SIW PL9. Extracción de información semántica.

Hugo Fonseca Díaz
uo258318@uniovi.es

Escuela de Ingeniería Informática. Universidad de Oviedo.

19 de noviembre de 2020

1. Información estructurada

En esta primera sección se obtendrá información estructurada de tres textos mediante el uso de tipos definidos en `schema.org` [1]. Dicha información se define a continuación:

- **Miles Davis:** es una entidad de tipo `Person`. Sus propiedades son:
  - `familyName`: Davis
  - `givenName`: Miles
  - `hasOccupation`: `Occupation`(`name`: jazz musician)
  - `nationality`: `Country`(`name`: United States of America)
  - `url`: `https://www.wikidata.org/wiki/Q93341`
- **Barack Obama:** es una entidad de tipo `Person`. Sus propiedades son:
  - `familyName`: Obama
  - `givenName`: Barack
  - `hasOccupation`: `Occupation`(`name`: President)
  - `url`: `https://www.wikidata.org/wiki/Q76`
- **European Union:** es una entidad de tipo `Organization`. Sus propiedades son:
  - `name`: European Union
  - `legalName`: European Union
  - `url`: `https://www.wikidata.org/wiki/Q458`
- **Washington:** es una entidad de tipo `City`. Sus propiedades son:
  - `name`: Washington DC

- `url`: `https://www.wikidata.org/wiki/Q61`

- **Euro a Dólar:** es una entidad de tipo `ExchangeRateSpecification`. Sus propiedades son:
  - `currency`: EUR
  - `currentExchangeRate`: `UnitPriceSpecification`(`priceCurrency`: USD, `price`: 1.3)
  - `url`: `https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/index.en.html`
- **The New York Times:** es una entidad de tipo `Newspaper`. Sus propiedades son:
  - `name`: The New York Times
  - `url`: `https://www.wikidata.org/wiki/Q9684`
- **John McCarthy:** es una entidad de tipo `Person`. Sus propiedades son:
  - `familyName`: McCarthy
  - `givenName`: John
  - `hasOccupation`: `Occupation`(`name`: computer scientist)
  - `url`: `https://www.wikidata.org/wiki/Q92739`
- **LISP:** es una entidad de tipo `ComputerLanguage`. Sus propiedades son:
  - `name`: LISP
  - `url`: `https://www.wikidata.org/wiki/Q132874`

