

# Bases de Datos Semánticas

Irving Caro Fierros, Víctor Hugo Menéndez Domínguez y María Enriqueta Castellanos Bolaños

Facultad de Matemáticas  
Universidad Autónoma de Yucatán  
Mérida, México

irving.cf@gmail.com, {mdoming, enriqueta.c}@correo.uady.mx

**Resumen**— En 1992, cuando Tim Berners-Lee dio a conocer la primera versión de la Web, su visión a futuro era incorporar metadatos con información semántica en las páginas Web. Es precisamente a principios de este siglo que inicia el auge repentino de la Web semántica en el ambiente académico e Internet. El modelo de datos semántico se define como un modelo conceptual que permite definir el significado de los datos a través de sus relaciones con otros. En este sentido, el formato de representación de los datos es fundamental para proporcionar información de carácter semántico. La tecnología enfocada en las bases de datos semánticas se encuentra actualmente en un punto de inflexión, al pasar del ámbito académico y de investigación a ser una opción comercial completa. En este artículo se realiza un análisis del concepto de base de datos semántica. También se presenta un caso de estudio donde se ejemplifican operaciones básicas que involucran la gestión de la información almacenada en este tipo de base de datos.

**Palabras Clave**—Modelo de datos semántico, Web semántica, bases de datos semánticas, RDF, SPARQL.

## I. INTRODUCCIÓN

Cuando Tim Berners-Lee presentó la primera versión de la Web en 1992, su visión a futuro era añadir metadatos con información semántica en las páginas Web, que permitieran describir el contenido del recurso, su significado y la relación que mantuviera con otros datos [1]. El modelo de datos semántico se define como un modelo conceptual que tiene la capacidad de expresar información que permite interpretar el significado de las instancias sin necesidad de conocer su verdadera representación. Este modelo, orientado a los hechos, brinda la posibilidad de definir el significado de los datos en contexto de sus relaciones con otros [2].

A inicios del siglo XXI comienzan a surgir en los ambientes académicos e Internet, artículos acerca del auge inminente de la Web semántica y la revolución que traería consigo, al permitir al usuario consultar información sobre un tema en particular en un sitio Web, y en caso de que éste careciera de ella, podría utilizar otras fuentes para resolver la petición [3].

El W3C (World Wide Web Consortium) es una comunidad internacional donde las organizaciones Miembro, el personal de tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web [4]. Dentro de las funciones de la organización se incluye la definición del concepto de Web semántica, nombre con el que se conoce al conjunto de tecnologías y formatos cuyo objetivo es permitir la publicación de datos que puedan ser legibles e interpretados por aplicaciones informáticas [5].

La pila de la Web semántica se compone de los siguientes elementos (en orden ascendente) [5]:

- Tecnologías de hipertexto IRI (Internationalized Resource Identifier) y XML (Extensible Markup Language). Proveen

la base a las demás capas. Esta es compartida con la Web tradicional.

- Tecnologías propias de la Web semántica. Contiene tecnologías estandarizadas por la W3C que facilitan construir aplicaciones Web semánticas. Entre estas tecnologías se encuentran formatos de representación de datos RDF (Resource Description Framework) junto con su lenguaje de consulta SPARQL (SPARQL Protocol and RDF Query Language), así como sus formatos de intercambio de reglas RIF (Rule Interchange Format), etcétera.
- Tecnologías semánticas faltantes. Aún no se encuentran estandarizadas o se encuentran en fase de concepto. Ejemplos de estas es la validación de las fuentes de datos, y la interfaz de usuario que sirva de enlace entre los usuarios y las aplicaciones.

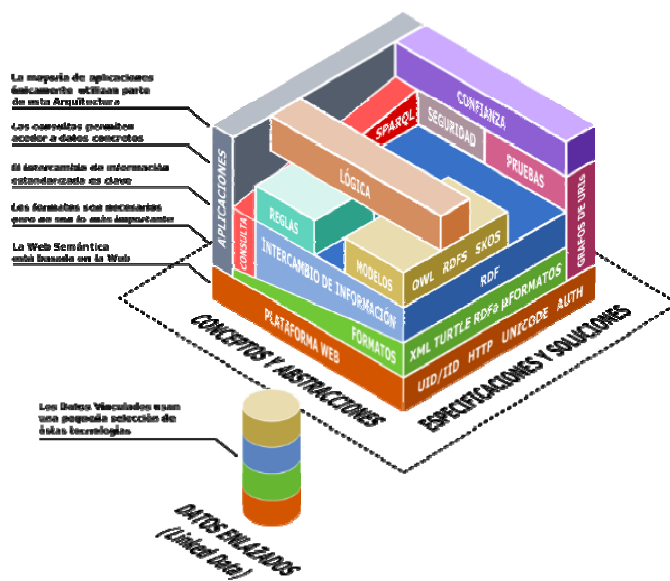


Fig. 1. Arquitectura Tecnológica de la Web Semántica [7]

Cuando se habla de Web semántica, el formato de representación de los datos es crucial para proporcionar información de orden semántico [5]. Debido a esto, han surgido una serie de tecnologías diferentes al modelo relacional que permiten almacenar y recuperar tanto la representación de los datos como la relación entre estos. Un modelo de base de datos es una especificación que describe cómo una base de datos es estructurada y usada (Figura 2). En el caso de las bases de datos semánticas, los datos se modelan usando grafos [2].

