



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO



CENTRO UNIVERSITARIO UAEM TEXCOCO

**PUBLICACIÓN DE CONTENIDOS SEMÁNTICOS EN FORMATO RDF
PARA BASES DE DATOS RELACIONALES**

T E S I S

**QUE PARA OBTENER EL GRADO DE:
MAESTRA EN CIENCIAS DE LA COMPUTACION**

**PRESENTA:
ÁNGELA RÍOS LUNA**

**TUTOR ACADÉMICO:
M EN ISC IRENE AGUILAR JUÁREZ**

**TUTORES ADJUNTOS:
DR. EN ED. JOEL AYALA DE LA VEGA
DR. EN C LETICIA DÁVILA NICANOR**

TEXCOCO, ESTADO DE MÉXICO, DICIEMBRE DE 2016.



DICTÁMEN PARA AUTORIZACIÓN DE GRADO DE MAESTRÍA

Texcoco, Méx., a 7 de noviembre de 2016



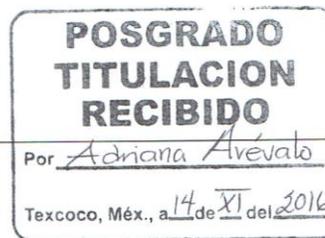
TÍTULO DEL PROYECTO:

Publicación de contenidos semánticos en formato RDF para bases de datos relacionales



TESISTA:

Angela Ríos Luna



DICTAMEN:

NO. DE REVISIÓN: 1

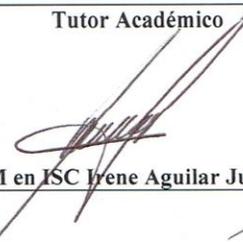
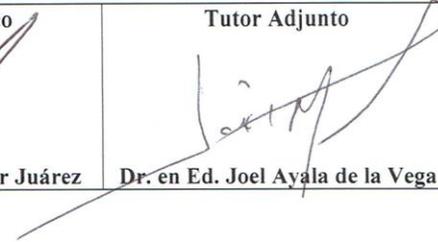
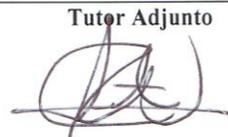


- RECHAZADO
- SUJETO A MODIFICACIONES
- ACEPTADO, CONDICIONADO
- ACEPTADO

OBSERVACIONES GENERALES: _____

ACEPTADO PARA IMPRESIÓN

ACEPTADO PARA PRESENTACIÓN DE EXAMEN DE GRADO

Tutor Académico  M en ISC Irene Aguilar Juárez	Tutor Adjunto  Dr. en Ed. Joel Ayala de la Vega	Tutor Adjunto  Dr. en C. Leticia Dávila Nicanor
---	--	---



Resumen

Los sitios de internet han evolucionado en tres etapas nombradas internet 1.0, 2.0 y 3.0. Los sitios de internet 1.0 se refiere a los primeros sitios estáticos en los que los contenidos son definidos en archivos interrelacionados por hipervínculos. Los sitios de internet 2.0 consideran mecanismos que permiten que los usuarios agreguen información que permanece referenciable permanentemente, tal como los blogs y las redes sociales. Finalmente, en los sitios de internet 3.0 los contenidos tienen además metadatos.

Los metadatos convierten un sitio de internet en un repositorio de información con valor semántico. Además, es posible hacerle consultas como si se tratara de una base de datos. Esto permite que los sitios de internet compartan información entre sí.

El problema de los metadatos es que es información intangible cuya definición requiere un esfuerzo extra de parte de los desarrolladores de sitios de internet estáticos.

En este trabajo se describe el desarrollo de un API que permite generar automáticamente archivos de metadatos RDF del estándar Dublin Core en formato semántico a partir de información contenida en una base de datos relacional, para así cumplir los requerimientos de la web 3.0.

Se ejemplifica una estrategia completa en la que un sistema de información dinámico almacena información en una base de datos que es posteriormente publicada como página de internet estática con metadatos.

Esta estrategia puede implementarse en empresas e instituciones en las que conviven sistemas de información internos (dinámicos) y sitios de internet corporativos (estáticos) y cuya información además requiere estar permanentemente sincronizada.

DEDICATORIA

Doy gracias a Dios por todo lo que me ha dado, gracias a mi familia, novio y amigos ya que ustedes me han demostrado que no existe ningún obstáculo cuando se quiere algo.

AGRADECIMIENTOS

Durante el desarrollo de mi maestría conté con el apoyo de personas e instituciones entre las que me permito mencionar:

Al Consejo Nacional de Ciencia y Tecnología, a la Universidad Autónoma del Estado de México y al Consejo Mexiquense de Ciencia y Tecnología por el apoyo económico que me brindaron para la formación de mis estudios de maestría y culminar satisfactoriamente esta investigación.

Como también a los profesores que contribuyeron en mi formación académica. M en ISC Irene Aguilar Juárez, Dr. en Ed. Joel Ayala de la Vega y la Dr. en C Leticia Dávila Nicanor por su apoyo y dedicación constante.

A mi familia que es un gran ejemplo de superación y que siempre me ha apoyado incondicionalmente en todo momento.

A mi novio por brindarme su cariño y comprensión en momentos críticos, por su optimismo que me impulsó a seguir adelante cada día.

A mis amigos que me apoyaron incondicionalmente, por todos sus consejos, sus valores y conocimientos.

CONTENIDO

Introducción.....	1
Problemática	4
1.1 Justificación	6
1.2 Objetivos	6
1.3 Metodología	7
Capítulo 2 Marco Teórico.....	8
2.1 Web semántica	8
2.1.1 Estado actual de los sistemas de información en la web.....	8
2.1.2 Que es la web semántica.....	10
2.1.3 Arquitectura de la web semántica (URI, XML, RDF, ONTOLOGIA, LOGIC, PROF, TRUST).....	10
2.1.4 Funcionamiento de la web semántica.....	12
2.1.5 Retos de la web semántica.....	13
2.2 Estándares de la web semántica	14
2.2.1 XML	14
2.2.2 RDF	16
2.2.3 RDF SCHEMA	19
2.2.4 OWL (Web Ontology lenguaje)	20
2.2.5 SPARQL	22
2.3 Trabajos relacionados.....	22
2.3.1 Actualización incremental de grafos RDF a partir de bases de datos relacionales.....	22
2.3.2 Generación de Datos Semánticos a partir de una Base de Datos Relacional de una Institución de Educación Superior.....	24
2.3.3 Creación de aplicaciones Web inteligentes para un mundo más inteligente	

2.3.4	TRIOO, estudio e implementación de modelos de datos RDF en lenguajes orientados a objetos.....	27
2.4	Software actual para implementar la web semántica	29
2.4.1	Semantic Web Builder	29
2.4.2	Protégé.....	31
2.4.3	Tecnología D2RQ Mapping Lenguaje.....	31
2.5	Sistemas de gestión de base de datos relacionales con soporte para RDF.	33
2.5.1	JENA	33
2.5.2	KOWARI	33
2.5.3	RDF Gateway	33
	Capítulo 3 diseño y desarrollo	35
3.1	Diagramas de secuencia.....	46
	Capítulo 4 Caso de estudio y resultados	56
	Capítulo 5 Discusión	60
	Capítulo 6 Conclusiones	62
	Glosario.....	64
	Anexos	66
	Bibliografía.....	99

CONTENIDO DE ILUSTRACIONES

Ilustración 1 Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. http://lod-cloud.net/	10
Ilustración 2 CRUZ-CUNHA M.M., (2009). Arquitectura de la Web Semántica, https://espacepolitique.revues.org/docannexe/image/3493/img-1.png	11
Ilustración 3 Grafo RDF (Elaboración propia)	17
Ilustración 4 Álvarez, L. (2014) Procedimiento para la actualización incremental de grafos RDF. Recuperado de http://biblioteca.sistedes.es/articulo/actualizacion-incremental-de-grafos-rdf-a-partir-de-bases-de-datos-relacionales/	23
Ilustración 5 León, F. (2014). Topología de implantación. Recuperado de https://www.researchgate.net/publication/262413150_Generacion_de_Datos_Semanticos_a_partir_de_una_Base_de_Datos_Relacional_de_una_Institucion_de_Educacion_Superior	25
Ilustración 6 Ramachandran, K. (2011) Arquitectura MVC de una aplicación Web inteligente. Recuperado de https://www.ibm.com/developerworks/ssa/library/wa-aj-smartweb/	26
Ilustración 7 Fernández, S, (2010). Arquitectura en modelos de datos RDF en lenguajes orientados a objetos. Recuperado de Http://trioo.wikier.org/resources/doc/trioo-tfm.pdf	28
Ilustración 8 Infotec, (2010). Arquitectura de semantic web builder. Recuperado de www.semanticwebbuilder.org.mx	30
Ilustración 9 Mapeo objeto-relacional, tabla y clase para la entidad libro (elaboración propia).....	37
Ilustración 10 Clase DTO con anotaciones.....	39
Ilustración 11 Diagrama de Clases (elaboración propia).....	43
Ilustración 12 Diagrama de clases del paquete Domain	44
Ilustración 13 Diagrama de clase ContextoRDF	45
Ilustración 14 Diagrama de secuencia de la clase ContextoRdf:.....	46
Ilustración 15 Diagrama de paquetes	47
Ilustración 16 Diagrama de Secuencia de la clase Parse	48
Ilustración 17 Diagrama de secuencia Invoke Getter.....	49
Ilustración 18 Arquitectura de esta propuesta	50
Ilustración 19 Diagrama de clases Contenido de marshalling	51
Ilustración 20 Diagrama de clases Contenido del dominio	53
Ilustración 21 Diagrama de clases Contenido de implementación.	54
Ilustración 22 Generación automática del RDF (elaboración propia).....	55
Ilustración 23 Implementación de la propuesta (elaboración propia).....	56
Ilustración 24 Vista de alta de un nuevo curso (elaboración propia).....	57

Ilustración 25 Vista Externa de la lista de cursos (elaboración propia)	58
Ilustración 26 Tripleta de un curso con DC (elaboración propia)	58
Ilustración 27 Tripleta de un curso (elaboración propia)	59
Ilustración 28 Representación de un curso en grafo (elaboración propia).	59

INTRODUCCIÓN

El área de trabajo conocida como **semiótica** o semiología es el estudio de los signos. Los signos se consideran los elementos que componen la comunicación, evidentemente la comunicación es un fenómeno social porque implica interacciones e intenciones humanas, se produce una comunicación eficaz cuando la intención del emisor se corresponde en gran medida con la interpretación del mensaje que realiza el receptor. Los signos y los sistemas de signos pueden agruparse en cuatro niveles interdependientes (Davies, 2014): pragmática, semántica, sintaxis y empírica. Estas constituyen las cuatro principales ramas de la semiótica.

La **pragmática** estudia el contexto y la cultura general de la comunicación o de las suposiciones compartidos que subyacen a la comprensión humana, la pragmática es el estudio de que exista un entendimiento mutuo. La **semántica** estudia el significado de los signos, la asociación entre los signos y el comportamiento. La semántica se puede considerar como el estudio de las relaciones que existen entre los símbolos y sus referentes o conceptos, especialmente de la forma en que los signos hacen referencia al comportamiento humano plasmado en normas. La **sintaxis** se ocupa del estudio de la lógica y la gramática de los signos, la sintaxis se dedica al estudio de la forma física más que al contenido de los signos. La **empírica** estudia las características físicas del medio de comunicación, esta se dedica a estudiar los canales de comunicación y sus características como el sonido, la luz y las transmisiones electrónicas. (Davies, 2014)

Por lo tanto, la pragmática se ocupa del aspecto de las intenciones, la semántica del significado de un mensaje, la sintaxis del formalismo utilizado para representar mensajes y la empírica de las señales utilizadas para codificar y transmitir un mensaje.

La semántica es el estudio de aquello a lo que se refieren los signos. La comunicación implica utilizar e interpretar signos. Cuando nos comunicamos, el emisor tiene que exponer sus intenciones utilizando algunos signos. En una conversación cara a cara, esto supondrá utilizar signos lingüísticos. El receptor del mensaje debe interpretar los signos, es decir, debe de asignar algún significado de los signos del mensaje. La semántica se ocupa de este proceso de asignar significado a los signos, un modelo sencillo de la semántica es aquel en el cual un signo puede ser descompuesto en tres componentes: el **símbolo(s)**, al que a veces se le llama significante. Esto es lo que está significando. El **referente**, al que a veces se llama significado. Aquello que está siendo significado, y el **concepto**, que es la idea del significado. En el lenguaje clásico de los sistemas de información, los símbolos equivalen a dos datos. Un dato es un conjunto de símbolos utilizado para representar algo.

Internet es un medio de comunicación. La **world wide web** (WWW) inició en el ámbito científico, su crecimiento en los años sucesivos dio lugar a los buscadores de Internet que nos permiten elegir de toda la inmensa cantidad de documentos aquellos que más nos puedan interesar. *“Estos buscadores localizan palabras en documentos mediante algoritmos sofisticados de búsqueda y con una gran inversión en hardware. Y el contenido crece de manera exponencial. La web ha tenido 3 etapas: en la primera etapa, **Web 1.0**, aparecen los primeros contenidos estáticos; en la **web 2.0** se permite que los usuarios agreguen información que permanece referenciable permanentemente, Un ejemplo común son las redes sociales”* (Codina, 2009), la **web 3.0** se enfoca en la interacción de los contenidos para volver más nutritiva la información. Mientras en el mundo del papel nos hemos resignado a este desbordamiento, en la era de los documentos en formato electrónico, somos miembros de la sociedad de la información.

Ante el gran volumen de contenidos que existe en internet, y que aumenta día con día, Tim Berners-Lee, padre del internet y director del **Consortio World Wide Web (W3C)** inició una solución para abordar esta explosión de contenido: dejar que los programas informáticos busquen y manejen la información de Internet por nosotros. Es decir, cambiar el prototipo actual de recuperación de documentos para y por humanos hacia la delegación de tareas a software.

Los estándares de la W3C fueron creados para guiar la Web hacia su máximo potencial mediante el desarrollo de protocolos comunes que promuevan su evolución y garanticen su interactividad. El modelo de utilización de la Web cambia de un repositorio de documentos a ser una gran base de conocimiento para avanzados sistemas capaces de ejecutar tareas complejas. La web se ha convertido en el almacén más grande de información que crece día a día. La Web ha revolucionado nuestra forma de trabajar, dándonos acceso a contenido en cualquier lugar del mundo a cualquier hora.

La Web 3.0 incluye la creación de repositorios, la construcción de vocabularios, y la escritura de reglas para el manejo de datos de manera semánticamente, ya que permite que los programas informáticos puedan procesar, entender y compartir información. El lenguaje de consulta de metadatos es el **Protocol and RDF QueryLanguage (SPARQL)**.

En este trabajo se menciona la evolución del internet hasta llegar a la web Semántica, se explica su arquitectura y sus componentes definidos por la W3C ya que es una parte importante para el desarrollo de una aplicación web. También se detalla una estrategia propia de publicación de contenidos semánticos en formato **Resource Description Framework (RDF)** que es uno de los estándares definidos por la W3C que se utiliza para el intercambio de datos.

La estrategia presentada permite la evolución de un sitio de internet a la web 3.0 mediante la publicación automática de contenidos digitales con metadatos a partir de información contenida en una base de datos relacional de acuerdo a las directrices de la

web 3.0. Específicamente se desarrolló una ***Application Programming Interface (API)*** que genera archivos de metadatos RDF a partir de objetos con anotaciones.

PROBLEMÁTICA

Hoy en día existe tanta información en internet que los usuarios necesitan herramientas que discriminen o promuevan automáticamente los contenidos. Una Web carente de semántica puede en ocasiones, convertir el acceso a la información en una tarea difícil y frustrante.

La web 3.0 plantea las directrices que le permiten a los sitios web interactuar entre sí para enriquecer la información presentada.

La publicación de contenidos semánticos para las empresas e instituciones es una tarea compleja para la tecnología que existe hoy en día. La solución se debe pegar a los estándares ya definidos por expertos; como también definir el proceso a seguir para la publicación de contenidos.

Existen algunos sistemas de información web que necesitan transformar esta información a un contenido semántico. Existe la posibilidad de transformar la información ya existente almacenada en una base de datos relacional mediante los estándares definidos por la W3C, los retos para esta transformación es obtener la información de una base de datos relacional y posteriormente ofrecer un contenido semántico.

La principal desventaja de los sitios de Internet estáticos es que al no estar su información almacenada en una base de datos no es posible realizarle consultas. Mientras que los sitios de Internet dinámicos presentan desventajas al momento de ser indexados, pues los contenidos son generados sobre demanda y se necesitan directrices que marquen los atributos primarios que deben solicitarse para su indexación. Para indexar un sitio dinámico no basta con dar permisos de acceso a la información privada, cada recurso debe ser accesible mediante un **UniformResourceLocator (URL)** única. Cabe mencionar que la información algunas veces es utilizada con más de un propósito; por ejemplo, en el sitio de Internet de una agencia de autos, los precios de los automóviles pueden ser utilizados por el departamento de marketing como información pública, o como información privada en el departamento de contabilidad. En los sitios estáticos, al no estar la información almacenada en una base de datos, este tipo de datos no son referenciales más que como contenido web limitando la utilidad de esta información.

Las instituciones y empresas necesitan por tanto una estrategia que combine contenidos estáticos y dinámicos, una combinación de **página de Internet corporativa** e **Intranet** privada. Para agilizar la administración de páginas de Internet corporativas (estáticas) puede utilizarse un **publicador de contenidos** tales como wordpress© (Mullenweg) o Joomla©(Mambo Open Source, s.f.). Los publicadores de contenido son herramientas que a partir de la captura de información cruda automatizan su publicación con formato

en un servidor. Sin embargo, esto supone capturar la información dos veces, una en la Intranet y otra en el publicador de contenidos. La administración de un publicador de contenidos requiere personal con un perfil diferente al que administra las herramientas propietarias, se necesita el doble del personal. Algunos publicadores de contenidos almacenan la información en una base de datos, pero bajo el diseño que la herramienta define y no bajo la arquitectura que la organización ya posee. La información dinámica y la estática no son actualizadas al mismo tiempo por lo que podrían ser diferentes y presentar inconsistencias.

Los últimos años se ha visto un enorme crecimiento en el tamaño y la variedad de datos RDF con la disponibilidad de datos semánticos aplicados para cualquier disciplina (Jyoti, 2014); La mayoría de los datos en los que se basa la Web actual se almacenan en bases de datos relacionales con mayor frecuencia. Los datos representados en RDF pueden ser interpretados, procesados y razonados por los agentes de software. De ahí que el cambio del modelo relacional al modelo RDF para la manipulación de la información sea uno de los retos de investigación actuales (Álvarez, 2014)

Así que, de acuerdo a las directrices de la web 3.0, nuestros contenidos requieren de una parte visual (para las personas) y de metadatos (para ser procesadas automáticamente por máquinas).

El estándar RDF proporciona información descriptiva simple sobre los recursos que se encuentran en la Web y que se utiliza, por ejemplo, en catálogos de libros, directorios, colecciones personales de música, fotos, eventos, etc. En la web semántica los contenidos se transforman para que sean comprensibles para las máquinas. Es decir, las máquinas se convierten en consumidores para que, entre otras cosas, redistribuyan solo información del interés del usuario o faciliten su búsqueda.

1.1 Justificación

La divulgación de datos estructurados almacenados en una base de datos relacional como contenido semántico que contiene datos semiestructurados genera nuevos canales de comunicación y comercio para las empresas e instituciones. Los sitios 3.0 posicionan a las instituciones como generadoras de contenidos confiables y referenciales, aumentando su presencia en internet.

Los sistemas informáticos semánticos con la arquitectura definida por esta investigación permiten:

- Modularidad: ensamblarse con otros vocabularios.
- Extensibilidad: el sistema se debe poder adecuar con elementos adicionales a las necesidades locales sin perjudicar sus contenidos semánticos.
- Refinamiento: vocabularios controlados.

