Plataforma para la visualización de datos multidimensionales basados en tecnologías semánticas

Andrea Daniela Morales Rodríguez[1,2], Víctor Hugo Saquicela Galarza[1]

[1] Universidad de Cuenca, Av. 12 de abril y Agustín Cueva, Cuenca, Ecuador 010203.

[2] Red Nacional de Investigación y Académica del Ecuador, Calle la Condamine 12 - 109 y Calle Larga, Cuenca, Ecuador, 010101.

Autores para correspondencia: {primero, segundo}@ucuenca.edu.ec, primero@cedia.edu.ec

Fecha de recepción: 26 de mayo 2017 - Fecha de aceptación: 22 de agosto 2017

RESUMEN

La plataforma "Repositorio Ecuatoriano de Investigadores", permite identificar las áreas de conocimiento mediante la agrupación de palabras claves definidas en las publicaciones científicas que realizan los investigadores ecuatorianos a través de la utilización de técnicas de *datamining* y tecnología semántica. Actualmente, la información de los autores es almacenada en un repositorio en formato *Resource Description Framework* (RDF). La visualización de la información en la actualidad se realiza mediante consultas estáticas desarrolladas por los programadores del proyecto, por lo que no es posible realizar consultas dinámicas por parte de los usuarios. Con esta problemática, el presente trabajo define una arquitectura de software para mejorar las herramientas actuales de visualización, basada en el concepto de cubos de datos multidimensionales utilizando el vocabulario *RDF Data Cube*, permitiendo implementar búsquedas dinámicas.

Palabras clave: Tecnología semántica, RDF, RDF data cube vocabulary, QB, Opencube toolkit, modelos multidimensionales.

ABSTRACT

The "Researcher's Ecuadorian Repository" platform allows the identification of knowledge's areas through the grouping of key words defined in the scientific publications carried out by Ecuadorian researchers using data mining techniques and semantic technology. Currently, the information of the authors is stored in a repository form called "Resource Description Framework" (RDF). The visualization of the information at present is made by static requests developed by the programmers of the project, reason why it does not allow to make dynamic requests by the users. Therefore, the present work defines a software architecture to improve the current visualization tools based on the concept of multidimensional data cubes described using the RDF Data Cube vocabulary, allowing to implement dynamic searches.

Keywords: Semantic technology, RDF, RDF data cube vocabulary, QB, Opencube toolkit, multidimensional models.

1. INTRODUCCIÓN

Actualmente la plataforma "Red Ecuatoriana de Investigadores (REDI[1])", permite identificar las áreas de conocimiento mediante la agrupación de palabras clave definidas en las publicaciones científicas que realizan los investigadores ecuatorianos a través de la utilización de técnicas de *datamining* y tecnología semántica. La información de los autores se almacena en un repositorio en formato *Resource Description*

[1] http://redi.cedia.org.ec

Framework (RDF) basado en tripletas (sujeto, predicado, objeto) que permiten la manipulación de grandes cantidades de datos y la interoperabilidad entre aplicaciones, a través del intercambio de información.

La visualización de esta información se realiza mediante consultas estáticas o predefinidas por los programadores del proyecto, es decir, si se desea incluir una nueva consulta ésta deberá ser implementada por los desarrolladores con conocimientos de tecnología semántica, requiriendo actualizaciones frecuentes del sistema. Con esta problemática, el presente estudio tiene como propósito principal mejorar las herramientas actuales de visualización de la información almacenada en el repositorio REDI, continuando el trabajo con la implementación de tecnología semántica, RDF y la utilización de cubos de datos multidimensionales basados en RDF para implementar búsquedas dinámicas.

Para lograr este cometido, se analizó la arquitectura de la plataforma REDI para entender su funcionamiento y determinar que mejoras se pueden implementar sobre ella, por lo tanto, en este trabajo se definió una arquitectura de software basado en tecnología semántica, que permite mejorar las herramientas de visualización para lograr que estas sean dinámicas. Adicionalmente, se desarrolló un proceso de transformación de la información, que actualmente se encuentra almacenada en el repositorio REDI en formato RDF, a formato RDF Data Cube. Esta información será almacenada en un *Data Warehouse Semántico* (DWS) para que, mediante consultas, se realice la visualización de la información disponible de forma dinámica.

Este artículo se encuentra estructurado en cinco secciones, la primera presenta una introducción al trabajo realizado. La segunda sección presenta los antecedentes y trabajos relacionados sobre la temática desarrollada. En la tercera se encuentra la definición de la arquitectura de software planteada para la implementación del prototipo. La cuarta sección presenta los resultados obtenidos del trabajo desarrollado y, finalmente, en la quinta sección se concluye con los descubrimientos y trabajos futuros a implementar.

2. ANTECEDENTES Y TRABAJOS RELACIONADOS

En la actualidad existe una gran cantidad de profesionales ecuatorianos que han realizado sus estudios de postgrados a nivel nacional e internacional con el afán de mejorar sus aptitudes. Una parte de estos profesionales son investigadores/docentes de las Instituciones de Educación Superior del Ecuador (IES), que regresan con el interés de realizar investigación en el país. Para esto, es necesario participar en proyectos de I+D+i con financiamiento provenientes de diferentes fuentes, sean estos públicos o privados, lo que permite trabajar de manera colaborativa con investigadores internos o externos a la institución que pertenece el investigador.

En muchos de los casos los investigadores desconocen con quien trabajar dentro de una misma área de conocimiento, esto se complica aún más si se trata de diferentes áreas puesto que no se dispone de mecanismos que permitan detectar de manera automática las áreas de conocimiento en las que se encuentren trabajando.