Otra forma de representar los datos y sus relaciones es mediante la especificación RDF de la W3C [6]. Los formatos de serialización más comunes son XML y Notation3 o N3 (N-tripletas). El formato XML tiene la ventaja de poder ser analizado rápidamente por una máquina.

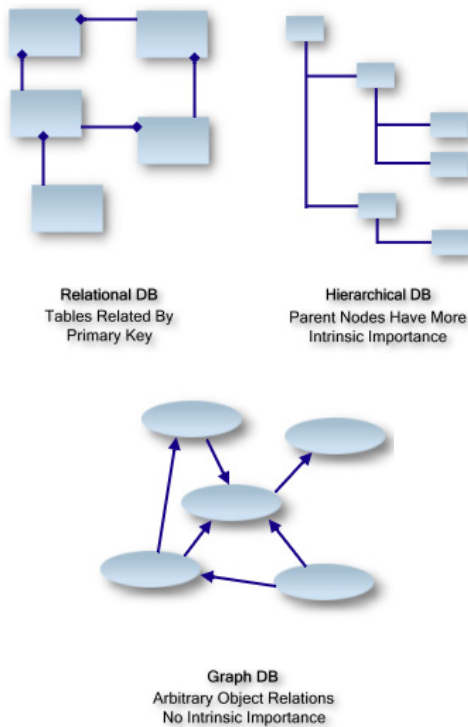


Fig. 2. Modelos de bases de datos [8]

Sin embargo, conforme la cantidad de datos descritos aumenta, tiende a ser complejo de leer para los seres humanos; de ahí el surgimiento de Notation3 como formato de lectura más amigable [7]. El tres en el nombre se debe a las tripletas del modelo en RDF, el cual al ser similar al de entidad-relación, está basado en la idea de generar sentencias acerca de recursos en la forma de expresiones sujeto-predicado-objeto. El sujeto denota el recurso, el predicado especifica los tratamientos o aspectos del recurso y expresa una relación entre el sujeto y el objeto. Nótese que un sujeto también puede ser referenciado como objeto en otra sentencia RDF [6].

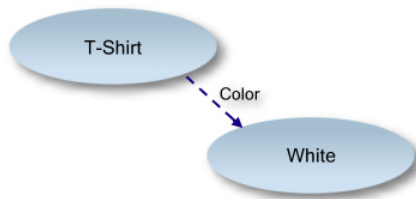


Fig. 3. Grafo RDF [8]

En el grafo que se muestra en la Figura 3, el sujeto es la camiseta, el predicado es el color y el objeto es el color blanco. Usando XML como formato de serialización, el modelo de datos queda representado así:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:feature="http://www.linkeddatatools.com/clothing-features#">
  <rdf:Description
rdf:about="http://www.linkeddatatools.com/clothes#t-shirt">
    <feature:color
rdf:resource="http://www.linkeddatatools.com/colors#white"/>
  </rdf:Description>
</rdf:RDF>
```

Al igual que los datos modelados mediante el modelo relacional, en los datos RDF es necesario utilizar lenguajes de

consulta propios [6]. Respaldo por la W3C, SPARQL (acrónimo recursivo de SPARQL Protocol and RDF Query Language) es un lenguaje de consultas RDF capaz de obtener y manipular datos en bases de datos semánticas. SPARQL brinda la posibilidad de construir consultas a múltiples recursos (a esta operación se le conoce como consulta federada) [9].

SPARQL proporciona constructores que efectúan consultas usuales como SELECT, CONSTRUCT, ASK y DESCRIBE presentes en los lenguajes SQL tradicionales [9]. La sintaxis general de las consultas se presenta en la Figura 4.