Todo esto, mediante el uso del estándar RDF que permite generar contenidos enriquecidos y ayuda a la organización de los datos y proporciona información descriptiva simple sobre los recursos que se encuentran en la Web; Cabe mencionar que agregar metadatos de manera artesanal es muy tardado y complicado.

1.2 Objetivos

GENERAL

Desarrollar una estrategia de generación, actualización y publicación automática de contenidos semánticos en formato RDF a partir de la información almacenada en una base de datos relacional.

PARTICULARES

- Analizar el estándar ResourceDescription Framework (RDF) definido por la W3C y la correspondencia del lenguaje de consultas SPARQL.
- Identificar los requerimientos de adecuación de la base de datos
- Codificar el módulo que permita la conversión de información de la base de datos a RDF
- Desarrollar las estrategias necesarias para la actualización de los contenidos semánticos y relacionales.
- Publicación de los contenidos semánticos en un entorno web

1.3 Metodología

Para esta investigación se utilizará la metodología SCRUM donde se plantean bloques temporales y cada iteración corresponde a un resultado. En este caso se debe realizar un análisis del estándar de vocabulario RDF, para poder utilizar el tipo de vocabulario conveniente para la investigación, como también la comparativa de estos RDF y visualizar en qué casos se deben de utilizar y que herramientas son de utilidad. Posteriormente seleccionamos los requisitos, planificamos la iteración y ejecutamos la iteración para posteriormente realizar las demostraciones, donde se analizara los requisitos, para posteriormente conectar estas iteraciones con una base de datos relacional.

Se utiliza una metodología que adopta una estrategia de desarrollo incremental, seguida de incrementos y versiones parciales del mismo, es importante mencionar que cada incremento tiene su propio ciclo de vida. Cada incremento agrega funcionalidad y mejora del desarrollo de investigación. Conforme se completa cada etapa, se verifica e integra la versión con las demás versiones ya completadas para la investigación. Durante cada incremento, el sistema se evalúa con respecto al desarrollo de versiones futuras. Las actividades se dividen en procesos y subprocesos para que el desarrollo de esta investigación sea exitoso.

La iteración se realizará con los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), y también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones. El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Posteriormente se realizarán las pruebas mediante un ambiente de servidor web local.

Capítulo 2

MARCO TEÓRICO

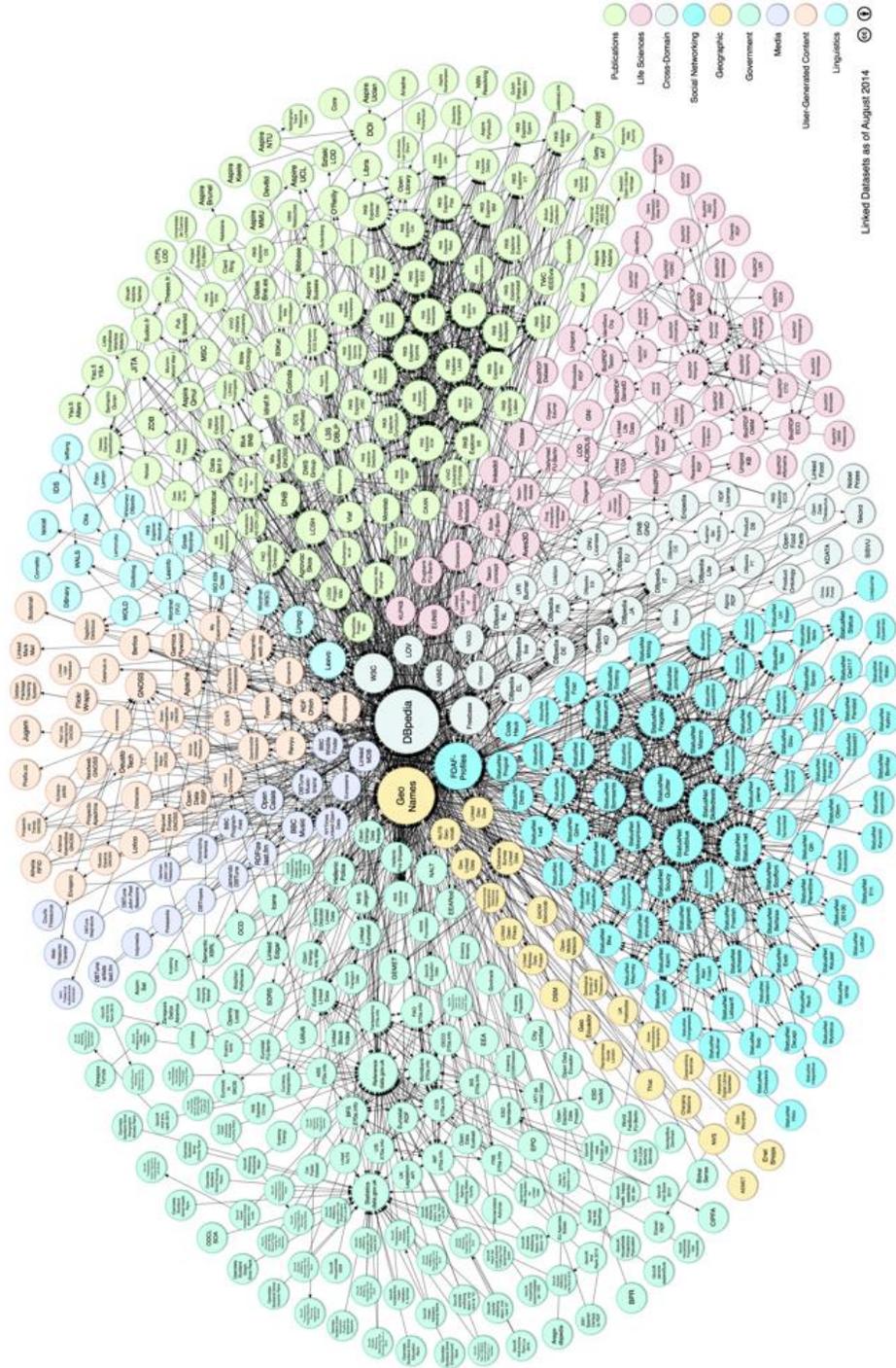
2.1 Web semántica

2.1.1 Estado actual de los sistemas de información en la web.

Los datos representados en RDF pueden ser interpretados, procesados y razonados por los agentes de software. En el año 2007 surgió, en el W3C, el proyecto **Linked Open Data**, con el fin de divulgar la Web semántica. En su web se publica cada año un grafo que muestra los enlaces creados entre todas las Webs semánticas. Si observamos la evolución, se puede apreciar como el ritmo de crecimiento ha sido exponencial. En el centro de un grafo tenemos a la **DBpedia**, la versión semántica de la Wikipedia y muy cerca Freebase que es actualmente propiedad de Google y ha sido la base para su Knowledgegraph. Este “grafo del conocimiento” sirve para complementar las búsquedas con información semántica **Ilustración 1.**(Saquete, 2013)

La evolución ha llevado desde un espacio global de información de documentos enlazados hasta uno en el que ambos, documentos y datos, estén enlazados. Por debajo de este proceso está un conjunto de buenas prácticas para publicar y conectar datos estructurados conocido como Linked Data, cuyo ejemplo más visible de adopción es el proyecto Linking Open Data.(Castro, 2012)

Se pasa así de una Web basada en documentos HTML, donde los enlaces son relaciones entre puntos de los documentos y en la que el usuario es el destinatario de la información publicada, a una Web de datos enlazados que están expresados en RDF, en la que sistemas y agentes software pueden explotar estos datos de forma automática (recopilándolos, agregándolos, interpretándolos, publicándolos, etc.).



2.1.2 Que es la web semántica

La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla.

“La colección de las tecnologías de la Web Semántica (RDF, OWL (Ontology Web Language), SKOS (Simple Knowledge Organization System), SPARQL, etc)” (Semántica) ofrece un ambiente donde la aplicación puede consultar los datos, extraer inferencias utilizando vocabularios, etc.

2.1.3 Arquitectura de la web semántica (URI, XML, RDF, ONTOLOGIA, LOGIC, PROF, TRUST)

Para tener una adecuada definición de los datos, la Web Semántica utiliza esencialmente RDF, SPARQL, y OWL, mecanismos que ayudan a convertir la Web en una infraestructura global en la que es posible compartir, y reutilizar datos y documentos entre diferentes tipos de usuarios. *“Otra tecnología que ofrece la Web Semántica para enriquecer los contenidos de la Web tradicional es RDFa”* (García Ruiz & Arroyo Gómez, 2008). Mediante RDFa se pueden representar los datos estructurados visibles en las páginas Web (eventos en calendarios, información de contacto personal, información sobre derechos de autor, etc.), a través de unas anotaciones semánticas incluidas en el código e invisibles para el usuario, lo que permitirá a las aplicaciones interpretar esta información y utilizarla de forma eficaz. Por ejemplo, una aplicación de calendario podría importar directamente los eventos que encuentra al navegar por cierta página Web, o se podrían especificar los datos del autor de cualquier foto publicada, así como la licencia de cualquier documento que se encuentre. *“Para extraer el RDF se podría utilizar un mecanismo Gleaning Resource Descriptions from Dialects of Languages (GRDDL), una técnica estándar para extraer la información expresada en RDF desde documentos extensible markup language (XML), y en particular, de las páginas extensible hypertext markup language XHTML”* (w3c, Web Semántica).

El desarrollo de la Web Semántica se está produciendo en forma de capas, de modo que el desarrollo de nuevas capas se realice sobre las ya existentes. “En la actualidad, el W3C maneja una estructura en lo que respecta a la Web Semántica que no dista mucho

de lo que estableciera Tim Berners-Lee en el año 2000: el estándar denominado XML.” (w3c, Estándares de la web semántica)

En segundo lugar, aporta descripciones (metadatos) de las páginas y sitios web con un formato que sea compatible con la estructura general de la Web y con diversas categorías de páginas e interoperable entre distintos sistemas informáticos, de esto se ocupa la norma RDF.

En tercer lugar, mediante un sistema de ontologías que permitan especificar conceptos de los diversos dominios del conocimiento mediante el uso de un lenguaje fuertemente basado en lógica simbólica y susceptible, por tanto, de ser eventualmente interpretado por un ordenador. De este aspecto se ocupa el denominado OWL. La **ilustración 2** muestra la figura de la arquitectura de la web semántica:

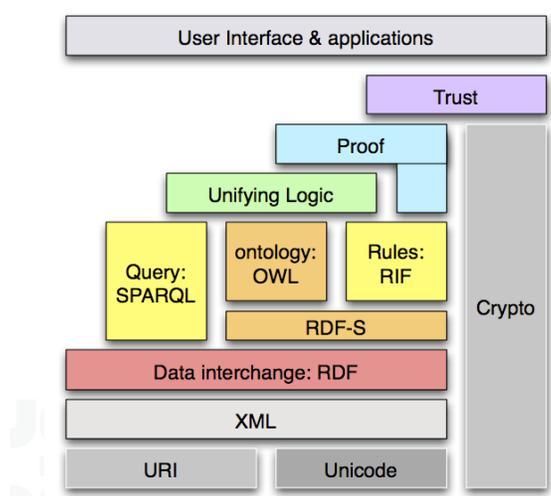


Ilustración 2 CRUZ-CUNHA M.M., (2009). Arquitectura de la Web Semántica, <https://espacepolitique.revues.org/docannexe/image/3493/img-1.png>

A continuación, se describen los módulos de la Arquitectura de la Web Semántica:

URI: las referencias. URI es el acrónimo de "uniform resource identifier" o Identificador Uniforme de Recursos, identificador único que permite la localización de un recurso que puede ser accedido vía Internet. Se trata del **URL**(descripción de la ubicación) más el **URN** (descripción del espacio de nombre).

Unicode: el alfabeto. Se trata de una codificación del texto que permite utilizar los símbolos de diferentes idiomas sin que aparezcan caracteres extraños. De esta forma, se puede expresar información en la Web Semántica en cualquier idioma.

XML: aporta la sintaxis superficial para los documentos estructurados, pero sin dotarles de ninguna restricción sobre el significado.

- XML Schema: es un lenguaje para definir la estructura de los documentos XML.RDF es un modelo de datos para los recursos y las relaciones que se

puedan establecer entre ellos. (Ruiz R. G., 2008) Aporta una semántica básica para este modelo de datos que puede representarse mediante XML.

- **RDFS** es un vocabulario para describir las propiedades y las clases de los recursos RDF, con una semántica para establecer jerarquías de generalización entre dichas propiedades y clases. Ejemplo, en catálogos de libros, directorios, colecciones personales de música, fotos, eventos, etc.

SPARQL: es lenguaje de consulta sobre RDF, que permite hacer búsquedas sobre los recursos de la Web Semántica utilizando distintas fuentes de datos.

OWL: es un mecanismo para desarrollar temas o vocabularios específicos en los que asociar esos recursos. Lo que hace OWL es proporcionar un lenguaje para definir ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas. Las ontologías, que se encargan de definir los términos utilizados para describir y representar un área de conocimiento, son utilizadas por los usuarios, las bases de datos y las aplicaciones que necesitan compartir información específica, es decir, en un campo determinado como puede ser el de las finanzas, medicina, deporte, etc. Las ontologías incluyen definiciones de conceptos básicos en un campo determinado y la relación entre ellos.

Rules RIF (Regla de Formato de Intercambio): Estándar que ofrece a los programadores web una nueva forma de escribir normas para intercambiar datos de diferentes sitios de Internet de manera exitosa.

unifying logic: Deben de existir reglas de inferencia. Una ontología puede expresar la regla de esta forma, un programa podría deducir.

Proof: será necesario el intercambio de "pruebas" escritas en el lenguaje unificador (se trata del lenguaje que hace posible las inferencias lógicas hecha posibles a través del uso de reglas de inferencia tal como es especificado por las ontologías) de la Web Semántica.

Crypto: bloque encriptado de datos que serán utilizados por los ordenadores y los agentes para verificar que la información adjunta ha sido ofrecida por una fuente específica confiable.

2.1.4 Funcionamiento de la web semántica

Para que la información de la web semántica funcione de una manera correcta debe de estar bien definida y etiquetada la información. Se necesitan lenguajes semánticos potentes que sean capaces de representar conocimiento basándose en el uso de metadatos y ontologías. RDF es un lenguaje para especificar metadatos, como también intercambia entre aplicaciones distintas

Según Tim Berners-Lee, los cuatro principios de diseño en los que se basa la web de datos enlazados son (Carmen, 2011):

- Utilizar URIs como nombres únicos para los recursos
- Utilizar el Hypertext Transfer Protocol (HTTP) para nombrar y resolver la ubicación de los datos identificados mediante esas URIs. Si las entidades están identificadas mediante URI's pueden buscarse sin tener referencia a la URI utilizando el protocolo HTTP.
- Representar los datos en RDF y utilizar SPARQL como lenguaje de consulta de dichos datos. Mientras que HTML proporciona un medio para estructurar y enlazar documentos en la Web, RDF proporciona un modelo de datos genérico, con el que estructurar y enlazar datos que describen cosas en el mundo en forma de triples sujeto-predicado-objeto.
- Incluir enlaces a otras URIs para permitir la localización de más datos enlazados, aprovechando que los elementos de un triple RDF pueden ser una URI o una cadena de literales.

De esta forma, si una aplicación desea obtener información sobre un dato identificado mediante una URI, cuando hace una llamada HTTP se referencia el recurso, obteniendo información fácilmente procesable en formato RDF. Cuando, además, se proveen puntos de consulta avanzada, con SPARQL, el resultado ante una consulta podrá ser interpretado de forma automática. Si enlazamos los datos no se quedarán aislados y se podrá compartir información con otras fuentes externas. Gracias a estos mecanismos, cualquier recurso es susceptible de ser enriquecido con cualquier tipo de información especializada, incluso la que no se espera que sea combinable.

2.1.5 Retos de la web semántica

Tener un nivel más alto de rendimiento de los motores de búsqueda, sistemas semánticos, desarrollando algoritmos cada vez más inteligentes. La implementación de vocabularios estructurados para controlar el contenido de estos metadatos y para organizar la información. De esta forma se puede conseguir una organización más eficaz de las colecciones que redundará en una recuperación de información más pertinente, se debe combinar el desarrollo de estructuras de meta información y la potencia del procesamiento informático de las mismas. Ambas tendencias coincidirían en el planteamiento de la Web semántica, en la que se persigue una búsqueda inteligente que debería aprovechar el conocimiento estructurado y el valor añadido de lo humano embebido en las estructuras de conocimiento.

El problema ahora es que la información representada de manera semántica utiliza las definiciones propias de cada empresa y es prácticamente imposible relacionarla de manera correcta, ya que se cuenta con silos de grafos distribuidos en toda la red. Es por ello que ha surgido la iniciativa de Linked Data, para enlazar toda la información disponible en los grafos de la Web y construir un único grafo que representa todo el conocimiento almacenado en la red.(Pacheco, 2015)

Conseguir que los contenidos estén descritos mediante metadatos y dotados explícitamente de semántica. Gracias a la estructuración y a la normalización de la información se espera que los agentes sean capaces de buscar por conceptos, en lugar de hacerlo por simple comparación de caracteres. Si, además, se consigue que la semántica esté explícita en una ontología, los agentes de software serán capaces de deducir e inferir conocimiento.

En los sistemas de las bibliotecas, archivos y centros de documentación tradicionalmente han utilizado los vocabularios controlados como un elemento fundamental de sus bases de datos y catálogos, los retos de estos es adaptar los vocabularios estructurados a los estándares y a las recomendaciones de la Web semántica definida por la W3C y así poder tener una semántica para proporcionar un potencial ilimitado para facilitar su utilización en múltiples nuevas formas.

Dado el volumen de información disponible en la Web, será inevitable automatizar algunos de los procesos que supone la elaboración de grandes estructuras de conocimiento, aplicando procedimientos con los que se ha experimentado desde la recuperación de la información. Actualmente existen empresas que desarrollan herramientas para la clasificación automatizada (Castro, 2012).

La publicación masiva de datos implica tener que preservar la privacidad de las personas e instituciones, garantizando que no sea posible deducir indirectamente determinada información confidencial. Además, el hecho de que cualquiera pueda publicar y enlazar datos en la web de datos implica que hay que tener en cuenta también aspectos sobre la procedencia de los datos, su calidad y la fiabilidad de las fuentes. (Schorlemmer, 2011).

2.2 Estándares de la web semántica

Actualmente existen diversos estándares que pueden contribuir a mejorar los sistemas de búsqueda, especialmente si son aplicados en entornos controlados como los repositorios, las bibliotecas digitales o las intranets.

2.2.1 XML

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML, pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. (Bray, 2006).

XML es un formato de datos muy popular. Existen grandes volúmenes de datos XML, estos se producen ya sea manualmente (como documentos HTML que existen hoy en día), o con herramientas de software. Los datos XML pueden ser almacenados en una

base de datos debido a la estructura que se puede manejar en este lenguaje. (Florescu, 2000).

XML fue desarrollado por un Grupo de Trabajo de XML perteneciente al W3C, conocido originalmente como el Comité de Revisión Editorial de standard generalized markup language (SGML), en 1996. Este grupo de trabajo estaba presidido por John Bosak de SUN. Los objetivos de diseño que este grupo se propuso fueron los siguientes:

- XML debía ser utilizable directamente sobre Internet
- XML debería soportar una amplia variedad de aplicaciones
- Tenía que ser compatible con SGML
- Tenía que ser sencillo escribir programas que procesasen documentos XML
- Debía existir el mínimo de características opcionales (mejor ninguna)
- Los documentos debían ser legibles para los seres humanos
- El diseño de los documentos tenía que ser formal y conciso
- Los documentos XML tenían que ser sencillos de crear.
- El formato de las etiquetas usadas tendría la mínima importancia. La primera recomendación XML del W3C salió en febrero de 1998.

