

## MODELO DE SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN SEMÁNTICA CON BASE EN COMPILADOR

J. Guadalupe Ramos Díaz<sup>1</sup>, Rogelio Ferreira Escutia<sup>2</sup>, Juan Carlos Olivares Rojas<sup>3</sup>, Adrián Núñez Vieyra<sup>4</sup>, Isela Navarro Alatorre<sup>5</sup>, Juan Jesús Ruiz Lagunas<sup>6</sup>

Eje1. La investigación en las Ciencias Básicas.

Mesa 2. Ciencias de la Ingeniería y Tecnología.

**Palabras Clave:** (Web Semántica, Datos abiertos, Compilador)

La publicación de información de interés público en la modalidad de datos abiertos, mediante archivos formateados a partir de lenguajes semánticos de la Web 3.0 tales como RDF, RDFS u OWL constituyó un hito en la difusión de información para consumo masivo de los seres humanos. Sin embargo, a pesar de ser una solución conveniente para compartir información precisa, no ambigua, en contraste con la publicación de información en lenguaje natural, no tuvo un éxito superlativo a nivel global. Esto se debe en parte a que la publicación de la información requería conocimientos especializados en los lenguajes semánticos en comparación con el hecho de publicar una página con redacción en lenguaje natural.

En este trabajo proponemos un modelo de sistema de recuperación de información en el que colocamos un compilador de dominio específico, es decir un compilador diseñado expreso para procesar archivos RDF de un dominio particular de aplicación con funcionalidad en dos sentidos, primero para convertir información hacia tripletas RDF y el segundo para compilar tripletas RDF. De esta manera, un usuario que desee publicar información en formato RDF no requerirá conocimientos especializados, haciendo más fácil la generación de un RDF store. Por otra parte, quien desee alimentar el RDF store podrá subir tripletas que serán validadas por el compilador e incorporadas al almacén. Este compilador, construido a partir de herramientas generadores de procesadores de lenguajes como ANTLR, está integrado en la propuesta del sistema de recuperación de información semántica.

### Introducción

Guardar y recuperar información de un medio de almacenamiento digital a través de un sistema informático es una tarea que comúnmente se le atribuye a un Sistema de

<sup>1</sup> Tecnológico Nacional de México / Instituto Tecnológico de Morelia; j.guadalupe@morelia.tecnm.mx

<sup>2</sup> Tecnológico Nacional de México / Instituto Tecnológico de Morelia; rogelio.fe@morelia.tecnm.mx

<sup>3</sup> Tecnológico Nacional de México / Instituto Tecnológico de Morelia; juan.or@morelia.tecnm.mx

<sup>4</sup> Tecnológico Nacional de México / Instituto Tecnológico de Morelia; adrian.nv@morelia.tecnm.mx

<sup>5</sup> Tecnológico Nacional de México / Instituto Tecnológico del Valle de Morelia; isela.na@morelia.tecnm.mx

<sup>6</sup> Tecnológico Nacional de México / Instituto Tecnológico de Morelia; juan.rl@morelia.tecnm.mx

Recuperación de Información: IRS, del inglés *Information Retrieval System*, (Baeza, et al., 2011). Con el advenimiento de la Web, las herramientas IRS cobraron un protagonismo sustantivo en cada una de sus etapas evolutivas. En la Web 1.0 y 2.0, las tareas del IRS consistían en recuperar información escrita fundamentalmente en lenguaje natural. Los buscadores web han permitido por lustros, localizar información en sus enormes índices y devolver al usuario enlaces hacia la propia información requerida en la petición de búsqueda.

Con el fin de dotar con significado explícito a la información publicada en la Web y con la intención de que las máquinas buscadoras pudieran interpretar dicho significado para responder de manera precisa a preguntas a un usuario, tal como lo hace un gestor de bases de datos, surgió la Web 3.0 o también llamada Web Semántica. La intención de la Web Semántica consiste en la definición de vocabularios formales escritos en lenguajes de etiquetas derivados del XML (Yu, 2011), posteriormente, dichos vocabularios se emplean para escribir sentencias formales. De este modo se escriben oraciones en lenguajes formales que pueden ser recuperadas por un lenguaje de consulta.

Aunque esto último parece una ventaja muy importante, desgraciadamente no favoreció la publicación masiva de información semántica, la razón, que generar sentencias mediante lenguajes formales requiere de un nivel de conocimiento que normalmente no posee un usuario general.

Es en este contexto, proponemos el uso de compiladores destinados a generar texto codificando sentencias formales a partir de enunciados simples del usuario. Eso persigue la idea de que se simplifique la generación de texto formal sin la necesidad de que el usuario tenga que ser experto en lenguajes semánticos.