```
PREFIX (Namespace Prefixes)
e.g. PREFIX plant: <http://www.linkeddatatools.com/plants>

SELECT (Result Set)
e.g. SELECT ?name

FROM (Data Set)
e.g. FROM <http://www.linkeddatatools.com/plantsdata/plants.rdf>

WHERE (Query Triple Pattern)
e.g. WHERE { ?planttype plant:planttype ?name }

ORDER BY, DISTINCT etc (Modifiers)
e.g. ORDER BY ?name
```

Fig. 4. Plantilla para consultas SPARQL [10]

Como se ha mencionado, resulta ideal almacenar datos en formato RDF mediante un sistema de base de datos en grafos. Este tipo de base de datos existe desde los años 70s, pero gracias a la Web semántica y a la estandarización del modelo de datos, han surgido en los últimos años soluciones de almacenamiento de datos que brindan una opción lo suficientemente madura para los estándares de la industria [2].

## II. ESTADO DEL ARTE

Al momento de seleccionar una base de datos para persistir y recuperar datos semánticos, se deben tomar consideraciones en cuanto al modelo de datos [2]. Se puede optar por un almacenamiento nativo usando el modelo relacional o bien uno NoSQL: mediante grafos, tripletas RDF, etcétera.

Diversos estudios han demostrado empíricamente que en general las bases de datos RDF son en promedio tres veces más lentas que sus contrapartes relacionales [11-13].

La comparativa entre las bases de datos RDF con sus homólogas NoSQL es más complicado de realizar al no existir un estándar en su implementación [11]. En general, las segundas tienden a ser más competitivas que las RDF en términos de tiempo de consulta. Además, la mayoría de las bases de datos NoSQL están optimizadas para realizar trabajos de carga de datos en paralelo y se pueden escalar con facilidad en comparación con las de tipo RDF [12]. Por último, en los escenarios de consulta SPARQL más complejos (el cual involucra la unión de varios nodos) las bases de datos RDF tienen un tiempo de respuesta mucho menor que las NoSQL [13]. A pesar de ello ofrecen beneficios en términos de funcionalidades.

Continuando la comparativa de las bases de datos semánticas mediante tripletas, en la Tabla I se presenta una lista de las principales implementaciones disponibles en el mercado al momento de escribir este artículo.

Es importante notar que la mayor parte de las implementaciones han sido actualizadas para ser compatibles con el estándar SPARQL 1.1, el cual fue lanzado en marzo de

TABLA I. PRINCIPALES BASES DE DATOS SEMÁNTICAS BASADAS EN TRIPLETAS [14]

| Nombre                                  | Almacenamiento nativo                                     | SPARQL | SPARQL Endpoint | APIs nativas                                                                   |
|-----------------------------------------|-----------------------------------------------------------|--------|-----------------|--------------------------------------------------------------------------------|
| 4store                                  | RDF                                                       | 1.1    | √               |                                                                                |
| AllegroGraph                            | Graph                                                     | 1.1    | √               | Para la mayoría de lenguajes de programación modernos                          |
| ARC2                                    | Propietario                                               | 1.1    | √               | PHP                                                                            |
| ARQ                                     | Propietario                                               | 1.1    |                 | Java                                                                           |
| BigData                                 | Triplestore                                               | 1.1    | √               | Java                                                                           |
| BrightstarDB                            | Graph data model in Heap file                             | 1.0    |                 | .NET Framework o Web Service                                                   |
| Corese                                  | Propietario                                               | 1.0    |                 | Java                                                                           |
| D2R Server                              | Propietario                                               | 1.1    | √               | Java                                                                           |
| Dydra                                   | Graph database in the cloudSaaS                           | 1.1    | √               | REST API                                                                       |
| Hercules                                | Stored in web browser                                     | 1.0    |                 | JavaScript                                                                     |
| Intellidimension Semantics Platform 2.0 | Propietario                                               | 1.0    |                 | .NET Framework                                                                 |
| Jena                                    | Tuple store                                               | 1.1    | √               | Java                                                                           |
| KAON2                                   | Propietario                                               | 1.0    |                 | Java                                                                           |
| SparkleDB                               | Triplestore/Quadstore                                     | 1.1    | √               | Para la mayoría de lenguajes de programación modernos                          |
| MarkLogic                               | Triplestore/Quadstore                                     | 1.0    | √               | REST API, SPARQL Endpoint, Graph Protocol Endpoint, Java API, XQuery, SQL/ODBC |
