



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Intercambio de Información y Web
Semántico

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:
DAVID MÉNDEZ LÓPEZ

DIRECTOR DE TESIS:
DR. SERGIO RAJSBAUM GORODEZKY



2010



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Hoja de Datos del Jurado

1. Datos del alumno Méndez López David 55 93 44 92 Universidad Nacional Autónoma de México Facultad de Ciencias Ciencias de la Computación 302017344
2. Datos del tutor Dr. Sergio Rajsbaum Gorodezky
3. Datos del Sinodal 1 Mat. Mónica Leñero Padierna
4. Datos del Sinodal 2 L. en C.C. Francisco Lorenzo Solsona Cruz
5. Datos del Sinodal 3 Dra. Elisa Viso Gurovich
6. Datos del Sinodal 4 Dr. Víctor Hugo Jacobo Armendáriz
7. Datos del trabajo escrito Intercambio de Información y Web Semántico 33 páginas 2010

*A mi familia, mis amigos y en especial a mi madre
Patricia López Zaragoza Q.E.P.D.*

Índice general

1. Introducción	5
1.1. Antecedentes	5
1.2. Objetivo	6
1.3. El Web Semántico y su surgimiento	6
1.4. Importancia de el Web Semántico	7
1.5. Tecnologías afines y definiciones	8
2. Desarrollo	13
2.1. Vocabularios	13
2.2. Plataformas de desarrollo	15
2.3. Programas y demostraciones	15
2.4. Productos finales	16
3. Manual de los productos	18
3.1. Rellenador de formas	18
3.1.1. Instalación	18
3.1.2. Funcionamiento	18
3.2. Generador de generadores	21
3.2.1. Instalación	21
3.2.2. Funcionamiento	21
3.3. Tipo de contenido Plone	26
3.3.1. Instalación	26
3.3.2. Funcionamiento	26
4. Conclusiones	30
4.1. Experiencias y aprendizaje	30
4.2. Conclusiones	31

Capítulo 1

Introducción

Durante el semestre 2010-1, que abarca de agosto a noviembre de 2009, se realizó el seminario de titulación llamado *Intercambio de Información y Web Semántico* impartido por el Dr. Sergio Rajsbaum. Tiene como objetivo crear un protocolo y un prototipo de su implementación para un sistema de comunicación entre distintas dependencias de la UNAM, usando las tecnologías de la Red Semántica. Entre las actividades realizadas se encuentran: la investigación sobre las bases y las tecnologías que constituyen a el Web Semántico, análisis de los problemas que se pueden resolver con el Web Semántico, análisis y evaluación de cómo montar las tecnologías sobre los sistemas de información ya existentes, pruebas de concepto sobre el funcionamiento y la utilización de estas distintas tecnologías, y el desarrollo de un producto utilizable para las instituciones de la UNAM, que resuelva o atienda alguno de los problemas de los sistemas de información usados actualmente en la UNAM. El seminario forma parte de uno de los proyectos del PAPIIT *Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica*, el cual lleva como nombre el mismo del seminario.

1.1. Antecedentes

En la UNAM existe un problema con la manera en que se administran los datos sobre el personal que labora en ella. Cada dependencia de la UNAM utiliza su propio sistema de manejo de información, ya sea personal académico o personal administrativo. En este ámbito hay un intercambio de información poco eficiente o nulo, lo que acarrea problemas como inconsistencia y replicación de datos. Así, una persona y muchos de sus datos se repiten a través de distintos sistemas, y en algunas ocasiones con información no actualizada. Otro problema muy recurrente que se puede solucionar fácilmente, es la práctica constante de que las personas llenen una y otra vez formas con la misma información. Datos como nombre, apellidos, teléfono, correo electrónico, son ingresados una y otra vez a bases de datos y son tecleados por las personas que quieren acceder a alguna institución. En el seminario de titulación queremos dar una solución a este

problema.

Ha habido distintos proyectos para centralizar los datos del personal de la UNAM. Por ejemplo, el RUPA (Registro Único de Personal Académico) creado por la DGAPA (Dirección General de Asuntos de Personal Académico). Estos proyectos tienen una manera de actualizar datos nada automatizada, que involucra instalar la base de datos en una computadora de ciertas características y luego mandar esa computadora a la DGAPA. Esto, en mi opinión, es un método anticuado para una tarea de esta magnitud. También si se necesita hacer una actualización de esa información la tarea se hace más bien un proceso burocrático y no uno automatizable¹.

1.2. Objetivo

El objetivo de mi participación en este seminario es colaborar con el diseño e implementación del protocolo, elaborando un prototipo que comunicará a la Facultad de Ciencias y al Instituto de Matemáticas intercambiando datos curriculares y cursos impartidos por los profesores.

Esta aplicación busca demostrar las distintas tecnologías que nos brinda el desarrollo de el Web Semántico y dar formas en las que se pueden utilizar para lograr una comunicación entre dos sistemas de información. También se busca evaluarlas en cuanto a desempeño y facilidad de aplicación, para un uso a nivel inter-institucional.

Como segunda parte se propone colaborar con el personal de DGAPA en el proyecto de intercambio de información y elaborar aplicaciones usando tecnologías de el Web Semántico.

1.3. El Web Semántico y su surgimiento

En los últimos 20 años la red de computadoras llamada Internet ha creado una gigantesca cantidad de datos que se comparten y se almacenan. Con casi mil quinientos millones de usuarios descargando y cargando varios miles de Megabytes de datos, el crecimiento de la cantidad de datos es cada segundo mayor. Crece aproximadamente un 100 % cada año [2]. El impacto que ha tenido en la sociedad el uso de esta gran red ha cambiado la forma en que la gente busca información, los profesores enseñan a sus alumnos, la sociedad interactúa consigo misma y sobre todo la rapidez con que la información sobre eventos, personas y conocimientos se puede dar a conocer.

Pero pasa algo con esta gran cantidad de datos que crece cada día más y más. Las conexiones que tenemos son solamente de computadoras y documentos, no de datos. No hay nada que me pueda decir que el archivo de imagen en donde aparece una persona sea una foto de esa persona. Eso lo podemos inferir nosotros como personas si es que vemos la imagen y conocemos a la persona, o si dicha foto está en una página de Internet y hay una etiqueta con el nombre de la persona. Pero no hay manera para que un programa lo sepa.

¹Ver oficio DGAP/1526/2009

El Web Semántico es exactamente eso, la forma en que se crean las conexiones entre los datos. La primera tarea fue crear el lenguaje con el cual se podría crear un estándar con el cual estos datos pudieran conectarse. Partiendo del XML (*Extensible Markup Language*, <http://www.w3.org/TR/REC-xml>) se creó el RDFS (*Resource Description Framework Schema*) un lenguaje con el que es posible crear relaciones entre distintos entes y valores primitivos ya predefinidos. A partir del RDFS surgió el OWL (*Web Ontology Language*) el cual es un lenguaje que se utiliza para compartir a través de la red, ontologías o vocabularios que describen objetos o datos. Usando OWL varias organizaciones, grupos y personas han empezado a generar los vocabularios que se van a utilizar para describir los datos. Estos vocabularios pueden describir desde personas (*FOAF*)², hasta organizaciones (*Wordnet Organization*)³, negocios (*SWEET Human Activities*)⁴, vinos (*Wine Ontology*)⁵, cervezas (*Beer Ontology*)⁶ o aun ataques terroristas (*Terrorist Attack Types*)⁷. El esfuerzo para crear y publicar todo tipo de vocabularios o esquemas esta recopilado por www.schemaweb.info, un sitio dedicado a la publicación y difusión de esquemas, que busca ser el directorio general para todos los esquemas de descripción de datos que se quieran utilizar para la construcción de el Web Semántico. Teniendo un lenguaje en común y estandarizado ya podemos empezar a recorrer esta red de datos. No necesitamos crear en sí las conexiones, ya que el lenguaje RDF las crea por nosotros. Al describir un elemento o un dato implícitamente estamos haciendo la conexión entre dos recursos a través de la red. Esto se verá a detalle más adelante.

