

# Metodología para el Desarrollo de Aplicaciones Web Utilizando Datos Enlazados

Taniana Rodríguez<sup>1</sup>, Ricardo Dos Santos<sup>1</sup>, Jose Aguilar<sup>1, \*</sup>

<sup>1</sup>Centro de Estudios en Microelectrónica y Sistemas Distribuidos (CEMISID)  
Universidad de Los Andes, Mérida, Venezuela  
{taniana, aguilar}@ula.ve, ricardojds@gmail.com

\*Prometeo researcher, Escuela Politécnica Nacional, Quito, Ecuador  
Universidad Técnica Particular de Loja, Ecuador

**Resumen**—El objetivo de este trabajo es proponer una Metodología para el desarrollo de Aplicaciones Web utilizando datos enlazados (Linked Data), con el fin de publicar los datos de la organización de una forma clara, precisa y más amigable. La metodología se basa en el uso de las nuevas tendencias tecnológicas de Datos Abiertos Enlazados (Linked Open Data), Para probar la Metodología, se presenta el caso de estudio de la producción intelectual de estudios de cuarto nivel de la Universidad de Los Andes (ULA), en específico, el programa del Postgrado en Computación y el Doctorado de Ciencias Aplicadas de la Facultad de Ingeniería de la Universidad de Los Andes de Mérida-Venezuela.

**Palabras Claves**—Producción Intelectual; Linked Open Data; Ontologías; Web Semántica;

## I. INTRODUCCIÓN

La Web ha venido evolucionando drásticamente estos últimos años, comenzando con una Web de Hipertexto que hizo posible enlazar y conectar los documentos (sitios web) compuestos por texto, imágenes, videos, etc. Inicialmente, la Web era interpretada por los humanos y no por las máquinas, pero con el surgimiento de la Web 2.0 se logró compartir los documentos y permitir la colaboración, aunque igualmente la interpretación seguía siendo realizada por los humanos. En la Web 3 o Web Semántica [3], se busca enlazar los metadatos semánticos y ontológicos que describen el contenido de los sitios web, a través del paradigma de “Datos Enlazados”, que permite identificar, describir, conectar y relacionar los elementos de datos de la Web.

La ventaja de utilizar el paradigma de Datos Abiertos Enlazados en la Web Semántica es poder conectar los datos de la web, independientemente de dónde residan. La vinculación se hace utilizando un identificador global llamado Identificador Uniforme del Recurso (URI, por sus siglas en inglés *Uniform Resource Identifier*). Ahora bien, en la actualidad no se cuenta con una metodología que permita desarrollar aplicaciones Web basada en datos enlazados.

Este trabajo tiene como objetivo el desarrollo de una Metodología para el desarrollo de Aplicaciones Web utilizando Datos Enlazados. Esto permite que los datos de la organización se puedan descubrir, reutilizar, por parte de otras aplicaciones. Además, la metodología se prueba en el caso de estudio de la

producción intelectual de cuarto nivel de la Universidad de Los Andes (ULA).

En particular, nosotros proponemos un conjunto de etapas a seguir para especificar una aplicación Web basada en Datos Enlazados, y proponemos una Metodología, llamado Metodología para el Desarrollo de Aplicaciones Web utilizando Datos Enlazados (MEDAWEDE), que facilitad el desarrollo de dichas aplicaciones.

## II. MARCO TEÓRICO

### A. Datos Enlazados

Los datos enlazados o vinculados (en inglés *Linked Data*) describen una forma de publicar los datos estructurados para que se puedan interconectar entre ellos. Para ello, se basa en tecnologías Web, tales como RDF<sup>1</sup> (Resource Description Framework) y los URIs, para compartir información que pueda ser leída automáticamente por los computadores, para conectar y consultar los datos desde diferentes fuentes.

Los Datos Enlazados se han aplicado con éxito en proyectos como Dbpedia<sup>2</sup>, que es un proyecto de la Universidad de Leipzig y la Universidad Libre de Berlín, con la finalidad de transformar de forma automática, los contenidos de Wikipedia en datos estructurados RDF; Europeana [8], que es una biblioteca digital semántica que constituye una agregación de los fondos editoriales de numerosas instituciones culturales de todos los países miembro de la Unión Europea, a través de una única interfaz, para tener acceso a millones de objetos digitales bibliográficos, archivísticos y de arte, que conforma el acervo cultural europeo; EDM, que es el modelo de datos de la Unión Europea, que incluye ontologías como SKOS<sup>3</sup> (Simple Knowledge Organization System), DC<sup>4</sup> (Dublin Core) y FOAF<sup>5</sup> (Friend Of A Friend); y Datos.bne.es<sup>6</sup> [9], que es el resultado del proyecto de datos enlazados de la Biblioteca Nacional de España.

<sup>1</sup> <https://www.w3.org/RDF>

<sup>2</sup> <http://wiki.dbpedia.org>

<sup>3</sup> <https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html>

<sup>4</sup> <http://dublincore.org/documents/dc-rdf>

<sup>5</sup> <http://xmlns.com/foaf/spec>

<sup>6</sup> <http://datos.bne.es/inicio.html>

## B. Principios de los Datos Enlazados

Linked Data describe un conjunto de prácticas para publicar, compartir y conectar piezas de datos, información y conocimiento en la Web Semántica, usando identificadores URIs y RDF para describir los recursos [4]. Berners-Lee propuso las siguientes reglas para la publicación de Datos Enlazados [5, 11,12]<sup>7</sup>:

- *Identidad*: Utilización de URIs para identificar las cosas
- *Accesibilidad*: Usar URIs HTTP (Hypertext Transfer Protocol) para que las personas puedan buscar recursos
- *Estructura*: Utilizar estándares RDF para describir recursos y SPARQL<sup>8</sup> (Query Language for RDF) para realizar consultas.
- *Navegación*: Incluir enlaces a otras URIs para descubrir más cosas.