2. Modelado RDF

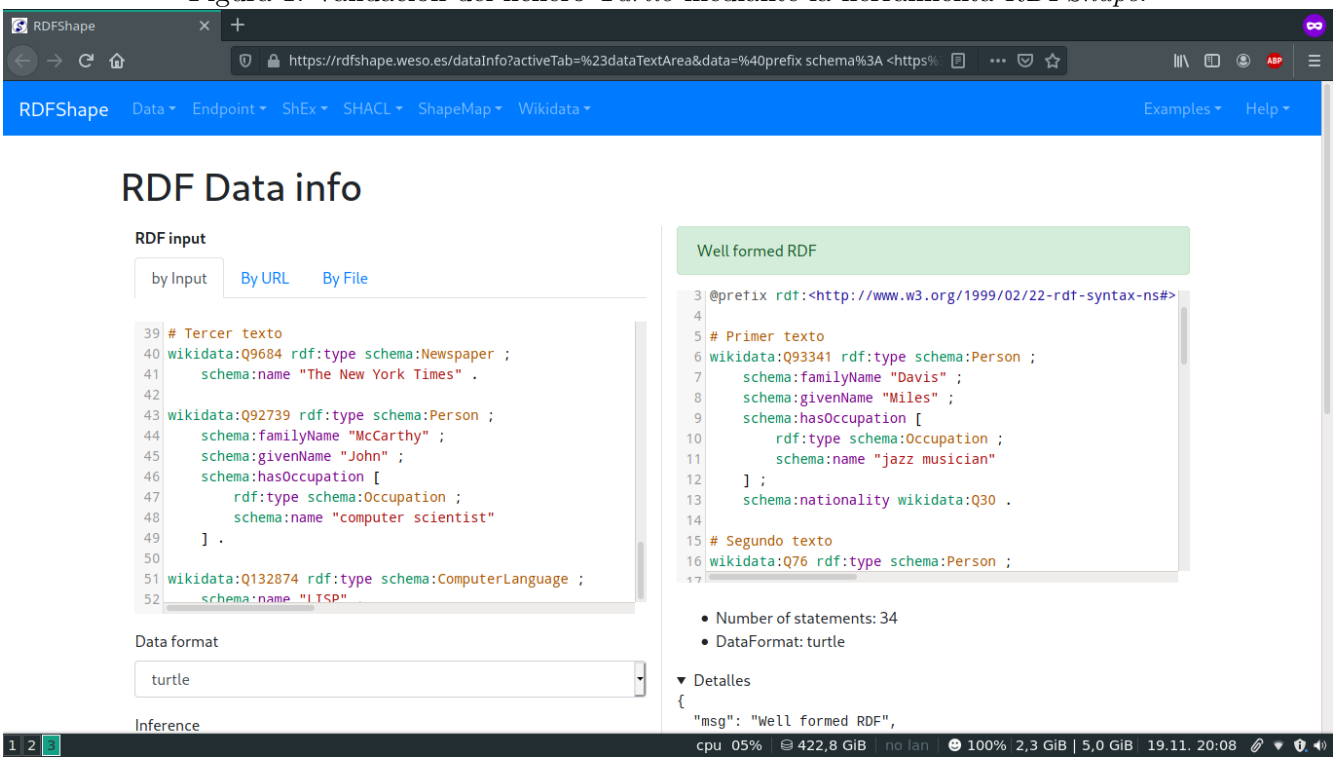
En esta sección se muestra el modelado *RDF* en formato *Turtle* con el que se ha representado la información listada previamente. Dicho modelado es el siguiente:

```
@prefix schema: <https://schema.org/> .
@prefix wikidata: <https://wikidata.org/> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

# Primer texto
wikidata:Q93341 rdf:type schema:Person ;
    schema:familyName "Davis" ;
    schema:givenName "Miles" ;
    schema:hasOccupation [
        rdf:type schema:Occupation ;
        schema:name "jazz musician"
    ] ;
    schema:nationality wikidata:Q30 .
```

```
# Segundo texto
wikidata:Q76 rdf:type schema:Person ;
    schema:familyName "Obama" ;
    schema:givenName "Barack" ;
    schema:hasOccupation [
        rdf:type schema:Occupation ;
        schema:name "president"
    ] .

wikidata:Q458 rdf:type schema:Organization ;
    schema:name "European Union" ;
    schema:legalName "European Union" .

wikidata:Q61 rdf:type schema:City ;
    schema:name "Washington DC" .

<https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/index.en.html
    rdf:type schema:ExchangeRateSpecification ;
    schema:currency "EUR" ;
    schema:currentExchangeRate [
        rdf:type schema:UnitPriceSpecification ;
        schema:priceCurrency "USD" ;
        schema:price 1.3
    ] .

# Tercer texto
wikidata:Q9684 rdf:type schema:Newspaper ;
    schema:name "The New York Times" .

wikidata:Q92739 rdf:type schema:Person ;
    schema:familyName "McCarthy" ;
    schema:givenName "John" ;
    schema:hasOccupation [
        rdf:type schema:Occupation ;
        schema:name "computer scientist"
    ] .

wikidata:Q132874 rdf:type schema:ComputerLanguage ;
    schema:name "LISP" .
```

En la **Figura 1** se observa el output de la herramienta *RDFShape* al introducir el fichero en formato *Turtle*. Se observa que el fichero está bien formado. Una vez hecho eso, convertimos el fichero a *JSON-LD* mediante el uso de la herramienta *RDF Translator* [2]. En la **Figura 2** se puede ver la validación de la *Google Structured Data Testing Tool* [3] sobre el *JSON-LD* generado previamente.