1.4. Importancia de el Web Semántico

Vivimos en un mundo donde se generan cada día cantidades gigantescas de datos. Desde los datos generados por un estudio estadístico o científico hasta el hecho de que alguna persona generó una base de datos de una pastelería. En la actualidad es muy fácil publicar documentos, imágenes, artículos, etc. pero el significado y la utilidad potencial de estas cosas pasa desapercibido. Pero ¿qué pasa cuando toda esta información la publicamos en un formato estándar y público? Podemos utilizar estos datos en formas que el creador o el que publicó estos datos jamás pensó.

En los últimos años ha venido en aumento el apoyo a que la gente publique sus datos, los estandarice y desarrolle aplicaciones que utilicen estos datos de manera interesante. Todo comenzó a partir de la publicación del Scientific American de mayo de 2001. En el artículo *The Semántic Web*[3] Tim Berners-Lee plantea los fundamentos, beneficios y ventajas que tendría una red de datos. Sobre todo plantea lo rápido y lo poco costoso que es conectarse a la red de datos.

No mucho tiempo después los gobiernos y universidades han empezado a darse cuenta

²<http://xmlns.com/foaf/0.1>

³<http://xmlns.com/wordnet/1.6/>

⁴http://sweet.jpl.nasa.gov/ontology/human_activities.owl

⁵ <http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine>

⁶<http://www.purl.org/net/ontology/beer>

⁷ <http://reliant.teknowledge.com/DAML/terroristAttackTypes.daml>

del potencial que existe en estas tecnologías y se ha empezado dar fondos para la investigación y desarrollo. Por ejemplo: el proyecto de la agencia estadounidense DARPA (*Defense Advanced Research Projects Agency*) llamado DAML (*DARPA Agent Markup Language*) que costó 45 millones de dólares, desarrolló varias tecnologías de donde se fundamenta el Web Semántico.[4] De este proyecto surgió la tecnología que después se convertiría en OWL, la cual explicaremos más adelante. En la actualidad varios proyectos de el Web Semántico en Estados Unidos son financiados por la **NSF National Science Foundation**.

La mayor importancia que tiene el Web Semántico es el cambio de la manera de pensar sobre los datos. Antes los datos se navegaban por donde o como estaban almacenados, ahora con el Web Semántico los datos están relacionados por su significado. En la UNAM la importancia radica no sólo en que se utilice para comunicar instituciones, sino para hacer algo útil con datos de personas, cursos, presupuestos, etc. Poder hacer cosas como identificar causas de problemas, correlaciones o comparar con otros conjuntos de datos fuera de la UNAM.

1.5. Tecnologías afines y definiciones

Ahora daremos una breve descripción de las tecnologías en las que se monta el Web Semántico, así como términos y acrónimos usados en este contexto.

La tecnología más importante y la más indispensable es el concepto de **URI** (*Uniform Resource Identifier*). El URI es una forma de crear identificadores usando cadenas de caracteres para recursos, ya sean abstractos o físicos, de manera que sean únicas. Es decir que un URI nunca puede usarse para identificar a dos elementos distintos. Está definido en el RFC (*Request for Comments*) 3986 [5] de 2005, actualizando el RFC original, el 1808, escrito en 1995 por R.Fielding. Un URI consta de 5 partes:

- **Esquema:** Nombre que se refiere a un espacio de nombres o al protocolo de acceso al recurso. E.g. *ftp*:, *http*:, *mailto*:.
- **Autoridad:** Elemento jerárquico que identifica a la parte que asigna los nombres. E.g. (*//www.worldofwarcraft.com*).
- **Ruta:** Cadena jerárquica de caracteres que identifica al recurso dentro de la autoridad. E.g. (*/classes/druid/intro.html*).
- **Consulta:** Cadena no jerárquica que da detalles sobre cómo debe ser accedido el recurso, inicia con el carácter '?', el famoso QueryString. E.g. (*?value=6780*).
- **Fragmento:** Identificador de una parte del recurso principal. Indicado por el carácter '#'

Técnicamente se utiliza un subconjunto de URI que es el **IRI** (*Internationalized Resource Identifier*)⁸ y que deja permitir caracteres internacionales (arabigos, cirílico, japonés,etc.).

⁸<http://www.w3.org/International/iri-edit/draft-duerst-iri-04>

La importancia de los URI recae en la unicidad que tienen, nos da una base para tener un lenguaje con elementos no ambiguos.

La siguiente tecnología en orden de importancia es el XML (*eXtensible Markup Language*). XML es un conjunto de reglas de codificación de documentos que serán compartidos de manera electrónica. Este estándar fue creado por el W3C (*World Wide Web Consortium*) en 1996. El diseño de XML tenía como objetivo hacer un lenguaje simple, general y que pudiera usarse ampliamente a través del Internet. Poco a poco también se convirtió en una manera de representar estructuras de datos arbitrarias gracias al concepto de DTD (*Document Type Definition*), lo cual impone reglas estrictas de validación a los documentos XML[6]. Un documento en XML tiene la siguiente estructura:

- **Declaración:** una cadena que nos dice que el documento es un documento XML.

```
<?xml version=1.0 encoding=UTF-8 ?>
```

- **Etiqueta:** es una cadena que empieza con el carácter '`<`' y termina con '`>`'. Hay de tres estilos: de inicio, como `<elemento>`, de cierre, como `</elemento>`, y la etiqueta vacía, como `<aquiacaba/>`.
- **Elemento:** Es un componente del documento que se encuentra entre una etiqueta de inicio y una de cierre. Esto incluye cualquier texto escrito entre las etiquetas, incluyendo otros elementos a los que se les llama Elementos hijo. En el caso de etiquetas vacías éstas también son Elementos.
- **Atributo:** es una parte de un Elemento formada por un pareja nombre/valor la cual se pone en una etiqueta de inicio o en una vacía. Por ejemplo

```
<img src='imagen.jpg' alt='dibujo'>
```

Como se puede ver, un documento XML nos da una estructura de Elementos anidados, lo que genera una estructura de árbol con Elementos como sus nodos. Algo muy importante es el hecho de que cada Elemento puede ser asociado a una URI. Por ejemplo: si tenemos en el archivo `http://servidor.org/archivos/ejemplo.xml` el siguiente elemento:

```
<seccion>... </seccion>
```

Podemos acceder a ese elemento con el siguiente URI:

```
http://servidor.org/archivos/ejemplo.xml#seccion
```