Entre las tecnologías disponibles se pueden destacar:

- **XSL**: Lenguaje Extensible de Hojas de Estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo móvil, o un conjunto de páginas de un catálogo, informe o libro.
- **XPath**: Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML.
- **XLink**: Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos XML.
- **XPointer**: Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.
- **XQL**: Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web.

El método de almacenamiento presenta inconvenientes cuando se trabaja con volúmenes grandes de datos. Por este motivo desde hace unos años existen una serie de productos que permiten almacenar tripletas RDF en bases de datos. (Matute)

2.2.2 RDF

La arquitectura de la Web Semántica está centrada en el modelo resource description framework (RDF), es un formato universal para datos en la web. Actualmente se utiliza para la descripción conceptual o modelado de la información que esta implementada en los recursos de la WEB, utilizando anotaciones de sintaxis.

El RDF provee una semántica para metadatos; una mejor precisión en la exploración de recursos que la obtenida por los motores de búsqueda que rastrean el texto completo; y mejores aplicaciones. Todo ello en tanto se desarrollen los esquemas – Schema - correspondientes. En general, el RDF brinda las bases para herramientas genéricas, tanto sea para crear, gestionar o buscar datos en la web de manera entendible por las computadoras, promoviendo la transformación de la web en un repositorio de información procesable por las computadoras. Los objetivos de linked data es usar los datos en la Web de la misma forma que los documentos, enlazar los datos entre sí, y usar los datos de la forma que esta colección de conjuntos de datos relacionados entre sí en la Web también puede ser referido como linked data. Para lograr y crear linked data, las tecnologías deben estar disponibles para un formato común RDF, ya sea para hacer la conversión o en la marcha el acceso a las bases de datos existentes (w3c, Estandares de la web semantica)

Hay muchas ventajas para la gestión de datos RDF como un objeto de base de datos

- Se hace más fácil para modelar aplicaciones RDF.
- Los datos RDF se pueden integrar fácilmente con otros datos de la empresa.
- Reutilización del objeto RDF hace que sea posible el desarrollo de aplicaciones de manera más eficiente.
- Abstracción de objetos y la encapsulación de comportamientos RDF-específicas hacen que las aplicaciones más fáciles de entender y mantener.
- No se requiere mapeo entre objetos del lado del cliente RDF y columnas de base de datos y tablas que contienen triples.
- No se requieren configuraciones adicionales para el almacenamiento de datos RDF.(Ravada, 2006)

1. Modelo de datos basados en grafos

La estructura de la sintaxis es conjunto de tres elementos llamadas tripletas y está compuesto de “un sujeto, predicado y un objeto” (w3c, RDF concepts, 2014), a este conjunto se llama grafo RDF, donde un grafo RDF se puede visualizar como nodo del grafo y arco del grafo, en el que cada triple se representa como un nodo de enlace. Cabe mencionar que estos tres elementos se representan mediante URIS Ilustración 3.



Ilustración 3 Grafo RDF (Elaboración propia)

1. En el sujeto: se describe de quien está hablando,
2. En el predicado que también se le conoce como propiedad: este representa una conexión entre un sujeto y un objeto, normalmente una relación del sujeto con algo o un atributo del sujeto
3. En el objeto el valor de lo declarado.

A continuación, se muestra un ejemplo de una tripleta en la tabla 1:

Tabla 1 Tripleta básica

SUJETO	PREDICADO o PROPIEDAD	OBJETO
Http://ejemplo.com/pepe	Http://ejemplo.com/nacimineto	"14/08/1988"

2. Propiedades del lenguaje RDF

Las propiedades básicas de RDF se dividen en distintas categorías fundamentales, para la definición, a continuación, se listan unas de ellas:

rdf: Resource: En RDF se denomina 'recurso', al igual que en la mayoría de sistemas de metadatos, a cualquier cosa que pueda poseer una URI, por lo que el potencial descriptivo de RDF está sujeto entre otros, a los objetos que puedan poseer.

rdf: Property: Son utilizados como predicados de las tripletas, la semántica de una tripleta depende del valor 'property' utilizado como predicado. RDF considera las propiedades como clases de primer orden.

rdf: Value: Es una propiedad que provee RDF para denotar valores primordiales.

rdf: Statement: Si se necesita declarar quien es el autor de una declaración en particular, necesario que la declaración sea un recurso. `rdf: Statement` sirve para retornar una declaración formulada anteriormente en un recurso. Estos recursos debe tener por lo menos 3 propiedades: `rdf:subject`, `rdf:object`, `rdf:Predicate`.

A continuación, se muestran otras propiedades importantes del RDF, véase en la tabla2:

Tabla 2 Propiedades de una tripleta

rdfs:domain	Un dominio de la propiedad del sujeto.	rdf:Property	rdfs:Class
rdfs:range	Un rango de la propiedad del sujeto.	rdf:Property	rdfs:Class
rdfs:label	Un nombre para el sujeto legible por seres humanos.	rdfs:Resource	rdfs:Literal
rdfs:comment	Una descripción del recurso sujeto.	rdfs:Resource	rdfs:Literal
rdfs:member	Un miembro del recurso sujeto.	rdfs:Resource	rdfs:Resource
rdf:first	El primer item en la lista RDF del sujeto.	rdf:List	rdfs:Resource
rdf:rest	El resto de la lista RDF del sujeto después del primer item.	rdf:List	rdf:List
rdfs:seeAlso	Más información sobre el recurso sujeto.	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	La definición del recurso sujeto.	rdfs:Resource	rdfs:Resource

3. Sintaxis de RDF:

Como resultado de la reunión de distintas comunidades que están de acuerdo en los principios básicos de la representación y transposición de metadatos, RDF está influido de varias fuentes diferentes. Las principales influencias provienen de la propia Comunidad de Normalización de la Web en forma de metadatos HTML y PICS, la comunidad bibliotecaria, la comunidad de los documentos estructurados en forma de SGML y sobre todo XML (w3c, 2001)Continuación se muestran algunas sintaxis en base a sus necesidades:

- **RDF/XML:** RDF provee de una sintaxis sobre XML para escribir la representación de gráficas o tripletas y a esto se le conoce como RDF/XML. Su sintaxis está definida en RDF/XML.(García)
- **N3:** La semántica de N3 se han definido, al igual que algunas de las propiedades de un operador incorporado que añaden inferencia lógica utilizando reglas de la lengua, y que permiten las reglas para definir la inferencia que se puede extraer de los documentos web específicos en la web, en función de otra información sobre esos documentos. (W3, Notacion 3)
- **N TRIPLETAS:** Subconjunto basado en línea. La sintaxis es una versión revisada de la N-Triples definida originalmente en los Casos de Prueba RDF [RDF-casos de prueba]. Su intención original era para escribir casos de prueba, pero ha

demostrado ser popular como un formato de intercambio de datos RDF. (W3, N-TRIPLES)

- **JSON:** Representa un conjunto de RDF como una serie de estructuras de datos anidadas. Cada sujeto único en el conjunto de triples se representa como una clave en el objeto JSON. El valor de cada clave es un objeto cuyas claves son los URI de las propiedades asociadas a cada tema. (HP)
- **RDFA:** Ofrece una serie de funciones avanzadas que permiten a los autores para expresar datos estructurados bastante complejas, como las relaciones entre las personas, lugares y eventos en un documento HTML o XML. (W3, RDF).

2.2.3 RDF SCHEMA

Proporciona un vocabulario de modelado de datos para los datos RDF Proporciona mecanismos para la descripción de los grupos de recursos relacionados y las relaciones entre estos recursos. El esquema RDF está escrito en RDF utilizando los términos descritos en el presente documento. Estos recursos se utilizan para determinar las características de otros recursos, como los dominios y rangos de propiedades. La clase y la propiedad del sistema Esquema RDF son similar a los sistemas de tipos de lenguajes de programación orientados a objetos como Java. Esquema RDF se diferencia de muchos de estos sistemas en los que en lugar de definir una clase en términos de las propiedades de sus casos puedan tener, Esquema RDF describe las propiedades en función de las clases de los recursos a los que se aplican. Este es el papel de los de dominio y rango mecanismos descritos en esta especificación. (McBride, 2014).

1. Clases

Los recursos pueden ser divididos en grupos llamados clases. Los miembros de una clase se conocen como las instancias de la clase. Las clases son en sí mismos recursos. A menudo se identifican por URI y pueden describirse mediante las propiedades RDF. El `rdf:type` característica puede ser utilizada para indicar que un recurso es una instancia de una clase.

RDF distingue entre una clase y el conjunto de sus instancias. Asociado con cada clase hay un conjunto, llamado la extensión de clase de la clase, que es el conjunto de las instancias de la clase. Dos clases pueden tener el mismo conjunto de casos, pero tenga diferentes clases. Por ejemplo, la agencia tributaria puede definir la clase de personas que viven en la misma dirección que el editor de este documento. La oficina de correos puede definir la clase de personas cuya dirección tiene el mismo código postal como la dirección del autor. Es posible que estas clases para tener exactamente los mismos casos, sin embargo, para tener diferentes propiedades. Sólo una de las clases tiene la

propiedad que se definió por la oficina de impuestos, y sólo el otro tiene la propiedad de que fue definido por la Oficina de Correos.

Una clase puede ser un miembro de su propia extensión de clase y puede ser una instancia de sí mismo. El grupo de recursos que son clases RDF Schema es en sí misma una clase llamada `rdfs: Class`.

2. Propiedades

Los Conceptos RDF y la especificación de sintaxis abstracta describen el concepto de una propiedad RDF como una relación entre los recursos y los recursos sujetos a objetos.

Esta especificación define el concepto de subpropiedad. El `rdfs: subPropertyOf` característica puede ser utilizada para indicar que una propiedad es una subpropiedad de otro. Si una propiedad P es un subpropiedad de propiedad P' , entonces todos los pares de los recursos que están relacionados por P están también relacionados por P' . El término súper-propiedad se utiliza a menudo como el inverso de subpropiedad. Si una propiedad P' es un súper -propiedad de una propiedad P , entonces todos los pares de los recursos que están relacionados por P también están relacionados por P' . Esta especificación no define una propiedad superior que es el súper-propiedad de todas las propiedades.

2.2.4 OWL (Web Ontology lenguaje)

El Lenguaje de Ontologías Web (OWL) está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos.

Diseño y población semiautomática de ontologías. (Cortes, 2012)

- Relaciones entre clases
- Cardinalidad
- Igualdad, más tipos para las propiedades,
- Características de las propiedades
- Clases enumeradas.

OWL facilita un mejor mecanismo de interpretación de contenido Web que los mecanismos admitidos por XML, RDF, y esquema RDF (RDF-S) proporcionando vocabulario adicional junto con una semántica formal. OWL tiene tres sublenguajes, con un nivel de expresividad creciente: OWL Lite, OWL DL, y OWL Full.

La Web semántica es una visión del futuro de la Web donde la información está dando un significado explícito, permitiendo que las máquinas puedan procesar automáticamente e integrar la información disponible en la Web. La Web semántica se basará en la capacidad de XML para definir esquemas de etiquetas a medida y en la aproximación flexible de RDF para representar datos. El primer nivel requerido por encima de RDF para la Web semántica es un lenguaje de ontologías que pueda describir

formalmente el significado de la terminología usada en los documentos Web. Si se espera que las máquinas hagan tareas útiles de razonamiento sobre estos documentos, el lenguaje debe ir más allá de las semánticas básicas del RDF Schema. (Group, 2012).

OWL ha sido diseñado para cubrir esta necesidad de un lenguaje de ontologías Web. OWL forma parte de un conjunto creciente de recomendaciones del W3C relacionadas con la Web semántica.

- XML proporciona una sintaxis superficial para documentos estructurados, pero no impone restricciones semánticas en el significado de estos documentos.
- XML Schema es un lenguaje que se utiliza para restringir la estructura de los documentos XML, además de para ampliar XML con tipos de datos.
- RDF es un modelo de datos para objetos ("recursos") y relaciones entre ellos, proporcionando una semántica simple para éste. Este tipo de modelo de datos puede ser representado en una sintaxis XML.
- RDF Schema es un vocabulario utilizado para describir propiedades y clases de recursos RDF, con una semántica para la generalización y jerarquización tanto de propiedades como de clases.
- OWL añade más vocabulario para describir propiedades y clases: entre otros, relaciones entre clases (por ejemplo, desunión), cardinalidad (por ejemplo, "uno exacto"), igualdad, más tipos de propiedades, características de propiedades (por ejemplo, simetría), y clases enumeradas.
- Los tres sublenguajes de OWL

OWL proporciona tres lenguajes, cada uno con nivel de expresividad mayor que el anterior, diseñados para ser usados por comunidades específicas de desarrolladores y usuarios.

- OWL Lite: está diseñado para aquellos usuarios que necesitan principalmente una clasificación jerárquica y restricciones simples. Por ejemplo, a la vez que admite restricciones de cardinalidad, sólo permite establecer valores cardinales de 0 ó 1. Debería ser más sencillo proporcionar herramientas de soporte a OWL Lite que a sus parientes con mayor nivel de expresividad y OWL Lite proporciona una ruta rápida de migración para tesauros y otras taxonomías.
- OWL DL está diseñado para aquellos usuarios que quieren la máxima expresividad conservando completitud computacional (se garantiza que todas las conclusiones sean computables), y reusabilidad (todos los cálculos se resolverán en un tiempo finito). OWL DL incluye todas las construcciones del lenguaje de OWL, pero sólo pueden ser usados bajo ciertas restricciones (por ejemplo, mientras una clase puede ser una subclase de otras muchas clases, una clase no puede ser una instancia de otra). OWL DL es denominado de esta forma debido a su correspondencia con la lógica de descripción

(DescriptionLogics, en inglés), un campo de investigación que estudia la lógica que compone la base formal de OWL.

- OWL Full está dirigido a usuarios que quieren máxima expresividad y libertad sintáctica de RDF sin garantías computacionales. Por ejemplo, en OWL Full una clase puede ser considerada simultáneamente como una colección de clases individuales y como una clase individual propiamente dicha. OWL Full permite una ontología para aumentar el significado del vocabulario preestablecido (RDF u OWL). Es poco probable que cualquier software de razonamiento sea capaz de obtener un razonamiento completo para cada característica de OWL Full.

2.2.5 SPARQL

Formato de datos para grafos dirigidos y etiquetados para representar la información en la Web. Esta especificación define la sintaxis y la semántica del lenguaje de consulta SPARQL para RDF. SPARQL se puede utilizar para expresar consultas que permiten interrogar diversas fuentes de datos, si los datos se almacenan de forma nativa como RDF o son definidos mediante vistas RDF a través de algún sistema middleware. SPARQL contiene las capacidades para la consulta de los patrones obligatorios y opcionales de grafo, junto con sus conjunciones y disyunciones. SPARQL también soporta la ampliación o restricciones del ámbito de las consultas indicando los grafos sobre los que se opera. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF.

2.3 Trabajos relacionados

Estudios recientes demuestran que es posible trasladar del modelo de base de datos relacionales al modelo basado en grafos RDF. Por medio de diferentes métodos que detecten estas actualizaciones y luego se trasladan al modelo de datos RDF.

2.3.1 Actualización incremental de grafos RDF a partir de bases de datos relacionales.

Este método se enfoca a la actualización incremental, cuando se produce un cambio en dicha base de datos, con el menor coste en tiempo y recursos computacionales posible. De ahí que sea relevante seguir un enfoque incremental, es decir evitar la repetición de procesos desde cero y aprovechar los resultados existentes antes del cambio.
Ilustración 4.



Ilustración 4 Álvarez, L. (2014) Procedimiento para la actualización incremental de grafos RDF. Recuperado de <http://biblioteca.sistedes.es/articulo/actualizacion-incremental-de-grafos-rdf-a-partir-de-bases-de-datos-relacionales/>.

Para la detección de cambios se realiza mediante la detección de cambios en la base de datos relacionales, posteriormente se generan las tripletas RDF que representan dichos cambios. Después se actualiza el grafo RDF con las tripletas generadas.

- Detectar cambios en la base de datos relacional (BDR)

Para detectar los cambios en la base de datos relacional se usan triggers que capturan los datos involucrados en las operaciones de inserción, eliminación y/o modificación realizadas sobre las tablas. Para almacenar los datos de dichos cambios se crea una tabla auxiliar, cuyos atributos hacen referencia al nombre de la tabla modificada, el identificador de la fila donde se realizó el cambio, así como el tipo de modificación realizado (inserción, eliminación o modificación). A cada tabla del modelo relacional se asocian dos triggers, uno para las operaciones de inserción y modificación y otro para la eliminación. La diferencia entre ambos radica en capturar el identificador nuevo de la fila creada o el identificador de la fila eliminada, que ya no existiría. Ambos triggers han de invocar la misma función que se encarga de insertar los datos descritos anteriormente en la tabla auxiliar. Cada vez que los cambios recogidos en la tabla auxiliar sean alineados con el modelo de datos RDF se borrarán las tuplas de la tabla, evitando el cúmulo de información innecesaria en la base de datos relacional.

- Para generar las tripletas RDF

Se utilizó el lenguaje de alineación RDB de RDF mapping language (R2RML), que es el lenguaje de alineación estándar del W3C. Para realizar la alineación mediante R2RML es necesario generar un documento de alineación que hace referencia a la estructura de las tablas de la base de datos relacional para obtener los datos contenidos en ellas. La

primera vez que se realiza la alineación, los datos se extraen directamente de las tablas del modelo relacional y se genera el grafo RDF correspondiente. Cuando se proceda a realizar la actualización de la base de datos relacional, las tripletas serán generadas a partir de la tabla auxiliar generada en la fase anterior.

- Actualizar tripletas en el Grafo RDF

Una vez que se tienen las tripletas RDF y se conoce a priori el tipo de operación que implicó el cambio en la base de datos relacional. Por lo tanto, es posible realizar las consultas SPARQL de inserción, eliminación o modificación correspondientes, que se encargarán de reflejar los cambios identificados en la base de datos relacional en el grafo RDF. (Sciences, 2014)

2.3.2 Generación de Datos Semánticos a partir de una Base de Datos Relacional de una Institución de Educación Superior

Esta investigación se enfoca en el análisis y migración de información académica, almacenada en bases de datos relacionales (RDB) hacia metadatos tipo Resource Description Framework (RDF). Para llevarlo a cabo se han desarrollado mapeos a cada una de las tablas posibilitando su interpretación y reutilización, por medio de vocabularios estandarizados o relacionados al desarrollo de la Web Semántica. Como contribución se ha implementado un proceso original, combinando técnicas para mejorar los datos de la Web tradicional.

El método utilizado para la obtención de datos abiertos RDF, consta de cuatro etapas:

- Análisis de los Datos: Se usó un gestor de Base de Datos (MySQL), para el proceso de migración y almacenamiento de los nuevos datos.
- Generación de mapeos: Se usaron las clases y propiedades definidas por la ontología DBpedia, así como también otros estándares como: Dublin Core (DC), Friend of a Friend (FOAF), entre otros.
- Obtención de datos RDF: Para esta etapa se realizó el volcamiento de los datos a formato RDF, lo que permitió su posterior publicación y reutilización. Estos estarían representados en forma de tripletas y contienen toda la información de la RDB original.
- Visualización de los datos: Se ejecutó consultas por medio de lenguaje SPARQL EndPoint (FrontEnd), que permite a los usuarios acceder y consultar datos RDF mediante el lenguaje SPARQL (León, 2014).

