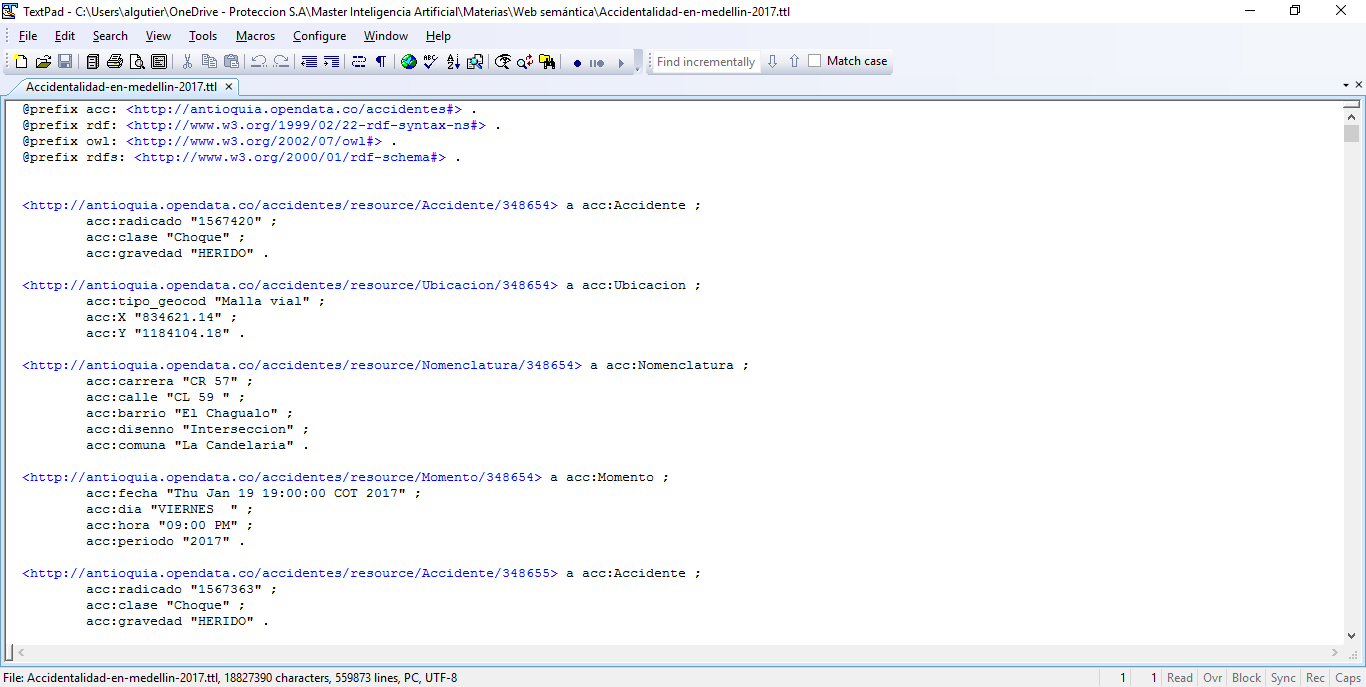
****

Figura 17. Export de data en formato turtle.

# Capítulo 2 Aplicación y explotación

En capítulo se concentra en la explotación de los datos enlazados generados a través de las consultas realizadas utilizando SPARQL. Para realizar esta labor se construyó un programa sencillo en java utilizando las librerías de Jena y realizando consultas SPARQL. En este ejercicio se realizaron solo consultas con la ontología propia, ya que como se explicó anteriormente la estructura de la data no se habilitaba para usar ontologías existentes. A continuación, se describe cada una de las consultas realizadas.

## Consulta de recursos

En esta consulta se buscaron los 10 primeros recursos de la data para cada uno de los recursos definidos (Ubicación, Nomenclatura, Accidente y Momento)

Para su correcto funcionamiento se realizaron los siguientes pasos:

1. Construcción de un modelo vacío de rdf-jena.
2. Lectura del archivo rdf de una ruta local.
3. Carga del archivo leído en el modelo creado en el paso 1.