Otra cosa que soporta XML es el hecho de manejar espacios de nombres de las etiquetas. Es decir, que se puede especificar una dirección URI de otro documento XML y hacer referencia a elementos que existen en ese documento de manera remota. Esto es muy importante ya que nos ayuda a tener una base para la siguiente tecnología. el RDF. **RDF** (*Resource Description Framework*) es una metodología para describir objetos y sus relaciones. RDF describe las cosas a partir de Enunciados en forma de tripletas de la siguiente manera:

Sujeto — Predicado — Objeto

Lo cual crea una lógica donde la tripleta nos dice que el sujeto está relacionado con el objeto de manera dada por el predicado. Como esta lógica crea una lógica de descripción tenemos estructuras y conceptos como Clase, Herencia y Propiedades. Juntando estos conceptos e implementandolos con XML y las URIs es que se construye la base de lo que es el Web Semántico, el RDF/XML, en donde cada tripleta se expresa como elementos de un archivo XML.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:contact="http://www.w3c.org/2000/10/swap/pim/contact">

    <contact:Person rdf:about="http://organizacion.org/contactos/johny.html">
        <contact:fullName>Johny Ramirez Ortiz</contact:fullName>
    </contact:Person>
</rdf:RDF>
```

En el parrafo anterior hay dos tripletas, la primera es

```
<http://organizacion.org/contactos/johny.html> | es un | contact:Person. La segunda es
<http://organizacion.org/contactos/johny.html> | contact:fullName | Johny Ramirez
Ortiz. Hay que remarcar que cada elemento de la tripleta es ya sea un URI o una literal de
XML; es decir, una cadena, un número o una fecha.
```

El RDF tiene distintas maneras de expresarse. Como RDF/XML no es muy legible para los humanos se crearon varias representaciones. La más usual es la notación N3⁹ y sus distintos subconjuntos (Turtle y N Triples) que cuentan con menos expresividad pero son más fáciles en su traducción a RDF/XML.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix contact: <http://www.w3c.org/2000/10/swap/pim/contact>.
```

```
http://organizacion.org/contactos/johnny.html      rdf:type
    http://www.w3c.org/2000/10/swap/pim/contact.
http://organizacion.org/contactos/johnny.html      contact:fullName
    Johny Ramirez Ortiz.
```

Teniendo como base esto, se desarrolló un vocabulario llamado RDF-Schema, que nos ayuda a generar otros vocabularios y poder hacer descripciones usando los vocabularios. En términos generales el RDF nos da la sintaxis del lenguaje y el RDF-Schema nos da la semántica.

⁹<http://www.w3.org/DesignIssues/Notation3.html>

Con este lenguaje es que describimos los datos que queremos compartir. La descripción abstracta de un tipo de datos se le llama **Ontología**. Por ejemplo, describir en general lo que constituye una persona:

- Nombre
- Apellidos
- Sexo
- domicilio
- teléfono
- correo electrónico

A la descripción de una persona en particular, apegandose a alguna Ontología previamente, creada se le llama **Ejemplar**. Por ejemplo, describir a una persona como

- Nombre = "Juanito"
- Apellidos = "B Belows"
- Sexo = "Masculino"
- domicilio = "Ciudad Madero"
- teléfono = "3287493"
- correo electrónico = "john@lol.com"

De manera abstracta una descripción en RDF crea conexión una entre datos usando URI, lo que crea una gráfica de datos llamada Gráfica de RDF. Con esto ya podemos recorrerla usando técnicas y algoritmos que recorren gráficas para poder buscar los datos que uno requiere. Afortunadamente ya existe un lenguaje de búsqueda y un protocolo que recorre la gráfica. El **SPARQL** (*SPARQL Protocol And RDF Query Language*)¹⁰ es un lenguaje y un protocolo creado por el *W3C RDF Data Access Working Group* con el fin de tener una interfaz estándar con la que pueden montarse todas las aplicaciones que quieran acceder a los datos dentro de el Web Semántico.

Para buscar una mayor expresividad, con los lenguajes se creó uno que nos da estructuras lógicas aun más poderosas. Este se llama **OWL** *Ontology Web Language* ¹¹ el cual parte de RDF y RDF-Schema y nos da expresiones más detalladas sobre los objetos que describimos. Por ejemplo, nos da la posibilidad de decir si algo pertenece o no a un conjunto, si algo es contrario a algo; si ciertas clases son excluyentes; las relaciones de equivalencia, etc. OWL tiene distintos niveles de expresividad, en donde con mayor expresividad se pierde poder computacional, es decir que en su forma más expresiva se

¹⁰ Acronimo recursivo como GNU

¹¹ <http://www.w3.org/TR/owl-features/>

pueden crear descripciones cíclicas. La única diferencia entre los tres niveles es qué tantas expresiones se pueden usar o no.

Una de las partes fundamentales de tener una red semántica es la posibilidad de poder inferir de manera lógica sobre los datos que conecta. Eso se debe al lenguaje estándar de descripción que usamos. Es por eso que se ha definido un lenguaje de creación de declaraciones RDF a partir de reglas lógicas llamado **SWRL** (*Semantic Web Rule Language*). Actualmente existen distintas implementaciones de este lenguaje, llamadas Razonadores. La mayoría de este trabajo fue realizado por la élite del W3C encabezado principalmente por Tim Berners Lee, uno de los creadores del HTTP (*HyperText Transfer Protocol*). Hay un grupo de trabajo por cada una de las tecnologías aquí expuestas, y cada una lleva más de 7 años en desarrollo e investigación, lo que implica que ya hay muchas aplicaciones concretas de todas estas tecnologías. Un ejemplo importante sería el proyecto **SIMILE** (*Semantic Interoperability of Metadata and Information in unLike Environments*) encabezado por el MIT. Este proyecto es una compilación de distintos proyectos que están dirigidos a crear y experimentar en todas estas tecnologías de intercambio y manejo de información. La documentación de todas estas tecnologías, lista de casos de uso de las tecnologías, así como noticias sobre los grupos de trabajo están recopiladas en el sitio *W3C Semantic Web Activity*¹² y una compilación de artículos y noticias sobre lo que sucede en el mundo sobre el Web Semántico se puede encontrar en *SemanticWeb.com*¹³.

¹²<http://www.w3.org/2001/sw/>

¹³<http://www.semanticweb.com>

Capítulo 2

Desarrollo

2.1. Vocabularios

El comienzo del proyecto involucró generar varias ontologías usando RDF como lenguaje de descripción. La información que queremos compartir es principalmente datos de académicos. Datos como nombre, apellido, edad, correo electrónico, número de expediente, proyectos en los que participa la persona, instituciones a las que pertenece, etc. Es por eso que definimos, después de varias juntas con el personal de cómputo de la DGAPA, las propiedades más relevantes que se manejan en las bases de datos de personal académico. Y con eso escribir las ontologías que describen a una persona que trabaja en la UNAM.

Para crear estos archivos nos valimos de una herramienta llamada *Protege*¹, un programa que autogenera ontologías y archivos de RDF de ejemplar apartir de una interfaz gráfica amistosa. Aunque todavía tiene problemas y dificultades, como por ejemplo al querer cambiar de URI donde tenemos nuestra ontología, o querer utilizar ontologías externas, donde tenemos que incluir todos los archivos de ontologías, este programa los pone de manera muy estorbosa en los archivos XML que generamos.