Por lo tanto, los Datos Enlazados se basan en la creación de recursos con información expresada directamente en RDF, relacionados entre sí para realizar consultas. Cada objeto dentro del recurso de los datos enlazados cuenta con un nombre único, su URI, que permite referenciarlo de forma unívoca.

La gran ventaja de los datos enlazados es que permite utilizar y combinar datos procedentes de diferentes fuentes (otros recursos de datos enlazados), y a partir de su integración, extrae nuevo conocimiento.

## III. METODOLOGÍA DE DESARROLLO DE APLICACIÓN WEB UTILIZANDO DATOS ENLAZADOS (MEDAWEDE)

### A. Fases de la Metodología

Los datos enlazados implican un proceso complejo con diferentes etapas no excluyentes, por lo tanto, el ciclo de vida de los datos bajo tecnologías dato enlazado no es un proceso estático sino dinámico [11]. Así, nosotros proponemos las siguientes etapas en nuestra metodología:

- *Especificación*: se centra en el análisis de fuentes de datos. En esta fase se selecciona el conjunto de datos, algunos datos es probable que se extraigan utilizando técnicas de Big Data. Además, se deben diseñar los URIs a utilizar en los dataset o metadatos, así como la definición de las licencias de los diferentes dataset o metadatos a usar.
- *Modelado*: se centra en la creación del modelo que describe el conocimiento del área de estudio. Para ello se reutilizan ontologías, se utilizan vocabularios estándares, e incluso, se diseñan ontologías propias al área de estudio, utilizando algunas de las metodologías de desarrollo de ontologías [1,2,14]. Para la implementación del modelo se utiliza un editor de ontología, por ejemplo protégé<sup>9</sup>
- *Generación*: se centra en la transformación de los datos al lenguaje de RDF. En la actualidad, existen muchas herramientas para la transformación de datos a RDF. Por

ejemplo, openRefine<sup>10</sup> para datos en formatos CVS o Excel, y el D2RQ<sup>11</sup> para base de datos relacionales.

- *Vinculación*: en esta etapa se vinculan los datos con otros conjuntos de datos o datasets para aumentar su valor, visibilidad y calidad. Con estos datos vinculados, se puede mostrar nuevo conocimiento que se encuentra oculto en los datasets. Una de las posibles herramientas utilizadas para la vinculación es Silk<sup>12</sup>, también se puede vincular de forma manual, a través de los siguientes vocabularios: owl:sameAs, rdfs:seeAlso, rdfs:subClassOf, foaf:knows, rdfs:subPropertyOf.
- *Publicación*: del conjunto de datos en un triplestore o almacenamiento RDF, también llamado almacen de tripletas. Para ello, se usan herramientas tales como Virtuoso<sup>13</sup>, Pubby<sup>14</sup>, etc.
- *Explotación de la información*: Esta etapa permite el manejo e integración de distintas interfaces para consumir los recursos publicados de manera agradable y sencilla. Para esta fase se propone el uso de una arquitectura orientada a servicios [15].

### B. Tareas Principales de la Metodología para Desarrollar Aplicaciones Web

Las etapas de nuestra metodología conllevan a la realización de dos grandes tareas, que facilitan el desarrollo de las aplicaciones Web. En la Figura 1 se muestran esas dos tareas en la Metodología propuesta:

- La primera es modelar los datos de la organización, utilizando los principios de Datos Enlazados
- La segunda es generar la aplicación Web en base a los datos generados en la primera tarea.

Las siguientes subsecciones describen las dos tareas.

### C. Modelar los Datos de la Organización

El modelado de los datos de la organización se realiza en dos fases, en la primera se modela la base de conocimiento de la organización, es decir, se realiza el modelo ontológico de la organización, y luego, en la segunda fase se transforman los datos de la organización con los principios de datos enlazados.

Las fases detalladas que se siguen en nuestra metodología en esta tarea se indican a continuación:

- *Modelo ontológico de la organización*: en el modelado se analiza la organización en base a la información que la organización posee, por ejemplo: sus páginas Web, sus sistemas computacionales, los documentos que describen a la organización, entre otros. Una vez obtenida toda la información, se genera el esquema conceptual ontológico de la organización de manera semiautomático, según donde ella se encuentre:

<sup>10</sup> <http://openrefine.org>

<sup>11</sup> <http://d2rq.org>

<sup>12</sup> <http://silkframework.org>

<sup>13</sup> <https://virtuoso.openlinksw.com>

<sup>14</sup> <http://wifo5-03.informatik.uni-mannheim.de/pubby>

<sup>7</sup> <https://www.w3.org/DesignIssues/LinkedData.html>

<sup>8</sup> <https://www.w3.org/TR/rdf-sparql-query>

<sup>9</sup> <http://protege.stanford.edu>

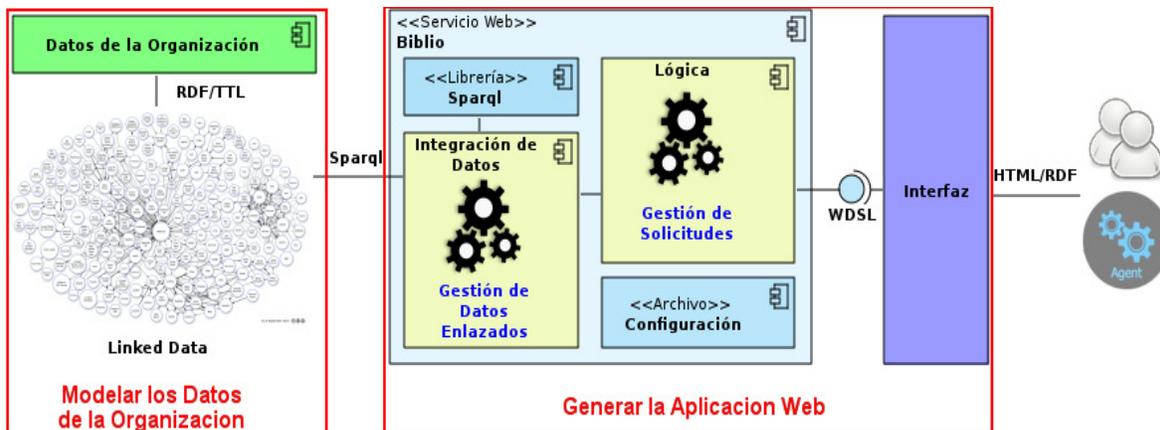


Figura 1. Metodología para el Desarrollo de Aplicación Web utilizando Datos Enlazados (MEDAWEDE)

- *Caso 1. La información está en páginas web o documentos:* en este caso se usa un sistema de extracción de entidades y relaciones [10], o técnicas de minería semántica [13, 14], para la generación del esquema conceptual.
- *Caso 2. La información se encuentra en bases de datos:* el esquema conceptual se extrae del esquema de la bases de datos.
- *Caso 3. La información se encuentra en páginas web, documentos y bases de datos:* en este caso, si se realiza el caso 1 y 2 por separado, luego se analizan los esquemas generados y se fusionan en el esquema conceptual

2. **Seleccionar los Datos:** los expertos toman el archivo generado del proceso anterior y analizan los datos. En este proceso se extrae la información que se requiere, es decir, cuáles de los datos se van a publicar con acceso abierto.
3. **Generar RDF y enlazado:** Este proceso tiene como entrada el archivo generado en el proceso anterior, y los diferentes vocabularios. Esos datos se transforman a RDF explotando esos vocabularios, dando como resultado el archivo rdf. Durante esta tarea se usan archivos OWL<sup>15</sup> (Web Ontology language), RDF, y los vocabularios, para realizar la vinculación de los datos con otros datos.

Una vez generado el esquema conceptual ontológico, se valida el esquema con los expertos del dominio.

Seguidamente, se pasa a explorar los vocabularios ya existentes para enriquecer al esquema del modelo ontológico de la organización. Por ejemplo, se pueden usar los vocabularios que describen a las personas (FOAF, VCARD, Schema#Person, etc), a la organización (schema.org#Organisation), en el Modelo Ontológico de la Organización.

- *Transformar los datos de la organización en los principios de datos enlazados:* en esta fase se transforma los datos en RDF usando el modelo ontológico generado por la tarea anterior. El objetivo de esta fase es la generación de RDF desde las fuentes de información asociadas a la organización, para transformar los datos originales a un formato estándar e interoperable en el contexto de la Web. En la Figura 2 se muestra el flujo de trabajo para la transformación.

Esta transformación se realiza en tres pasos, que se describen a continuación

1. **Convertir los datos en Formato CSV (Comma-Separated-Values):** Este proceso tiene como entrada las fuentes de datos sin procesar, en distintos formatos. Usualmente, la mayoría de los datos está en base de datos relacionales,

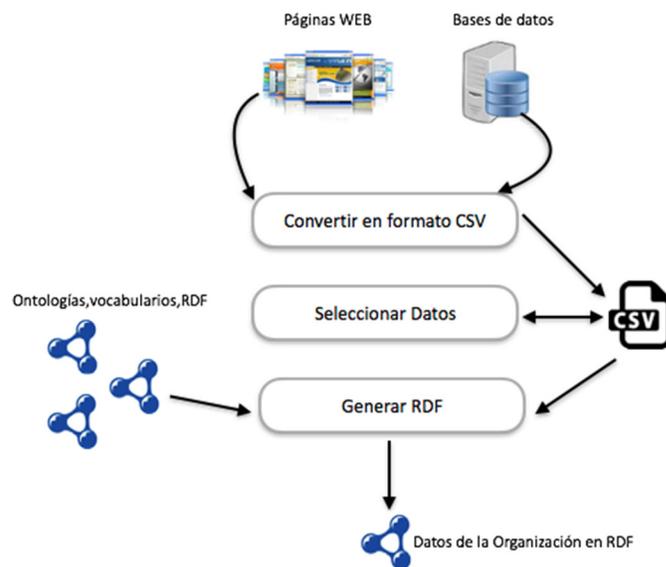


Figura 2. Flujo de Trabajo para la Transformación de los Datos de un Organización en RDF

<sup>15</sup> <https://www.w3.org/OWL>

#### D. Generar la Aplicación Web

La generación de la aplicación web se desarrolla en dos fases, en la primera fase se crea el servicio web que ofrecerá una serie de métodos o funciones para la gestión de solicitudes y de datos enlazados, que permitirá consumir los datos enlazados de la organización. En la segunda fase se crea la interfaz visual, que servirá como acceso web a los datos, a los usuarios finales y a los agentes maquinas [15]. La fases detalladas de esta tarea son:

- **Crear el servicio web:** En esta fase se procede a seleccionar el lenguaje de programación y las librerías para la implementación del servicio web. Luego se generan las operaciones o métodos para la Gestión de Solicitudes que ofrecerá el servicio web, que permitirán consumir el conocimiento publicado por la organización. Por último, se crean las operaciones privadas para la Gestión de los Datos Enlazados. Allí se definen las consultas a los repositorios de los recursos publicados por la organización, a través de la tecnología de Linked Data, y para las consultas se usa el lenguaje Sparql.
- **Crear la interfaz:** Lo primero que se define en esta fase es el protocolo que se va a usar para consumir el servicio web creado en la fase anterior. Luego, conociendo los métodos disponibles en el servicio web, se implementa la navegabilidad de la interfaz, para obtener las características deseadas.