La figura 18 muestra los 3 primeros pasos.

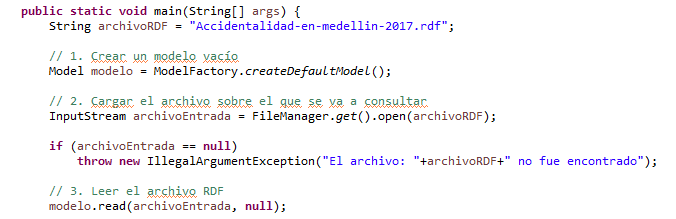


Figura 18. Pasos iniciales en la realización de consultas a documentos RDF.

1. Construcción de 4 consultas SPQRQL, una por cada tipo de recurso. Ver la figura 19.

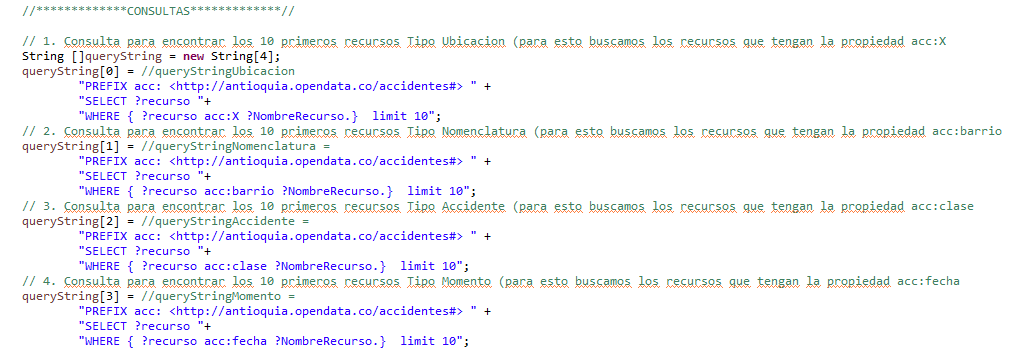


Figura 19. Consultas SPARQL para conocer los recursos por cada tipo en el archivo RDF.

1. Ejecutar los Queries. Para esto se itera en un ciclo for cada uno de los queries y en otro ciclo por cada resultado, se imprimen los valores correspondientes. En las figuras 20 y 21 se observan estos pasos.

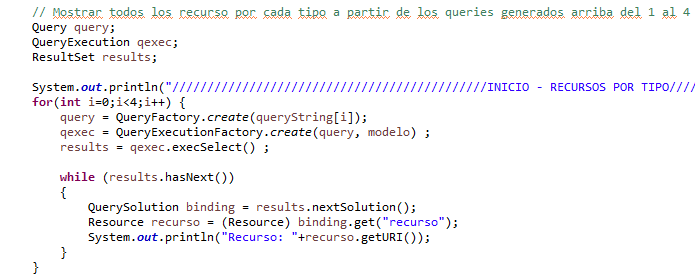
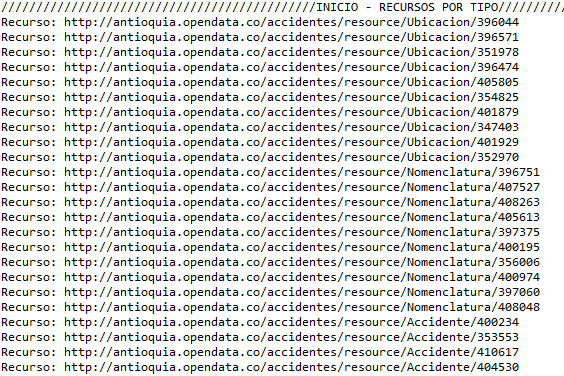


Figura 20. Ciclos de ejecución de queries SPARQL.

  
Figura 21. Los 10 primeros recursos por tipo.