Para tratar de resolver este problema, la Red Nacional de Investigación y Educación del Ecuador – RedCEDIA, con el apoyo de la Universidad de Cuenca, desarrollaron una plataforma basada en tecnología semántica, a la que llamaron Red Ecuatoriana de Investigadores (REDI), que permite identificar las áreas de conocimiento mediante la agrupación de palabras claves, a través de la utilización de técnicas de minería de datos (*datamining*), y visualizando información sobre las publicaciones realizadas por autores y co-autores ecuatorianos. Actualmente, toda la información está almacenada en formato RDF y posee un acceso a través de un SPARQL Endpoint[2].

[2] https://www.w3.org/wiki/SparqlEndpoints

2.1. Fortalezas y limitaciones de la plataforma REDI

Del análisis realizado a la plataforma REDI se han identificado algunas fortalezas y limitaciones con respecto a la manipulación y visualización de información:

Fortalezas:

- localiza la fuente de las publicaciones de los investigadores que han sido integrados a los respectivos repositorios digitales de las IES;
- permite detectar áreas similares de conocimiento de los investigadores ecuatorianos; y
- permite agrupar las publicaciones por palabras claves para identificar el universo de investigadores que trabajan sobre cierta área de conocimiento.

Limitaciones:

- las búsquedas son predefinidas por lo que no es posible incluir una nueva búsqueda de manera inmediata, la misma debe ser desarrollada para su implementación;
- para la generación de una nueva búsqueda es necesario que el desarrollador tenga conocimientos sobre la tecnología utilizada; y
- no es posible realizar búsquedas de manera dinámica puesto que no se dispone actualmente de herramientas necesarias para realizarlo.

Con la plataforma REDI se ha tratado de cubrir en parte el problema general encontrado, pero las búsquedas que se realizan actualmente son preestablecidas, si bien cumplen su función de visualizar la información deseada, actualmente no permiten que el usuario realice búsquedas dinámicas sobre el repositorio. Esta es una de las grandes limitaciones de la plataforma REDI, que se pretende superar incorporando nuevas herramientas de búsquedas utilizando modelos multidimensionales basados en tecnología semántica.

2.2. Trabajos relacionados

En esta sección se realiza un análisis de la literatura sobre: arquitectura de software, proceso de transformación de diferentes fuentes de datos a RDF Data Cube y la tecnología para la visualización de los datos transformados.

Arquitectura de software

Kämpgen & Harth (2011), Vdovjak & Houben (2001), Aggoume *et al.* (2016) y Ghasemi (2014) proponen una arquitectura basada en capas:

- en la capa 1, se define la fuente de datos para trabajar sobre ellos o para almacenarlos;
- en la capa 2, Kämpgen & Harth (2011) indican que se debe poblar el modelo multidimensional, y Vdovjak & Houben (2001) y Ghasemi (2014) realizan el proceso de transformación;
- para la capa 3, cada autor ha definido sus necesidades; Kämpgen & Harth (2011) serializa los datos obtenidos de la capa anterior en las tablas de hechos y dimensiones, Vdovjak & Houben (2001) definen las reglas necesarias para inferir los resultados obtenidos de la capa anterior, y según Aggoume *et al.* (2016) debería existir una capa denominada "*wrapper*" que realizará la consulta inicial sobre los datos, y para la capa final los autores indican que se realizarán las consultas sobre los datos para realizar la visualización; adicionalmente, Ghasemi (2014) obtiene un grafo de la transformación realizada en la capa anterior.

Transformación a RDF Data Cube

Para visualizar de manera dinámica la información que ofrece la plataforma REDI mediante la utilización de modelos multidimensionales basados en tecnología semántica, es necesario realizar un proceso de transformación de la información que actualmente se encuentra en un repositorio en formato RDF a formato QB, que permite la visualización de modelos multidimensionales basados en tecnología semántica.

De acuerdo a Helmich (2013), Rivera Salas *et al.* (2012), Martin *et al.* (2015) y *OpenCube* Project (2013) es posible realizar el proceso de transformación a QB desde diferentes fuentes de datos como CSV, XLS, PX, KML, OLAP, RDB, RDF, el mismo que está relacionado con la visualización de los datos transformados. Según Kämpgen & Harth (2011), Helmich (2013) y Nebot *et al.* (2009), es necesario definir el conjunto de datos, la estructura de datos, definición de dimensiones, medidas, instancia de datos. Otros datos de apoyo como "*label*" y "*comment*", son necesarios de acuerdo a OpenCube Project (2013).

La información que se encuentre en QB se almacenará en un DWS. Según Nebot *et al.* (2009) es necesario definir cuatro fases para determinar un modelo conceptual para la generación de un DWS. La primera fase consiste en establecer una ontología multidimensional integrada - MIO, la segunda consiste en identificar y extraer las ontologías necesarias, la tercera es un validador de los cubo de datos, y la cuarta es el análisis de la información que se encuentra en el DWS mediante la utilización de las ontologías seleccionadas. De acuerdo a Bellatreche *et al.* (2013), para diseñar un DWS, es necesario seguir cinco pasos: 1. Análisis de los datos; 2. Diseño conceptual; 3. Diseño lógico; 4. Proceso ETL; y 5. Desarrollo y diseño físico.