La ventaja de separar la lógica de negocio en una fase con tecnología de servicio web, radica en que la lógica de negocio pasa a ser un módulo, lo que permite ser invocado por múltiples interfaces y distintas aplicaciones, ya sea de ventana, móviles o web.

#### IV. CASO DE USO DE MEDAWEDE: PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE LOS ANDES

En esta sección describimos el uso de nuestra metodología, en un caso de estudio. El caso de estudio usa toda la información y datos de los programas del Postgrado en Computación y del Doctorado de Ciencias Aplicadas de la Facultad de Ingeniería de La Universidad de Los Andes.

##### A. Modelar los Datos de la Organización

El modelado se realiza en dos fases.

###### Fase 1: Modelo ontológico de la organización

En la Figura 3 se muestra la red de ontologías, que es una colección de ontologías entrelazadas, que incluye el uso de vocabularios existentes, que describe el modelo ontológico del caso de estudio.

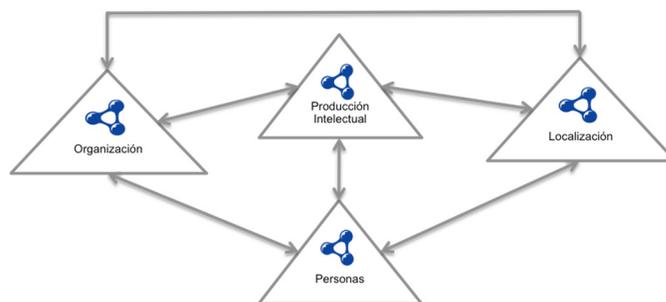


Figura 3. Red de Ontologías Utilizada en el Caso de Estudio

A continuación se describe brevemente la red de ontología del caso de estudio: la *producción intelectual* es producida por *personas*, en este caso las personas son autores y tutores, los autores son los que generan la producción intelectual y los tutores orienta a los autores para la generación de la producción intelectual. Además, la producción intelectual es publicada por una *organización*, y por ultimo; la organización, la producción intelectual y las personas están ubicados en una *localización*.

Para el modelado de los datos de la organización se utilizan vocabularios, tales como los vocabularios dublin core, foaf, Bibo, que contienen varias propiedades y conceptos que requiere nuestro modelo, tales como:

- **Bibo:** Representa los conceptos y propiedades que permiten describir las citas y referencia bibliográficas, y puede ser usado para describir la producción intelectual de una organización
- **DublinCore:** Dublin Core Metadata Initiative (DCMI) fue desarrollado para representar recursos y atributos que se utilizan en el ámbito bibliográfico.
- **FOAF-Vcard:** fue desarrollado para representar/describir atributos y relaciones entre entidades.

A partir del modelo conceptual de la Figura 3, se diseñó la ontología de la organización. En la Figura 4 se muestra parte de la base de conocimiento de la producción intelectual, la cual describe que una tesis es creada por un autor (CodigoTesis dc:creator CodigoAutor), el CodigoAutor es de tipo Autor (CodigoAutor rdf:type Autor), la tesis es tutoriado por un tutor (CodigoTesis dc:contributor CodigoTutor), además, el CodigoAutor tiene los siguientes atributos: Nombre, Primer Nombre, Apellidos etc., al igual que el CodigoTutor. Adicionalmente, indica que el CodigoAutor y Codigo Tutor son personas, y CodigoTesis es una TesisMaestria o es una TesisDoctorado, y la TesisMaestria y TesisDoctorados es una Tesis. En el modelado se utilizó el vocabulario Dublin Core que tiene el prefijo dc, y foaf que tiene el prefijo foaf.

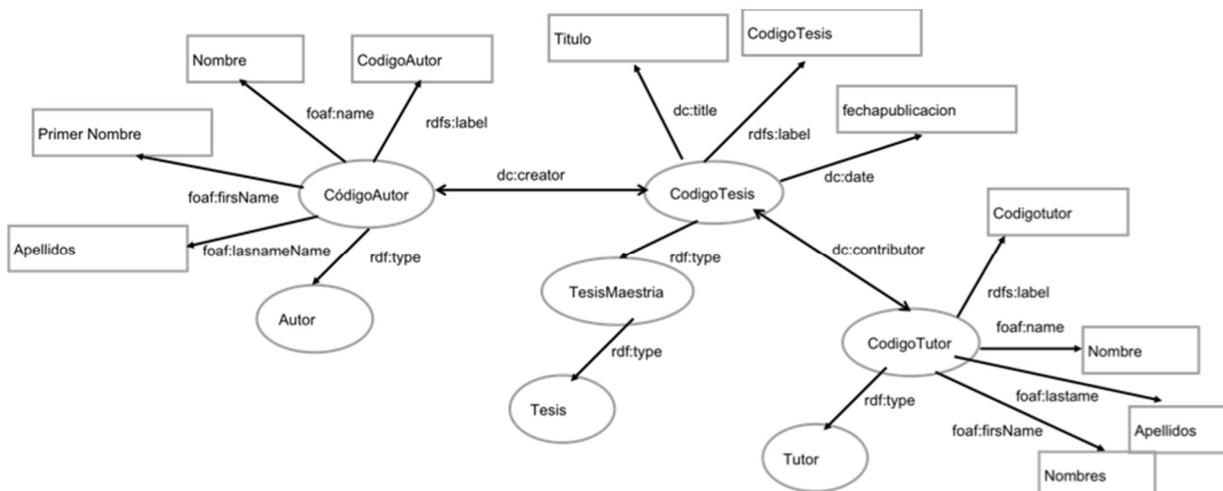


Figura 4. Parte de la Base de Conocimiento de la Producción Intelectual

Luego que se tiene la base de conocimiento, se realiza la segunda fase, que se describe a continuación.