A continuación, en la Sección Antecedentes se enuncian las tecnologías que fundamentan la propuesta y enseguida, en la Sección “Modelo de IRS con base en compilador” se describe la propuesta de la presente investigación, para finalmente, en la Sección Conclusiones presentar las ideas de cierre del presente trabajo.

### Antecedentes

El área de aplicación más importante de los lenguajes asociados a la Web Semántica es la corriente “Open Data”, Gobierno de México (2022). Los denominados **datos abiertos** (Open Data Charter, 2015) son parte de un mecanismo global para que una organización pueda publicar información acerca de sus procesos de forma meta etiquetada. La intención de los datos abiertos en una organización es publicar información en la Web Semántica acerca de algún contexto (o un proceso), para que esté disponible al mundo, con la finalidad de que los creadores de aplicaciones puedan construir interfaces que hagan una vista más amigable de los datos y por tanto, que se incremente la transparencia y acceso a la información de la organización que los publica (Open Data

Charter, 2015). Regularmente los datos abiertos se publican en lenguajes formales de la Web Semántica como los descritos en Yu (2011).

Entre las tecnologías para la Web Semántica sobresalen estándares tales como el lenguaje extensible de marcado (XML del inglés *extensible markup language*), el marco de descripción de recursos (RDF del inglés *Resource Description Framework*) y el lenguaje de ontologías web (OWL del inglés *Ontology Web Language*), Yu (2011). El primero proporciona un lenguaje de marcado de datos universal, el segundo es propiamente un subconjunto de XML especializado para la descripción de recursos de información y el tercero proporciona un medio para la definición de vocabularios formales (ontologías) para emplear en la descripción de recursos.

En palabras llanas, para escribir información en la Web Semántica, primero definimos un vocabulario formal (ontología) o bien tomamos uno que ya haya sido creado en alguna parte del mundo. El vocabulario nos proveerá de un conjunto de conceptos acerca de los cuales hablar y de las propiedades que de ellos podemos referenciar. Este conjunto de vocabularios está escrito en OWL o RDFS, después, para escribir enunciados acerca de dichos conceptos y propiedades emplearemos construcciones en formato RDF. El que sigue es un fragmento de código RDF:

```
<rdf:Description rdf:about="#Lizbeth">
  <rdf:type rdf:resource="#Honorarios"/>
  <mipersonal:tiene_de_jefe rdf:resource="#Vicente"/>
  <mipersonal:salario>15600</mipersonal:salario>
</rdf:Description>
```

En el código se describe a “Lizbeth” que es profesora por “Honorarios, cuyo jefe es “Vicente” y tiene un salario de 15600. Estamos hablando de 3 enunciados o tripletas dispuestas de la siguiente manera: <Lizbeth, type , Honorarios>, <Lizbeth, tiene\_de\_jefe , Vicente>, < Lizbeth,salario,15600>. Cuando se almacenan enunciados masivamente, suele darse el nombre a ese banco de tripletas como triple-store, esto es almacén de tripletas, análogo a “base de datos”.

Una vez que hemos escrito oraciones en formato RDF la gran ventaja es que podremos llevar a cabo extracción de información empleando para ello el estándar de consulta denominado SPARQL, Yu (2011). Propiamente, a través del lenguaje de consulta podremos extraer conocimiento preciso, sin ambigüedad. Es como tener acceso a la Base de Datos completa de la organización, pero, sin requerir una interfaz funcional (API) como ocurre actualmente en la mayoría de los sitios de Internet.

La propuesta sugiere la creación de un compilador para producir etiquetas formales, si bien crear un procesador formal de lenguajes es una tarea sofisticada, también es cierto que hoy en día hay herramientas maduras para su generación como, por ejemplo: ANTLR, Parr(2013).

## Modelo de IRS con base en compilador

En la Figura 1, se puede observar la posición del compilador cuya tarea es en dos sentidos: a) producir código RDF a partir de oraciones en lenguaje natural, que podrían provenir de una base de datos y b) compilar código RDF provisto por un usuario. En ambos casos lo que se pretende es contar con sentencias formales, es decir, sentencias correctas que puedan ser consultadas por cualquier usuario en el mundo a través del lenguaje de consulta SPARQL, Yu (2011). La función del compilador también es simplificar la generación de sentencias formales y con ello alentar la masificación de la codificación semántica de la información.