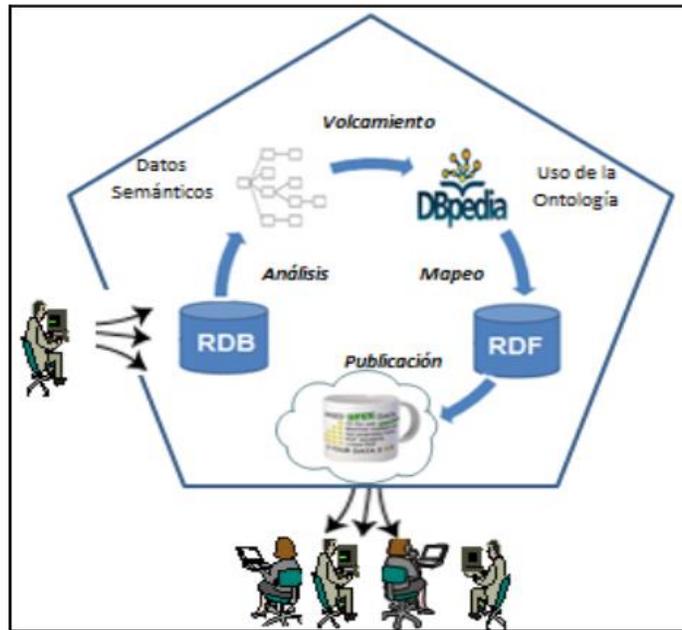


Ilustración 5 León, F. (2014). Topología de implantación. Recuperado de https://www.researchgate.net/publication/262413150_Generacion_de_Datos_Semanticos_a_partir_de_una_Base_de_Datos_Relacional_de_una_Institucion_de_Educacion_Superior.

2.3.3 Creación de aplicaciones Web inteligentes para un mundo más inteligente

Esta propuesta ofrece un visión de alto nivel Model-View-Controller (MVC) de una aplicación Web inteligente, y describe los módulos centrales en la presentación, el procesamiento y las capas de acceso de datos.

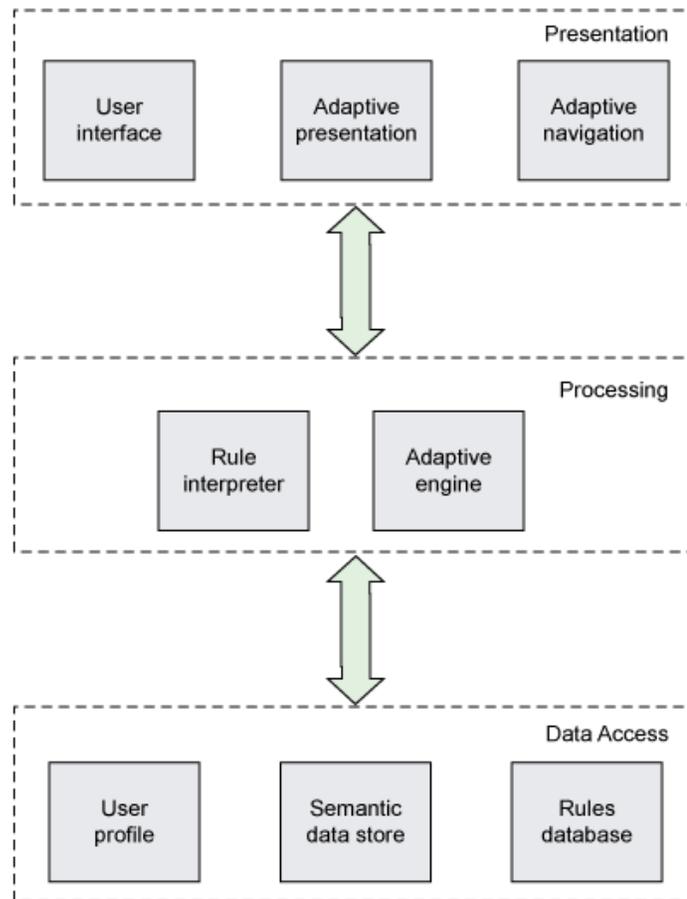


Ilustración 6 Ramachandran, K. (2011) Arquitectura MVC de una aplicación Web inteligente. Recuperado de <https://www.ibm.com/developerworks/ssa/library/wa-aj-smartweb/>.

- Capa de acceso de datos

La capa de acceso de datos suministra un almacenamiento de datos semánticos, un perfil de usuario y la base de datos de reglas. El almacenamiento de datos semánticos corresponde a la representación semántica de los datos de aplicaciones. Una diferencia importante entre la mayoría de las aplicaciones Web actuales y las aplicaciones Web inteligentes es que la representación de datos en una aplicación Web inteligente se define semánticamente y no sólo lógicamente. El perfil de usuario almacena las preferencias del usuario en términos de preferencia de contenidos y preferencia de interfaz de usuario, como así también detalles del usuario como edad, género, etc. La base de datos de reglas almacena el conjunto de reglas que vincula las preferencias del usuario o el perfil de usuario a los contenidos, las interfaces de usuario y los temas en la aplicación. Los datos de la aplicación, sus relaciones, perfiles de usuario, preferencias y reglas se pueden almacenar usando cualquier base de datos relacional como DB2, Oracle, etc.

- Capa de procesamiento

La capa de procesamiento suministra el intérprete de reglas y también el motor adaptable.

El intérprete de reglas lee las reglas almacenadas en la base de datos y las activa adecuadamente según el estado del usuario en el sistema. El estado de un usuario se puede definir según el tema/subtema actual, el desempeño en temas, etc. El motor adaptable cumple la función de adaptar la presentación y la navegación a las preferencias del usuario y las reglas, como así también al estado del usuario en la aplicación. El intérprete de reglas y el motor adaptable se pueden implementar usando lenguaje estándar orientado a objetos como código Java, C++, .NET, etc. con subprocesos o temporizadores que se pueden usar para aplicar una regla específica en el escenario adecuado.

- Capa de presentación

La capa de presentación suministra la interfaz de usuario y la presentación y navegación adaptables. La interfaz de usuario en una aplicación Web inteligente deberá ser lo suficientemente flexible como para crear la página según los elementos de presentación y navegación disponibles en lugar de un diseño de página fijo como ocurren con la mayoría de las aplicaciones Web actuales. La presentación adaptable se encarga de adaptar los contenidos de la página según el perfil y las preferencias del usuario. La función de la navegación adaptable es adaptar los elementos y los escenarios de navegación según el perfil y las preferencias del usuario. Una interfaz de usuario adaptable con presentación adaptable y navegación adaptables se puede implementar usando un marco de interfaz de usuario avanzado como Websphere Portal, java server faces (JSF), Struts o Tapestry, brindando capacidades dinámicas para adaptar los contenidos de la interfaz de usuario a los diferentes usuarios en variados escenarios. (Ramachandran, 2011)

2.3.4 TRIOO, estudio e implementación de modelos de datos RDF en lenguajes orientados a objetos

Se propone una tecnología que permite de una manera sencilla utilizar datos RDF directamente desde lenguajes orientados a objetos, permitiendo que el origen y la forma de los datos no tenga consecuencias para los diseños orientados a objetos, a la vez que la semántica de los datos se refleje de la manera más fiel posible. Es importante mencionar que en esta investigación se tiene una arquitectura bien definida basada en los argumentos. Lamentablemente por falta de tiempo estas pruebas no se realizaron de manera sistemática; por lo tanto estas pruebas no han sido totalmente completadas y documentadas.

El catálogo de anotaciones se compone de:

- RdfResource: es la anotación principal que identifica que un objeto Java se puede mapear a un recurso RDF. Dado que el nivel de RDF se ha sacado fuera del sistema de tipos de Java, esta anotación permite mapear una clase Java con n clases RDFS/OWL.
- RdfProperty: es la anotación que permite indicar que un atributo sustenta también una relación en un grafo RDF.
- Literal: anota un atributo para un literal RDF, un tipo de dato simples en definitiva (primitivos, fechas, etc.).
- DynamicRdfProperty: permite cierta dinámica en la definición de los mapeos
- URI: provee una forma de declarar como se compone las URI de los recursos en función del valor de determinados atributos en los objetos, que como es lógico previamente son codificados para formar una URI válida.

Registro de clase:

El paquete `java.lang.reflection` provee al lenguaje Java de una potente API de reflectividad. De esta manera se puede interrogar en tiempo de ejecución a los objetos sobre su estructura y estado. Este paquete incluye determinados componentes para describir abstractamente los recursos RDF en Java de una forma lo más flexible posible. (Fernandez, 2010).

A continuación, se muestra la arquitectura planteada por trio:



Ilustración 7 Fernández, S, (2010). Arquitectura en modelos de datos RDF en lenguajes orientados a objetos. Recuperado de [Http://trioo.wikier.org/resources/doc/trioo-tfm.pdf](http://trio.wikier.org/resources/doc/trioo-tfm.pdf)

2.4 Software actual para implementar la web semántica

2.4.1 Semantic Web Builder

Es una suite de productos que sirven como plataforma para el desarrollo de aplicaciones y portales semánticos, creada por Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación (INFOTEC), centro público de innovación y desarrollo tecnológico del CONACYT. Está conformada por una serie de herramientas que permiten la evolución de los sitios web convencionales (sin significado), hacia los portales que cumplan con la visión de la Web Semántica (con significado), permitiendo exponer en formatos estándar como RDF todo el conocimiento generado, además de contar con un Modelo (ontología) que permita el intercambio de esta información entre los diferentes sistemas.

Permite definir en una ontología (OWL) la estructura de información, la arquitectura de objetos, sus dependencias y su representación gráfica, de forma que mediante estos elementos se pueda ejecutar un proceso automatizado que genera el código del modelo de objetos definido y su persistencia en una base de datos semántica (basada en RDF). (Builder)

Los mecanismos desarrollados para representar conocimiento, principalmente los de la Web Semántica, evidencian las limitantes de algunos de los paradigmas de desarrollo de software actuales. Estas limitantes están relacionadas principalmente con los lenguajes que usan estructuras de datos predefinidas y el uso de la orientación a objetos con herencia múltiple restringida o no permitida. Por otro lado, las bases de datos relacionales, que son aún de las más utilizadas para persistir la información en los sistemas de software, no cuentan con la flexibilidad para modelar escenarios donde la naturaleza y estructura de los datos cambia constantemente y requiere recuperarse con un alto nivel de desagregación. Dada esta problemática, INFOTEC ha creado el Desarrollo Orientado a la Semántica. El desarrollo orientado a la semántica es un framework que pone en práctica las ideas y estándares de la Web Semántica dentro del proceso de desarrollo de software. Es decir, toma como punto de partida el conocimiento de un dominio capturado en una ontología para generar de manera automática la arquitectura y funcionalidad básica de un sistema de software cuyos objetos son persistentes de manera nativa en formato RDF. Lo anterior permite contar con un mecanismo acelerado de construcción semiautomática de aplicaciones, permitiendo responder de manera rápida a los requerimientos cambiantes de cada proyecto de desarrollo y facilitando el mantenimiento de las aplicaciones. (Solis, 2013)

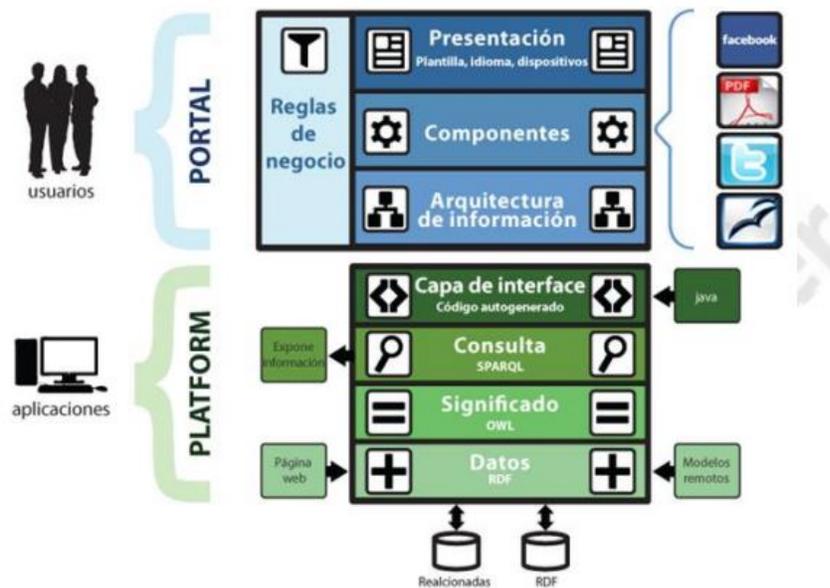


Ilustración 8 Infotec, (2010). Arquitectura de semantic web builder. Recuperado de www.semanticwebbuilder.org.mx.

La plataforma semántica maneja una organización jerárquica de objetos, este nivel jerárquico está hecho en base a la función que desempeñan. A continuación, se listan estos objetos:

- SemWB Base: Este objeto es la base de SWB y se encarga de manejar las utilerías E-mail e I/O; control de errores en Logs, manejo XML (Extensible MarkupLanguage), Connection Pool.
- RDF Library: Es el segundo nivel y contiene las librerías RDF (Resource Description Framework) el cual es un modelo de datos que representa recursos y relaciones que se puedan establecer entre ellos.
- SemWB Triple Store: Es el tercer nivel, es una herramienta que almacena y realiza búsquedas sobre triplas RDF usando el framework de JENA.
- SemWB Platform: Es el cuarto nivel, y se encarga de la construcción de modelos semánticos. La unión entre el objeto SemWBTriple Store y SemWB Platform la realizan la Sincronización de aplicaciones y el mantenimiento en memoria cache de SWB
- Core Model: Es el quinto nivel y mantiene las ontologías que definen a SWB. Se modela una Ontología, se genera código y finalmente se personaliza código Core Model mantiene el código generado

2.4.2 Protégé

Es un programa informático de tipo open source. Su función consiste en la creación, edición y mantenimiento de ontologías, posee la capacidad de generar de manera automática el código fuente de cualquier ontología creada con Protégé en formato RDF/OWL.

Jena es un framework Java desarrollado por HP Labs que sirve para construir aplicaciones para la Web Semántica. Proporciona un entorno de programación para RDF, RDFS y OWL, incluyendo un motor de inferencia basado en reglas. El código de Jena es libre.

El framework de Jena incluye:

- Una API RDF
- Lectura y escritura de RDF en formato RDF/XML, N3 y N-Triples.
- Una API OWL.
- Almacenamiento en memoria y persistente.

Jena1 almacena las tripletas de una forma normalizada. Una tabla de sentencias almacena referencias al sujeto, predicado y objeto, y el valor real de las URIS y de los literales se almacena en dos tablas adicionales. Este diseño es eficiente en espacio, porque los valores de texto se almacenan una única vez, independientemente del número de veces que aparezcan en las tripletas. Sin embargo, una unión a tres bandas se requiere para operaciones de búsqueda.

Además, la tabla de sentencias no es escalable para grandes bases de datos. Jena 2 utiliza una visión des normalizada, multi-modelo y de triple almacenamiento.

El módulo de bases de datos de Jena proporciona una implementación de la interface del modelo Jena, pero con la capacidad de almacenar y recuperar datos RDF utilizando una base de datos. Actualmente, soporta MySql, Oracle y PostgreSQL para almacenamiento persistente (Ruiz R. G., 208).

2.4.3 Tecnología D2RQ Mapping Lenguaje

D2RQ es una plataforma que permite al acceso a la base de datos relacionales de una forma de solo lectura, El acceso se realiza mediante RDF para realizar lectura de la información almacenada en la base de datos Ofrece acceso basado en RDF con el contenido de las bases de datos relacionales y sin tener que replicar en un almacén RDF. Usando D2RQ puede consultar una base de datos no RDF usando SPARQL acceder al contenido de la base de datos como datos enlazados en la Web crear vertederos personalizadas de la base de datos en los formatos de RDF para cargar en una

información de acceso al almacén de RDF en una base de datos no RDF utilizando el Apache Jena API D2 RQ es un software de código abierto y publicado bajo la licencia Apache. El código fuente está disponible en GitHub.

Realiza una construcción automática del mapeo analizando la estructura de la Base de Datos.

Resumiendo, haciendo uso de D2RQ se puede:

1. Hacer consultas sobre una base de datos que no sea de tipo RDF haciendo uso de SPARQL.
2. Acceder al contenido de la base de datos, como linked data sobre la web.
3. Crear volcados personalizados de la base de datos en formato RDF.
4. Acceder a información de una base de datos que no sea de tipo RDF, haciendo uso de la API Apache Jena.

Funcionamiento d2rq

Conexión de una base de datos Se debe de definir la conexión java databaseconnectivity(JDBC) a una base de datos relacional.

Propiedades para la conexión de la BD: Para realizar la conexión con la base de datos es necesario seguir la estructura de D2RQ para Base de datos, como también es necesario especificar el tipo de columnas de la base de datos utilizadas por D2RQ.

Estas son algunas propiedades:

- **d2rq:jdbcDSN:** la dirección de la base de datos JDBC
- **d2rq:jdbcDriver:** Controlador de la base de datos JDBC
- **d2rq:username:** El nombre de usuario de la base de datos
- **d2rq:password:** La contraseña de la base de datos
- **d2rq:resultSizeLimit:** Establece un límite superior para el número de resultados devueltos desde grandes bases de datos.
- **d2rq:fetchSize:** Valor que se recupera para cada una de las solicitudes de la base de datos
- **d2rq:startupSQLScript:** Sirve para la conexión y pruebas. Para cargar desde el sistema de archivos en relación a la ubicación del archivo de asignación, utilice esta sintaxis: `d2rq: startupSQLScript<archivo: script.sql>; #`
- Para una conexión de una base de datos a un largo plazo existe una propiedad:
- **jdbc: keepAlive:** Este define el valor en segundos de conexión.

2.5 Sistemas de gestión de base de datos relacionales con soporte para RDF.

2.5.1 JENA

Para desarrollar aplicaciones basadas en RDF, OWL o lenguajes similares se precisan librerías para leer y procesar las ontologías definidas en estos lenguajes. Sin embargo, con diferencia, el parser de RDF y OWL más popular es Jena2, desarrollado por Hewlett Packard, que permite leer, recorrer y modificar grafos tanto RDF como OWL desde un programa Java. Jena2 permite además guardar las ontologías tanto en RDF textual como en formato de base de datos, lo que es importante para grafos muy grandes.

2.5.2 KOWARI

Es una base de datos construida con el propósito de almacenar, recuperar y analizar metadatos. Está escrita en Java, y su código es libre. Soporta tanto metadatos RDF como OWL. (kowari)

A continuación, se muestran algunas características:

- Soporte nativo de RDF
- Múltiples modelos de bases de datos por servidor.
- Lenguaje de consultas similar a SQL: TucanaQuery Lenguaje, aunque se pretende dar soporte a SPARQL en el futuro.
- Funcionalidad búsqueda completa de texto.
- Soporta tipos de datos.
- Soporta las especificaciones de W3C y cumple sus recomendaciones de uso.

Conectividad

- JRDF: Java RDF, un conjunto de API e implementaciones base para RDF utilizando Java.
- Software Developers Kit (SDK)

2.5.3 RDF Gateway

Es una plataforma diseñada para la Web Semántica. Es un servidor de aplicaciones, un servidor Web, y un servidor de bases de datos RDF. El motor de base de datos de RDF Gateway no es como los de los Sistemas de Gestión de Bases de datos Relacionales

(RDBMS). Soporta un lenguaje de comandos similar a SQL, múltiples bases de datos, tablas almacenadas en memoria o disco, cursores y transacciones.

La principal diferencia entre RDF Gateway y una base de datos relacional es como se representa la semántica de un modelo de datos. Una base de datos relacional codifica la semántica implícitamente utilizando elementos de almacenamiento como tablas, columnas, y claves únicas. Las aplicaciones cliente tienen que ser desarrolladas para comprender la semántica de la información relacional. Esto hace que sea difícil compartir información entre sistemas que no comparten un esquema relacional idéntico. RDF proporciona un estándar abierto para especificar la semántica de un modelo de datos. Además, RDF Gateway no necesita codificar la semántica utilizando tablas, columnas y claves ajenas. Con RDF Gateway todas las tablas tienen el mismo formato y las claves ajenas no son necesarias.