| Mulgara                                 | Propietario                                               | 1.0    |                 | Java, REST API                                                                 |
| OntoBroker                              | Triplestore                                               | 1.1    | √               | Java                                                                           |
| Ontotext OWLIM                          | Propietario                                               | 1.1    | √               | Java                                                                           |
| Open Anzo                               | Propietario                                               | 1.0    | √               | Java, JavaScript, .NET Framework                                               |
| OpenLink Virtuoso                       | Hybrid (Relational Tables and Relational Property Graphs) | 1.1    | √               | Para la mayoría de lenguajes de programación modernos                          |
| Parliament                              | Propietario                                               | 1.1    | √               | Java, C++                                                                      |
| Pellet                                  | Propietario                                               | 1.0    |                 | Java                                                                           |
| Profium Sense                           | In-memory triplestore                                     | 1.0    | √               | Java                                                                           |
| RAP                                     | In-memory triplestore or heap file                        | 1.0    |                 | PHP                                                                            |
| RDF API for PHP                         | Propietario                                               | 1.0    |                 | PHP                                                                            |
| RDF::Query                              | Propietario                                               | 1.1    | √               | Perl                                                                           |
| RDF-3X                                  | Triplestore                                               | 1.0    |                 |                                                                                |
| Redland, Redstore                       | Propietario                                               | 1.1    | √               | C                                                                              |
| SemWeb.NET                              | Propietario                                               | 1.0    | √               | .NET                                                                           |
| Sesame                                  | Propietario                                               | 1.1    | √               | Java                                                                           |
| SPARQL Engine                           | Propietario                                               | 1.0    |                 | Java                                                                           |
| Stardog                                 | Triplestore                                               | 1.1    | √               | Java, Groovy                                                                   |
| StrixDB                                 | Triplestore                                               | 1.1    | √               | Lua                                                                            |
| Twinql                                  | Propietario                                               | 1.0    |                 | Lisp                                                                           |

2013. Este estándar, permite realizar actualizaciones a la base de datos mediante nuevas formas de consulta.

Algunas bases de datos admiten configurar la fuente de datos para alimentarse de otro motor [11]. Por ejemplo, se podría configurar una base de datos Sesame para que tome como fuente de datos una MySQL y realizar consultas SPARQL sobre esa base de datos externa mediante el intérprete de Sesame, el cual se encargaría de traducir las consultas a un lenguaje SQL. Sin embargo, lo anterior no es recomendable, ya que el tiempo requerido por el intérprete puede repercutir negativamente en la velocidad de respuesta en

la consulta [11]. Se espera que conforme las bases de datos en RFD maduren, este tipo de compatibilidad caiga en desuso.

Adicional a los servidores de bases de datos semánticos, existen en Internet un conjunto enorme de bases de conocimiento que pueden ser consultados a través de servicios Web mediante consultas SPARQL. A este tipo de interfaces se le conoce como *SPARQL Endpoints*. Un *Endpoint* (punto final o puerto) indica la ubicación específica para acceder a un servicio Web usando un protocolo y formato de datos estándar de los antes mencionados (N3, RDF, etcétera) [9, 15].

### III. CRÍTICA Y VALORACIÓN

Es aceptable que las bases de datos en grafos se encuentren en desventaja dentro de un mercado compuesto por el modelo relacional, con una madurez de más de 50 años.

La tecnología de base de datos semánticas se encuentra actualmente en un punto de inflexión, al pasar del ámbito académico y de investigación a posicionarse como una opción comercial completa. La problemática actual que afrontan las bases de datos en grafos es su poca documentación, si las comparamos con las de tipo relacional. A pesar de que el lenguaje de consulta SPARQL se encuentra ya en su versión 1.1, no existe la suficiente bibliografía para que los desarrolladores puedan documentarse en la construcción de consultas efectivas. De la misma forma, los cursos de las universidades están enfocados en su mayoría, en la enseñanza del modelo relacional. Esta situación origina que sea difícil alcanzar a la masa crítica de desarrolladores para darle el empuje a la tecnología a nivel comercial.