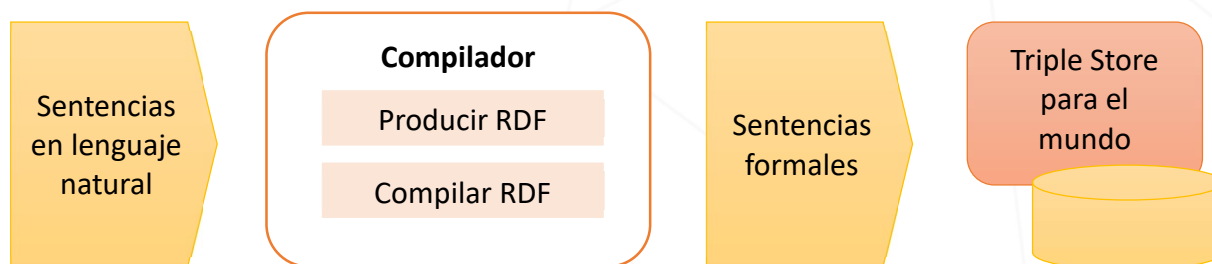


Figura 1: Modelo de Sistema de Recuperación de Información con compilador

En la Figura 2, se observa la definición de la gramática para producir el código RDF de la Sección de antecedentes

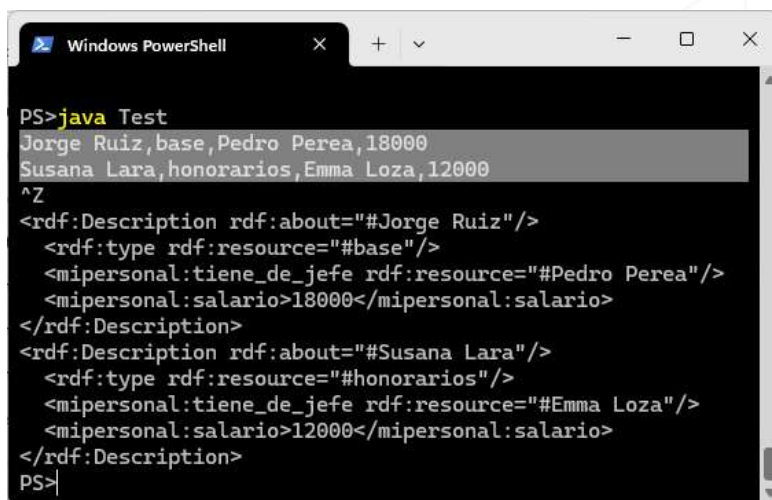
```

grammar compilador;
rule: employee+ ;
employee:
  t1=STR {System.out.println("<rdf:Description rdf:about=\"#" + $t1.text + "\"/>");}
  COMMA
  type {System.out.println("  <rdf:type rdf:resource=\"#" + $type.text + "\"/>");}
  COMMA
  t2=STR {System.out.println("  <mipersonal:tiene_de_jefe rdf:resource=\"#" + $t2.text + "\"/>");}
  COMMA
  salary {System.out.println("  <mipersonal:salario>" + $salary.text + "</mipersonal:salario>"
    + "\n" + "</rdf:Description>");};
  
```

Figura 2: Gramática para generar el compilador del ejemplo

A continuación, en la Figura 3, se observa la ejecución del compilador en el que se asigna como entrada dos tuplas de datos y se obtiene como salida el código RDF.





```

PS>java Test
Jorge Ruiz,base,Pedro Perea,18000
Susana Lara,honorarios,Emma Loza,12000
^Z
<rdf:Description rdf:about="#Jorge Ruiz"/>
<rdf:type rdf:resource="#base"/>
<mipersonal:tiene_de_jefe rdf:resource="#Pedro Perea"/>
<mipersonal:salario>18000</mipersonal:salario>
</rdf:Description>
<rdf:Description rdf:about="#Susana Lara"/>
<rdf:type rdf:resource="#honorarios"/>
<mipersonal:tiene_de_jefe rdf:resource="#Emma Loza"/>
<mipersonal:salario>12000</mipersonal:salario>
</rdf:Description>
PS>
  
```

Figura 3: Ejecución del compilador y código producido.

## Conclusiones

En este trabajo hemos planteado que es viable la construcción de un compilador que se puede adherir como un componente intermediario de un Sistema de Recuperación de Información cuya tarea es en dos sentidos, producir código en el lenguaje formal RDF a partir de sentencias en lenguaje natural o bien, compilar código RDF que le sea provisto, de tal manera que, al garantizar código RDF correcto pueda ser colocado en un *triple store*, y desde ahí, contestar cualquier consulta que se pudiera plantear un usuario genérico en cualquier parte del mundo, favoreciendo con ello la posibilidad de explotación de la información y con ello el despliegue de “datos abiertos”.

## Referencias

- Gobierno de México (2022), Datos abiertos del Gobierno de México, <https://datos.gob.mx/>
- Open Data Charter, (2015) <https://opendatacharter.net/principles-es/>, consultado en enero de 2022
- Baeza-Yates, Ricardo, and Ribeiro-Neto, Berthier, (2011), Modern Information Retrieval, Addison Wesley.
- Parr, Terence (2013), The Definitive ANTLR 4 Reference, Pragmatic Bookshelf, 2nd edition
- Yu, L., (2011), A Developer's Guide to the Semantic Web. Springer