## Consulta de información para un tipo de recurso

En esta consulta se definió la necesidad de encontrar la clase, gravedad y radicado de un Accidente en el cual la clase sea un incendio. Para lograr esta consulta se realizaron los siguientes pasos:

1. Uso de los pasos de la consulta anterior del 1 al 4.
2. Construcción del query SPARQL. Para ello fue necesario incluir en la sentencia Where los campos a través de los cuales se obliga a traer como resultado al tipo de recurso Accidente, además se definen 3 variables. En la figura 22, se puede observar este paso.

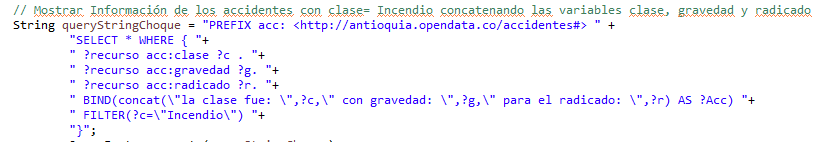
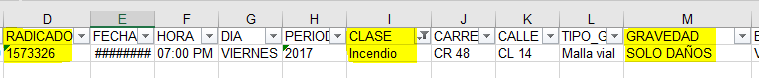


Figura 22. Consulta SPARQL con creación de variable a partir de otras.

1. Haciendo uso de la funcionalidad BIND, se crea una nueva variable de respuesta, a partir de las variables definidas en el punto anterior.
2. Se hace uso de la función concat, para concatenar los textos y las variables con el fin de generar una variable con este contenido
3. Se renombra la nueva variable con el operador AS.
4. Se ejecuta el query
5. Se imprime en pantalla el resultado. Como se observa en la figura 23 y 24, el resultado del query es el mismo al mirar los datos originales del archivo de Excel.



Figura 23. Resultado de la Consulta SPARQL con creación de variable a partir de otras.

 Figura 24. Resultado de filtro en Excel similar a la Consulta SPARQL con creación de variable.

# 

# Conclusiones

A partir del trabajo realizado, se puede concluir que el uso de datos enlazados representa un eje fundamental a la hora de consultar información en la web de manera oportuna.

En este sentido, se pueden observar las ventajas que dan los Datos Enlazados en la descripción de los recursos de bibliotecas que deseen compartir sus datos, tanto para entidades o instituciones como usuarios afines.

Es claro que la aplicación tecnológica de la Web Semántica y Datos enlazados en la web debe tener ajustes que permitan mejorar algunos aspectos, buscando corregir algunas limitaciones y problemas que se pueden presentar por la estructura de algunos datos, sin embargo, también cabe mencionar que existen muchas herramientas que nos permiten construir datos enlazados con calidad. Para ello se requiere cierta experiencia que se adquiere con la manipulación constante y efectiva de estas herramientas y datos.

La práctica fue muy enriquecedora, ya que se ponen a prueba los conocimientos adquiridos durante todo el curso y permite descubrir las falencias que aún se puedan tener. En general considero esta metodología uso de datos enlazados debe ser promovida para alcanzar unos niveles considerables en los que diferentes aplicaciones puedan obtener muchos beneficios.

# Lista de referencias

GREL String Functions. (s.f.). OpenRefine Expression Language. Recuperado de <https://github.com/OpenRefine/OpenRefine/wiki/GREL-String-Functions>

Ontology Pitfall Scanner (s.f.). Recuperado de <http://oops.linkeddata.es/response.jsp>

Anatomy of a simple SPARQL query. (s.f.). Recuperado de <https://www.ibm.com/developerworks/library/j-sparql/>

Cicourel. I., b. (agosto 29 de 2016). Semantic Web 101: SPARQL. Recuperado el 30 de marzo de 2018 de <https://www.codementor.io/isaib.cicourel/sparql-query-using-endpoints-101-du10884g8>

Fariz. D.,Werner N. (2014-2015). Handling RDF in Java: The Jena Framework. Recuperado el 30 de marzo de 2018 de <http://www.inf.unibz.it/~nutt/Teaching/SemTechs1415/SemTechsLabs/exercises-2.pdf>

1. https://www.arcgis.com/home/item.html?id=b7f7158734f64e7f849df6ac4a6b10f1 [↑](#footnote-ref-2)