Motor de Base de datos Nativo:

- Utiliza comandos similares a SQL para realizar consultas de datos RDF.
- Razonamiento deductivo vía reglas de inferencia (librerías para OWL y RDFS)
- Alto rendimiento en disco y tablas de memoria.
- Transacciones con niveles aislados.
- Tablas virtuales que integran fuentes de datos externas (páginas Web, documentos locales, bases de datos relacionales, etc.)
- Consultas federadas a través de múltiples fuentes de datos internas y externas.
- Búsquedas de texto.
- Encriptación de datos
- Soporte para la serialización RDF/XML, N3 y NTriples
- Seguridad a nivel de tabla y de sentencias.

Capítulo 3

DISEÑO Y DESARROLLO

Tras analizar las actuales herramientas que permiten que las organizaciones publiquen contenidos en internet con valor semántico se detectó que dichas opciones requieren que la información que poseen las empresas se migre a una nueva plataforma, o que se trabaje con bases de datos diferentes de las que las empresas ya poseen.

Ya se ha abordado previamente que cuando la información es almacenada en más de un lugar, no se garantiza su homogeneidad e integridad.

El objetivo de este trabajo es diseñar una estrategia de evolución de las plataformas de las empresas e instituciones a la web 3.0 con el menor impacto a su infraestructura.

Es por ello que se realizó un nuevo componente de software bajo las siguientes premisas:

- Permita migrar los sistemas de información existentes de las organizaciones a la web 3.0
- Obtenga la información de las mismas bases de datos relacionales con las que la organización ya cuenta.
- Pueda incorporarse al código fuente de aplicaciones existentes.
- Sea de uso general.
- Permita que los contenidos estáticos se presenten actualizados y sincronizados con su equivalente en la base de datos.

La estrategia presentada a continuación se desarrolló con el lenguaje Java para sistemas de información empresariales bajo el patrón MVC (modelo-vista-controlador).

Fue necesario desarrollar una API para la generación de archivos de metadatos RDF a partir de clases con anotaciones. Un API es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

En el patrón MVC, la interfaz gráfica muestra el estado que guarda el modelo. La información es obtenida de la base de datos y enviada a las capas superiores como objetos DTO (objeto de transferencia de datos). Son estos objetos los que se pretende presentar como información pública, estática con metadatos y como información privada, dinámica y operativa al mismo tiempo.

El modelo

El modelo de objetos permite integrar en un único procedimiento de modelado tanto la dinámica de datos como los diversos aspectos sobre la estructura de los datos. Un modelo de objetos consiste en una serie de objetos. Los objetos se pueden considerar como entidades extendidas, entidades con un comportamiento determinado. El modelado de objetos utiliza los siguientes constructores:

Objeto: *Un objeto es cualquier cosa que tiene una existencia independiente procedente de un universo de discurso que se modela.*

Clase objeto: *Una clase objeto es una abstracción de las características comunes de un grupo de objetos. Los objetos pueden tener en común atributos, relaciones o métodos, los modelos de objetos no suelen indicar objetos pero si sus clases.*

Atributo: *un atributo es una propiedad de una clase objeto. Relación. Una relación es una conexión entre clases objeto.*

Método. *Un método establece un comportamiento para una clase objeto. (Davies, 2014)*

Cabe mencionar que cuando se desea almacenar el estado de los objetos en una base de datos se utiliza un API de persistencia. La persistencia se entiende como la capacidad de los objetos a sobrevivir más allá del tiempo que dura la ejecución de la aplicación (o rutina) en la que se crean.

El API de persistencia encapsula el acceso a la base de datos y le permite al programador trabajar orientado a objetos; En esta investigación se reutilizan dichos objetos sin afectar la funcionalidad del sistema.

Cuando se diseña una base de datos que representa un ámbito determinado, las entidades son representadas como tablas. Del mismo modo, en programación orientada a objetos, cada entidad del dominio de un problema es codificada como una clase.

A la correspondencia entre tablas y clases de una aplicación se le llama mapeo objeto-relacional. Son precisamente estas clases las que interesa publicar como archivos RDF pues cada una de ellas representa un concepto en particular; En la ilustración 9 se muestra un ejemplo del Mapeo Objeto-Relacional.

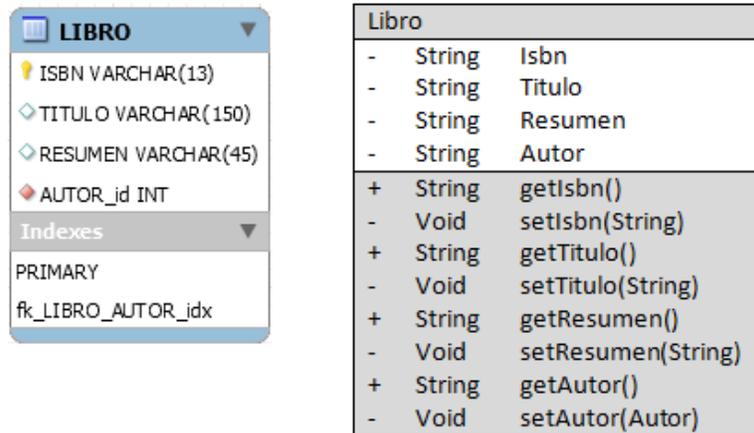


Ilustración 9 Mapeo objeto-relacional, tabla y clase para la entidad libro (elaboración propia)

Por cada tabla de la base de datos se crea una clase en la aplicación y por cada columna de cada tabla se le agrega una variable a cada clase. La capa de persistencia nos permitirá encapsular el comportamiento necesario para leer, escribir y borrar objetos en el almacenamiento persistente (base de datos).

Generación de archivos RDF desde clases con anotaciones

Las API de persistencia requieren del programador que especifique explícitamente cual clase corresponde a cuál tabla y cuál atributo de la clase corresponde a cuál campo y llave de la base de datos; esto mediante el uso de anotaciones. En la **Ilustración 10** la clase “Ubicación” tiene encima del nombre de su clase una anotación **@Table** que indica que hay una tabla homóloga en la base de datos “capacitación”. También muestra que la variable “id” es la llave primaria de la base de datos pues la clase tiene una anotación **@id** encima de su método accesor. Las tres variables de la clase (id, edificio, salón) de tipo *String* tienen un equivalente en la base de datos de tipo *varchar*.

Las anotaciones **@table**, **@column**, **@id** forman parte del estándar JPA (Java persistence API) del JDK Java.

El API de persistencia permite realizar operaciones CRUD orientadas a objetos que afectan la base de datos. Así por ejemplo la instrucción:

- Insert into UBICACIÓN (edificio,salon) values ('edificio de posgrado', 'auditorio');

Se trabaja orientada a objetos de la siguiente manera:

- Ubicación ubicación = new Ubicación();
- ubicación.setEdificio(“edificio de posgrado”);
- ubicación.setSalon(“auditorio”);
- sesión.guardar(ubicacion);

Las anotaciones en Java se incluyeron a partir de Java 5. Éstas son metadatos que se pueden asociar a clases, miembros, métodos o parámetros. Nos dan información sobre el elemento que tiene la anotación y permiten definir cómo queremos que sea tratado por el framework de ejecución. Además, las anotaciones no afectan directamente a la semántica del programa, pero sí a la forma en que los programas son tratados por las herramientas y librerías.

En Java, una anotación se define por medio de la palabra reservada `@interface` (estas clases están en el módulo de `coreRDF` en el package de anotaciones). Hay que tener ciertas consideraciones en cuenta a la hora de crearlas:

- Cada método dentro de la anotación es un elemento.
- Los métodos no deben tener parámetros o cláusulas `throws`.
- Los tipos de retorno están restringidos a tipos primitivos, `String`, `Class`, `enum`, anotaciones, y arrays de los tipos anteriores.
- Los métodos pueden tener valores por defecto

Pero no sólo disponemos de las anotaciones por defecto de Java, si no que podemos crear las nuestras propias y así disponer de metadatos útiles en tiempo de ejecución.

`java.lang.reflection` una herramienta del JDK Java que nos permite inspeccionar clases y ejecutar acciones. De esta manera se puede interrogar entiendo de ejecución a los objetos sobre su estructura y estado, como también obtener las anotaciones que se definieron.

Cabe mencionar que los DTO (Objetos de transferencia de Datos) son sólo contenedores de datos que serán utilizados para el transporte de datos entre capas y niveles. Los DTOs son serializables con el fin de brindar transferencia de datos a través de la JVM.

Las clases de java en las que se ejecutan las operaciones CRUD se llaman DAO (data acces object). No es obligatorio que cada DAO implemente todas las operaciones CRUD.

Siguiendo la misma estrategia del API de persistencia, se desarrolló un conjunto de anotaciones que describe la correspondencia entra las clases y las entidades RDF

Hay que recordar que la sintaxis RDF define un conjunto de tres elementos llamado tripleta compuesto de un “sujeto, un predicado y un objeto” (w3c, RDF concepts, 2014), a este conjunto se le llama grafo RDF. En el sujeto se describe de quien se está hablando; en el predicado que también se le conoce como propiedad se representa una conexión entre un sujeto y un objeto, normalmente una relación del sujeto con algo o un atributo del sujeto; en el objeto se almacena el valor de lo declarado.

Cabe mencionar que estas anotaciones no afectan el funcionamiento del API de persistencia así que se pueden implementar en los mismos DTO a consideración del programador.

Se muestra en la ilustración 10 un ejemplo de un DTO cuando se implementan las anotaciones:

```

// RDF
@RDFSujeto(url="Http://localhost:8080/public/cursos", extension="html")
// JAXB
@XmlRootElement
// JPA
@Entity
@Table(name="Ubicacion",catalog="capacitacion")
public class Ubicacion {

    private Integer id;|
    private String edificio;
    private String salon;

    @RDFIdentifier
    @Transient
    public String getIdentificador(){
        return "Ubicacion"+this.id;
    }

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="id",nullable=false)
    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    @RDFPropiedad(
        url="Http://localhost:8080/cursos",
        prefix="uaem",
        nombre="edificio" )
    @Column(name="edificio")
    public String getEdificio() {
        return edificio;
    }
}

```

Ilustración 10 Clase DTO con anotaciones

Para señalar las clases y campos a publicar, se diseñaron: Un conjunto de 4 anotaciones RDF (Tabla 3) y un conjunto de 15 Anotaciones Dublin Core (Tabla 4).

Tabla 3 Anotaciones desarrolladas

Anotación	Definición
@RDFSujeto(url, extension):	Esta es una anotación que se le coloca a las clases que se van a convertir a archivos RDF. En esta se especifica la URL del repositorio en la que se van a ubicar los archivos y la extensión del mismo. Ej.

	http://biblioteca.com/libros, http://biblioteca.com/autores
@RDFIdentifier	Se le coloca al método que retorna una cadena de texto cuyo valor es utilizado como nombre del archivo RDF. Es por tanto el identificador y no debe producir valores repetidos. Ej. libro32165.rdf; por lo que lo hace único.
@RDFPropiedad (url, prefix, nombre)	Se le coloca a las variables alfanuméricas cuyo valor se insertará como nodo en el archivo RDF. Es necesario especificar el prefijo, etiqueta y namespace del nodo a generar.
@RDFResource(url, prefix, nombre)	Se le coloca a las variables que hacen referencia a otras clases. En el archivo RDF se agregará un hipervínculo a otro archivo RDF. El hipervínculo se construye dependiendo de las anotaciones de la clase referenciada.

La anotación RDFPropiedad sirve para especificar campos personalizados en el archivo RDF. Los campos del estándar Dublin Core se contienen en un conjunto de anotaciones adicionales a continuación se muestra en la tabla 4.

Tabla 4 Anotaciones del conjunto Dublin Core

Anotación	Definición
@RDFDublinCoreTitle	Título
@RDFDublinCoreSubject	Tema
@RDFDublinCoreDescription	Descripción
@RDFDublinCoreSource	Fuente
@RDFDublinCoreLenguaje	Lenguaje
@RDFDublinCoreRelation	Relación
@RDFDublinCoreCoverage	Cobertura
@RDFDublinCoreCreator	Autor
@RDFDublinCorePublisher	Editor
@RDFDublinCoreContributor	Colaboradores
@RDFDublinCoreRights	Derechos
@RDFDublinCoreDate	Fecha
@RDFDublinCoreType	Tipo de recurso
@RDFDublinCoreFormat	Formato
@RDFDublinCoreIdentifier	Identificador

Ejemplo de clase (diagrama) con anotaciones a continuación se muestra en la tabla 5:

Tabla 5 Diagrama de clases con anotaciones.

@RDFSujeto(url=" http://www.uaemex.com/cursos ", extensión="rdf")
Curso
Privateid:Integer
Privatenombre:String
Privatedescripción:String
Privatefechalnicio:Date
Privateprofesor:Profesor
@Transient
@RDFIdentifier
Public getIdRdf:String
@RDFDublinCoreTitle
Public getNombre:String
@RDFDublinCoreDescription
Public getDescripcion:String
@Transient
@RDFPropiedad(url=" http://www.uaemex.com/cursos " ,prefix="uaem" , nombre="fechaCurso")
Public getFechaCurso:String
@Transient
@RDFDublinCoreCreator
Public getCreador:String
@RDFResource(url=" http://www.uaemex.com/profesores " , prefix="uaem" , nombre="profesor")
Public getProfesor:Profesor
Otros getters&setters

Podemos observar, que cumplir con todas las etiquetas del estándar Dublin Core depende de que la información esté disponible en la base de datos. De otro modo deberán añadirse los campos faltantes. Aunque algunos atributos derivados pueden tratarse orientados a objetos, mediante métodos extra que concatenen variables. En el ejemplo anterior los getters marcados @transient son métodos cuyo valor no es exactamente igual al ofrecido por los atributos y no son del interés del api de persistencia, como pueden ser campos numéricos y fechas que deben ser convertidas a cadenas de texto.

Las clases con anotaciones son procesadas mediante el *API REFLECTION* de Java. Esta suite contenida en el JDK permite inspeccionar objetos y clases y obtener información de sus variables y métodos. El API reflection permite además invocar la ejecución de sus métodos por su nombre. Las anotaciones y los valores obtenidos de la

inspección de los objetos generan temporalmente una estructura de datos de la cual se genera finalmente el contenido RDF como cadena de texto.

El resultado puede por ejemplo publicarse en una sección pública del servidor de aplicaciones y el resto del sitio web hará referencia a ellos mediante hipervínculos. O puede generarse un archivo de noticias RSS para anunciar los cursos vigentes, etc. Más adelante se muestra un ejemplo de archivo RDF generado

Se creó un modulo para tener acceso a la base de datos esta contiene un paquete llamado coreRDF donde contiene tres paquetes:

1. package anotaciones
2. package domino
3. packageutilis

La **ilustración 11** muestra el diagrama de clases del paquete de anotaciones. Para crear la anotación dentro del fichero "RDFPropiedad.java" simplemente escribimos como en todas las clases "public" seguido del nombre de la anotación. Posteriormente se escribe la palabra que contiene arroba "@interface", Dentro de esta clase tenemos que declarar las variables que utilizamos al usar la anotación, que fueron:

- publicStringprefix();
- publicStringurl();
- publicStringnombre();

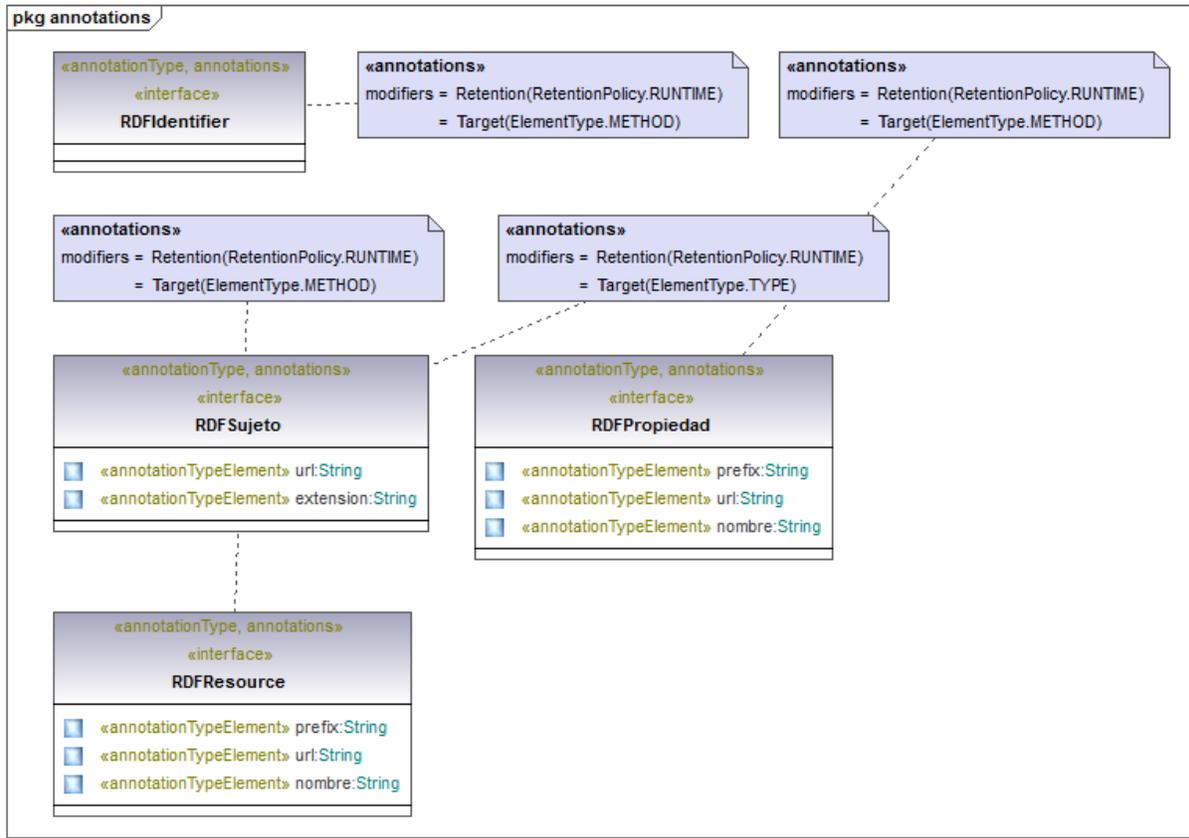


Ilustración 11 Diagrama de Clases (elaboración propia)

Las clases con anotaciones, son procesadas para formar una estructura de datos temporal que más tarde será convertida a nodos RDF. La **ilustración 12** muestra el Diagrama de clases del paquete *domain* correspondiente a las clases que forman dichas estructuras de datos.

