Universidade Vigo

GRADO EN INTELIGENCIA ARTIFICIAL
WEB SEMÁNTICA
ENTREGABLE: RDF
Milagros Fernández Gavilanes • *Octubre 2025*

1. CREAR UN PERFIL EN FOAF

A modo de ejemplo, puedes descargar un ejemplo – mi perfil – aquí en: `https://github.com/mfgavilanes/WS/blob/main/perfil.ttl`. Puedes editarlo para crear tu perfil personal.
Tendrás que:

- Cambiar el `@base` para un lugar donde puedas subir tu perfil en la Web (debe ser el URL del archivo). Si no tienes acceso a un espacio Web, puedes poner `http://tu-nombre.org/perfil.ttl` reemplazando "`tu-nombre`" con tu nombre y apellidos sin espacios. Lo más importante es que tenga un valor único para el `@base` en la clase, y que sea un URL valido.
- Cambiar todos los datos que figuran en el fichero y adecuarlos a tu situación: (`birthday: MM-DD`), edad (`age`), género (`gender`), pagina web (`homepage`, puedes utilizar un Facebook, LinkedIn, Twitter, o inventar una). Puedes mentir.
- La propiedad `based_near` indica una ubicación. Para encontrar el identificador de su ubicación, vete a `http://www.geonames.org/`, busca el nombre del lugar de tu procedencia, haz click en el lugar y encuentra su numero y reemplaza `6359872` en mi perfil con ese nuevo número. Si no puedes encontrar un número, puedes utilizar el mismo que yo, `6359872` que es el de Ourense.
- Añadir al perfil el contacto de al menos tres compañeros de clases con la propiedad `knows`. Para hacer eso, tienes que preguntar a tus compañeros por sus IRIs. Tu IRI es el valor de `@base+#yo` (el `string #yo` es el subject y es un IRI relativo que va a ser agregado a la base). En mi caso, por ejemplo, mi IRI es `https://github.com/mfgavilanes/WS/perfil.ttl#yo`. Usa una coma para separar valores y un punto y coma para terminar los valores para esa propiedad.
- Indicar al menos tres de tus intereses. Para hacer eso, deber usar IRIs de Wikidata. Estos se encuentan en `https://www.wikidata.org/` y buscar tus intereses. Puedes cambiar el idioma a español si quieres buscar en español; no importa el idioma que usas (el resultado va a ser el mismo). Busques lo que busques siempre empezará por `Q` y tendrás que remplazar los códigos por tus intereses. Agrega al menos tres intereses. Usa una coma para separar valores y un punto y coma para terminar los valores para esa propiedad.
- Agregar al menos tres películas que te gustan. De nuevo, deberías encontrar los IRIs de Wikidata (como antes).
- Finalmente, cambia el nombre del perfil a tu nombre.

Para verificar que la sintaxis de tu perfil es correcto y convertirlo a formato `RDF/XML`, copia y pega tu perfil en: `https://rdfshape.weso.es/dataConvert`.

2. Generar RDF con Apache Jena

Estas instrucciones son para crear el proyecto. Es necesario introducir en el ID de grupo del proyecto: `entregable.jena`. En el ID de artefacto, escribe `websem`.

Una vez creado, el objetivo del proyecto es generar datos RDF a partir de fuentes no RDF. Utiliza los mini guiones aportados en las prácticas anteriores sobre Jena para generar el grafo RDF mediante programación.

Descarga el conjunto de datos sobre Renfe. Este conjunto de datos, publicado como datos abiertos por Renfe, describe los horario de los servicios de viajeros prestados por los trenes de Alta Velocidad, Larga Distancia y Media Distancia.

En el archivo `stops.txt` del conjunto de datos que descargaste se describen los nombres y la ubicación de las estaciones de tren con parada. Lo relevante son los `stop_id`, `stop_name`, `stop_lat` y `stop_lon`.

2.1 Parte 1

Utiliza Jena para crear un grafo RDF que describa los nombres y las ubicaciones de las estaciones de tren del archivo mencionado anteriormente. Los elementos geolocalizados se pueden describir como instancias (`rdf:type`) de la clase `http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing`, generalmente abreviadas como `geo:SpatialThing`. El vocabulario de geoposicionamiento `WGS84` también proporciona propiedades RDF para latitud (`geo:lat`) y longitud (`geo:long`). Genera una IRI para cada parada según su `stop_id`.

A partir de una sola línea del archivo, el RDF resultante debería ser (en Turtle):

```
@prefix ex: <http://www.ejemplo.com/> .
@prefijo geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:17000 a geo:SpatialThing;
        rdfs:label  "Madrid-Chamart n-Clara Campoamor"@es;
        geo:lat  "40.4711789"^^xsd:decimal;
        geo:long "-3.6829524"^^xsd:decimal .
```

El conjunto de datos de entrada sigue el formato de la *General Transit Feed Specification* (GTFS). La documentación técnica de cada campo (cada columna) está disponible públicamente. Amplía tu código para que el grafo RDF de salida también capture otros campos del conjunto de datos y usando contenedores.

Recomendación: Para probar tu código de forma más rápido, haz un archivo que contenga solo una muestra de los datos (200 líneas). Esto también evitará que se sature la memoria.

2.2 Parte 2

El punto de partida es el fichero RDF obtenido a partir del enunciado anterior, el cual contiene información sobre distintos lugares geográficos de las estaciones, representados como instancias de `geo:SpatialThing` del vocabulario WGS84. Cada recurso dispone de propiedades de latitud (`geo:lat`) y longitud (`geo:long`).

El objetivo es crear un programa con la API de Jena que recorra el modelo RDF (sin emplear SPARQL) y dadas las coordenadas de un rectángulo, filtre los recursos que se encuentren dentro de ese rango específico de coordenadas geográficas.

3. Referencias útiles

1. El sintaxis que vamos a ver es RDF/XML. La documentación está aquí si la necesitas: `https://www.w3.org/TR/REC-rdf-syntax/`.
2. También veremos la sintaxis Turtle. La documentación está aquí si la necesitas: `https://www.w3.org/TR/turtle/`
3. Finalmente, el vocabulario que vamos a utilizar es FOAF. La documentación está aquí: `http://xmlns.com/foaf/spec/`