Figura 1: Validación del fichero *Turtle* mediante la herramienta *RDFShape*.



3. Obtención automática de información estructurada

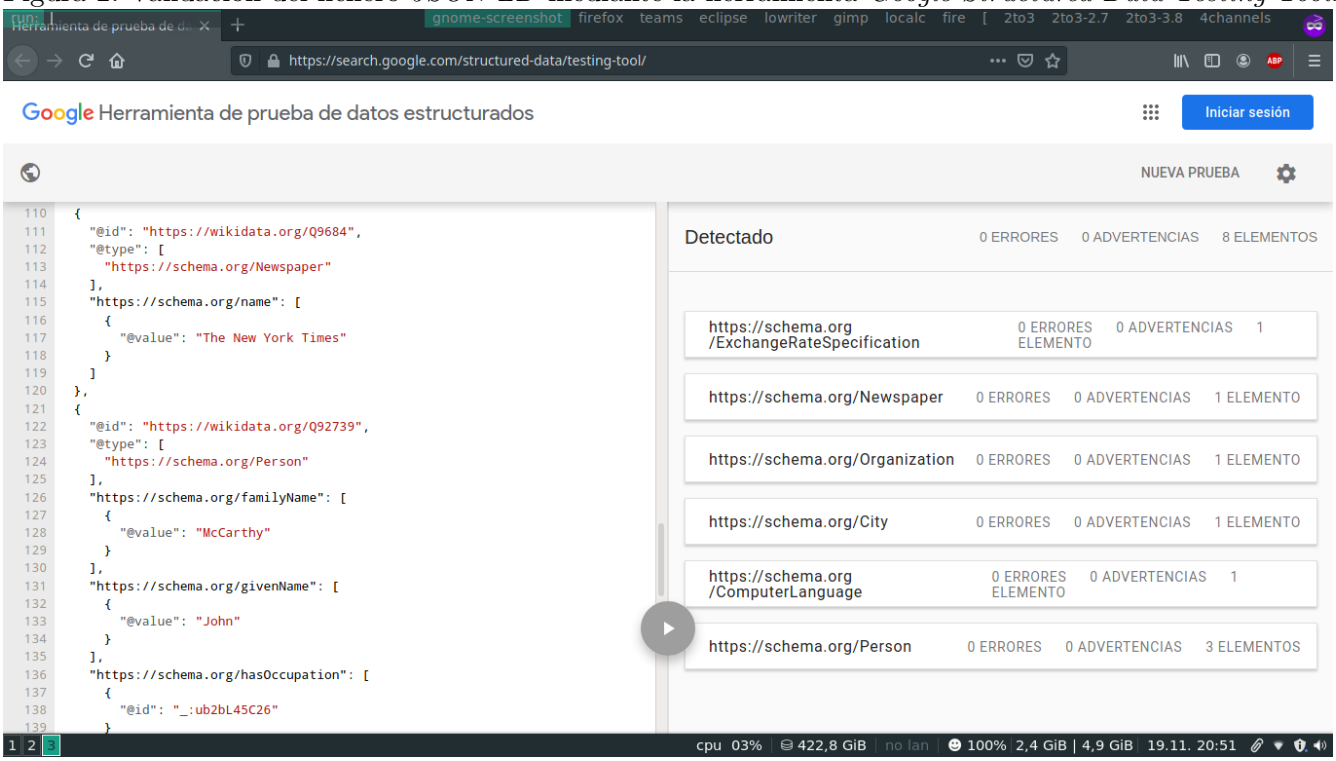
En esta parte del documento se procederá a obtener información estructurada mediante el uso de varias herramientas. Estas son *Intelligent Tagging* de *PermID* [4], *DBpedia Spotlight* [5] y *FRED* [6].