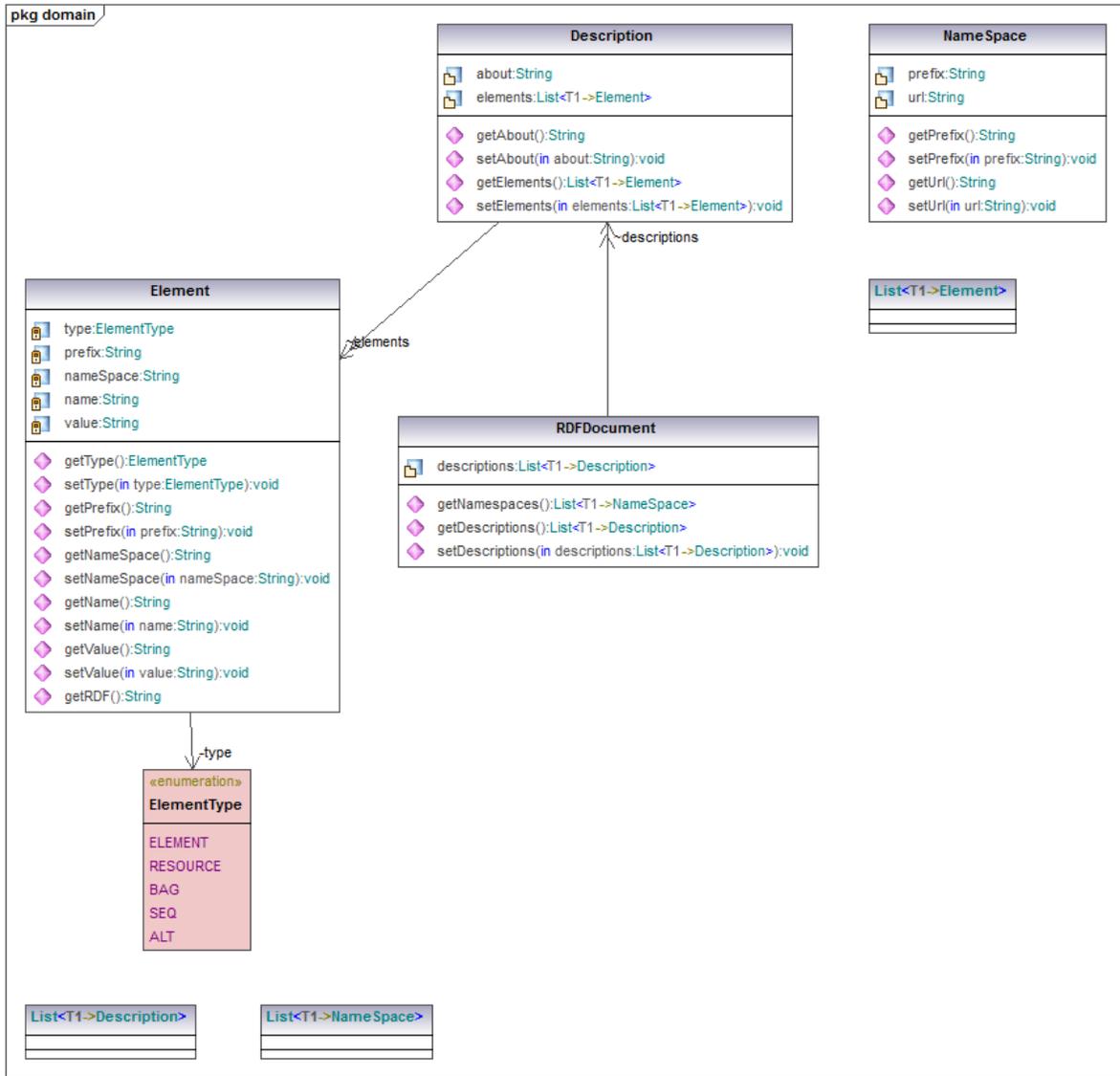


Ilustración 12 Diagrama de clases del paquete Domain

Las clases que ayudan a procesar los objetos con anotaciones y obtener de ellos el RDF se hallan en el paquete *utils* del API desarrollada (**ilustración 13**). Se debe crear un objeto del tipo “ContextoRDF” pasándole una lista de las clases a convertir. Dicha clase generará un archivo xml por cada clase:

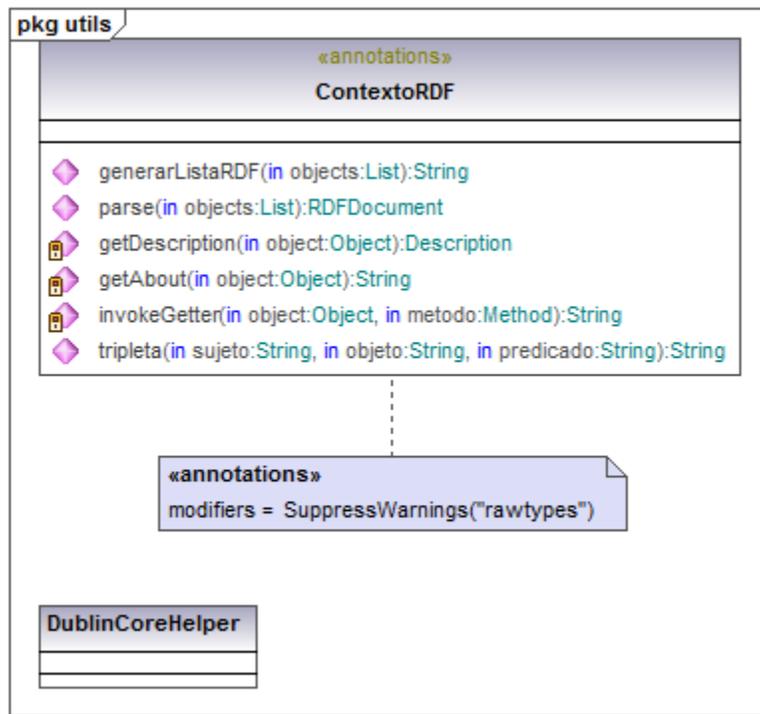


Ilustración 13 Diagrama de clase ContextoRDF

3.1 Diagramas de secuencia

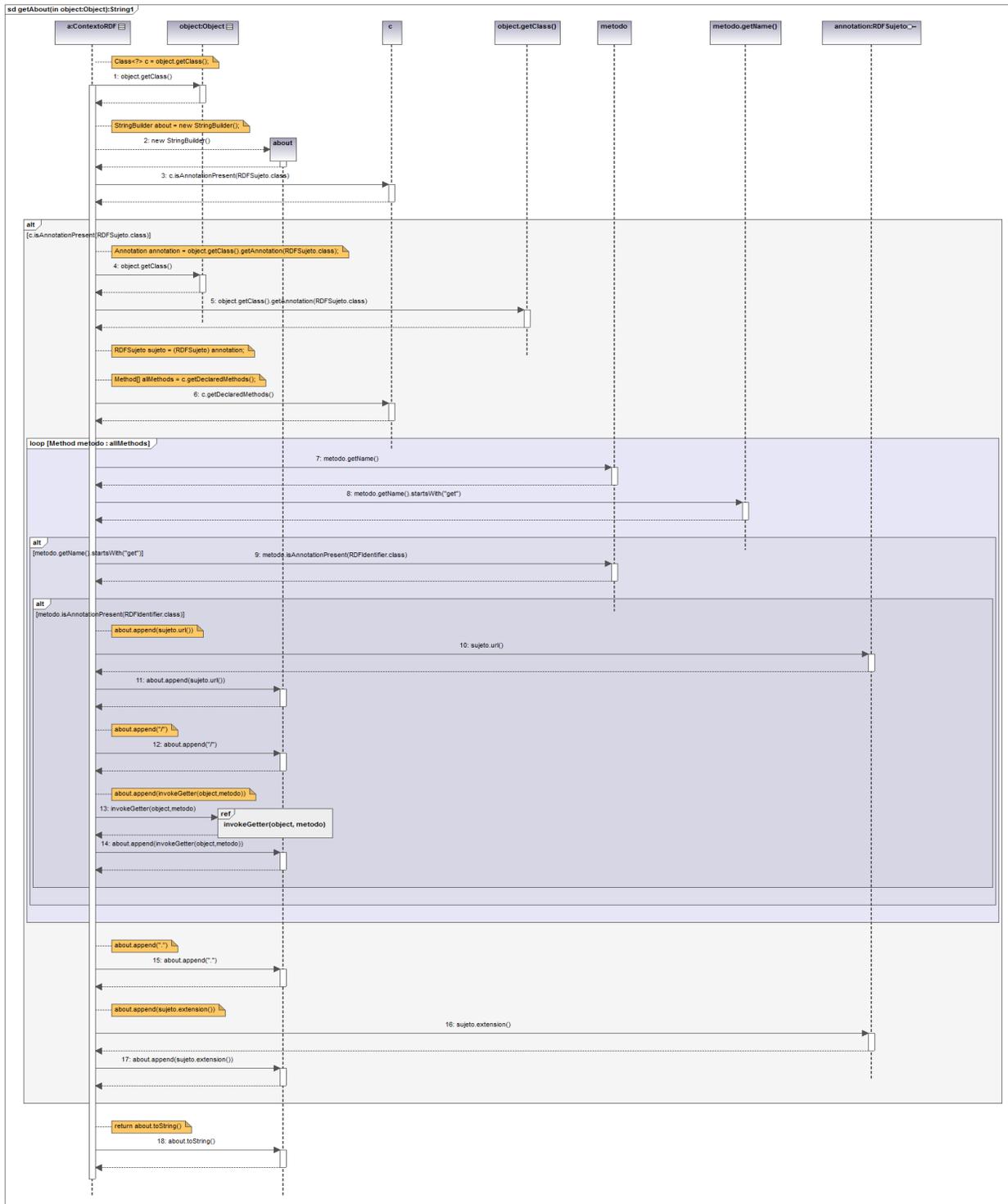


Ilustración 14 Diagrama de secuencia de la clase ContextoRdf:

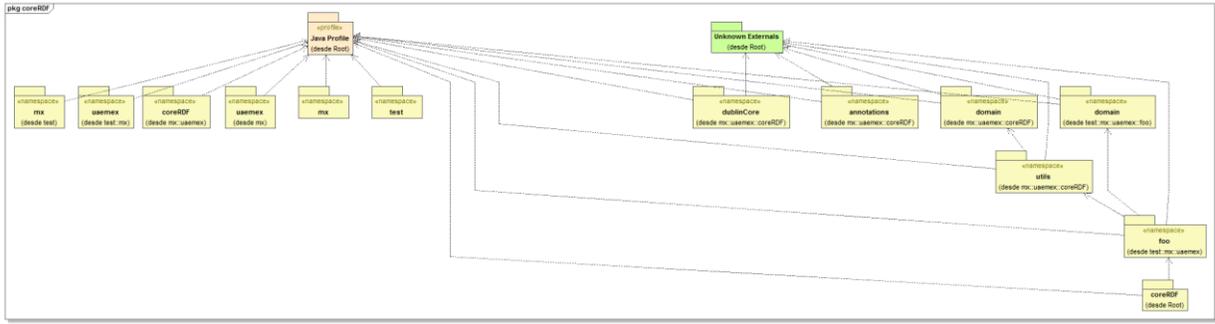


Ilustración 15 Diagrama de paquetes

propietario	uaemex
propiedades	nombre completo coreRDF::mx:uaemex::coreRDF nivel de acceso Public «namespace» True
diagramas Propios	Contenido de coreRDF
miembro Propio	annotations domain dublinCore utils
origen de la relación	Dependencia Java Profile
aparece en el diagrama	Contenido de uaemex Dependencias entre paquetes de coreRDF El contenido de coreRDF y todos los subpaquetes
hipervínculos	Contenido de coreRDF

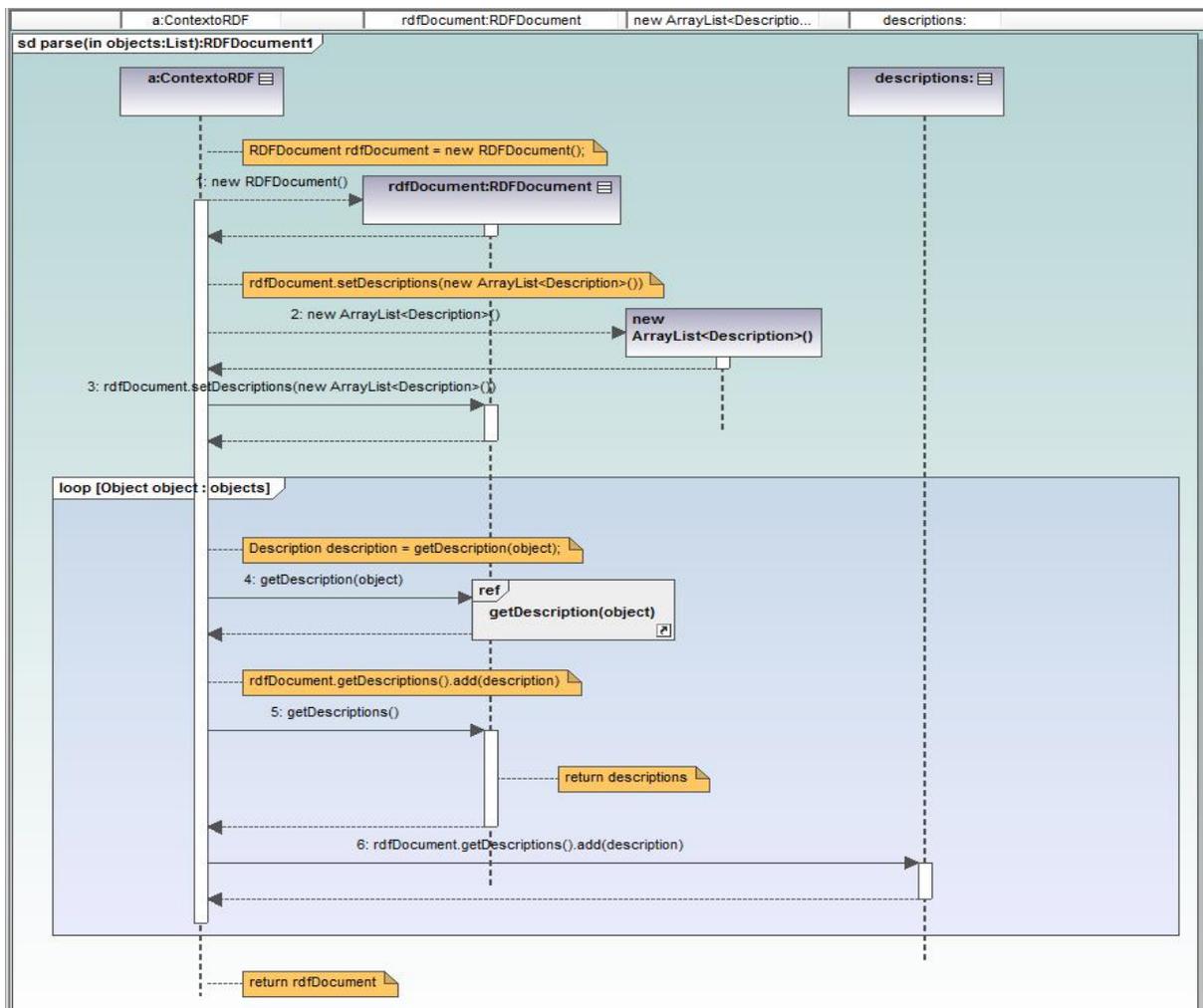


Ilustración 16 Diagrama de Secuencia de la clase Parse

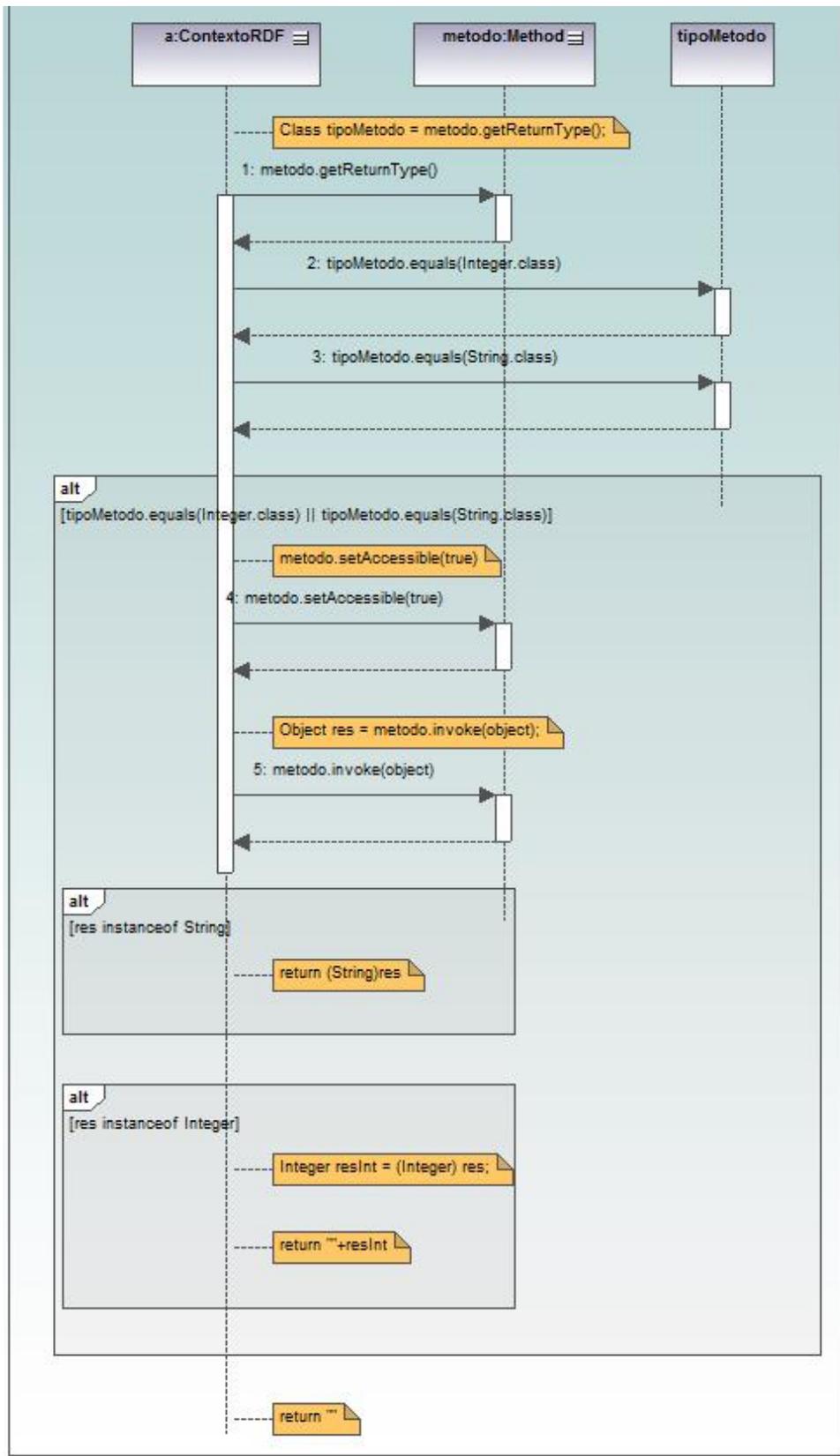


Ilustración 17 Diagrama de secuencia InvokeGetter

Panorama general

Para implementar la generación de los archivos de metadatos se crea un módulo en la capa lógica de negocios para los RDF; para que posteriormente se publique la información de manera estática en un sitio web público y de forma paralela se guarde el RDF en un repositorio; En la **ilustración 18** se muestra la arquitectura de esta propuesta;

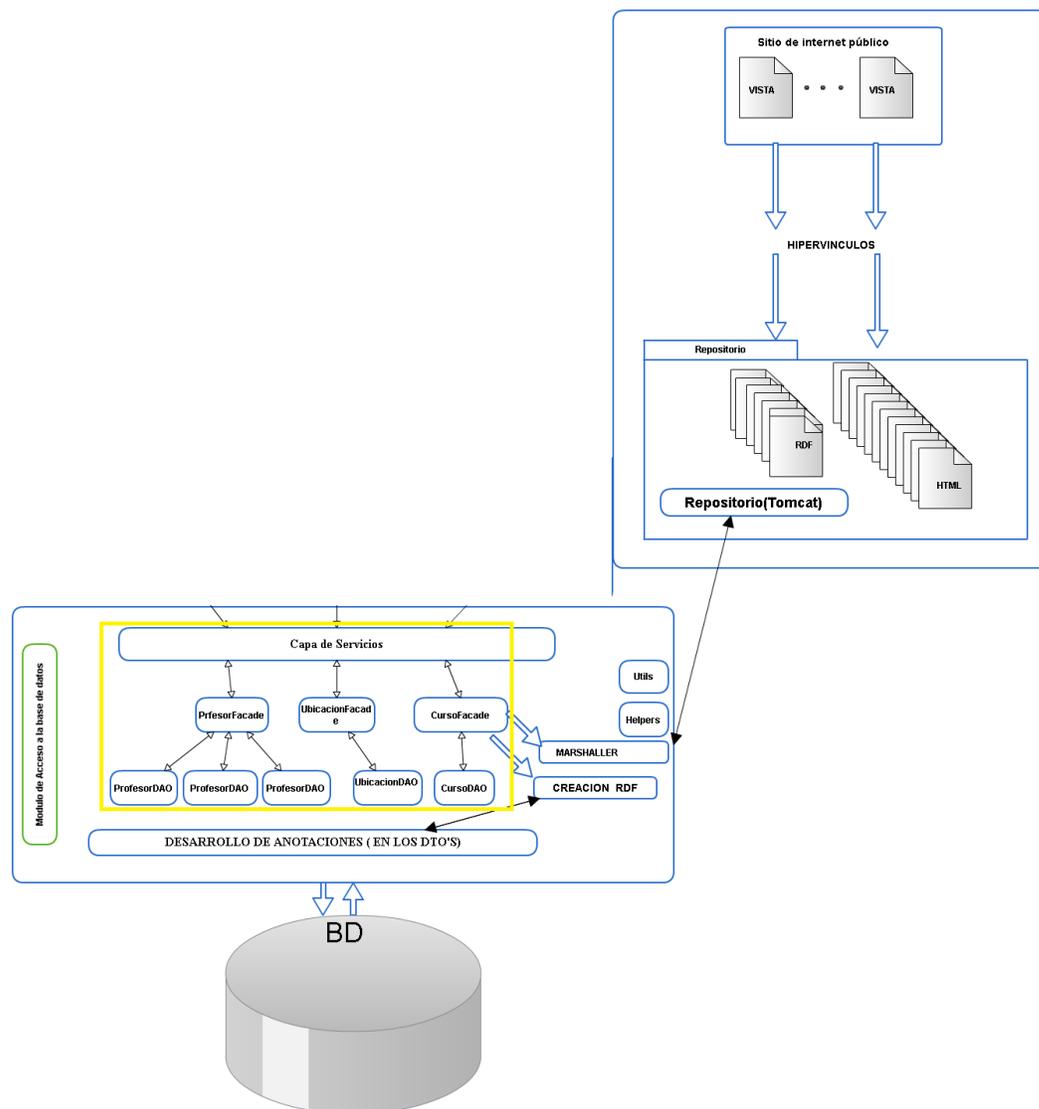


Ilustración 18 Arquitectura de esta propuesta

Las vistas presentan el estado del 'modelo'. Para el caso de las vistas se creó una página estática por cada una de las clases, se utilizó la api de JAXB (Java Architecture for XML Binding) que nos permitió generar archivos XML para la conversión de objetos en archivos XML a esto se le llama **marshalling**, cabe mencionar que esta api también utiliza anotaciones a partir de objetos.