Visualización de datos transformados a QB

Una vez con los datos transformados y almacenados es necesario realizar la visualización, como se mencionó anteriormente, la transformación y visualización de datos a QB están muy relacionados. Según Rivera-Salas *et al.* (2012), Helmich (2013) y Martin *et al.* (2015), para realizar la visualización en herramientas como CubeViz, Payola, OpenCube Toolkit, es necesario definir el conjunto de datos, la estructura de los datos, dimensiones, medidas, instancias, etc., para continuar con las consultas requeridas.

3. ARQUITECTURA DE SOFTWARE PLANTEADA

De acuerdo al análisis del estado del arte y las necesidades acorde al desarrollo de este estudio, se ha establecido que la arquitectura de software a utilizar será la basada en capas y flujo de datos. La arquitectura de capas está compuesta por la capa datos, capa de aplicaciones, capa de presentación y la capa del cliente; la arquitectura de flujo de datos estará compuesta por cuatro componentes: extracción de datos SPARQL, proceso de transformación de RDF a QB, SPARQL para almacenamiento en SDW y extracción de datos QB (Figura 1).

3.1. Capa de datos

Actualmente, se dispone de un repositorio RDF, donde se almacenan los datos de los autores con sus publicaciones en un formato de tripletas (sujeto, predicado, objeto). Este repositorio se encuentra en la plataforma *Apache Marmotta*[3] y se actualiza de forma periódica. Adicionalmente, en el repositorio se encuentra el SDW donde se almacenan los datos transformados a QB, y estos son extraídos por el componente de la capa superior.

3.2. Capa de aplicaciones

La capa de aplicaciones se encuentra organizada dentro de una arquitectura de flujo de datos, la misma que se conforma por cuatro componentes, que son:

[3] http://marmotta.apache.org/

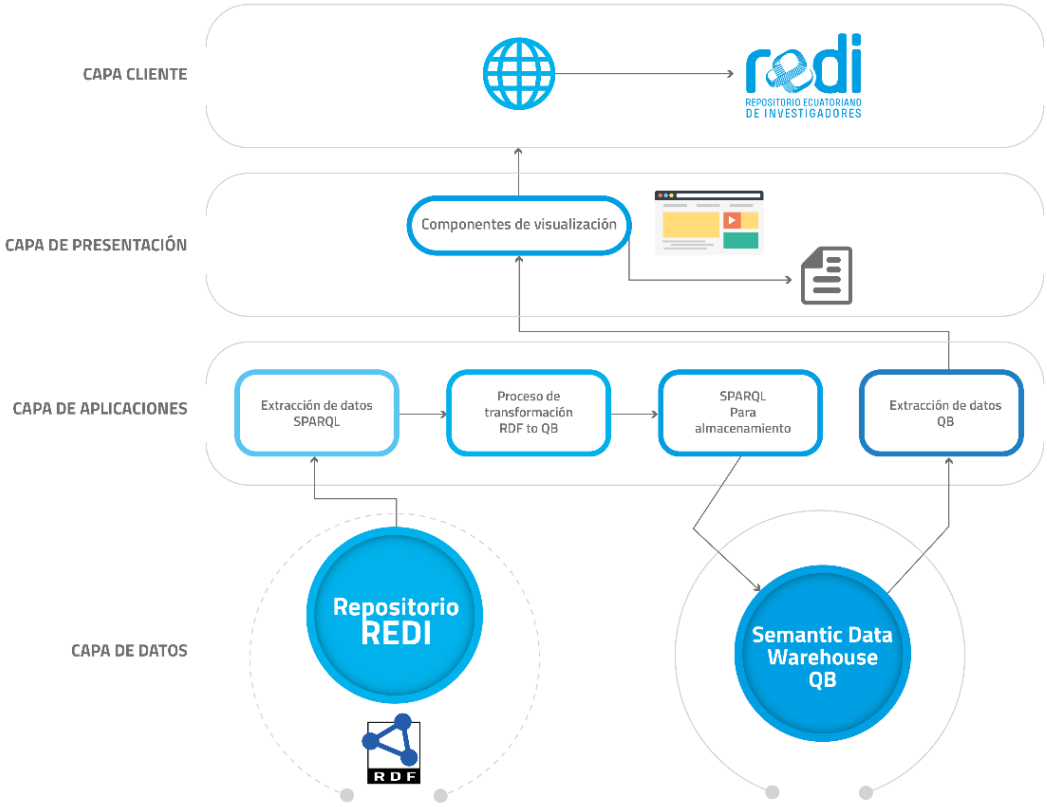

Figura 1. Nueva arquitectura propuesta.

Componente 1: Extracción de datos SPARQL

El primer filtro iniciará con el proceso de extracción de los datos que se encuentran almacenados en el repositorio RDF mediante consultas SPARQL. Para realizar este proceso, es necesario determinar qué información va a ser consultada, como se encuentra estructurada y su disponibilidad, para esto se consultó a los usuarios finales sobre los requerimientos de información que desean visualizar en el REDI, de este proceso se estableció las siguientes preguntas:

- ¿Cuál es el número de publicaciones realizadas por autor, en un período determinado?
- ¿Cuál es el número de publicaciones generadas por cada Institución de Educación Superior, por año?
- ¿Cuál es el número de publicaciones generadas por año?
- ¿Cuáles son las áreas de conocimiento en las que trabaja un autor, en un período determinado?
- ¿Cuáles es el número de publicaciones por año, por autor de la Universidad de Cuenca en un período determinado?

Con las preguntas identificadas, se procedió extraer la información del repositorio REDI mediante consultas SPARQL (Tabla 1), para confirmar su disponibilidad y responder las preguntas planteadas.