Nos dimos primero a la tarea de ver qué vocabularios se necesitarían para el proyecto. Se llegó a la conclusión de que queríamos hacer una extensión de *FOAF (Friend Of A Friend)* a la cual le añadiremos términos y datos que utilizan todas las personas que viven y trabajan en México. A esta ontología la llamamos FoafMexico. La ontología FOAF es la que más se usa para describir personas y generar perfiles. Incluye valores que queremos utilizar para describir a académicos, como son: nombre, correo electrónico, página Web, edad y teléfono. Para extender este vocabulario simplemente creamos una clase llamada foafMexico:PersonaMexico dentro del archivo FoafMexico y la hicimos subclase de foaf:Person. Aquí, le agregamos valores como primerApellido, segundoApellido, CURP, RFC y genero. Después, creamos un vocabulario enfocado a describir a personas que laboran en la UNAM. Esta ontología se llama personaUNAM, y tiene las siguientes clases: personaUNAM,

¹<http://protege.stanford.edu/>

que es una subclase de foafMexico:PersonaMexico, personaUNAM:investigador y personaUNAM:alumno que las dos heredan de personaUNAM. Aquí agregamos detalles como el grado máximo de estudios, resumen curricular, entidad de la UNAM donde trabaja, número de expediente y en el caso de un alumno el número de cuenta. Decidimos hacer los vocabularios de esta manera ya que así seguimos el espíritu de la red semántica el cual es uno de reutilización y universalidad del lenguaje de descripción, y con esto contamos con la ventaja de que un solo archivo RDF de una persona puede ser utilizado por, distintos sistemas aparte del nuestro. Dado que una descripción en RDF no tiene por qué sólo ser de una única ontología, se puede describir a una persona utilizando varios vocabularios y éstos no interfieren con los demás ya que si hay un elemento XML que no se entiende éste se ignora.

Las primeras versiones del vocabulario se generaron en editor de texto en formato N-Triples, haciendo un análisis superficial de los datos que se necesitan para describir a una persona. Esta versión se usó sólo para una presentación. La segunda versión tuvo modificaciones y agregamos campos después de platicar con el personal de DGAPA y utilizar el análisis que hicieron para crear su base de datos de personal académico.

A partir de la tercera versión empezamos a utilizar Protege, ya que los traductores de N-triples a RDF/XML no funcionaban de manera adecuada, y no se traducían de manera completa la descripción de las cosas. Usando la interfaz gráfica de Protege fácilmente creamos las clases y las propiedades de personaMexico y personaUNAM con prácticamente la misma estructura de la versión dos.

En la cuarta versión agregamos aun más propiedades a las clases personaUNAM, varias de las cuales se piden en el sitio de registro de proyectos que en este momento esta contruyendo la DGAPA. Agregamos la clase Direccion así como las propiedades resideEn y trabajaEn. A personaUNAM agregamos las propiedades fechaDeNacimiento y estadoCivil. A los investigadores les agregamos la propiedad nombramientoAcademico, y a los alumnos agregamos las propiedades ultimoSemestreCursado, porcentajeCreditos y promedio.

En cuanto a la parte del proyecto que involucra a la Facultad de Ciencias, creamos una ontología bastante amplia. Después de discutir con los encargados del sistema de información de la facultad, creamos la ontología mapeando más o menos la estructura de información de los cursos impartidos por la facultad. Sólo modificamos aspectos que en un ambiente semántico orientado a objetos no tendrían sentido (i.e. Profesor fuera propiedad de un Horario).

La ontología Curso consiste en 5 clases: Asignatura, Grupo, Clase, Horario y Plan de Estudios. La estructura es la siguiente: Asignatura guarda datos como nombre de la asignatura, clave de la asignatura, clave del plantel, etc. Esta clase se relaciona con Grupo con la propiedad selmparte; aquí es donde se guarda la información del grupo en donde se imparte la asignatura, como clave del grupo, periodo en que se imparte. La clase Grupo se relaciona con la clase Clase que a su vez se relaciona con la clase PersonaUNAM y la clase Horario, esto es para decir que una clase tiene un encargado, ya sea ayudante, profesor titular, etc. y que cada uno tiene un horario al que asiste. La clase Plan de Estudios se relaciona con Asignatura, ya que tiene información sobre la carrera en que se imparten las asignaturas y el año en que fue creada.

2.2. Plataformas de desarrollo

Durante el transcurso del seminario desarrollamos los programas, demostraciones y pruebas de concepto de dos distintas maneras. La primera fue mediante Java y su tecnología para desarrollo de aplicaciones Web Java Servlet Pages. En algunas partes se utilizaron scripts de Javascript que facilitaron el reconocimiento de XML y el manejo de eventos del lado del cliente (e.g. click del mouse, recarga de página, etc.). Utilizamos para la creación de código en Java y jsp el IDE (*Integrated Design Environment*) Netbeans 6.7. Para los scripts de Javascript se utilizó Aptana Studio 2.0 y el *plugin* de Firefox llamado Firebug 1.4 para probar y depurar el código. En cuanto al manejo de RDF utilizamos el *framework* para web semántico llamado JENA. Este framework incluye implementaciones de distintas tecnologías de la red semántica tal como generación de RDF, conversión entre N3 y RDF/XML, manejo de consultas de SPARQL, razonadores y manejo del lenguaje SWRL. Para fines del seminario sólo utilizamos el motor de SPARQL. El motivo por el cual decidimos utilizar esta estrategia es por nuestra experiencia previa en el desarrollo de aplicaciones en Java. Por tanto fue muy rápido y ágil construir las herramientas y las demostraciones.

La segunda manera en que desarrollamos fue utilizando el CMS (*Content Management System*) Plone 3, el cual es un manejador de contenido que sirve para publicar páginas web, blogs, documentos, etc. Plone 3 trabaja sobre un servidor de aplicaciones llamado Zope (*Z Object Publishing Environment*) el cual está escrito en el lenguaje de programación Python. Por tanto el desarrollo en este ambiente involucra escribir scripts en Python y utilizar para la creación y modificación de interfaces los ZPT (*Zope Page Templates*) y sus lenguaje para crearlos, TAL (*Template Attribute Language*) y METAL (*Macro Expansion Template Attribute Language*). Para el manejo de archivos RDF utilizamos la biblioteca de Python rdflib que al igual que JENA trae manejo de consultas SPARQL, creación y parseo de RDF/XML y N3. Esta plataforma de desarrollo se eligió ya que el sistema de información del Instituto de Matemáticas esta construido con Plone y Zope, y esto requiere un enfoque distinto al tradicional y por tanto un desarrollo totalmente distinto.

2.3. Programas y demostraciones

En la segunda semana de septiembre se elaboró un pequeño programa de demostración, el cual rellenaba una forma HTML con información obtenida especificando un archivo RDF que estuviera en la red y que cumpliera con el vocabulario FOAF. Los archivos de ejemplo se generaron con Protege y fueron publicados en un servidor Apache. La aplicación se encuentra en un proyecto de netbeans dentro de un archivo zip en la siguiente URL

<http://proyectos.matem.unam.mx:8085/papiit/aplicaciones/demos/RellenarForma.zip>

Después, este mismo demo se adaptó para que funcionara con archivos RDF que cumplieran nuestros vocabularios. Esta aplicación fue una prueba de concepto de una de las funciones que el personal de DGAPA estaba interesado en implementar en su sistema.

