

Adaptación de la base de datos de Jamendo a una estructura RDF usando la librería RDFLib de Python

Cerna Espiritu, Roberto Alexis
Universidad Nacional de Ingeniería
Lima, Perú
roberto.cerna.e@uni.pe

Cruz Coro, Cristhian Elian
Universidad Nacional de Ingeniería
Lima, Perú
cecruz@uni.pe

Resumen—En este informe se da una breve introducción a la web semántica, al modelo conceptual de RDF y como manipular una estructura RDF ya existente, relacionada a la industria de la música, para poder realizar inferencias a partir de ella.

Índice de Términos— RDF, web semántica, music ontology.

I. INTRODUCCIÓN

El propósito de la web semántica es simple, facilitar el intercambio de datos, brindando estándares para la publicación de estos. World Wide Web Consortium es el encargado de este tema, aunque este proceso resulte mucho más laborioso y costoso, facilita enormemente el intercambio de datos en aplicaciones; además de brindar mejor organización de ellos.

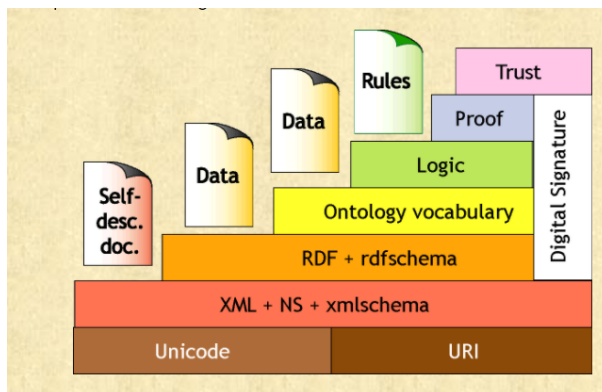


Figura 1. Web semántica

Una definición interesante de la web semántica planteada por Tim Berners-Lee es la siguiente *"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."*

Como lo dice Tim Berners-Lee de esta manera personas que no se conocen, ni han planteado un estándar para el intercambio de datos; podrían hacerlo sin problemas, poniendo un ejemplo práctico, tenemos dos

aplicaciones: la primera es una biografía sobre la vida de diferentes artistas con diferente información como lugar de nacimiento, fecha de nacimiento, lugar de residencia y la segunda es una plataforma en la que solo podemos escuchar música de diferentes artistas. En caso la plataforma de música desee incorporar una opción para poder ver información sobre la vida del artista del cuál estás escuchando su música, le sería muy útil utilizar los datos de la primera aplicación. Si este ha seguido tecnologías planteadas por la web semántica sería reconocida por los desarrolladores de la plataforma de música y podrían realizar la mejora con la data que se obtiene sin mucha dificultad.

Existen varias tecnologías que se utilizan como XML, UNICODE, URI pero en este trabajo realizaremos un enfoque en RDF.

Por otra parte, se partió de una estructura de música ya existente en este caso se utilizó Jamendo el cual contiene bastante música bajo la licencia de Creative Commons además de artistas, conciertos, discos, etc.

II. ESTRUCTURA RDF

La estructura inicial sobre la cuál se trabajó se obtuvo de la plataforma Jamendo [6], que es repositorio gigante de música, está provee distintas URL con data en formato rdf para su utilización. Esta estructura fue almacenada en el archivo all-artists.rdf.

Con la ayuda de un script en Python, se extrajeron todas las estructuras relacionadas al archivo inicial, almacenándolas en el archivo music.rdf; en el cual se encuentran instancias de las estructuras MusicArtist, Record, Track y Tag.

Una representación gráfica de estas estructuras, sus atributos y como se relacionan entre sí se muestran en la siguiente figura.

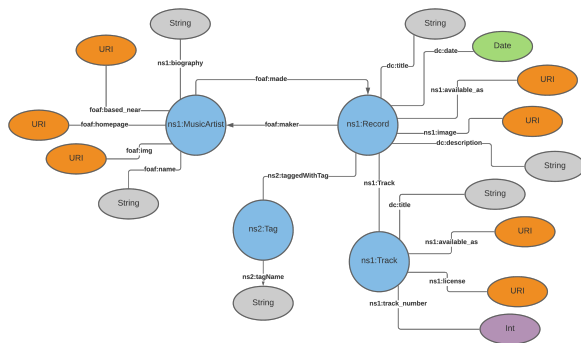


Figura 2. Representación gráfica de las estructuras RDF

Utilizando la herramienta de W3C RDF Validation Service se obtuvo los grafos que representan a las instancias de las estructuras RDF con las que trabajamos. Dichos imágenes se encuentran agregadas en el repositorio del proyecto, en el directorio Images.