Tabla 1. Extracción de información del repositorio REDI.

```
PREFIX bibo: <http://purl.org/ontology/bibo/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?Nombre ?Apellido ?Institucion ?Publicaciones ?Titulo ?AnioPublicacion ?AreaConocimiento
WHERE {?algo a foaf:Person;
        foaf:firstName ?Nombre;
        foaf:lastName ?Apellido;
```

```
        dct:provenance ?provenance;
        dct:subject ?AreaConocimiento;
        foaf:publications ?Publicaciones.
        ?Publicaciones a bibo:Document;
                dct:title ?Titulo;
                dct:created ?AnioPublicacion.
        ?provenance <http://ucuenca.edu.ec/ontology#name> ?Institucion.
}ORDERBY ?Apellido ?Nombre
```

Basado en estas preguntas se definió un modelo multidimensional que refleja las dimensiones y medidas para este estudio: Dimensión: "Autor", se refiere a la persona que genera publicaciones; Dimensión: "Área de conocimiento", es el área de conocimiento en la que se encuentra trabajando el autor; Dimensión: "Año de publicación", se refiere al año en el que se generó la publicación; Dimensión: "Institución de Educación Superior", es la institución a la que pertenece el autor; y la medida: Número de publicaciones, se refiere al número de publicaciones que ha generado un autor, la misma que puede ser calculada por cada dimensión propuesta (Figura 2).

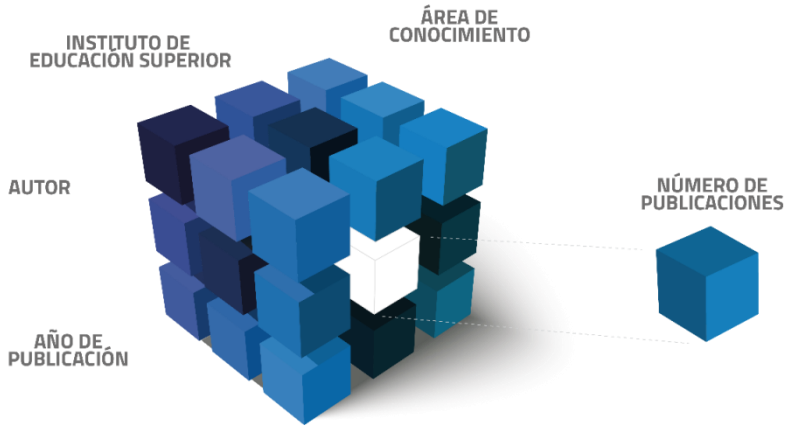

Figura 2. Modelo Multidimensional, dimensiones y medida.

Componente 2: Proceso de transformación de RDF a QB

Con la información recibida del filtro anterior, inicia el proceso de transformación de la información de RDF a QB, basados en el vocabulario QB simplificado (Figura 3). De acuerdo a la revisión de la literatura, es necesario identificar los pasos a seguir para realizar la transformación, (Figura 4), esto permitirá realizar un mapeo entre la fuente de REDI y el vocabulario QB.

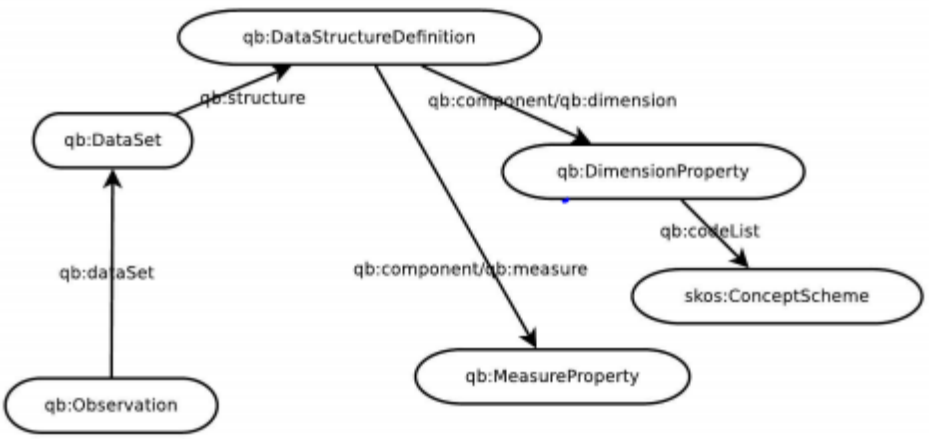

Figura 3. Vocabulario RDF Data Cube - QB, simplificado.

Paso 1: Definir el conjunto de datos (Tabla 2). Para definir que un recurso es un conjunto de datos se utiliza la clase *qb:DataSet*, al cual se asigna un nombre para identificarlo dentro de la estructura "*dataset-np1*"; con la propiedad *rdfs:label* se define un nombre único al conjunto de datos para identificarlos en el momento de la visualización; con la propiedad *rdfs:comment* se brinda más información sobre el conjunto de datos; con la propiedad *qb:structure* se indica que pertenece a esa estructura del conjunto de datos que, en este caso sería, "*pub:dsd-np*" nombre de la estructura de datos que será definida más adelante.