Se utilizó la tecnología de JSP por lo cual se necesita montar en un servidor *Tomcat* o *Glassfish* para que funcione.

El segundo programa fue una aplicación web generadora de ejemplares de RDF. Este funciona a partir de rellenar una forma y al dar *submit* te da un archivo RDF de ejemplar de persona UNAM:investigador con los valores que damos en la forma.

Una semana después creamos algo más complicado y útil. Ampliamos este mismo programa y creamos una aplicación web que a partir de cualquier archivo de ejemplar, reconoce las clases y da a escoger qué clase y que propiedades queremos. Esto nos regresa un generador de archivos RDF de ejemplar para la clase que escogimos con los campos de propiedad que también escogimos. Esta fue una aplicación que surgió como parte de una idea de simplificar la creación de ejemplares de nuestros vocabularios, tanto para que lo usen personas de las instituciones participantes o instituciones que eventualmente quieran unirse al proyecto. También se decidió que el desarrollo de esta idea se dejaría para más adelante en el proyecto.

De la misma manera en el relleno de formas se utilizó JSP y por tanto también se necesita que este instalado en un servidor *Tomcat* o *Glassfish*.

Estas aplicaciones también se encuentran en el sitio en las siguientes URLs:

- <http://proyectos.matem.unam.mx:8085/papiit/aplicaciones/demos/RDFgenerator.zip>
- <http://proyectos.matem.unam.mx:8085/papiit/aplicaciones/demos/RDFgeneratorPro.zip>

Por el lado de Plone también desarrollamos un tipo de datos que genera un archivo de ejemplar de RDF a partir de un llenado de forma. Esto aunque sencillo requirió más esfuerzo que los programas pasados. Esto es ya que no solo involucra programar en python sino también aprender una tecnología distinta de publicación de datos. Y sobre todo la falta de un estándar a la hora de hacer cosas como por ejemplo: formas, intercambio de archivos, interfaces gráficas, etc. Es decir hay muchas maneras de hacer casi cualquier cosa y es un razonamiento muy distinto a la programación Web tradicional. Al final utilizamos los paquetes *z3c.form* y *ArchGen* para generar nuestro producto en Plone. El desarrollo del producto de Plone fue de lo más problemático y de lo más tardado. Esto fue debido a que la manera de desarrollar aplicaciones web en Plone es muy distinta a la manera tradicional, la cual tiene una pendiente de aprendizaje demasiado elevada. Esto implicó que varias funciones que queríamos no fueran realizadas por falta de tiempo.

2.4. Productos finales

Se entregan al final de este seminario aplicaciones que juntan todos los conocimientos adquiridos al experimentar y desarrollar los demos ya mencionados. Estas aplicaciones están enfocadas a mostrar la utilidad de esta tecnología para resolver problemas simples pero en ocasiones molestos.

La primera aplicación que hicimos fue la de implementar un producto de Plone el cual facilite la generación y la publicación de los RDF dentro de un sitio de Plone; en especial generar RDF que tengan la estructura de la información de los investigadores del Instituto de Matemáticas, y a la vez saque cierta información del sistema de la Facultad de Ciencias. Lo que se quiere lograr con este producto es que se muestre de manera dinámica la información de los cursos que da un investigador del Instituto de Matemáticas en la Facultad de Ciencias. Con esta aplicación que publica los RDF/XML, los programadores de la DGAPA ya tendrán acceso a la información de los investigadores del Instituto de Matemáticas. Mejor aún, los investigadores ya tendrán publicado un archivo RDF con un url el cual le podrán agregar datos sobre sus personas utilizando otros vocabularios. Como parte importante del seminario también se desarrolló el sitio donde se documenta todas las actividades del proyecto de intercambio de información académica, en donde documentamos y publicamos los vocabularios generados, los demos y los productos. También aquí se incluye ligas a sitios y documentos importantes sobre la red semántica, ligas de referencias hacia cosas que se utilizaron para desarrollar las pruebas, así como lo que se utilizó como bibliografía y base teórica. Otra cosa que contiene este sitio son las distintas presentaciones que se realizaron durante el transcurso del proyecto, así como documentos importantes como este trabajo en su versión digital. La liga es la siguiente:

<http://proyectos.matem.unam.mx:8085/papiit>

Proyecto de Intercambio de In...

Buscar en el Sitio buscar
 solo en la sección actual

Universidad Nacional Autónoma de México

Inicio vocabularios documentación aplicaciones enlaces discusiones miembros

Proyecto de Intercambio de Información Académica

usted está aquí: inicio

Inicio

Página principal del proyecto de intercambio de información. Y página que recopila el trabajo hecho durante el seminario de titulación: Intercambio de Información y Web Semántico. Dirigido por el Dr. Sergio Rajsbaum y en donde participan: David Méndez López y Hugo Rodríguez Peña.

En este sitio encontrarán todos los productos y documentos desarrollados en el transcurso del semestre 2010-1. Se encuentran tanto las aplicaciones web de pruebas de concepto como los vocabularios y ejemplos de uso.

entrar
Nombre de Usuario
Contraseña
entrar
¿Ha olvidado su contraseña?
¿Nuevo usuario?

Capítulo 3

Manual de los productos

En este capítulo se verá paso a paso instrucciones para instalar y utilizar los productos creados durante el seminario. Los productos que consideré más importantes fueron: el rellenedor de formas, el generador de generadores de RDF/XML y el tipo de contenido de Plone que genera y publica RDF/XML. Esta sección es para las personas que estén interesadas en entender el funcionamiento de los productos que elaboramos y ver ejemplos de la implementación y el uso básico de las tecnologías de la red semántica.

3.1. Rellenador de formas

3.1.1. Instalación

Este producto fue generado utilizando Java Servlet Pages y por tanto debe ser instalado dentro de un servidor que soporte la plataforma J2EE versión 1.4; por ejemplo Tomcat versión 5 en adelante, el nuevo servidor Glassfish Enterprise v3 o J2EE SDK versión 1.4 en adelante. Se puede descargar el archivo war (*Web application ARchive*), listo para ser instalado en un servidor en la siguiente dirección:

<http://proyectos.matem.unam.mx:8085/papiit/aplicaciones/demos/rellenador-de-formas-usando-rdf/Prueba.war>

Dependiendo del servidor es el lugar en dónde se coloca el archivo. Busque la documentación de su servidor favorito.

3.1.2. Funcionamiento

El funcionamiento es relativamente sencillo. El primer paso se hace al introducir una url válida de un archivo RDF de instancia, que contenga una descripción usando el vocabulario personaUNAM:investigador (*figura 1*), al dar click en el botón . Por dentro un objeto de la clase RDFUnam toma el url adquiere el archivo y, usando el

Rellenar información

A continuación puede rellenar los siguientes datos o puede ingresar el url de un archivo RDF.

Dirección RDF

- **Grado Máximo de Estudios**
- **Nombre**
- **Primer Apellido**
- **Segundo Apellido**
- **Teléfono**
- **Correo**
- **Homepage**
- **CURP**
- **Institución Actualmente Adscrita**
- **Resumen Curricular**

Figura 3.1: En el primer campo se escribe la url

framework de JENA, genera un modelo y hace consultas de SPARQL para extraer la información de los campos.