A continuación se muestra una serie de ejemplos en forma de capturas de pantalla. Estos ejemplos ilustrarían el proceso realizado para obtener la información estructurada por medio de la herramienta *FRED* para el texto *"Miles Davis was an american jazz musician."*. En la **Figura 3** se muestra la herramienta *FRED* con el texto en lenguaje natural. En la **Figura 4** se muestra la conversión a *JSON-LD* del fichero *RDF-XML* obtenido en el previo paso (también se transforma a *N3* para una inspección manual, aunque no se muestra en la captura). Por último, en la **Figura 5** se muestra la validación del fichero *JSON-LD*. Estos pasos se realizan para todos los textos y los ficheros obtenidos pueden observarse en las carpetas adjuntadas a este documento.

4. Cuestiones sobre la obtención automática

En esta última sección se responderá a una serie de cuestiones relacionadas con la obtención automática de información semántica. Dichas cuestiones tratan sobre las herramientas utilizadas pero también sobre temas más generales como la creación y similitud de ontologías.

Figura 2: Validación del fichero *JSON-LD* mediante la herramienta *Google Structured Data Testing Tool*.



4.1. ¿Qué ontologías usa cada servicio para "tipar"las instancias detectadas en el texto?

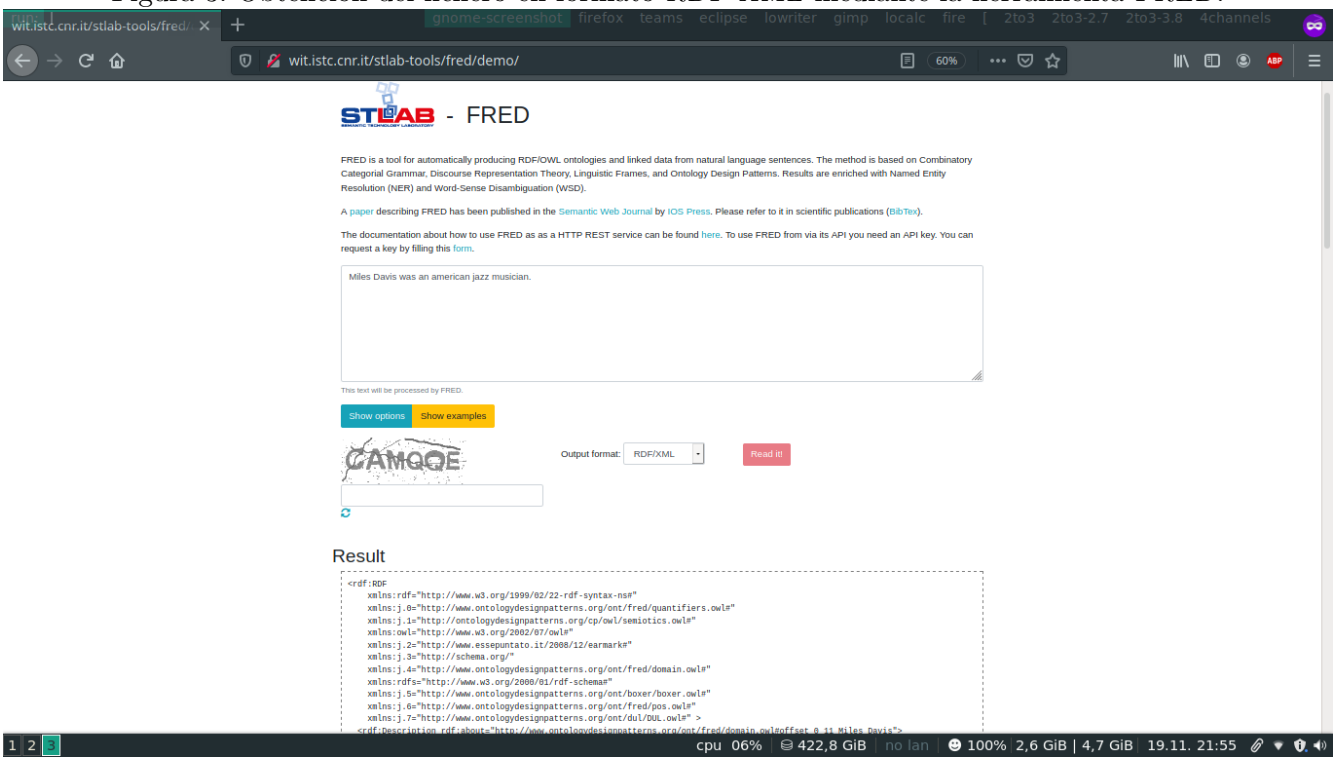
Intelligent Tagging utiliza una ontología propia, mientras que *FRED* y *DBpedia* utilizan una mezcla de ontologías propias y externas. En el caso de *FRED* se tiene, además de la suya propia, ontologías como *DOLCE+DnS Ultralite (dul)*, *schema.org* o la propia *DBpedia* mientras que en esta se encuentra *owl* o *schema.org*.