III. DESCRIPCIÓN DE RECURSOS USANDO RDF

Como la estructura RDF a manipular está relacionada con la música, se debe seguir un API que brinda un modelo a seguir para la publicación de esta data por la página The music Ontology [2].

III-A. Clases

III-A1. Music Artist: Un Music Artist representa a algún artista, banda o computadora que produce música

- name: Describe el nombre del Artista
- made: Describe un track que el artista ha realizado

```
<rdf:Description rdf:about="http://dbtune.org/jamendo/artist/15">
  <rdf:type rdf:resource="http://purl.org/ontology/mo/MusicArtist"/>
  <foaf:made rdf:resource="http://dbtune.org/jamendo/record/32"/>
  <foaf:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Various Artists</foaf:name>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <foaf:made rdf:resource="http://dbtune.org/jamendo/record/35"/>
  <foaf:made rdf:resource="http://dbtune.org/jamendo/record/203"/>
</rdf:Description>
```

Figura 3. Music Artist

III-A2. Record: Un record es una manifestación de música, para fines prácticos lo podemos interpretar como un album de música o un disco, como se puede ver consta de varios predicados, para este informe solo utilizaremos los siguientes:

- date: Describe la fecha de lanzamiento del record
- title: Es el nombre que recibe dicho record
- track: Una canción(track) que se encuentra en dicho record
- maker: El MusicArtist que realizo dicho record
- taggedWithTag: Representan tags que se asocian a dicho record

```
<rdf:Description rdf:about="http://dbtune.org/jamendo/record/7571">
  ns1:available as rdf:resource="http://www.jamendo.com/get/album/id/album/p2p/redirect/7571/p2pnet-ed2k&are=ogg2"/>
  ns1:available as rdf:resource="http://www.jamendo.com/get/album/id/album/p2p/redirect/7571/p2pnet-bittorrent&are=ogg2"/>
  ns1:taggedWithTag rdf:resource="http://dbtune.org/jamendo/tag/rock"/>
  ns1:image rdf:resource="http://img.jamendo.com/albums/7571/covers/1.300.jpg"/>
  <dc:date>2007-09-19 12:04:10</dc:date>
  ns1:track rdf:resource="http://dbtune.org/jamendo/track/82296"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Secret's Of Nonlain</dc:title>
  ns1:taggedWithTag rdf:resource="http://dbtune.org/jamendo/tag/gothic"/>
  ns1:image rdf:resource="http://img.jamendo.com/albums/7571/covers/1.100.jpg"/>
  ns1:track rdf:resource="http://dbtune.org/jamendo/track/82301"/>
  ns1:available as rdf:resource="http://www.jamendo.com/get/album/id/album/p2p/redirect/7571/p2pnet-bittorrent&are=mp32"/>
  ns1:taggedWithTag rdf:resource="http://dbtune.org/jamendo/tag/metal"/>
  ns1:available as rdf:resource="http://www.jamendo.com/get/track/id/album/audio/xspf/7571/item-o-track-no-asc&are=ogg2&n=all"/>
  ns1:track rdf:resource="http://dbtune.org/jamendo/track/82297"/>
  ns1:image rdf:resource="http://img.jamendo.com/albums/7571/covers/1.500.jpg"/>
  ns1:available as rdf:resource="http://www.jamendo.com/get/album/p2p/redirect/7571/p2pnet-ed2k&are=mp32"/>
  ns1:track rdf:resource="http://dbtune.org/jamendo/track/82294"/>
  ns1:track rdf:resource="http://dbtune.org/jamendo/track/82295"/>
  ns1:track rdf:resource="http://dbtune.org/jamendo/track/82300"/>
  ns1:taggedWithTag rdf:resource="http://dbtune.org/jamendo/tag/melodique"/>
  ns1:available as rdf:resource="http://www.jamendo.com/get/track/id/album/audio/play/7571/item-o-track-no-asc&are=ogg2&n=all"/>
  <rdf:type rdf:resource="http://purl.org/ontology/mo/Record"/>
  ns1:track rdf:resource="http://dbtune.org/jamendo/track/82297"/>
  <foaf:maker rdf:resource="http://dbtune.org/jamendo/artist/7907"/>
  ns1:image rdf:resource="http://img.jamendo.com/albums/7571/covers/1.0.jpg"/>
</rdf:Description>
```

Figura 4. Record

III-A3. Tag: Un tag corresponde a una etiqueta, se puede asociar un género o algún tipo de característica en particular

- tagName: Describe la característica

```
<rdf:Description rdf:about="http://dbtune.org/jamendo/tag/charme">
  ns2:tagName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">charme</ns2:tagName>
  <rdf:type rdf:resource="http://www.holygoat.co.uk/owl/redwood/0.1/tags/Tag"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdf:Description>
```

Figura 5. Tag

III-A4. Track: Un track corresponde a una canción, esta es un elemento de algún record.