Por último, algunos de los sistemas de bases de datos en grafos han sufrido el abandono por parte de sus creadores. Lo cual ha propiciado que los nuevos usuarios deban analizar con detenimiento qué proveedor de base de datos deberán elegir, ya que en un par de años éstas podrían dejar de recibir soporte o no ser actualizadas a la última versión.

Lo anterior ha ocasionado que constantemente se compare las bases de datos semánticas con las relacionales. Consideramos que esta práctica es errónea, ya que ambos tipos de bases de datos fueron diseñadas para cumplir propósitos diferentes. Por consiguiente, es natural que la debilidad de una (e.g. muy alto tiempo de consulta en las bases de datos semánticas) sea la fortaleza de otra. De tal modo que al reflexionar sobre qué base de datos utilizar, es importante tener presentes los objetivos y metas que queramos lograr con ella.

Hemos visto la facilidad con la que es posible modelar datos y sus relaciones con el modelo semántico, lo que nos permite construir soluciones para problemas específicos. Ejemplo de ellos son los sistemas que relacionan un conjunto de materiales educativos heterogéneos con una temática, profesor o asignatura dada. La temática de estos artefactos no tendría que ser dada de alta por una persona, ya que a través de sus metadatos sería posible catalogarlos.

### IV. CASO DE ESTUDIO

Empezar a utilizar una base de datos semántica es bastante sencillo, ya sea instalando nosotros mismos un servidor de base de datos semántica, o bien haciendo peticiones a una base de conocimiento remota mediante un servicio Web llamado *SPARQL Endpoint*.

Como caso de estudio se describirán los pasos necesarios para instalar *OpenRDF Sesame* (<http://www.openrdf.org/>), un servidor de bases de datos semánticos, así como la creación de un repositorio de datos; se proporcionará un conjunto de datos ejemplo, se explicará la forma en cómo insertarlos dentro del repositorio y por último, se presentará un ejemplo de consulta sobre estos datos.

*OpenRDF Sesame* es una herramienta para procesamiento de datos en formato RDF. Incluye un intérprete de datos RDF, soluciones de almacenamiento en memoria, soluciones de almacenamiento para bases de datos relacionales (mediante un convertidor), así como la consulta de datos usando SPARQL. Opcionalmente posibilita exponer los repositorios usando *SPARQL Endpoints*. Para el lado del desarrollador, proporciona una API de Java que posibilita ligar clases de Java a repositorios.

Los pasos para instalar Sesame se describen a continuación. Antes de comenzar es necesario verificar que Java así como un contenedor de servlets como Tomcat o JBoss se encuentren instalados; para nuestro ejemplo usaremos Apache Tomcat 7.

- 1) Descargar la versión más reciente de Sesame (que corresponda a la distribución “SDK”) desde <http://www.openrdf.org/download.jsp>.
- 2) Descomprimir el archivo con extensión “.zip” de Sesame en un directorio de su elección.
- 3) Iniciar Tomcat y a través de un navegador Web acceder al administrador de aplicaciones mediante la siguiente ruta: <http://localhost:8080/manager/html>. Es posible que sean solicitados un nombre de usuario y una contraseña; se recomienda verificar estas credenciales en el archivo `tomcat-users.xml` de la configuración del servidor Tomcat.
- 4) Una vez que haya sido registrado el administrador, ubicar la opción “Archivo WAR a desplegar” y elegir “seleccionar archivo”.
- 5) Seleccionar el archivo `openrdf-sesame.war` ubicado dentro de la carpeta `war` del directorio Sesame que fue descomprimido anteriormente. Posteriormente, seleccionar la opción “desplegar”. La página volverá a cargarse y será posible observar que ahora la aplicación `openrdf-sesame` aparece en la lista de aplicaciones instaladas en el Tomcat.
- 6) Repetir el paso anterior, seleccionando ahora el archivo `openrdf-workbench.war`.

La siguiente actividad consistirá crear un repositorio donde serán almacenados los datos RDF. Sesame nos permite hacer este almacenamiento en memoria o en disco duro. La primera opción posee tiempos de lectura menores, pero los datos desaparecen cuando el servidor es detenido. Por lo tanto, se procederá a crear un repositorio en disco duro.

- 1) Acceder a la URL <http://localhost:8080/openrdf-workbench/>. En el menú del lado izquierdo seleccionar “New Repository” (Figura 5).
- 2) Configurar las opciones del nuevo repositorio. Seleccionar “In Memory Store RDF Schema” como tipo, asignar un valor entero como 20 a ID y escribir el texto “SesameRepo” en Title.
- 3) Elegir la opción “Siguiente” y confirmar la persistencia como “Sí”. Por último, seleccionar “Crear” y se tendrá un nuevo repositorio listo para almacenar datos RDF.

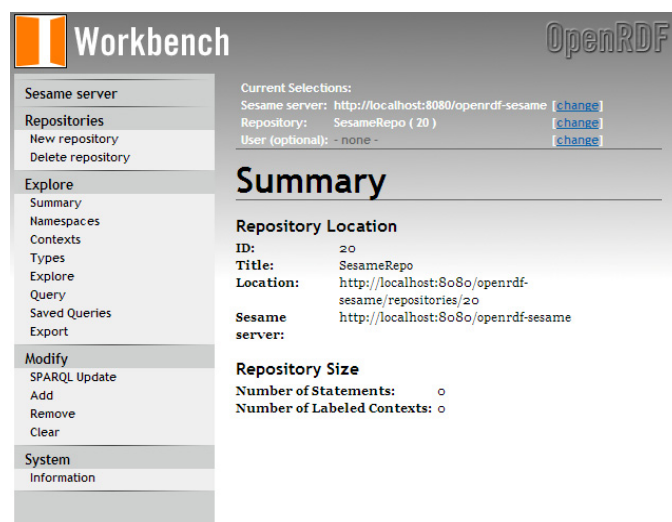


Fig. 5. Resumen del repositorio de datos



La siguiente actividad consiste en insertar un conjunto de datos en el repositorio. Para ello se define un documento FOAF (Friend of a Friend, Amigo de un amigo), el cual es una ontología que describe a personas, sus actividades y sus relaciones con otras personas y objetos. En el espacio de nombres foaf se describen los atributos disponibles como nombre, edad, correo electrónico, etcétera. Un espacio de nombres XML (Namespace) es una recomendación W3C para proporcionar elementos y atributos con nombre único en un archivo XML. Recordemos que para comprender mejor la estructura de estos documentos es necesario conocer acerca de XML y sus Namespaces.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:foaf="http://xmlns.com/foaf/0.1/">
```

```
<foaf:Person rdf:ID="aza">
  <foaf:name>Azanet</foaf:name>
  <foaf:mbox
rdf:resource="mailto:aza@rdfweb.org"/>
</foaf:Person>
<foaf:Person rdf:ID="andree">
  <foaf:name>Andree</foaf:name>
</foaf:Person>
<foaf:Person rdf:ID="irving">
  <foaf:name>Irving</foaf:name>
  <foaf:weblog
rdf:resource="http://www.example.com"/>
  <foaf:mbox
rdf:resource="mailto:irving@rdfweb.org"/>
  <foaf:knows rdf:resource="#aza"></foaf:knows>
  <foaf:knows
rdf:resource="#andree"></foaf:knows>
</foaf:Person>
</rdf:RDF>
```