Una vez que se tiene el XML se utilizó el lenguaje **XSLT** ya que nos permite generar documentos a partir de documentos XML, cabe mencionar que la hoja de estilo XSLT es el documento que contiene el código fuente del programa, es decir, las reglas de transformación que se van a aplicar al documento inicial.

En este caso se desarrolló una clase marshaller RDF que inspecciona las propiedades y datos de un objeto utilizando el API "reflection" de Java y que accede a los valores de sus métodos utilizando el método invoke. A partir de la obtención de estos valores y de las configuraciones especificadas en las anotaciones se concatenan las secciones del archivo RDF en una cadena de texto. La **ilustración 19** muestra el diagrama de clases del paquete marshalling.

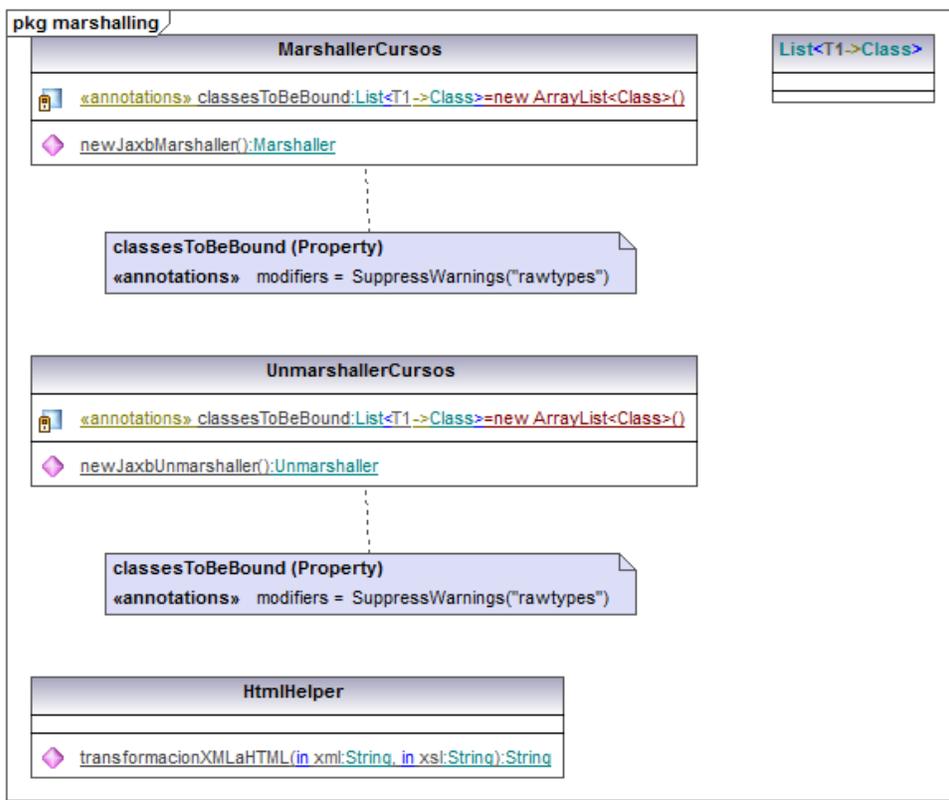


Ilustración 19 Diagrama de clases Contenido de marshalling

Es requerido que la conexión de la base de datos se encuentre desarrollada bajo un api de persistencia; esto presupone que las entidades del dominio están modeladas con objetos, por lo que nos será fácil mandar a llamas a las anotaciones que se crearon. La **ilustración 20** muestra el diagrama de clases del dominio del ejemplo a exportar.

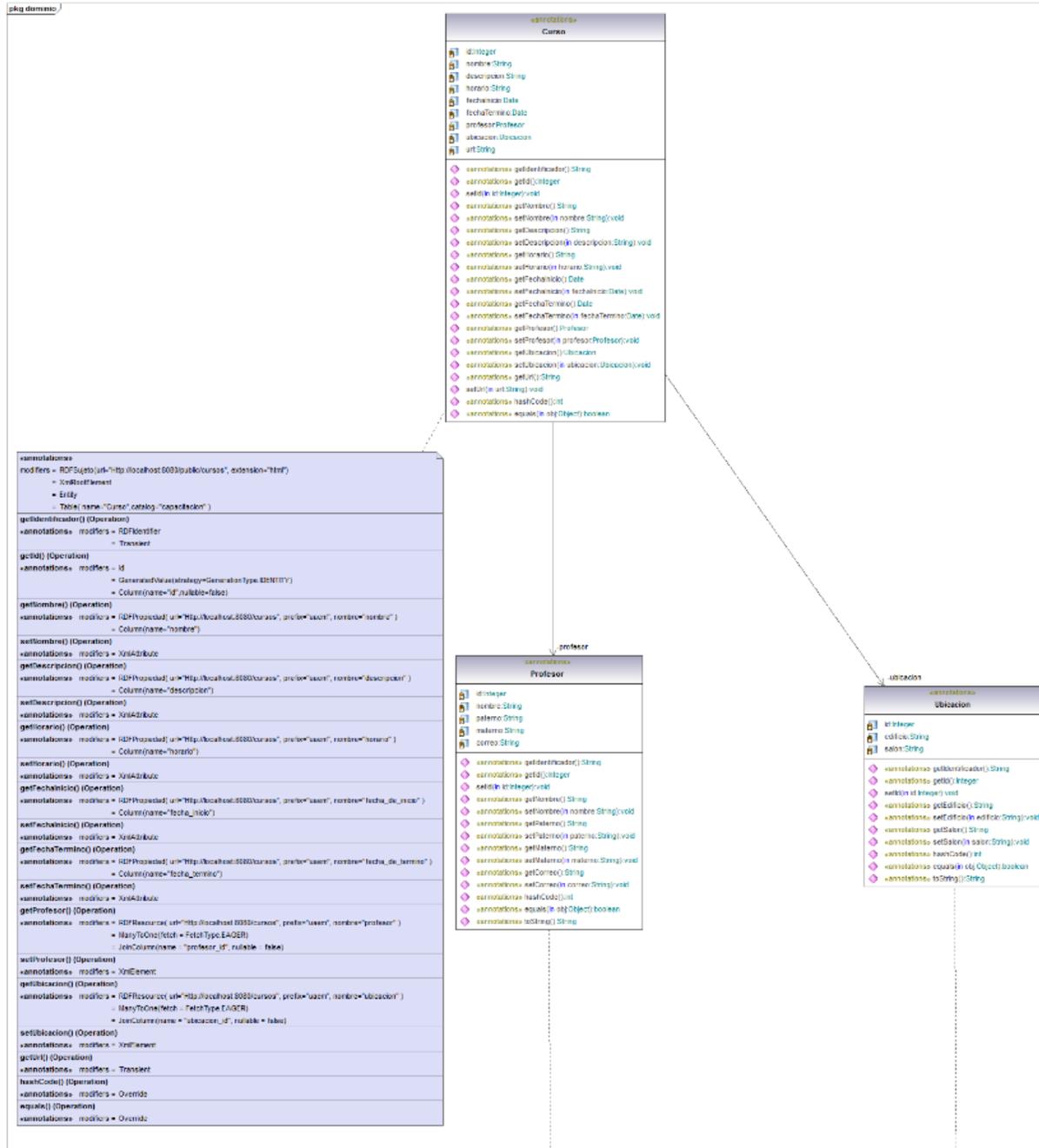




Ilustración 20 Diagrama de clases Contenido del dominio

Mientras que las clases DAO realizan tareas específicas de acceso a la base de datos, La capa superior *facade* puede desencadenar para una sola instrucción varias acciones. Los métodos CRUD de la capa facade invoca a su vez la ejecución de una o más acciones en los DAO. Pero además es aquí donde los DTO interceptados son convertidos a RDF para su publicación. La **ilustración 21** muestra el diagrama de clases del paquete facade.



Ilustración 21 Diagrama de clases Contenido de implementación.

Los contenidos RDF son almacenados en una sección pública del servidor de aplicaciones y el resto del sitio web hará referencia a ellos mediante hipervínculos. Con la finalidad que los usuarios externos y para que los robots puedan leer los RDF se creó un repositorio donde se almacenan los en formato RDF XML de cada curso.

Para cada uno de los cursos se guarda en una carpeta pública dentro del servidor Tomcat; La **ilustración 22** muestra el RDF generado.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:uaem="Http://localhost:8080/cursos">
<rdf:Description rdf:about="Http://localhost:8080/public/cursos/Curso2.html">
<uaem:profesor rdf:resource="Http://localhost:8080/public/cursos/Profesor1.html"></uaem:profesor>
<uaem:ubicacion rdf:resource="Http://localhost:8080/public/cursos/Ubicacion1.html"></uaem:ubicacion>
<uaem:nombre>BASES DE DATOS</uaem:nombre>
<uaem:descripcion>BASICO</uaem:descripcion>
<uaem:fecha_de_inicio></uaem:fecha_de_inicio>
<uaem:fecha_de_termino></uaem:fecha_de_termino>
<uaem:horario>LUNES A VIERNES 9 A12</uaem:horario>
</rdf:Description>
</rdf:RDF>
</xml>
```

Ilustración 22 Generación automática del RDF (elaboración propia)

Capítulo 4

CASO DE ESTUDIO Y RESULTADOS

Para dar un ejemplo de la implementación del componente de software generado se desarrolló un sistema con el patrón arquitectura de software MVC (Modelo, Vista, Controlador) donde en la capa de presentación se muestra la interfaz de usuario, la capa lógica de negocios recibe las peticiones del usuario y se envían las respuestas tras el proceso, y la capa de datos es donde residen los datos. Implementar el patrón MVC permite dar un mantenimiento fácil a las aplicaciones web debido a una clara separación entre las capas de presentación, lógica de negocio y acceso a datos. La **ilustración 23** muestra la arquitectura general del sistema y la implementación del api.

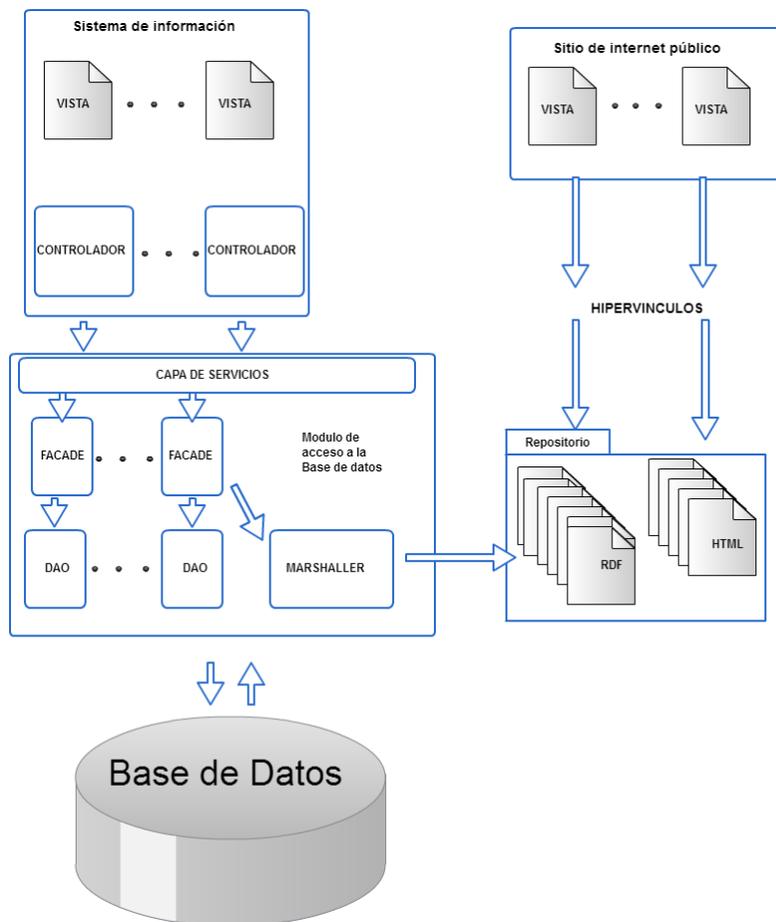


Ilustración 23 Implementación de la propuesta (elaboración propia)

Para la implementación de esta técnica se desarrolló un Sistema de Información que administra los cursos ofertados por profesores en una Universidad. Esta aplicación requiere que sea una aplicación web dinámica en la que cierta parte de su información necesita de su divulgación como contenido abierto, en formato estático y semántico para los usuarios externos, referenciable por una URL única. La aplicación web se desarrolló con la tecnología JSF 2.1 (Java Server Faces) implementada con la biblioteca de componentes de Primefaces 5.2; Se utilizó el framework de desarrollo de aplicaciones web Spring 4 y el API de persistencia Hibernate 4. La información es almacenada en una base de datos relacional MySQL. Todo esto, en un servidor Tomcat 7 sobre JVM 1.7 configurado para permitir la escritura en el servidor.

Se desarrollaron las clases descriptoras de los objetos mediante la API Reflection, para ello se definieron las notaciones necesarias para la aplicación. Se usó JAXB para obtener páginas HTML y los XML que permiten publicar los datos que se deseados. La aplicación web se modificó en la ejecución de sus operaciones CRUD (Create -Read -Update -Delete; las operaciones que se pueden realizar en cualquier elemento en sus catálogos, (Dittawit, 2012)). Las modificaciones permiten generar los respectivos archivos HTML estáticos que son guardados en una carpeta pública del servidor Tomcat.

La aplicación web permite darle seguimiento a la administración de cursos a continuación se muestra la interfaz de alta de un curso; En la **ilustración 24** se muestra Vista de alta de un nuevo curso.



Curso

Nombre:* BASE DE DATOS

Descripcion:* BASICO

Horario:* LUNES A VIERNES 9 A 12

Profesor:* ANGELA RIOS LUNA

Ubicacion:* edificio=1, salon=1

Guardar Regresar

Ilustración 24 Vista de alta de un nuevo curso (elaboración propia)

Para el alta de un curso se deberá tener acceso al sistema ingresando usuario y contraseña. Una vez que el administrador da de alta el curso; se muestra de manera estática para que los usuarios externos que no se loguen al sistema puedan visualizar la lista de cursos que se ofertan; En la **ilustración 25** se muestra la Vista Externa de la lista de cursos.

Cursos

Lista de cursos

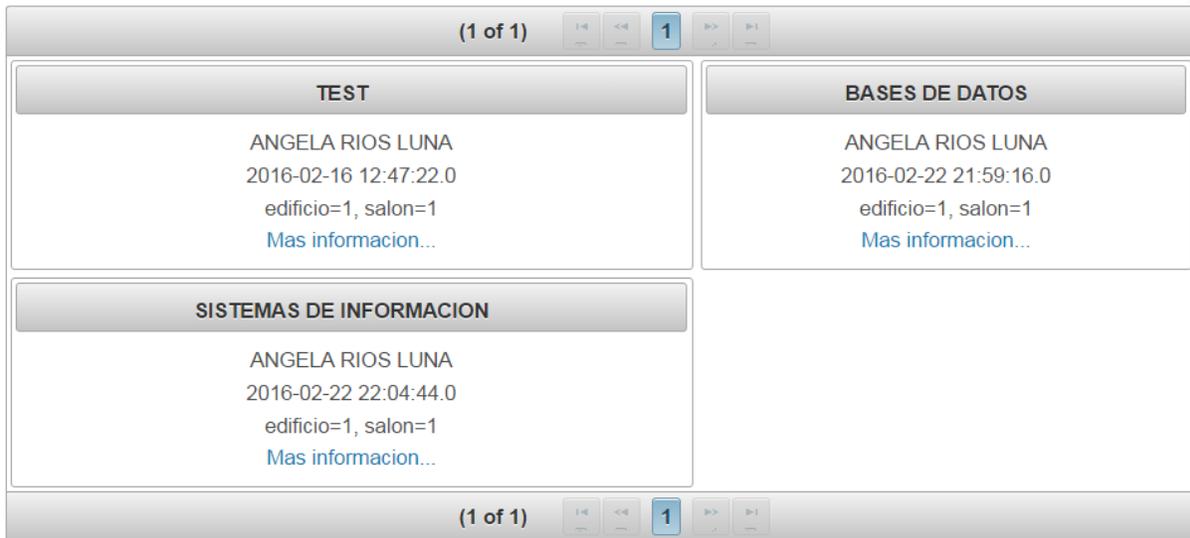


Ilustración 25 Vista Externa de la lista de cursos (elaboración propia)

Con el software desarrollado en este trabajo se logra generar de forma automática el RDF para cada uno de los cursos y este se guarda en una carpeta pública dentro del servidor Tomcat; a continuación, se muestra en la **Ilustración 26** el RDF generado.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/terms"
  xmlns:uaem="http://www.uaemex.mx/cursos">
  <rdf:Descriptionrdf:about="http://www.uaemex.mx/cursos/Curso1.html">
    <dc:title>matemáticas 1</dc:title>
    <dc:description>algebra</dc:description>
    <dc:creator>Juan_Perez_Hernandez</dc:creator>
    <uaem:fecha_de_inicio>Mon Oct 24 00:03:59 CDT 2016</uaem:fecha_de_inicio>
    <uaem:profesor
  rdf:resource="http://www.uaemex.mx/profesores/Profesor_Juan_Perez_Hernandez.html"></uaem:profesor>
    <uaem:horario>L-V 7:00 a 9:00</uaem:horario>
  </rdf:Description>
</rdf:RDF>
```

Ilustración 26 Tripletas de un curso con DC (elaboración propia)

La información generada mediante los RDF permite que sea visualizada de dos formas distintas, de acuerdo a la necesidad de la empresa, puede visualizarse en formato de tripleta (véase **ilustración 27**) o en forma de grafo (véase la **Ilustración 28**).