4.2. ¿Existe algún tipo de dichas ontologías que pudiera considerarse equivalente a otro tipo en *schema.org*?

Existen varios tipos en las ontologías citadas previamente similares a los que nos podemos encontrar en *schema.org*. A continuación se citan varios de ellos:

- **Person**: presente en todas las herramientas
- **ComputerLanguage** en *schema.org*
  - **Programming_languages**: propia de *DBpedia*
  - **ProgrammingLanguage**: propia de *FRED*

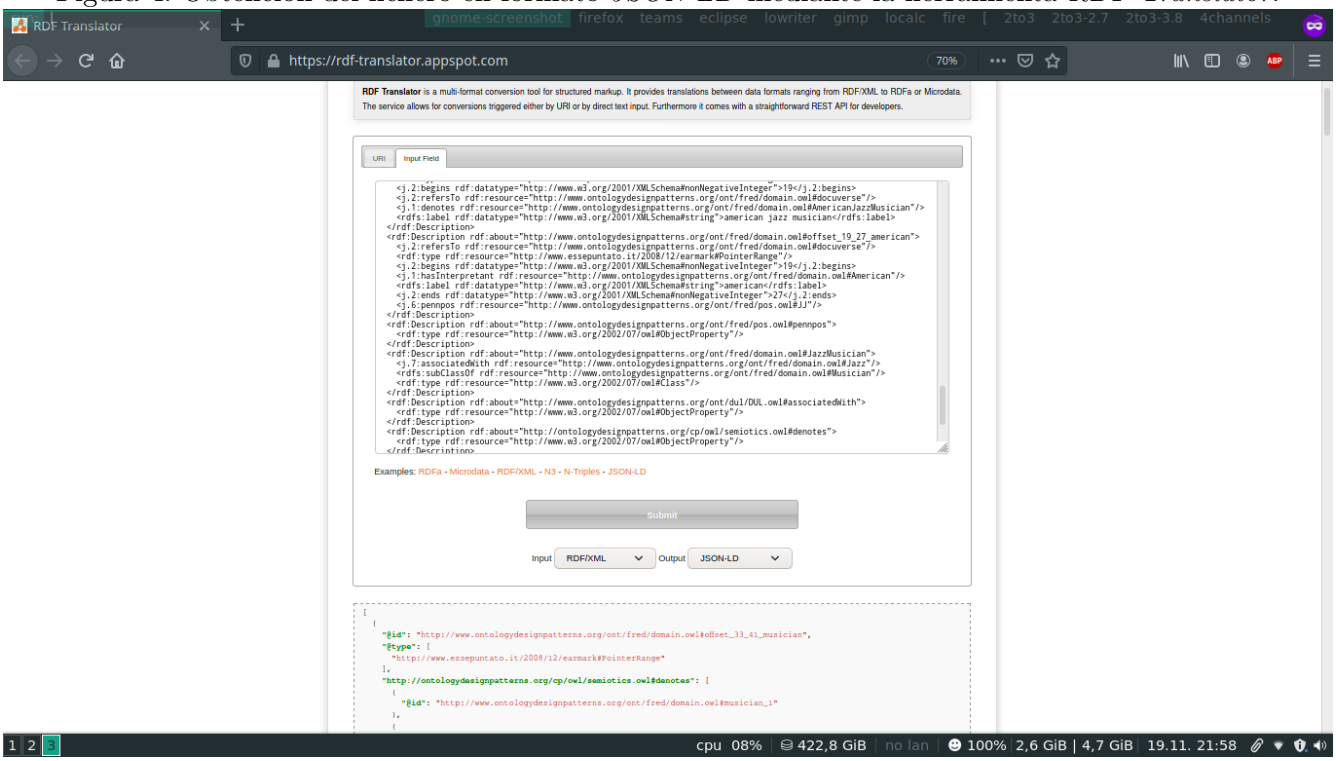
Figura 3: Obtención del fichero en formato *RDF-XML* mediante la herramienta *FRED*.