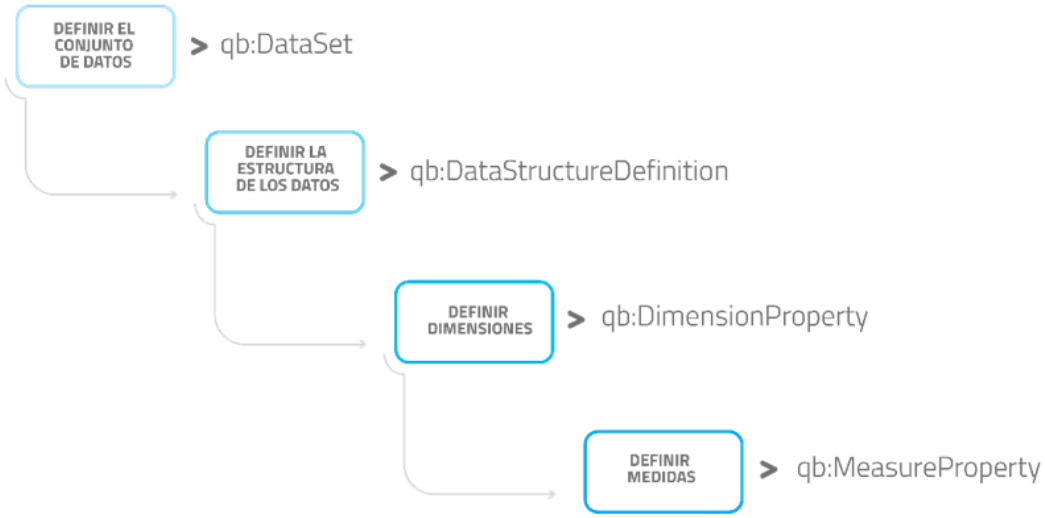

Figura 4. Proceso de transformación de los datos de RDF a QB.

Paso 2: Definir la estructura de los datos (Tabla 2). Para definir que un recurso es una estructura de datos se utiliza la clase *qb:DataStructureDefinition*, se asigna un nombre para que sea identificada dentro de la estructura del cubo "*dsd-np*". Las dimensiones, medidas y atributos son componentes, por lo que es necesario declararlos con la propiedad *qb:component*; con la propiedad *qb:dimension* se indica que es una dimensión y se asigna un nombre, adicionalmente, se declara un orden o nivel de importancia de cada dimensión creada con la propiedad *qb:order*. Con la propiedad *qb:measure* se indica que es una medida, y se asigna un nombre. Con la propiedad *qb:attribute* se indica que es un atributo y se declara la unidad de la medida. Con la propiedad *qb:componentAttachment* se indica que esta definición de estructura de datos pertenece a un *DataSet*.

Tabla 2. Generación de la estructura del cubo de datos multidimensionales QB.

```
1  prefix pub: <http://190.15.141.66:8899/ucuenca/>
2  prefix : <http://purl.org/dc/elements/1.1/>
3  prefix foaf: <http://xmlns.com/foaf/0.1/>
4  prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5  prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6  prefix dct: <http://purl.org/dc/terms/>
7  prefix qb: <http://purl.org/linked-data/cube#>
8  prefix skos: <http://www.w3.org/2004/02/skos/core#>
9  prefix xsd: <http://www.w3.org/2001/XMLSchema#>
10 prefix sdmxattribute: <http://purl.org/linked-data/sdmx/2009/attribute#>
11 prefix sdmxmeasure: <http://purl.org/linked-data/sdmx/2009/measure#>
12 prefix sdmxdimension: <http://purl.org/linked-data/sdmx/2009/dimension#>
13 prefix sdmxcode: <http://purl.org/linked-data/sdmx/2009/code#>
14 prefix sdmxconcept: <http://purl.org/linked-data/sdmx/2009/concept#>
15 insert data {
16  graph <http://lapproy01.cedia.org.ec:8080/marmotta/context/Redi1QB>
```

```
{
#Definición del Conjunto de Datos DataSet
pub:dataset-np1 a qb:DataSet;
 rdfs:label "Total de publicaciones1"@es;
 rdfs:comment "Conteo de publicaciones de los autores del REDI"@es;
 qb:structure pub:dsd-np ;
 sdmxattribute:unitMeasure <http://dbpedia.org/page/Number> .
#definición de la Estructura de Datos DataStructureDefinition
pub:dsd-np a qb:DataStructureDefinition;
# Dimensiones
qb:component [qb:dimension pub:refAutor; qb:order 1];
qb:component [qb:dimension pub:refAreaConocimiento; qb:order 2];
qb:component [qb:dimension pub:refFuente; qb:order 3];
qb:component [qb:dimension pub:refPeriod; qb:order 4];
#Medida
qb:component [qb:measure pub:numPublicaciones];
# Atributos
qb:component [qb:attribute sdmxattribute:unitMeasure; qb:componentAttachment
qb:DataSet;] .
# definición de Dimensiones
pub:refAutor a rdf:Property, qb:DimensionProperty;
 rdfs:label "Autor de Publicaciones"@es;
 rdfs:subPropertyOf sdmxdimension:refAutor;
 rdfs:range skos:Concept;
 qb:concept sdmxconcept:refAutor .
#
pub:refAreaConocimiento a rdf:Property, qb:DimensionProperty;
 rdfs:label "Area de conocimiento"@es;
 rdfs:subPropertyOf sdmxdimension:refAreaConocimiento;
 rdfs:range skos:Concept;
 qb:concept sdmxconcept:refAreaConocimiento .
#
pub:refFuente a rdf:Property, qb:DimensionProperty;
 rdfs:label "IES"@es;
 rdfs:subPropertyOf sdmxdimension:refFuente;
 rdfs:range skos:Concept ;
 qb:concept sdmxconcept:refFuente .
#
pub:refPeriod a rdf:Property, qb:DimensionProperty;
 rdfs:label "Anio de la publicacion"@es;
 rdfs:subPropertyOf sdmxdimension:refPeriod;
 rdfs:range skos:Concept;
 qb:concept sdmxconcept:refPeriod .
#definición de Medidas
pub:numPublicaciones a rdf:Property, qb:MeasureProperty;
rdfs:label "Total de publicaciones Autores"@es;
rdfs:subPropertyOf sdmxmeasure:obsValue;
rdfs:range xsd:integer .
#
pub:extraccion-de-preferencias-televisivas-desde-los-perfiles-de-redes-sociales
pub:numPublicaciones "1";
 qb:measureType pub:numPublicaciones ;
 pub:refAreaConocimiento "Performance Art" ;
 pub:refFuente "UCUENCA" ;
 pub:refPeriod "2014" ;
 pub:refAutor <http://ucuenca.edu.ec/resource/author/victor-saquicela>;
 qb:dataSet pub:dataset-np1 ;
 a qb:Observation.

pub:adding-semantic-annotations-into-geospatial-restful-services
pub:numPublicaciones "1";
```

```
75  qb:measureType pub:numPublicaciones;
76  pub:refAreaConocimiento "Tecnología Semántica";
77  pub:refFuente "UCUENCA";
78  pub:refPeriod "2012";
79  pub:refAutor <http://ucuenca.edu.ec/resource/author/victor-saquicela>;
80  qb:dataSet pub:dataset-np1;
81  a qb:Observation.
```