### Fase 2: Transformación de los datos de la organización

En esta fase se transforma los datos a RDF, y se realiza el enlazado. En particular, se genera en RDF las fuentes de información asociadas a la producción intelectual, tal que queden transformados los datos originales a un formato estándar e interoperable en el contexto de la Web. Para el proceso de transformación se realizaron los pasos de la Figura 2:

1. Convertir a Formato CSV: Las fuentes de datos sin procesar de la producción intelectual son páginas Web y bases de datos, se procedió a convertir los datos al formato cvs.
2. Seleccionar los Datos: los expertos toman el archivo generado del proceso anterior y analizan los datos. En este proceso se extraen los datos que se van a publicar con acceso abierto.
3. Generación de RDF y enlazado: Una vez obtenido los datos, se transforma los datos a RDF con la herramienta OpenRefine (<http://openrefine.org>), usando los vocabularios respectivos, generando el archivo RDF. En la Figura 5 se muestra el esqueleto de RDF para los autores.

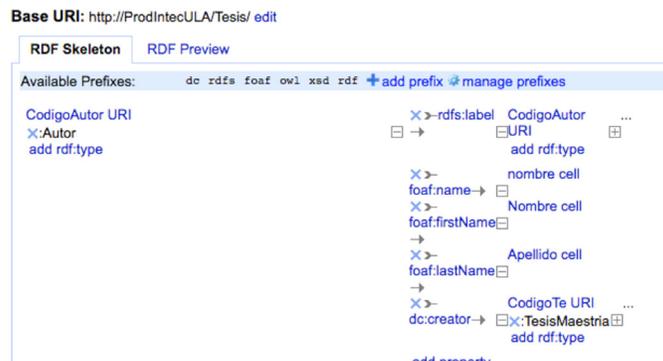


Figura 5. Esqueleto RDF para el Autor de Tesis

Todos los datos son transformados (tutores, las tesis, etc.) siguiendo el modelo conceptual de la Figura. 4, dando como resultado un archivo como el que se muestra en la Figura 6.

Una vez obtenidos los datos transformados en RDF, se almacenan los datos en un repositorio de datos RDF, En nuestro caso, utilizamos OpenLink Virtuoso y Fuseki. Ya terminado esta etapa, se pasa a la fase de explotación de la información, en la que por ejemplo, se puede generar una aplicación web.

```

@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://ProdIntecULA/Tesis/Autor_1> a <http://ProdIntecULA/Tesis/Autor> ;
  rdfs:label <http://ProdIntecULA/Tesis/Autor_1> ;
  foaf:name "Juan Luis Chaves Sanabria" ;
  foaf:firstName "Juan Luis" ;
  foaf:lastName "Chaves Sanabria" .

<http://ProdIntecULA/Tesis/Tesis_1> a <http://ProdIntecULA/Tesis/TesisMaestria> .

<http://ProdIntecULA/Tesis/Autor_1> dc:creator <http://ProdIntecULA/Tesis/Tesis_1> .

<http://ProdIntecULA/Tesis/Autor_2> a <http://ProdIntecULA/Tesis/Autor> ;
  rdfs:label <http://ProdIntecULA/Tesis/Autor_2> ;
  foaf:name "Hungria Zulema Berbesi Quintero" ;
  foaf:firstName "Hungria Zulema" ;
  foaf:lastName "Berbesi Quintero" .

<http://ProdIntecULA/Tesis/Tesis_2> a <http://ProdIntecULA/Tesis/TesisMaestria> .

<http://ProdIntecULA/Tesis/Autor_2> dc:creator <http://ProdIntecULA/Tesis/Tesis_2> .

<http://ProdIntecULA/Tesis/Autor_3> a <http://ProdIntecULA/Tesis/Autor> ;
  rdfs:label <http://ProdIntecULA/Tesis/Autor_3> ;
  foaf:name "Rosita del Socorro Bruscano López" ;
  foaf:firstName "Rosita del Socorro" ;
  foaf:lastName "Bruscino López" .

<http://ProdIntecULA/Tesis/Tesis_3> a <http://ProdIntecULA/Tesis/TesisMaestria> .

<http://ProdIntecULA/Tesis/Autor_3> dc:creator <http://ProdIntecULA/Tesis/Tesis_3> .

<http://ProdIntecULA/Tesis/Autor_4> a <http://ProdIntecULA/Tesis/Autor> ;
  rdfs:label <http://ProdIntecULA/Tesis/Autor_4> ;
  foaf:name "Emiro Antonio Coronado Cabrera" ;
  foaf:firstName "Emiro Antonio" ;
  foaf:lastName "Coronado Cabrera" .
    
```

Figura 6. Datos Transformados de los Autores de las Tesis del Postgrado en Computación