- **Programming Languages**: propia de *Itelligent Tagging*

- **Newspaper** en *schema.org*
  - **Newspaper**: propia de *DBpedia*
  - Subclase de **CreativeWork**: propia de *FRED*
  - **Newspaper Publishing**: propia de *Itelligent Tagging*
- **Occupation** en *schema.org*
  - **Occupation**: propia de *DBpedia*
  - **Position**: propia de *Itelligent Tagging*
- **City** en *schema.org*
  - **City**: propia de *DBpedia*
  - **City**: propia de *Itelligent Tagging*
- **Organization** en *schema.org*
  - **Organisation**: propia de *DBpedia*

Figura 4: Obtención del fichero en formato *JSON-LD* mediante la herramienta *RDF Translator*.



- **Organization**: propia de *Itelligent Tagging*

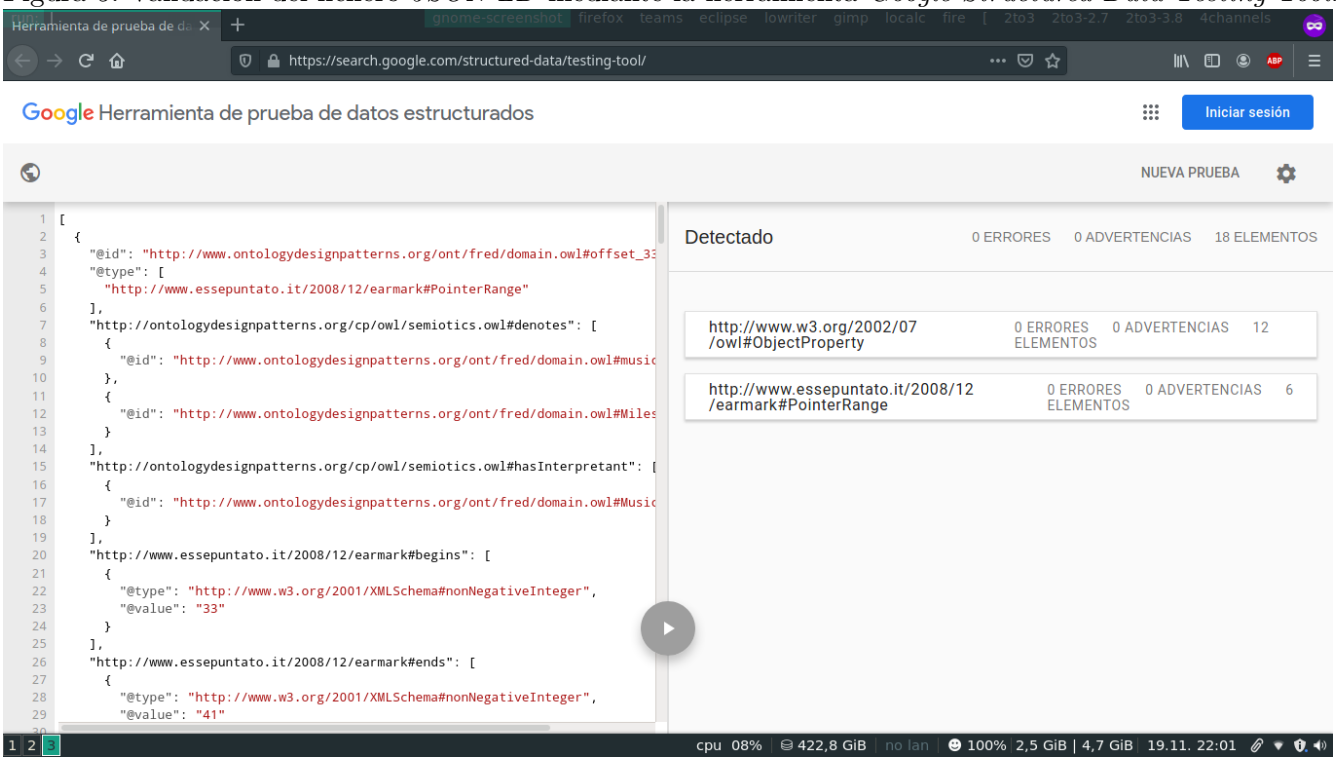
4.3. Reflexión sobre los motivos por los cuales un equipo de desarrollo crea su propia ontología