Paso 3: Definir dimensiones (Tabla 2). Las dimensiones identificadas previamente fueron renombradas para realizar su definición en el código, la dimensión Autor se renombró a "refAutor"; la dimensión: Área de conocimiento, se renombró a "refAreaConocimiento"; la dimensión: Año de publicación, se renombró a "refPeriodo", y la dimensión: Institución de Educación Superior, se renombró a "refFuente". Para declarar que un recurso es una dimensión se indica que es parte de la clase *rdf:Property* y *qb:DimensionProperty*, se indica el nombre de cada dimensión creada previamente en el "*dsd-np*" y se asigna a cada dimensión con un nombre para la visualización de los datos con la propiedad *rdfs:label*, se indica que es parte del vocabulario SDMX con las propiedades *sdmxdimension* y *sdmxconcept*.

Paso 4: Definir medida. De igual manera se renombró la medida Número de publicaciones a "numPublicaciones". Para declarar que es una medida, se indica que forma parte de la clase *rdf:Property* y *qb:MeasureProperty*, se asigna un nombre para que sea identificado en el momento de realizar la visualización de los datos, se indica que es un valor a ser observado con la propiedad *sdmxmeasure:obsValue*, e indica la unidad de la medida con *xsd:integer* (Tabla 2). Con la estructura del cubo lista se procede a generar las observaciones. Para identificar que es una observación se utiliza la clase *qb:Observation*. Para la visualización de la información en este documento, se ha recogido información del autor "Víctor Saquicela" de la Universidad de Cuenca, con dos publicaciones (Tabla 2).

Componente 3: SPARQL para almacenamiento

La información generada en el filtro anterior es la entrada para publicar por primera vez en la plataforma *Apache Marmotta*. La estructura de los datos y las observaciones son almacenadas en formato de tripletas en el SDW que se encuentra en la capa de datos. Para realizar la inserción de los datos es necesario especificar el grafo donde se almacena la información requerida, mediante consultas SPARQL (Tabla 2).

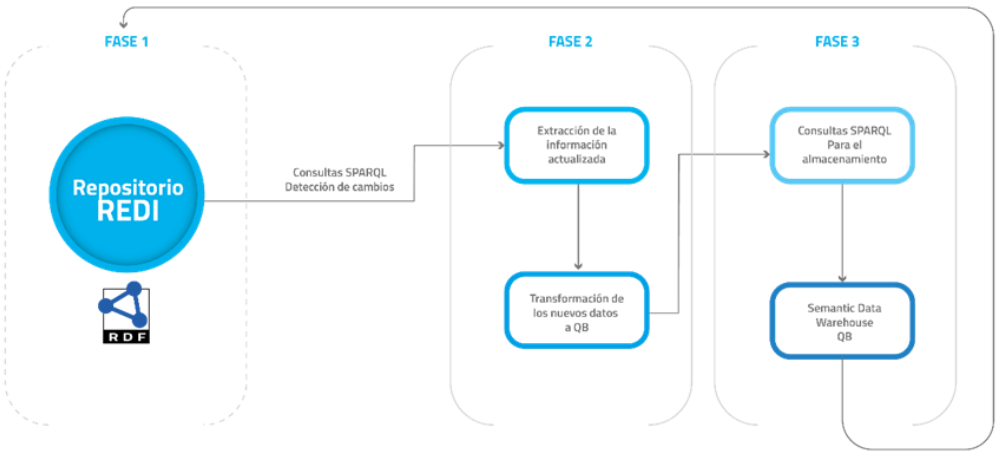

Figura 5. Proceso de detección para la actualización de la información.

Adicionalmente, es necesario definir políticas para actualización de la información en el SDW*:* esto se puede realizar de manera inicial e incremental. La actualización inicial consiste en detectar un cambio en el repositorio RDF mediante una revisión periódica de la información, mediante consultas SPARQL. Una vez detectado el cambio se elimina toda la información almacenada y se carga nuevamente toda la información actualizada. La actualización incremental consiste en trabajar con los resultados obtenidos previos al cambio y continuar con el proceso de actualización de los nuevos datos. De acuerdo a Reyes-Álvarez *et al.* (2014) para realizar una actualización incremental se debe realizar este proceso en tres fases, la primera consiste en detectar un cambio, la segunda fase es generar las tripletas con los nuevos cambios y la tercera fase es la actualización del grafo con las nuevas tripletas (Figura 5).