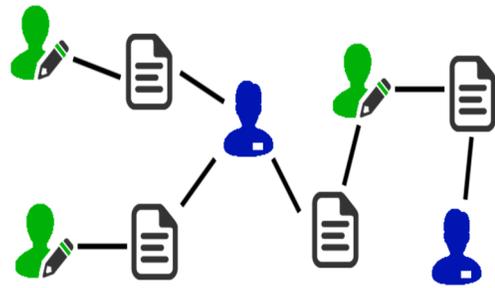


Figura 8. Representación de los Datos Enlazados de la Organización

Para este caso de estudio, se usó Python<sup>16</sup> como lenguaje de programación para implementar el Servicio Web. Las librerías usadas son las siguientes: `spyne`<sup>17</sup> para el Servicio Web y `sparqlwrapper`<sup>18</sup> para el uso de Sparql. En la Figura 9 se observa parte de los protocolos configurados, los puntos de montaje, y la creación del servicio web.

```

yml = Application([service], tns=tns,
  in_protocol=HttpRpc(),
  out_protocol=YamlDocument())

rest = Application([service], tns=tns,
  in_protocol=HttpRpc(), out_protocol=HttpRpc())

xml = Application([service], tns=tns,
  in_protocol=HttpRpc(), out_protocol=XmlDocument())

html = Application([service], tns=tns,
  in_protocol=HttpRpc(), out_protocol=HtmlMicroFormat())

soap = Application([service], tns=tns,
  in_protocol=Soap11(validator='lxml', pretty_print=True),
  out_protocol=Soap11())

root = WsgiMounter({
  'rest': rest,
  'xml': xml,
  'html': html,
  'soap': soap,
  'json': json,
  'jsonl': jsonl,
  'jsonil': jsonil,
  'yaml': yml,
})

from wsgiref.simple_server import make_server
server = make_server(host, port, root)
Debug()
server.serve_forever()
    
```

Figura 9. Servicio Web con spyne

Luego se definen los métodos o funciones de la Gestión de Solicitudes, que servirán como interfaz para la solicitud de información. Entre los métodos definidos están los siguientes: `listar_autores`, `listar_articulos`, `listar_tesis`, `listar_tutores`, `ver_información` y `buscar`. En la Figura 10 se observa un ejemplo de la definición de los métodos.

```

class ULA_Biblio(ServiceBase):

    @srpc(_returns=Iterable(String))
    def listar_autores():
        return buscar_autores()
    
```

Figura 10. Definición de los Métodos o Funciones del Servicio Web

## B. Generación de la Aplicación Web

### Fase 1: Crear Servicio Web

Esta fase se apoya en el paradigma SOA (ver Figura 7). En este caso en particular, se necesita que la aplicación tenga métodos que permitan mostrar el listado de tesis, artículos, autores y tutores, como también, que permitan ver la información relacionada a cada uno de ellos.

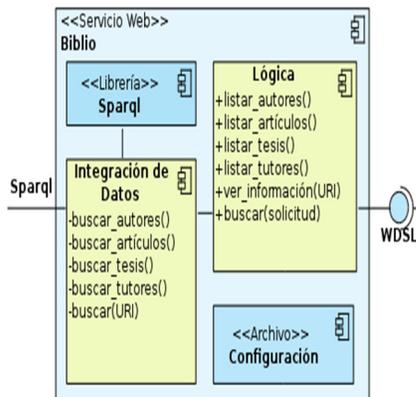


Figura 7. Servicio Web

Un aspecto a considerar para el diseño de la aplicación, es su capacidad para consumir los recursos publicados de la organización bajo la tecnología de datos enlazados. En la Figura 8 se muestra una representación gráfica de los datos de la organización, donde se puede observar de color verde los autores, de azul los tutores, y en forma de documento los artículos y tesis. También se observan las relaciones entre ellos.

<sup>16</sup> <https://www.python.org>

<sup>17</sup> <http://spyne.io>

<sup>18</sup> <https://rdflib.github.io/sparqlwrapper>

Por otro lado, se diseña un módulo de integración de los datos, donde se definen los métodos privados que permiten buscar la información usando Linked Data. Entre estos métodos se tiene los siguientes: buscar\_autores, buscar\_artículos, buscar\_tesis, buscar\_tutores, buscar por la URI, y en cada uno de estos métodos se especifican las consultas en lenguaje sparql que permiten obtener la información de los datos enlazados, como se observa en la Figura 11 el método buscar\_autores.

```
from SPARQLWrapper import SPARQLWrapper, JSON
def buscar_autores():
    sparql = SPARQLWrapper("http://127.0.0.1:8099/sparql")
    sparql.setQuery("""
        SELECT * WHERE {
            ?uri a <http://ProdIntecULA/Tesis/Autor> .
            ?uri foaf:lastName ?lastname
        }
    """)
    sparql.setReturnFormat(JSON)
    results = sparql.query().convert()
    for result in results["results"]["bindings"]:
        yield { 'uri': result["uri"]["value"], 'nombre': result["lastName"]["value"] }
```

Figura 11. Consultas en Lenguaje sparql del Servicio Web

Finalmente, se observa en la Figura 12 dos resultados obtenidos usando el Servicio Web de este caso de estudio. El primer resultado en la parte superior, es una respuesta SOAP con la especificación del servicio (WSDL) en formato XML, dicha especificación es muy importante para la construcción de la interfaz que consuma este servicio, ya que se indica los métodos disponibles, con sus parámetros. El segundo resultado que se muestra en la parte inferior, es una respuesta REST en formato JSON, donde se muestran todos los autores devueltos por el método listar\_autores.

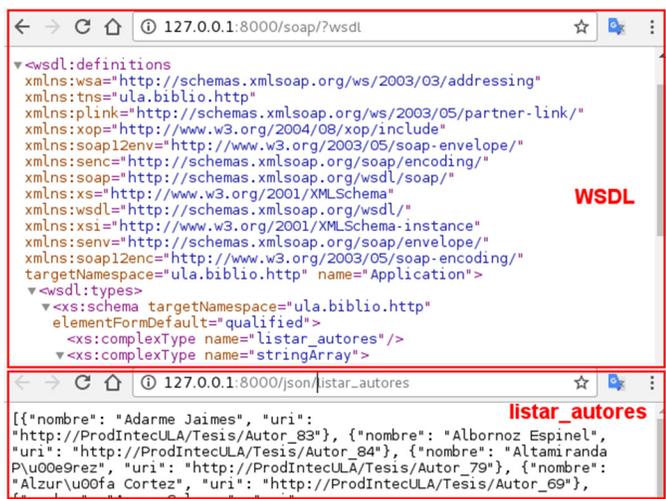


Figura 12. Respuesta de la Fase Servicio Web

## Fase 2: Crear la Interfaz Web

Aquí se implementa la o las interfaces visuales que sean necesarias para facilitar la visualización de la información de la organización, que esta publicada como datos enlazados.