En el campo de la Web Semántica la necesidad de crear una ontología estándar es algo tremendamente importante. Esto está relacionado con su objetivo principal, el hacer de la Web un enorme grafo de conocimiento entendible por máquinas. Por ello, cuantos mas datos estén marcados con una misma ontología, mas sencillo sera relacionarlos. Sin embargo, debido a la naturaleza heterogénea de la Web, la creación de esta ontología estándar no es tarea sencilla, puesto que hay tal nivel de especificación de algunos datos que pretender cubrir la Web entera con marcados muy detallados se convierte en algo muy complejo.

Es por esto que muchos equipos deciden crear su propia ontología, para poder especializar más el marcado de los datos y no quedarse en la generalidad de esa ontología universal, dotándolos de mayor riqueza. No obstante, no todo son ventajas, al haber creado una ontología nueva se podrán marcar ciertos datos con mayor riqueza, pero esto hace que no pueda alinearse correctamente con otras ontologías, y por lo tanto es probable que acabe en desuso o usada por equipos muy pequeños.

En resumen, aunque la creación de una ontología estándar se enfrente a grandes retos, es necesaria para

Figura 5: Validación del fichero *JSON-LD* mediante la herramienta *Google Structured Data Testing Tool*.



garantizar su propia longevidad y esto a su vez atraerá a desarrolladores a utilizarla o a alinearla con sus propias ontologías, de forma que se construya de manera colaborativa.

Referencias

[1] Página de *Schema.org*, `https://schema.org/`. Última vez accedido 19 de noviembre de 2020.

[2] Página del *RDF Translator*, `https://rdf-translator.appspot.com/`. Última vez accedido 19 de noviembre de 2020.

[3] Página de la *Google Structured Data Testing Tool*, `https://search.google.com/structured-data/testing-tool/`. Última vez accedido 19 de noviembre de 2020.

[4] Página de la herramienta *Intelligent Tagging* de *Refinitiv*, `https://permid.org/onecalaisViewer`. Última vez accedido 19 de noviembre de 2020.

[5] Página de la herramienta *DBpedia Spotlight*, `https://www.dbpedia-spotlight.org/demo/`. Última vez accedido 19 de noviembre de 2020.

[6] Página de la herramienta *FRED*, `http://wit.istc.cnr.it/stlab-tools/fred/demo/`. Última vez accedido 19 de noviembre de 2020.