Componente 4: Extracción de datos QB

Con la información almacenada en el grafo (Tabla 2, capa de datos) a través de la ejecución de consultas SPARQL, se puede extraer la estructura y las observaciones para que esta información sea utilizada por la capa de presentación.

3.3. Capa de presentación

Es la encargada de visualizar la información que se encuentra almacenada en el SDW, para esto, es necesario utilizar una herramienta que permita la visualización de datos multidimensionales semánticos. De acuerdo a la literatura estudiada y para el presente trabajo se utiliza la herramienta "*OpenCube Toolkit*". Mediante la utilización de la opción "*Data Provider Setup*" se realizó la conexión con el grafo que se encuentra en *Apache Marmotta*, adicionalmente, se trabajó con el *widget* "*OpenCube Compatibily Explorer*", que permite comprobar la compatibilidad de cubo creado para realizar los cálculos con las medidas y dimensiones establecidas para iniciar el manejo de los datos; el *widget* "*OpenCube Aggregator*", permite generar las agregaciones de suma y promedio. Una vez escogido el tipo de comportamiento para la medida, se procede a generar el cubo. Finalmente, con el *widget* "*OpenCube Browser*" se procede a la visualización del modelo multidimensional basados en tecnología semántica.

3.4. Capa cliente

En esta capa el usuario cliente podrá acceder a la información almacenada en el cubo multidimensional, basado en tecnologías semánticas, desde un navegador ingresando a la dirección http://redi.cedia.org.ec/#/es/data/datacube. Basado en la arquitectura propuesta, los componentes definidos en la capa de aplicación fueron automatizados con el fin de procesar todos los datos del REDI utilizando las consultas definidas en este trabajo.

4. RESULTADOS

Para demostrar la utilidad del proceso descrito anteriormente, en esta sección se presenta un caso de uso, que a través de la ejecución de una consulta SPARQL se detecta lo siguiente: autores que dispongan 10 publicaciones generadas en el período 2006-2016, que los autores pertenezcan a tres IES diferentes, (Tabla 3), acorde a esta información se responden a las preguntas planteadas en la sección 3.2.

Tabla 3. Ejemplo1: Datos para la ejecución del cubo multidimensional basado en tecnología semántica.

Autor	Institución	Publicaciones	Período de publicación

Víctor Saquicela	UCUENCA	10	2012 - 2016
Mauricio Espinoza	UCUENCA	10	2006 - 2016
Rodrigo Fonseca	ESPE	10	2006 - 2013
Xavier Ochoa	ESPOL	10	2008 - 2016
Lorena Alvarez	ESPOL	10	2007 - 2016

4.1. ¿Cuál es el número de publicaciones realizadas por autor, en un período determinado?

En la Tabla 4 se aprecia el número total de publicaciones de cada uno de los cinco autores ingresados al cubo por año de acuerdo al período de tiempo indicado. El autor "Lorena Alvarez", ha generado una publicación por año: 2007, 2008, 2012, 2014, 2015 y 2016, y publicó tres artículos en el año 2013. El autor, "Mauricio Espinoza", generó una publicación en el año 2006, 2007, 2010, 2011, 2015, 2016 y dos publicaciones en el año 2009 y 2014. El autor "Víctor Saquicela", publicó dos artículos en el año 2012, 3 en el año 2014 y finalmente 5 en el año 2015. El autor "Xavier Ochoa", publicó un artículo en el año 2008 y 2013, dos en el año 2011, tres en el año 2014 y 2016. El autor "Rodrigo Fonseca", generó una publicación por año en 2004, 2005, 2008, 2012, en el año 2007 generó cuatro publicaciones, mientras que en el año 2009 y 2013 generó dos publicaciones por año.

Tabla 4. Visualización del número de publicaciones por autor por año.

Año de publicación	Lorena Alvarez	Mauricio Espinoza	Rodrigo Fonseca	Victor Saquicela	Xavier Ochoa
2004	-	-	1	-	-
2005	-	-	1	-	-
2006	-	1	-	-	-
2007	1	1	4	-	-
2008	1	-	1	-	1
2009	-	2	2	-	-
2010	-	1	-	-	-
2011	-	1	-	-	2
2012	1	-	1	2	-
2013	3	-	2	-	1
2014	1	2	-	3	3
2015	1	1	-	5	-
2016	1	1	-	-	3

4.2. ¿Cuál es el número de publicaciones generadas por cada IES por año?

En la Tabla 5 se aprecia el número total de publicaciones que cada IES ha generado por año. La ESPE ha generado cuatro publicaciones en el año 2007, una publicación en el año 2008 y 2012 y dos publicaciones en el año 2009 y 2013. La ESPOL ha publicado un artículo en el año 2007, 2012 y 2015, dos publicaciones ene l año 2008, 2011 y finalmente cuatro publicaciones en el año 2013 y 2014. La UCUENCA, ha publicado un artículo en el año 2006, 2007, 2010, 2011, 2012, cinco en el año 2014 y seis en el año 2015, de acuerdo a la información que se encuentra almacenada en el cubo.

El resto de preguntas fueron contestadas visualizando sus resultados correctos, debido a la restricción del número de imágenes y tablas, estas no han sido incorporadas en este trabajo.

Tabla 5. Número de publicaciones generadas por IES por año.