En la Figura 6 se describen tres personas: Irving, Azanet y Andree. Los datos anteriores se pueden representar de igual manera mediante un grafo.

El proceso para insertar los datos en el repositorio que se ha creado, se describe a continuación (Figura 7):

- 1) Seleccionar la opción Add (Agregar) en la página de resumen.
- 2) Escribir en el campo Base URI la URL: `http://hosting.friends/database/`. Con esto se estaría indicando que el contenido de los datos a insertar estarán disponibles mediante esa dirección electrónica.
- 3) Mantener seleccionada la opción "usar la URI Base como identificador de contexto".

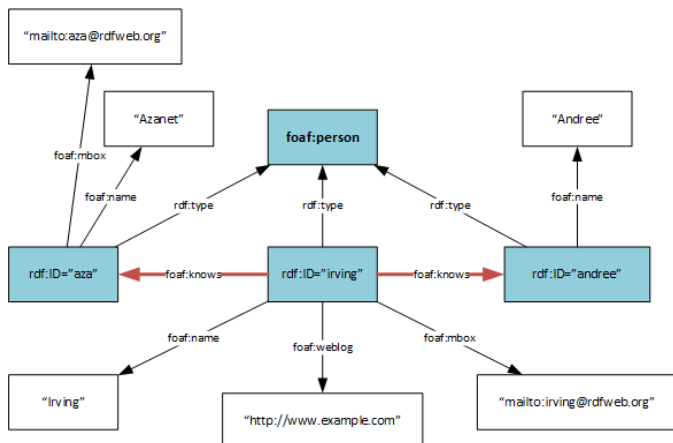


Fig. 6. Representación gráfica de un espacio de nombres

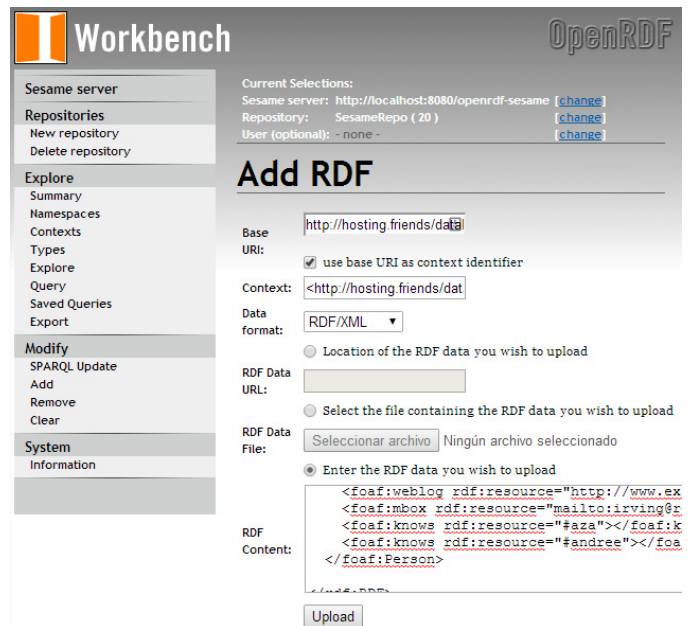


Fig. 7. Inserción de datos RDF al repositorio

- 1) Seleccionar "RDF/XML" como Data format (Formato de datos).
- 2) Activar la opción "Enter the RDF data you wish to upload" (Introduce los datos RDF que deseas cargar). En el formato de datos seleccionamos "RDF/XML" y seleccionamos "Ingresar los datos que deseas ingresar".
- 3) Insertar los datos en el área de texto inferior.
- 4) Presionar el botón "Upload" y observar que en la página de resumen se ha incrementado el número de sentencias en el repositorio.

Ahora que se tienen los datos en el repositorio ya es posible empezar a realizar consultas sobre ellos.

En la página de resumen del repositorio se deberá seleccionar la opción Query. En la siguiente pantalla se podrá elegir el lenguaje de consulta (SPARQL por defecto), la consulta por sí misma, el número de resultados por mostrar y las acciones a realizar (ejecutar la sentencia o restablecer los campos).