Estos datos se guardan dentro de atributos de la clase RDFUnam y se regresan utilizando sus respectivos métodos *getter*. Después, nuestro index.jsp saca los valores del objeto RDFUnam y rellena la forma. Sólo rellena los campos que encontró dentro del RDF (figura 3.2).

El segundo botón de la forma no está implementado ya que solo se muestra una prueba de concepto de cómo utilizar RDF para rellenar automáticamente una forma.

Rellenar información

A continuación puede rellenar los siguientes datos o puede ingresar el url de un archivo RDF.

Dirección RDF

- **Grado Máximo de Estudios**
- **Nombre**
- **Primer Apellido**
- **Segundo Apellido**
- **Teléfono**
- **Correo**
- **Homepage**
- **CURP**
- **Institución Actualmente Adscrita**
- **Resumen Curricular**

Figura 3.2: Vemos cómo los campos de la forma se rellenan

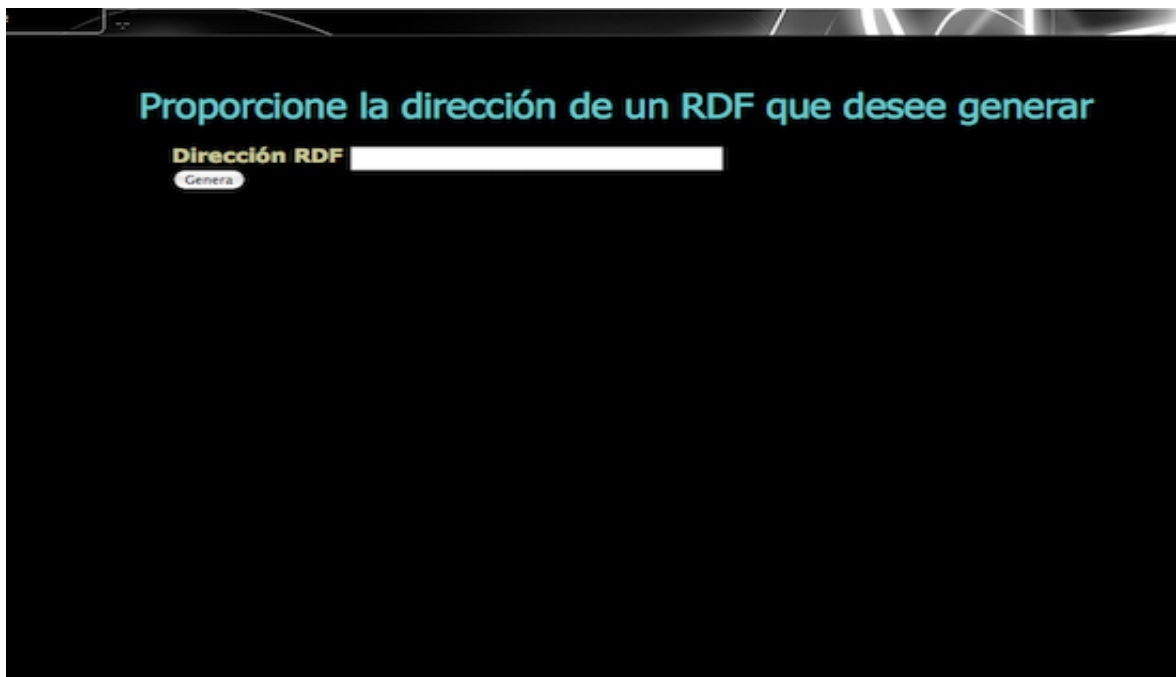


Figura 3.3: Otra vez se ingresa la url

3.2. Generador de generadores

3.2.1. Instalación

Al igual que el rellanador de formas, esta aplicación se generó utilizando tecnología JSP. Por tanto, de la misma manera necesita un servidor que soporte la plataforma J2EE 1.4. El archivo war se encuentra en la siguiente liga:

<http://proyectos.matem.unam.mx:8085/papiit/aplicaciones/demos/generador-de-generadores/RDFgeneratorPro.war>

3.2.2. Funcionamiento

Lo que hace esta aplicación es, al igual que el rellanador, tomar un archivo de instancia en RDF/XML, pero al contrario del rellanador, puede ser una instancia de cualquier vocabulario. Igual comenzamos introduciendo la url de un RDF de instancia en XML, en el campo dirección RDF (figura 3.3).

Esa url la toma un script de javascript, que analiza el RDF y ve que clases se describen y que propiedades tiene cada clase. Después se genera una forma la cual te da la opción de qué clase y con que propiedades quieres hacer tu generador de RDF (figura 3.4). También

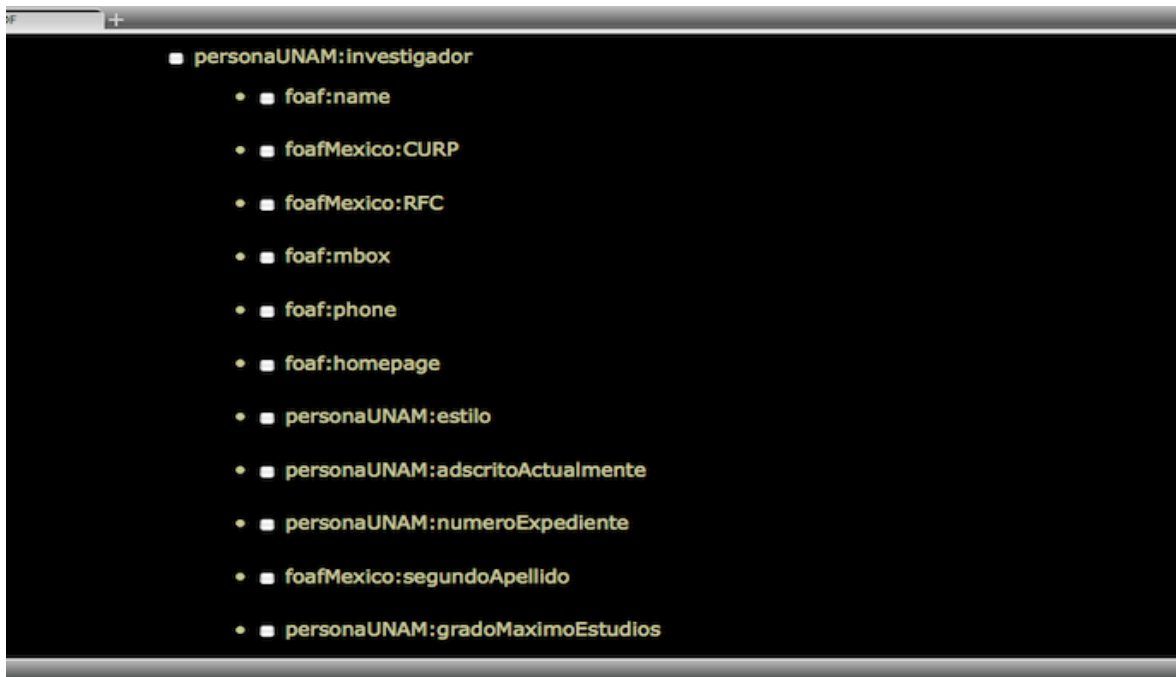


Figura 3.4: Seleccionamos la clase y las propiedades

se analiza los espacios de nombres definidos en el RDF y se ponen en campos escondidos de la forma.