Para implementar la interfaz, primero hay que definir cómo se va a consumir el servicio web. Las dos formas más comunes

son implementarla con PHP<sup>19</sup> (PHP: Hypertext Preprocessor), o con el lenguaje de programación Python.

En la Figura 13 se muestra el código en PHP que invoca/usa el método listar\_autores del servicio web, allí se implementa el uso del protocolo REST con respuesta en formato de intercambio Json.

```
<?php
$url="http://127.0.0.1:8000/json/listar_autores";
$contentido = file_get_contents($url);
$contentido = utf8_encode($contentido);
$resultados = json_decode($contentido);

echo "<h2>Lista de Autores</h2>";
foreach ($resultados as $resultado) {
    echo "<b>URI</b>: ", $resultado->uri, "<br><b>Nombre:</b> ", $resultado->nombre, "<br><br>";
}
```

Figura 13. Implementación de la Interfaz Web con PHP

La segunda implementación que se muestra en la Figura 14 hace uso del mismo método lista\_autores, pero en este caso usando python. Esta implementación es un poco más elaborada, usa el protocolo SOAP con un formato de intercambio XML, por lo que necesita implementar un cliente web que lea la descripción del servicio (WSDL) y se comunique a través del protocolo SOAP. Para este fin, se usa la librería suds<sup>20</sup>, y para ver el resultado como una interfaz web, se usa la librería Flask<sup>21</sup>, que es un microFramework para Python que despliega un servidor web que permite mostrar la interfaz desarrollada en este lenguaje.

```
import suds
from flask import Flask
app = Flask(__name__)
@app.route("/")
def listar_autores():
    url = "http://localhost:8000/soap/?wsdl"
    cliente = suds.client.Client(url)
    resultados = cliente.service.listar_autores()
    salida = "<h2>Lista de Autores</h2>"
    for resultado in resultados[0]:
        salida += "<b>URI</b>: " + resultado[1] + "<br><b>Nombre:</b> " + resultado[0] + "<br><br>"
    return salida
if __name__ == "__main__":
    app.run()
```

Figura 14. Implementación de la Interfaz Web con Python

Como se observa en la Figura 15, ambas implementaciones generan los mismos resultados al mostrar la lista de autores, aunque se estén usando distintos protocolos de intercambio de mensajes, lenguajes de programación y/o librerías.

Finalmente, en la Figura 16 se muestra la interfaz de la aplicación usando los distintos métodos requeridos en la fase del desarrollo del Servicio Web, que permite la navegabilidad usando los datos enlazados. Allí se observa un menú desplegable con los distintos tipos de listas, acompañado de un buscador de información relacionada a una URI de la organización, y en el cuerpo principal se muestra la lista de autores a través del método listar\_autores, donde cada autor posee un enlace a su respectiva información. En la Figura 17 se muestra la informa de una URI a través del método ver\_información, la cual retorna todas las relaciones que existe

<sup>19</sup> https://secure.php.net

<sup>20</sup> https://pypi.python.org/pypi/suds

<sup>21</sup> http://flask.pocoo.org

con dicha URI. En este caso, se muestra la información de una tesis, y allí se observa el título de la tesis, el tutor, la fecha, entre otras cosa más. Hay que recalcar que cada información que es mostrada esta enlazada a otros datos, característica fundamental en los datos enlazados.



Figura 15. Resultados de las Distintas Implementaciones del Consumo del Servicio Web