- license: La licencia bajo la que fue producida
- title: El nombre de el track
- track_number: Indica el número de ese track en el record al que pertenece

```
<rdf:Description rdf:about="http://dbtune.org/jamendo/track/75936">
  ns1:available as rdf:resource="http://www.jamendo.com/get/track/id/track/audio/xspf/75936"/>
  ns1:license rdf:resource="http://creativecommons.org/licenses/by-nc-sa/3.0"/>
  <rdf:type rdf:resource="http://purl.org/ontology/mo/Track"/>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">05 - DementialCore - Nichilismo comercial</dc:title>
  ns1:track_number rdf:datatype="http://www.w3.org/2001/XMLSchema#int">5</ns1:track_number>
  ns1:available as rdf:resource="http://www.jamendo.com/get/track/id/track/audio/play/75936"/>
</rdf:Description>
```

Figura 6. Track

III-B. Reglas del Esquema RDF

- ns1 : MusicArtist foaf : made ns1 : Record
- ns1 : Record foaf : maker ns1 : MusicArtist

- *ns1 : Record ns1 : track ns1 : Record*
- *ns1 : Record ns2 : taggedWithTag ns2 : Tag*

IV. MANIPULACIÓN DE RDF

La experimentación se ejecutó en un cuaderno de Colab, usando la librería *RDFLib* de Python para la manipulación de RDF. El experimento consistió en realizar inferencias sobre la información presentada en la estructura RDF de acuerdo a las reglas mencionadas en la sección anterior.

IV-A. Primera Inferencia

Premisas:

- *< MusicArtist >< made >< Record >*
- *< Record >< track >< Track >*

Resultado:

< MusicArtist >< made >< Track >

IV-B. Segunda Inferencia

Premisas:

- *< MusicArtist >< made >< Record >*
- *< Record >< taggedWithTag >< Tag >*

Resultado:

< MusicArtist >< madeWorkWithTag >< Tag >

IV-C. Tercera Inferencia

Premisas:

- *< MusicArtist >< made >< Record >*
- *< Record >< track >< Track >*
- *< Record >< license > (URI)*

Resultado:

< MusicArtist >< madeWorkUnderLicense > (URI)

IV-D. Cuarta Inferencia

Premisas:

- *< MusicArtist >< made >< Record >*
- *< Record >< track >< Track >*
- *< track >< int >< track_number >*

Resultado:

< MusicArtist >< int >< track_number >

V. CONCLUSIONES

- Con los conceptos de RDF y web semántica pudimos obtener datos y relaciones entre ellos sin necesidad de una previa coordinación con el que la brinda (Jamendo).
- Se realizaron inferencias en la estructura con las cuales se aprovecharon las relaciones que existen entre estos datos.
- La estructura RDF en si es un grafo, escrito no es tan entendible para las personas pero si para la computadora y gráficandolo se obtiene una mejor idea de como está estructurado
- A pesar de ser un poco trabajoso presentar la data en este formato y que otras personas lo entiendan, facilita y agiliza la extracción de una gran cantidad de datos, además *RDFLIB* permite hacerlo en pocas líneas de código.

REFERENCIAS

- [1] *rdflib 5.0.0 — rdflib 5.0.0 documentation.* (s. f.). *Rdflib 5.0.0 Documentation Website.* <https://rdflib.readthedocs.io/en/stable/>
- [2] *The Music Ontology – Specification.* The Music Ontology website. <http://musicontology.com/specification/index.html>
- [3] Madrid, U. C. (2001). *La Web Semántica. Hacia la Web semántica.* http://www.hipertexto.info/documentos/web_semantica.htm
- [4] *RDF Schema 1.1.* (2014, 25 febrero). *RDF Schema 1.1.* <https://www.w3.org/TR/rdf-schema/>
- [5] DuCharme, B. (2013). *Learning Sparql: Querying and Updating with Sparql 1.1* (2nd ed.). O'Reilly Media. <https://www.oreilly.com/data/free/learning-sparql-2ed-sample-marklogic.pdf>
- [6] *DBTune - Jamendo RDF dump.* (s. f.). *DBTune.Org.* <http://dbtune.org/jamendo/>