La información de esa forma pasa al archivo generaxml.jsp, el cual genera un XML que nos dice cuales son los espacios de nombre, el tipo de la instancia que generaremos y que propiedades tendrá. Colocamos el XML en un area de texto para que el usuario confirme lo que quiere generar (figura 3.5).



Figura 3.5: Vemos el XML generado y confirmamos que esté correcto

Después de confirmar se escribe el XML en un archivo temporal y pasamos a `generador.jsp`. Éste lee el archivo y primero mete los espacios de nombre y el nombre de la clase en campos escondidos de una nueva forma. Después se generan campos de entrada de acuerdo al nombre de la propiedad que se quiera insertar (figura 3.6). Esa forma manda sus datos al archivo `genera.jsp` el cual usa un objeto de la clase `GeneradorRDFPro` el cual paso a paso lee la forma y va generando su etiqueta en RDF/XML, por cada espacio de nombre, clase y propiedad. Al final escribe el RDF generado en un area de texto (figura 3.7).

Ingrese sus datos para crear un ejemplar de la clase RDF
personaUNAM:investigador

- foaf:name
- foafMexico:CURP
- foafMexico:RFC
- foafMexico:primerApellido

Figura 3.6: Se muestra la forma para introducir valores a las propiedades

Aquí está su RDF

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:foafMexico="http://proyectos.matem.unam.mx:8085/papiit/vocabularios/current/foafMexico/foafMexico.owl#"
xmlns:ontosem="http://morpheus.cs.umbc.edu/aksl/ontosem.owl#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:personaUNAM="http://proyectos.matem.unam.mx:8085/papiit/vocabularios/current/personaunam/personaUNAM.owl#"
xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:swrl="http://www.w3.org/2003/11/swrl#"
xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
xml:base="http://dummy.base"><!--Favor de poner aqui la URL de donde se vaya colocar el archivo-->
<personaUNAM:investigador rdf:about="me">
<foaf:name>David</foaf:name>
<foafMexico:CURP>MLJ890769</foafMexico:CURP>
<foafMexico:RFC>MLHLJ8998</foafMexico:RFC>
<foafMexico:primerApellido>MÃ©ndez</foafMexico:primerApellido>
</personaUNAM:investigador>
</rdf:RDF>
```

Figura 3.7: Aquí se muestra el RDF generado

3.3. Tipo de contenido Plone

3.3.1. Instalación

Éste es un producto para Plone, por tanto se necesita que este instalado dentro de una servidor de Zope 2. Necesita también tener instalado el paquete z3c.form. También requiere el uso de Python 2.4 ya que la biblioteca rdflib solo existe para esa versión. Lo primero que se debe hacer es poner el directorio `data.persona` dentro del directorio `src` del servidor de Zope. Después tenemos que modificar el archivo `buildout.cfg` e insertar las siguientes líneas en las siguientes secciones:

```
[buildout]
....
eggs= ...
    data.persona

develop= ...
src/data.persona

[instance]
...
zcml= ...
    data.persona
```

Después de hacer eso sólo se necesita darle `bin/buildout` a el servidor de Zope.

3.3.2. Funcionamiento

Este producto de Plone lo que hace es crear un tipo de dato `persona`, basándose en datos que tiene el vocabulario `personaUNAM:investigador`. Lo que tiene de diferente es que puede rellenar la forma para agregar `persona` con ayuda de un archivo RDF de instancia que use el vocabulario `personaUNAM:investigador`. Y también tiene la modalidad de generar el RDF de `personaUNAM:investigador` y darle un url dentro del sitio Plone. Dentro de la forma de agregar `persona` tenemos la opción de agregar una url de un archivo RDF (figura 3.8).

Al dar click en [Usar RDF](#) se manda la url a un script de Python que a la vez usa la biblioteca `rdflib` para hacer las consultas de SPARQL necesarios para sacar la información; a continuación rellena los campos de la forma (figura 3.9).

Al ya tener el tipo de dato creado, tenemos la opción de generar el `rdf` y publicarlo (figura 3.10), lo que manda a llamar un script de Python que genera un XML y lo guarda dentro de la base de datos de Zope, dandonos el url de dónde fue guardada.

Plone™

Home Users News Events Nuvco una-persona

You are here: Home → una-persona

View Edit Sharing

Edit Persona

Description of the Example Type

Direccion RDF:
Direccion RDF

Usar RDF

Nombre

Primer Apellido

Segundo Apellido

Correo

Telefono

Sitio Web

CURP

RFC

Adscrito Actualmente

Figura 3.8: Forma para agregar contacto

The image shows a web browser window displaying a Plone interface. At the top, the Plone logo is visible. Below it is a navigation menu with items: Home, Users, News, Events, Nuwco, and una-persona. A breadcrumb trail reads "You are here: Home → una-persona". Below the breadcrumb are three buttons: View, Edit, and Sharing. The main content area is titled "Edit Persona" and contains a "Description of the Example Type". The form fields are pre-filled with the following data:

- Direccion RDF:** (with a "Usar RDF" button)
- Nombres:**
- Primer Apellido:**
- Segundo Apellido:**
- Correo:**
- Telefono:**
- Sitio Web:**
- CURP:**
- RFC:**
- Adscrito Actualmente:**

At the bottom of the form are two buttons: "Guardar" and "Cancelar".

Figura 3.9: Vemos los datos rellenos automáticamente

The image shows a screenshot of a Plone web interface. At the top left is the Plone logo. Below it is a navigation menu with items: Home, Users, News, Events, Nuwco, una-persona, Perfil de Drew, RDF de asd, and RDF de Hugo. Below the navigation menu is a breadcrumb trail: "You are here: Home → una-persona". Underneath the breadcrumb are three buttons: View, Edit, and Sharing. The main content area displays the title "una-persona" and the text "by admin — last modified Jan 09, 2010 03:33 AM". Below this, several fields are listed with their values: "Nombres: Hugo", "Primer Apellido: Rodríguez", "Segundo Apellido: Peña", "Correo: sn4kes@ciencias.unam.mx", "Telefono: 56107335", "Sitio Web: www.sn4ke.com", "CURP: ROPH861109HDFDXG00", "RFC: ROPH861109", and "Adscrito Actualmente: Fac. Ciencias". At the bottom of this section is a button labeled "Generar RDF".

Figura 3.10: Vemos la información y damos la opción de generar el RDF.

Capítulo 4

Conclusiones

4.1. Experiencias y aprendizaje

Este proyecto fue uno de los primeros intentos por utilizar las tecnologías de el Web Semántico dentro de la Universidad. Por tanto fue un reto desde el principio en cuanto que tuvimos que investigar desde las tecnologías más básicas que fundamentan el Web Semántico (i.e. XML, URIs) e investigar a fondo no sólo qué es el Web Semántico sino también cómo se contruye. Una parte muy importante fue el experimentar las distintas maneras de construir las ontologías, desde escribir RDF en formato N3 y luego traducirlo a RDF/XML hasta utilizar una herramienta automática como Protege. Lo primero no resultó satisfactorio ya que la traducción de N3 a XML generaba, en algunos casos, tripletas erróneas.

También resultó el hecho de darnos cuenta lo problemático que es ponerse de acuerdo en qué estructura debe tener la información para ser compartida; esto es la definición adecuada de las distintas ontologías que elaboramos. Durante semanas nos sentabamos a discutir con gente de las instituciones involucradas en el proyecto (Facultad de Ciencias, DGAPA, Instituto de Matemáticas), para ver de qué manera cada quien manejaba la información y datos de un profesor/investigador, y buscar de qué manera ellos necesitaban que las ontologías describieran a las personas. Esto también trajo beneficios, ya que se compartían experiencias de problemas que se tuvieron al hacer sus sistemas de información. Por ejemplo, hay personas que sólo tienen un apellido, o hay gente que tiene distintos tipos de direcciones, no sólo casa y oficina.

Una experiencia satisfactoria fue aprender a usar el *framework* de desarrollo JENA. Esta es una herramienta en Java que aprovecha e implementa la mayoría de las tecnologías de el Web Semántico. Tiene cosas como la validación de RDF, un motor de SPARQL, encapsulación de resultados de búsquedas en objetos de java y razonadores con soporte SWRL. Esto nos ayudó mucho a la hora de crear los generadores de RDF, donde usabamos el motor de SPARQL de JENA. Aunque para propósitos de este proyecto usamos sólo una fracción del *framework*, ello nos ayudó mucho para saber cómo se hacen las cosas a la hora de programar sobre el Web Semántico.

Otra experiencia muy reveladora, y a la vez inquietante, fue trabajar en Plone. Esto por el hecho de que la programación web en Plone es totalmente distinta a la manera tradicional (i.e. tecnologías como HTML, AJAX, CSS). Ello implica investigar, experimentar y batallar para encontrar la manera de volver a hacer cosas como formas, widgets, web services, etc., ya que todo se ve como un objeto o una estructura de Python. Donde encontramos muchos problemas fue sobre todo en la falta de estándares de cómo hacer las cosas en esta plataforma y la forma en la que fue construido Plone. Ello significa que hay muchas maneras de hacer cualquier cosa. Ese fue un problema que tuvimos, ya que al principio andábamos perdidos sobre como hacer una forma y, más adelante, en tareas relativamente sencillas como lo es escribir un archivo sobre la base de datos de Zope o peor aún ejecutar scripts de python desde un evento provocado por un usuario. La única desventaja a esto fue que consumió demasiado tiempo del seminario poder dominar o entender como funciona esto.

En total la experiencia de todo el proyecto fue interesante, ya que no solo se necesitaba aprender a manejar nuevas tecnologías, sino también utilizarlas en distintos lenguajes de programación (Java, Javascript, Python). También se requirieron habilidades sociales ya que, desde un principio también fue nuestra tarea explicar lo que es el Web Semántico, sus ventajas y desventajas, cómo funciona y cómo hacer que funcione.

4.2. Conclusiones

Sin duda éste fue un proyecto muy importante, ya que aunque los productos finales no son de una gran magnitud, se tuvo que hacer un trabajo de investigación y experimentación a fondo. Esto por varias razones: la primera, por ser un conjunto de tecnologías emergentes que, aunque rápidamente están siendo adaptadas y su desarrollo soportado aún siguen relativamente desconocidas dentro de la UNAM. La segunda, que esto representa una nueva manera de pensar sobre cómo representar y adquirir información. Vi mucho que a los individuos que están muy enfocados en bases de datos modernas les cuesta trabajo pensar en la información con un enfoque semántico. También me di cuenta del escepticismo que hay sobre el Web Semántico, ya que varios desarrolladores siguen esperando que alguien haga algo revolucionario para entrarle de lleno al desarrollo, aunque admiten el potencial que existe. A la hora de crear las ontologías, me di cuenta que no hay ontología perfecta, y como todo en la vida, no se puede hacer feliz a todo mundo; por tanto optamos por lo más sano y sólo enfocarnos en la información mínima que se necesitaba para trabajar. Por fortuna eso no es problema, ya que es muy fácil expandir y adaptar las ontologías.

Hay que recalcar que en este seminario sólo se utilizó una fracción de lo que se puede hacer con el Web Semántico. Únicamente aprovechamos la utilidad de un lenguaje universal. Por falta de tiempo no pudimos explorar la parte semántica de la información. Me hubiera gustado buscar una aplicación en donde pudiéramos utilizar razonadores y hacer búsquedas de SPARQL más interesantes. Pero se hizo lo más importante que se requiere en el Web Semántico, que es publicar la información.

En cuanto a las metas planteadas para el seminario hay que remarcar que éstas no son las

planteadas para el proyecto de intercambio de información de personal académico, aunque en un principio así se planeó. Esto se debió al hecho de que a medida que fuimos investigando y experimentando con las tecnologías pudimos apreciar de mejor manera los requisitos para el desarrollo e implementación y, por tanto, ciertos objetivos cambiaron o se aplazaron. Por ejemplo, una meta inicial fue crear un protocolo estándar para el intercambio de información que pudiera utilizar cualquier institución de la UNAM que decida utilizar tecnologías de el Web Semántico. Esta meta fue aplazada a causa de la falta de tiempo y la logística que involucra llegar a un consenso entre las distintas instituciones de la UNAM. Queda como trabajo futuro la implementación de este protocolo. Para concluir, en el seminario se logró realizar una tarea muy importante, la cual fue: elaborar herramientas que faciliten el desarrollo del Web Semántico. La importancia de esto recae en el hecho de que muchas personas no saben, o no quieren, exponer los datos que posee un sistema de información, y de esta manera ayudamos a dar una visión concreta de cómo publicar estos datos.

Bibliografía

- [1] John Hebel, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez. *Semantic Web Programming*. Wiley Publishing. Primera edición, 2009.
- [2] Miniwatts Marketing Group. *Internet World Usage Stats - Internet and Population Statistics*. Enero 2010. <<http://www.internetworldstats.com>>.
- [3] Tim Berners-Lee, James Hendler, Ora Lassila. *The Semantic Web*. Scientific American Magazine. Publicado mayo 2001. <<http://www.scientificamerican.com/article.cfm?id=the-semantic-web>>.
- [4] Jennifer Zaino. *Follow the Money, Part 1*. Semanticweb.com. Publicado enero 2008. <http://www.semanticweb.com/main/follow_the_money_part_1_138849.asp>.
- [5] Tim Berners-Lee. *RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax*. Network Working Group. Publicado enero 2005. <<http://tools.ietf.org/html/rfc3986>>.
- [6] Tim Bray, Jean Paoli. *Extensible Markup Language (XML) 1.0*. W3C recommendation. Publicado junio 2006. <<http://www.w3.org/TR/xml>>.
- [7] Andy Mckay, Alan Runyan. *The Definitive Guide to Plone*. Apress Publishing. Segunda Edición. Septiembre 2007.
- [8] Dave Beckett. *RDF/XML Syntax Specification (Revised)*. W3C recommendation. Publicado febrero 2004. <<http://www.w3.org/TR/REC-rdf-syntax/>>.
- [9] Thomas Baker, Tom Heath, Susie Stephens. *Semantic Web Case Studies and Use Cases*. W3C document. Publicado agosto 2009. <<http://www.w3.org/2001/sw/sweo/public/UseCases/>>.