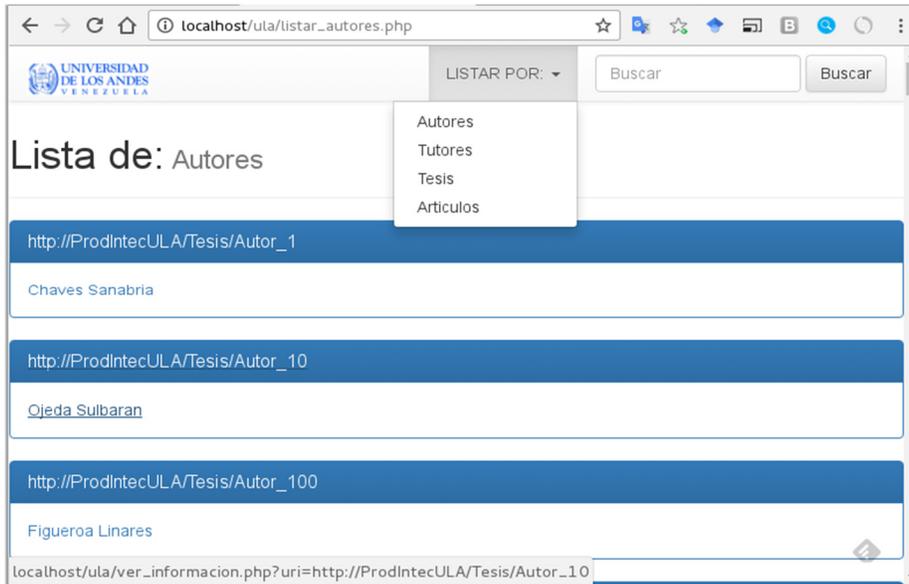


Figura 16. Interfaz Mostrando la Lista de Autores

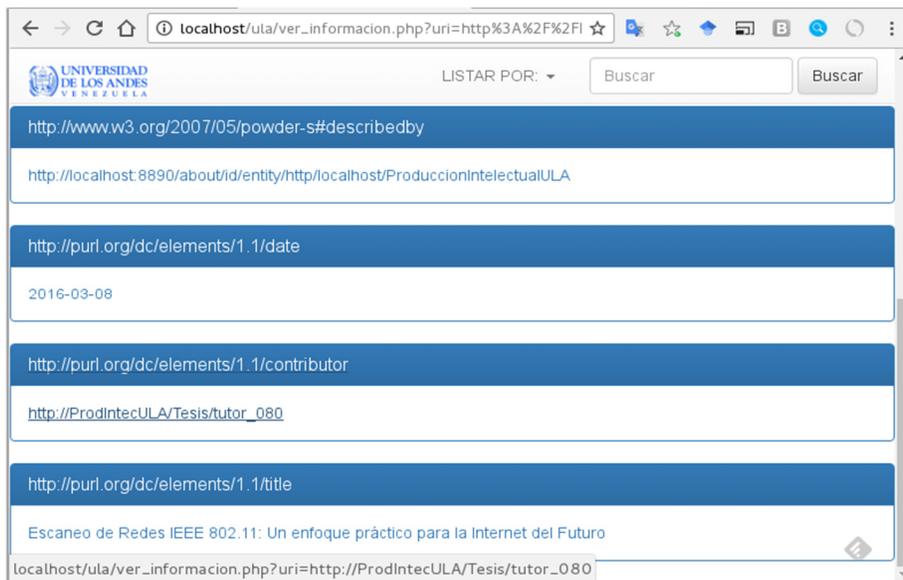


Figura 17. Interfaz Mostrando la Información de una Tesis

## V. CONCLUSIONES

Este trabajo presentó una metodología para explotar los datos enlazados de una organización. Además, en particular, define una Metodología para el desarrollo de Aplicaciones Web usando datos enlazados. Este Metodología tiene dos grandes tareas, la primera es el modelado de los datos de la organización, y la segunda es la generación de la aplicación web usando los datos enlazados, aprovechando la fortaleza de los servicios web a nivel de interoperabilidad e independencia de plataformas.

La metodología se probó en un caso de estudio, vinculado a la producción intelectual de una universidad, usando los principios de datos enlazados y el estándar RDF, para así lograr que la producción intelectual que se produce en la universidad sea más accesible y visible en la Web.

Entre los trabajos futuros se tiene la incorporación de toda la producción intelectual generada en la Universidad de Los Andes. También, se aspira utilizar la metodología en proyectos más complejos.

## AGRADECIMIENTO

Dr. Jose Aguilar ha sido parcialmente financiado por el Proyecto Prometeo del Ministerio de Educación Superior, Ciencia, Tecnología e Innovación de la República de Ecuador).

## REFERENCIAS

- [1] M.C. Suárez Figueroa. "NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse." PhD Thesis. Universidad Politécnica de Madrid, 2010.
- [2] A. Gómez-Pérez, M. Fernández-López, O. Corcho. "Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web." Springer. 2004.

- [3] T. Berners-Lee, J. Hendler, O. Lassila. "The Semantic Web," Scientific American, Vol. 284, No. 5, pp. 28-37, 2001.
- [4] O. Hartin, A. Langegger. "A Database Perspective on Consuming Linked Data on the Web." <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.2688&rep=rep1&type=pdf>, 2010.
- [5] R. Míguez Pérez, J. M. Santos Gago, V. M. Alonso Rorís, L. M. Álvarez Sabucedo y F. A. Mikic Fonte. "Linked Data como Herramienta en el Ámbito de la Nutrición". *Nutrición Hospitalaria.*, Vol. 27. No. 2, pp. 323-332, 2012.
- [6] C. Hernández. "La Aplicación de Linked Data en la Agrupación Datos Bibliográficos" MSc. Tesis. Universidad de Salamanca. Salamanca, 2012.
- [7] A. Rotem-Gal-Oz. "SOA Patterns", Manning. 2012.
- [8] M. Doerr, S. Gradmann, S. Henricke, A. Isaac, C. Mehini y H. Van de Sompel. "El Modelo de Datos de Europea (EDM). World Library and Information Congress: 76th IFLA General Conference and Assembly, Gotemburgo, Suecia, 2010.
- [9] D. Vila-Suero, B. Villazón-Terrazas y A. Gómez-Pérez. "datos.bne.es: a Library Linked Data Dataset". *Semantic Web.* IOS Press. 00. 1-6. 2012.
- [10] T. Rodríguez, J. Aguilar, "Aprendizaje Ontológico para el Marco Ontológico Dinámico Semántico", *DYNA*, Vol. 81, No. 187, pp. 56-63, 2014.
- [11] D. Wood, M. Zaidman and L. Ruth with M. Hausenblas. "Linked Data Structured Data on the Web", Manning Publications Co. 2014.
- [12] T. Heath and C. Bizer. "Linked Data: Evolving the Web into a Global Data Space". *Synthesis Lectures on the Semantic Web: Theory and Technology.* Morgan & Claupool Publishers series. 2011.
- [13] C. Rangel, J. Altamiranda, J. Aguilar; "Semantic Mining based on Graph Theory and Ontologies. Case Study: Cell Signaling Pathways", *CLEI Electronic Journal*, Vol. 19, No. 2, pp. 7-7, 2016.
- [14] J. Aguilar, O. Portilla, "Framework Basado en ODA para la Descripción y Composición de Servicios Web Semánticos (FODAS-WS)", *Latin American Journal of Computing*, Vol. 2, No. 2, pp. 15-24, 2015.
- [15] V. Bravo, J. Aguilar, F. Rivas, M. Cerrada, "Diseño de un Medio de Gestión de Servicios para Sistemas Multiagentes", *XX Conferencia Latinoamericana de Informática*, pp. 431-439, 2004.