A continuación se formulan una serie de expresiones de consulta para comprobar que es posible resolverlas con facilidad a través del modelo semántico:

A. "Lista los nombres de todas las personas registradas en el repositorio"

Dentro del campo Query se escriben las siguientes sentencias:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?name WHERE{
  ?person foaf:name ?name.
}
```

La consulta devuelve el resultado que se presenta en la Tabla II.

TABLA II. NOMBRES DE LAS PERSONAS REGISTRADAS EN EL REPOSITORIO

| Name     |
|----------|
| "Azanet" |
| "Andree" |
| "Irving" |

Tal como se aprecia, en la primera parte de la consulta se escribe el prefijo *foaf*, que nos sirve para abreviar la URL donde se encuentran los atributos del sujeto *person*. Si se omitiera el prefijo, sería necesario reescribir la consulta de la siguiente manera:

```
SELECT ?name WHERE{
  ?person <http://xmlns.com/foaf/0.1/name> ?name.
}
```

**B. “Lista el correo electrónico y el weblog de la persona llamada Irving”**

Dentro del campo Query se escriben las siguientes sentencias:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?email ?weblog WHERE{
  ?person foaf:name "Irving";
  foaf:mbox ?email;
  foaf:weblog ?weblog.
}
```

La consulta devuelve el resultado que se presenta en la Tabla III.

TABLA III. CORREO ELECTRÓNICO Y WEBLOG DE LA PERSONA LLAMADA IRVING

| Email                                                           | Weblog                                                      |
|-----------------------------------------------------------------|-------------------------------------------------------------|
| <a href="mailto:irving@rdfweb.org">mailto:irving@rdfweb.org</a> | <a href="http://www.example.com">http://www.example.com</a> |