Año de publicación	ESPE	ESPOL	UCUENCA
2006			1
2007	4	1	1
2008	1	2	
2009	2		2
2010			1
2011		2	1
2012	1	1	1
2013	2	4	
2014		4	5
2015		1	6

5. CONCLUSIONES Y TRABAJOS FUTUROS

Para trabajar con modelos multidimensionales basados en tecnología semántica se concluye que: primero, es necesario identificar qué información de interés se visualizará; segundo, establecer el proceso de transformación de información en RDF a RDF *Data Cube*, definiendo el conjunto de datos, la estructura de los datos, dimensiones, medida (s) y generar las observaciones de acuerdo a la estructura del cubo; finalmente, definir la herramienta de visualización idónea que permita publicar información de modelos semánticos basado en QB.

Adicionalmente, se puede concluir que la visualización de la información mejoró notablemente debido a que pasó de realizar consultas estáticas a consultas dinámicas definidas por el usuario a través de una herramienta de visualización, mediante la utilización modelos multidimensionales. En el caso del presente estudio se trabajó con un ejemplo, que utilizó cuatro dimensiones y una medida.

En base al estudio realizado se han detectado nuevas actividades para mejorar la herramienta, como por ejemplo generar la visualización de la información de manera geográfica, utilizando ontologías como *"geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>"* que permiten utilizar propiedades como "geo:lat" y "geo:long" para la visualización de la información de manera geográfica.

Actualmente el vocabulario "QB4OLAP" permite realizar operaciones agregadas como "sum, min, avg, cout, max" directamente desde la estructura del vocabulario. Se puede realizar a futuro pruebas sobre éste, pero con la limitante de que aún no es un estándar de la W3C.

Mejorar la presentación de la capa cliente haciéndola más intuitiva para el usuario es un reto importante, incluyendo más de un conjunto de datos, generando *slices,* incorporando el número de dimensiones y medidas como co-autores, palabras clave, incorporar más áreas de conocimiento, para que el usuario pueda realizar un detalle minucioso sobre las publicaciones generadas, para que en el momento de buscar investigadores ecuatorianos que trabajen sobre un área de conocimiento en específico disponga de toda la información necesaria.

AGRADECIMIENTOS

Este trabajo fue realizado en la Universidad de Cuenca enmarcado en la tesis de maestría "*Gestión Estratégica de las Tecnologías de la Información*" y ha contado con el apoyo del Departamento de Ciencias de la Computación. Se ha trabajado dentro del proyecto REDI apoyado y financiado por el Consorcio Ecuatoriano para el Desarrollo del Internet Avanzado - CEDIA.

REFERENCES

Aggoume, A., Bouramoul, A., & Khiereddine Kholladi, M. (2016). *Big data integration: A semantic mediation architecture using summary*. 2nd International Conference on Advanced Technologies for Signal and Image Processing, Monastir, Tunesia, pp. 21-25.

Bayerl, S., & Granitzer, M. (2015). *Data-transformation on historical data using the RDF data cube vocabulary*. Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business, 20 pp. Disponible en http://swib.org/swib15/slides/bayerl_data-transformation.pdf

Bellatreche, L., Khouri, S., & Berkani, N. (2013). *Semantic data warehouse design: From ETL to deployment à la carte*. DASFAA: International Conference on Database Systems for Advanced Applications, pp. 64-83.

Ghasemi, S. (2014). M2RML: *Mapping multidimensional data to RDF*. Thesis of Master of Science, School of Computing Science, Faculty of Applied Sciences, Simmon Fraiser University, Burnaby, British Columbia.

Helmich, J. (2013). *Analysing and visualizing statistical linked data*. Tesis de Maestría, Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, 125 pp.

Kämpgen, B., & Harth, A. (2011). *Transforming statistical linked data for use in OLAP systems*. I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Graz, Austria, 8 pp. Disponible en http://www.aifb.kit.edu/images/2/28/Kaempgen_harth_isem11_olap.pdf

Martin, M., Abicht, K., Stadler, C., Auer, S., Ngibga Ngomo, A-C., & Soru, T. (2015). *Cubeviz: Exploration and visualization of statistical linked data*. WWW '15 Companion Proceedings of the 24th International Conference on World Wide. Florencia, Italia. pp. 219-222. Disponible en http://www.www2015.it/documents/proceedings/companion/p219.pdf

Nebot, V., Berlanga, R., Pérez, J. M., Aramburu, M. J., Pedersen, T. B. (2009). *Multidimensional integrated ontologies: A framework for designing semantic data warehouses*. Journal on Data Semantics XIII, Springer, 36 pp.

OpenCube Toolkit (2013). *OpenCube Project*. Disponible en http://opencube-toolkit.eu/

Reyes-Álvarez, L., Hidalgo-Delgado, Y., Roldán-García, M., Aldana-Montes, J. F. (2014). *Exploring incremental reasoning approaches based on module extration.* Proc. of XII Congreso Internacional de Información. Palacio de convenciones de la Habana, Cuba, 12 pp. Disponible en http://ceur-ws.org/Vol-1219/paper1.pdf

Rivera Salas, P. E., Maia Da Mota, F., Martin, M., Auer, S., Breitman, K., & Casanova, M. A. (2012). *Publishing statistical data on the web*. Sixth International Conference on Semantic Computing, IEEE Computer Society, pp. 285-292. Disponible en https://svn.aksw.org/papers/2012/ESWC_PublishingStatisticData/public.pdf

Vdovjak, R., & Houben, G-J. (2001). *RDF based architecture for semantic integration of heterogeneous information sources*. Workshop on information integration on the Web, pp. 51-57.