**C. “Lista los nombres de los amigos de la persona llamada Irving”**

Dentro del campo Query se escriben las siguientes sentencias:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?friendname WHERE{
  ?person foaf:name "Irving";
  foaf:knows ?friend.
  ?friend foaf:name ?friendname.
}
```

La consulta devuelve el resultado que se presenta en la Tabla IV.

TABLA IV. NOMBRES DE LOS AMIGOS DE LA PERSONA LLAMADA IRVING

| Friendname |
|------------|
| "Azanet"   |
| "Andree"   |

**V. CONCLUSIONES**

Las bases de datos semánticas ya son una realidad tal como lo predijo Tim Berners-Lee. Posiblemente no es tan evidente para el usuario final, pero cada vez más empresas están agregando esa capa semántica a los datos que producen y consumen día a día. Tanto empresas con años en el sector de las bases de datos relaciones (Oracle) como otras nuevas están expandiendo las especificaciones del modelo semántico con la finalidad de ofrecer productos maduros, listos para ser integrados en desarrollos de software y con la interoperabilidad que demanda el mercado.

Como hemos visto, el modelo semántico nos permite inferir relaciones entre entidades, a pesar de que en su almacenamiento físico éste no sea evidente. Por ejemplo, una

entidad *perro* y una entidad *gato* que se encuentren almacenados en dos repositorios de datos sin relación alguna (llave primaria, externa, etcétera) pueden ser catalogados como mascotas si esos datos son indexados en una base de conocimiento. Es entonces cuando los datos adquieren un valor más allá del dado por una aplicación determinada, es decir, se transforma en información con un valor duradero.

Si tratáramos de implementar el caso de estudio presentado con anterioridad bajo un esquema relacional, veríamos que es sumamente complejo, dado que el modelo relacional asume que la relación entre los objetos está en los objetos; dicho de otra forma, asume que debe existir un objeto que vincule directamente la entidad *perro* con la entidad *gato* para determinar si existe alguna relación entre ellas.

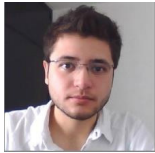
Hace un par de años, los detractores del modelo semántico argumentaban que la principal barrera para que los desarrolladores de software adoptaran este nuevo paradigma era la falta de herramientas, lenguajes y metodologías para construir aplicaciones Web semánticas 2.0. En los últimos años ha habido una explosión de herramientas para construir la pila Web semántica. Prueba de ello es que la mayor parte de las bases de datos presentadas en este artículo tienen menos de 3 años. El impulso que están adquiriendo las bases de datos NoSQL (entre las que se encuentran las basadas en tripletas), está logrando que los desarrolladores consideren este nuevo paradigma para integrarlo en sus desarrollos de software.

**REFERENCIAS**

- [1] T. Segaran, C. Evans y J. Taylor, “Programming the semantic Web. Semantic Web services processes and applications”, vol. 54, p. 300, 2009. Recuperado de <http://www.amazon.com/Programming-Semantic-Web-Toby-Segaran/dp/0596153813>
- [2] C. J. Date, “An introduction to database systems, vol. 1, p. 839.
- [3] D. Allemang, y J. Hendler, “Semantic web for the working ontologist: effective modeling in RDFS and OWL”, 2a ed., 2011. Recuperado de [http://books.google.com/books?hl=en&lr=&id=\\_qGKPOIB1DgC&oi=fnd&pg=PP2&dq=Semantic+web+for+the+working+ontologist:+effective+modeling+in+RDFS+and+OWL&ots=-ZfgLVAMkM&sig=jbR1BFgLuaUILoS262WXcj\\_OJ1s](http://books.google.com/books?hl=en&lr=&id=_qGKPOIB1DgC&oi=fnd&pg=PP2&dq=Semantic+web+for+the+working+ontologist:+effective+modeling+in+RDFS+and+OWL&ots=-ZfgLVAMkM&sig=jbR1BFgLuaUILoS262WXcj_OJ1s)
- [4] W3C, “About W3C”, Mayo 2016. Recuperado de <http://www.w3.org/Consortium/>
- [5] Q. Quboa y M. Saraee, “A state-of-the-art survey on semantic Web mining”, Intelligent Information Management, num. 5, vol. 1, pp. 10–17.
- [6] K. Coyle, “Linked data tools: connecting on the Web”, Library Technology Reports, num. 4, vol. 48, pp. 6–9, 2012. Recuperado de <http://books.google.com.mx/books?id=2kALpInkMtEC&lpq=P A44&ots=9B1jeih4MB&dq=www.linkeddatatools.com%2Fsemantic-web-basics&pg=PA44#v=onepage&q=www.linkeddatatools.com/semantic-web-basics&f=false>
- [7] L. C. Fernández. “Procedimiento semi-automático para transformar la Web en Web semántica”, Universidad Nacional de Educación a Distancia, 2009. Recuperado de <http://dialnet.unirioja.es/servlet/tesis?codigo=20625>
- [8] LinkedDataTools, “Introducing graph data”, Mayo 2014. Recuperado de <http://www.linkeddatatools.com/introducing-rdf>
- [9] J. Hebler, M. Fisher, Matthew, R. Blace, A. Perez-Lopez y M. Dean. “Semantic Web programming”. Indianapolis, IN: Wiley Publishing Inc, 2009, p. 406.
- [10] LinkedDataTools, “Querying semantic data”, Mayo 2016. Recuperado de <http://www.linkeddatatools.com/querying-semantic-data>
- [11] M. Schmidt, T. Hornung, N. Küchlin, G. Lausen y C. Pintel, “An Experimental Comparison of RDF Data Management

Approaches in a SPARQL Benchmark Scenario”, in Lecture Notes in Computer Science, vol. 5318, 2008, pp. 82–97.

- [12] M. Schmidt, T. Hornung, M. Meier, C. Pinkel y G. Lausen, “SP2bench: A SPARQL performance benchmark”, in *Semantic Web Information Management: A Model-Based Perspective*, Springer-Verlag Berlin Heidelberg, 2010, pp. 371–393.
- [13] Y. Zhang, P.M. Duc, O. Corcho y J.-P. Calbimonte, “SRBench : A Streaming RDF / SPARQL Benchmark”, *The Semantic Web – ISWC 2012*, Springer, 2012, pp. 641–657.
- [14] Wikipedia, “TripleStore”, Junio 2016. Recuperado de <http://en.wikipedia.org/wiki/Triplestore>
- [15] W3C Wiki, “SparqlEndpoints”, Marzo 2016. Recuperado de <http://www.w3.org/wiki/SparqlEndpoints>



**Irving Caro Fierros** es Ingeniero de Software por parte de la Universidad Autónoma de Yucatán (UADY). Es desarrollador de aplicaciones. Ha colaborado en proyectos de investigación relacionados con Tecnologías Web. Su trabajo de investigación se centra en Ingeniería Web y Bases

de Datos.



**Víctor Hugo Menéndez Domínguez** es Doctor en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha (UCLM), España; tiene un Máster en Tecnologías Informáticas por la misma institución. Además, cuenta con una Especialización en Docencia y una Licenciatura en Ciencias de la Computación por

parte de la Universidad Autónoma de Yucatán (UADY), México. Es profesor titular en la Facultad de Matemáticas de la UADY. Su trabajo de investigación se centra en temas relacionados con la representación del conocimiento y el aprendizaje, así como la gestión de Objetos de Aprendizaje y Repositorios.



**María Enriqueta Castellanos Bolaños** es Maestra en Gestión de Tecnología de Información por la Universidad Anáhuac Mayab, México, con líneas de especialidad en Ingeniería de Software, Redes de Computadoras y Seguridad de Sistemas. Además, cuenta con una Especialización en Docencia y una Licenciatura en Ciencias de la

Computación por parte de la Universidad Autónoma de Yucatán (UADY), México. Es profesora de tiempo completo en la Facultad de Matemáticas de la UADY. Su trabajo de investigación se centra en temas relacionados con la gestión del conocimiento y la Ingeniería Web.