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://localhost:8080/public/cursos/Curso2.htm	http://localhost:8080/cursosprofesor	http://localhost:8080/public/cursos/Profesor1.html
2	http://localhost:8080/public/cursos/Curso2.htm	http://localhost:8080/cursosubicacion	http://localhost:8080/public/cursos/Ubicacion1.html
3	http://localhost:8080/public/cursos/Curso2.htm	http://localhost:8080/cursosnombre	"BASES DE DATOS"
4	http://localhost:8080/public/cursos/Curso2.htm	http://localhost:8080/cursosdescripcion	"BASICO"
5	http://localhost:8080/public/cursos/Curso2.htm	http://localhost:8080/cursoshorario	"LUNES A VIERNES 9 A 12"

Ilustración 27 Tripleta de un curso (elaboración propia)

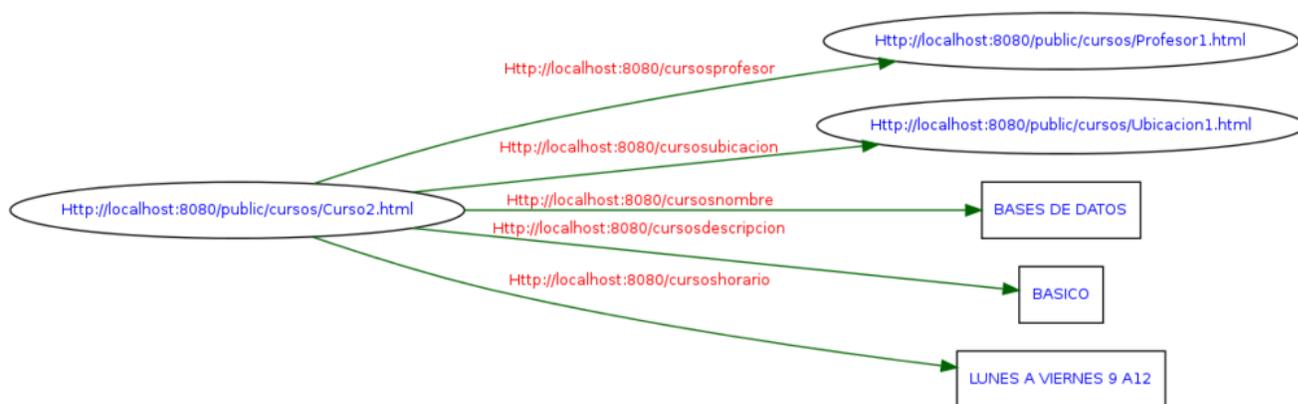


Ilustración 28 Representación de un curso en grafo (elaboración propia).

Capítulo 5

DISCUSIÓN

Las distintas estrategias y herramientas que permiten generar y publicar contenidos en formatos semánticos cubren distintas necesidades y tienen distintos requerimientos. La plataforma que las organizaciones deben seleccionar dependen de varios factores:

- ¿El código fuente de la aplicación puede modificarse?
- ¿La base de datos puede modificarse?
- ¿La organización ya tiene información?
- ¿Qué parte de la información es pública?
- ¿Cuánto presupuesto se enfocará a la migración hacia la web 3.0?
- ¿Qué estándar se requiere?

El desarrollo presentado en este trabajo surge de la necesidad de implementar la web 3.0 en sistemas de información que ya existen en instituciones que ya poseen información almacenada en bases de datos relacionales, esto requiere un impacto mínimo en su base de datos, software e infraestructura.

Para sistemas en los que el código fuente no puede modificarse, puede haber una aplicación independiente que publique la información como archivos RDF en otra ubicación, pero esto genera un problema de sincronización, pues hay que detectar el momento en el que hay cambios en la base de datos. Esto puede solucionarse a través de la implementación de columnas de estatus, triggers y tablas virtuales.

En el caso contrario, en el que el código fuente pueda modificarse, pero la base de datos no, las operaciones de transformación y sincronización pueden realizarse a nivel de objetos.

Por ejemplo, en una tabla el nombre y apellidos pueden estar en diferentes campos de una tabla, pero el DTO puede concatenarlos en una sola variable cuando se requiere el nombre completo en un solo valor string. Y no solo eso, no necesariamente deben corresponder las clases mapeadas a las tablas de la base de datos, sino solo a aquellas estructuras que cumplen con los requerimientos de la web semántica. En estos casos se necesitaría un paso intermedio de transformación de objetos de persistencia a objetos de mapeo RDF.

Es además diferente la estrategia de generación de archivos RDF *a priori* que *a posteriori*. Cuando la institución va a partir de 0 es más fácil adecuar la base de datos que cuando

ya tiene información. Además de que es diferente la generación de la primera generación de archivos que la sincronización entre las nuevas entradas y el contenido ya migrado.

Cuando no toda la información de la base de datos es pública, o cuando los campos de la base de datos no se corresponden directamente a lo requerido, pueden generarse tablas virtuales que no permiten la agregación, edición ni eliminación de información a la vez que no es necesario otorgarle a los programadores acceso completo a la base de datos o a la información contenida.

Las tablas virtuales también permiten la ejecución de aplicaciones de propósito general. Por ejemplo, imagine que con el API se desarrollara un módulo que sirve para hacer públicos los medios de contacto de los empleados. Si la aplicación se desarrolla independientemente de quien lo valla a utilizar, al momento de implementarlo en una organización bastará con diseñar un conjunto de tablas virtuales que cumpla con las especificaciones del módulo.

Capítulo 6

CONCLUSIONES

Se desarrolló una API que facilita la publicación de contenidos semánticos a partir de objetos obtenidos de bases de datos relacionales, y esta sirve para aquellos desarrolladores que requieran convertir su sitio web estático o dinámico a un sitio web 3.0; este representa una herramienta de desarrollo que transforma información almacenada en una base de datos relacional, datos estructurados a contenido semántico que contiene datos semiestructurados; genera nuevos canales de comunicación y comercio para las empresas e instituciones.

El programador puede establecer una estrategia que le permita actualizar los datos en la base de datos a través de la aplicación, también que actualice la información de los archivos de metadatos involucrados, manteniéndolos de esta forma sincronizados.

También se podrá desarrollar para los usuarios finales que no sean programadores aplicaciones de propósito general que funcione con requerimientos específicos y genere contenido semántico. Una de las ventajas de las anotaciones RDF es que reutilizan las clases de persistencia sin entrar en conflicto con los sistemas ya desarrollados, disminuyendo así el esfuerzo necesario para su implementación y los riesgos de modificar la operación del sistema.

La incorporación de metadatos permite la interacción entre sitios de internet y crea nuevos canales de comunicación y comercio para las empresas e instituciones. Los sitios semánticos posicionan a las instituciones como generadoras de contenidos confiables y referenciales, aumentando su presencia en internet.

El estándar de definición de metadatos Dublin Core es reconocido a nivel internacional. Es flexible, interoperable, semántico y se encuentra en constante crecimiento y evolución a través de una institución formal consorciada por la DCMI, cabe mencionar que el Dublin Core tiene lazos con Linked Data con la finalidad de lograr estandarizar los metadatos de la cabecera de los documentos se facilitaría su ordenación automática y mejoraría la efectividad de los motores de búsqueda

Agregar metadatos a un sistema de información aumenta su presencia en los motores de búsqueda. La implementación de vocabularios estructurados para controlar el contenido de estos metadatos y para organizar la información permite conseguir una organización más eficaz de las colecciones que redundará en una recuperación de información más pertinente.

Se debe combinar el desarrollo de estructuras de meta información y la potencia del procesamiento informático de las mismas. Ambas tendencias coincidirían en el planteamiento de la Web semántica, en la que se persigue una búsqueda inteligente que debería aprovechar el conocimiento estructurado y el valor añadido de lo humano embebido en las estructuras de conocimiento.

GLOSARIO

API: application programming interface. Conjunto de clases relacionadas y reutilizables diseñadas para proporcionar funcionalidad útil de propósito general, Nos permite como desarrolladores, evitar re codificar funcionalidad común.

CRUD: Create, Read, Update, Delete

DAO: data acces object

DC: Dublin Core

DTO: data transfer object

Facade: Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar.

Framework: Es un conjunto de clases cooperantes que constituyen un diseño reutilizable para una clase especifica de software, el framework determina la arquitectura de nuestra aplicación. Definirá la estructura general, su partición en clases y objetos. (Gamma, 2009)

FOAF: friend of a friend

JAXB: java architecturefor XML binding

JSF: java server faces

JPA: java persistence API

LOD: Linked Open data

MVC: Consiste en tres tipos de objetos. El Modelo es el objeto de la aplicación la Vista es su representación en pantalla y el Controlador define el modelo en que la interfaz reacciona a la entrada del usuario.

JDK: Java development kit

RSS: really simple syndication

RDF:Resource description framework

SPARQL: Protocol and RDF Query Language

SKOS: Simple Knowledge Organization System

OWL:ontology web language

URL: uniforma resourcelocator

WWW: worldwide web

W3C: consorcio worldwide web

XML: eXtensibleMarkupLanguage

XHTML:eXtensiblehypertextmarkuplenguaje

HTML:hypertextmarkup lenguaje

XSL: extensible stylesheetlanguage

XSLT: transformaciones XSL

ANEXOS

Search by Title or ISSN:

Select language 

INDEX  COPERNICUS
INTERNATIONAL



[Home](#) ⇒ [Journal passport](#) ⇒ [Journal content](#)

ICI Journals Master List 2014
Now available! Annual Report ICI Journals Master List 2014 summarizing the 2014 year with full list of journals and publishers from database of Index Copernicus.

[Index Copernicus Search Articles](#)

International Journal of Modern Engineering Research (IJMER)

ISSN:
2249-6645

Volume 5, 2015

Dublin Core RDF Metadata Automatic Generation from Stored Information in A Relational Database

Angela Ríos Luna¹, Irene Aguilar Juarez², Joel Ayala De La Vega³,
Leticia Dávila Nicanor⁴

^{1,2,3} (Centro Universitario UAEM Texcoco, Universidad Autónoma Del Estado De México, México)

⁴(Centro Universitario UAEM Valle De México, Universidad Autónoma Del Estado De México, México)

ABSTRACT: When a website is created, it is designed the way texts, images and sounds will be presented to the people. But it is equally important to establish the channels through other sites and software agents consume such information. Part of the information of Internet 3.0 sites is available in open formats or is described in metadata files. The structure of the semantic web sites enables filtering and automatic promotion of relevant content to users, and allows internet sites to interact. This paper describes the development of a software component for Java applications that generates Dublin Core standard RDF metadata files from classes with annotations. This would allow its integration into existing information systems with minimal or no changes to the database, with the additional advantage of synchronization between the information in the database and its semantic equivalent published online.

I. INTRODUCTION

The work area known as semiotics is the study of signs. Signs are considered the elements that compose communication; evidently communication is a social phenomenon as it involves human interactions and intentions, effective communication occurs when the intension of the issuer corresponds largely with the message interpretation made by the receiver. Signs and sign systems can be grouped into four interdependent levels: pragmatics, semantics, syntax and empirical. Pragmatics deals with the aspect of intentions, semantics with the meaning of a message, syntax with the formalism used to represent messages, and empirical takes on the used signals to encode and transmit a message. [1] Semantics is the study of signs reference. Communication involves using and interpreting signs. When we communicate, the issuer must state its intentions using some signs. In a face to face conversation, this will involve using linguistic signs. The receiver of the message must interpret the signs, that is to say, it must assign some meaning to the signs in the message. Semantics deals with this process [1].

Semantic Web is an extended web, provided with greater significance in which Internet users can find answers to their questions faster and easier thanks to a better-defined information [2]

Companies that offer Internet search services, generate keys whereby a site will appear as a search result. The sites are indexed starting with the home page and recursively following the hyperlinks that connect files together. Based on the large volume of content that exists on the Internet, and that increases day by day, Tim Berners-Lee, father of the internet and director of the World Wide Web Consortium, initiated a solution for dealing with this content explosion: let software find and manage Internet information for us. It means, changing the current document retrieval prototype for and from humans to delegate tasks to software.

W3C (World Wide Web Consortium) standards were created to lead the Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. The model of using the Web changes from being a document repository to be a great base of knowledge for advanced systems capable of performing complex tasks. Semantic Web, an approach by the creators of the WWW that attempts to create a new network for web applications, details a way to publish semantic content in RDF (Resource Description Framework) format as it is one of the standards defined by the W3C used for data exchange.

In 2007 the project *Linked Open Data* arose in the W3C, with the aim of spreading the semantic web. Each year a graph showing the links created between the semantic web network is published, it is possible to see that the growth has been exponential [3]. A current project is the DBpedia instance, this extracts information from Wikipedia to propose a semantic version.

Dublin Core is an organization dedicated to promoting the adoption of interoperable metadata standards, in order to be implemented XML (Extensible Markup Language) is generally used and it is based on the RDF Core was created in 1995 through initiatives by the associations of American librarians, and sponsored by OCLC (On Line Computer Library Center). It has its origin in an intellectual circle of Dublin, in the state of Ohio in the United States. Dublin core is associated with 52 expert researchers in the fields of library science, computer science, text encoders and related areas, with the aim of promoting the development of descriptive records of online information resources. Its progress has occurred in parallel with the development of the XML and the RDF; in October 2001, it was published the set of elements of Dublin vocabulary (DCMES, Dublin Core Metadata Element Set) which is defined by ISO in the ISO 15836 norm in 2009 and the Z39.85-2012 NISO standard [4].

This paper describes a strategy for automatic generation of semantic content (resources + metadata) from information stored in a relational database using the Java programming language. The RDF standard and international metadata format Dublin Core are used as they are the most cited and accepted, this is a product format an international and interdisciplinary effort with a very intense life, and the most influential in relation to the development of the theory of the use of metadata for information retrieval on the net.

II. PROBLEM

The web helps us to easily communicate with everyone at any time and at low cost, yet we generated an overload of information and diversity of information; organizations currently store in its databases confidential operational information that is not shown on their websites, and public information that needs to be spread. It is necessary to publish authorized information of an organization so that search services providers can index content and spread to the general public.

The need for sharing information between institutions and companies, without duplicated information. In recent years the number and size of RDF files with semantic data availability applied to any discipline has increased [5]; the data represented in RDF can be interpreted, processed and reasoned by software agents. Hence the change from the relational model to the RDF model for handling information is one of the challenges of current research [6]. So, according to the guidelines of the Web 3.0, our content requires a visual side (for people) and metadata (to be processed automatically by machines).

The information represented semantically uses different definitions owned by each company and is virtually impossible to relate it correctly as there are many distributed graphs all along the network. Therefore, the Linked Data initiative has emerged in order to link all available information into the graphs of the Web and build a unique graph that represents all the knowledge stored in the net [7]. Given the volume of information available on the Web, it is inevitable to automate some of the processes involved in the development of large structures of knowledge, applying the procedures that have been experimented since the retrieval of information. Currently there are companies that develop tools for automated classification [8]. The massive publication of data involves having to preserve the privacy of people and institutions, ensuring that it is not possible to deduce certain confidential information indirectly. Moreover, the fact that anyone can publish and link data on the web means that some aspects about the origin of the data, its quality and reliability of sources must be taken into account [9].

The web-based systems face challenges such as interoperability, the use of domain ontologies, contextualization and consistency of metadata. Such problems are related to the attempt to represent the information on the web in a way that computers can understand and manipulate. The main goal of semantics is to reuse the resources available in Web-based technologies through norms [10].

III. RDF

The Semantic Web architecture is focused on the RDF model, which is a universal format for data on the web. Currently, it is used for conceptual description or modeling the information that is implemented in web resources using syntax annotations.

The RDF provides a semantics for metadata; a better precision in resource exploration than the achieved by the search engines that track the full text, and better applications. All this while the corresponding schemes are developed. In general, the RDF provides the basis for generic tools that create, manage or search data on the web in an understandable way for computers, promoting the transformation of the web into a repository of information manageable by computers. The objectives of linked data are using data on the Web in the same way as the documents, linking data together, and using data as a collection of a self-related datasets, technologies must be available in a common RDF, in order to access existing databases either through a conversion or during runtime [12].

Advantages of the RDF data management as a database object:

- It is easier to model RDF applications.
- RDF data can be easily integrated with other company data.
- Reuse of RDF objects makes it possible to develop applications more efficiently.
- Objects abstraction, and encapsulation of RDF-specific behaviors make applications easier to be understood and maintained [1]
- No mapping between the RDF client-side objects, database columns and tables containing triplets is required.
- No additional configurations for storing RDF data are required [11].

1. Data model based on graphs

The structure of the syntax is set of three elements called triplets and is composed of a "subject, predicate and an object" [12], this set is called RDF graph, where an RDF graph can be displayed as a node in the graph and an arc of the graph, where each triplet is represented as a link node.

It is noteworthy that these three elements are represented by a URI (Uniform Resource Identifier).

- In the subject: it is described whom is being talking about,
- In the predicate, is also known as property: this represents a connection between a subject and an object, usually a subject's relationship with something or an attribute of the subject.
- In the object: the value of the declared.

Next, an example of a triplet is shown in Table 1:

Table 1 Basic Triplet

SUBJECT	PREDICATE OR PROPERTY	OBJECT
Http://ejemplo.com/pepe	Http://ejemplo.com/pepe/propiedad	"14/08/1988"

Dublin Core is an organization focused on "core metadata" for simple descriptions and generic resources. It comprises fifteen elements, it is noteworthy that "Dublin Core" has been ratified as IETF RFC 5013, ANSI / NISO standard Z39.85-2007, and ISO 15836: 2009.

Since 2000, the Dublin Core community focused on "application profiles" - the idea is that the Dublin Core metadata records would be used along with other specialized vocabularies to meet specific application requirements. It has worked with the World Wide Web Consortium in a metadata data generic model, the RDF. As part of an extended set of metadata terms DCMI, Dublin Core became one of the most popular vocabularies to be used with RDF [13], supporting the Linked Data movement.

Dublin Core aims to define a basic set of attributes that serve to describe all existing resources on the network, this format will help search engines in global retrieval of online information, and this goal turns it into a general purpose format.

Dublin Core defines fifteen elements which can be modified and scaled due to its flexibility, this allows authors of Web pages to encode their documents at the time of generating them.

Elements mainly related to the resource content:

1. Title
2. Subject
3. Description
4. Source
5. Language
6. Relation
7. Coverage

Elements mainly related to the resource when viewed as an intellectual property:

8. Creator (author)
9. Publisher (editor)
10. and other collaborations
11. Contributor (other authors / collaborators)
12. Rights.

Elements mainly related with the instantiation of the resource:

13. Date
14. Type (resource type)
15. Format
16. Identifier

V. RELATED WORKS

Recent studies show that it is possible to transport the relational model to the RDF graph-based model and the implementation of the Dublin Core metadata model. Through different methods and tools it is possible to do upgrades that reduce the cost of transferring information from databases to the RDF data model. The Research [6] focuses on the incremental update of RDF graphs from relational databases. When a change happens RDF triplets are generated representing updates, this information is updated on the RDF graph. This approach reduces the cost in time and computational resources compared to traditional techniques. This work shows the importance of following an incremental approach, which avoids processes from scratch and takes advantage of existing results before the changes.

Another research project is "TRIOO" in this work were studied and implemented RDF data models in an object-oriented language [14]"; the author proposes a technology that allows a simple way to use RDF data directly from object-oriented languages, allowing the origin and form of the data does not modify the object-oriented design, while the data semantics is reflected as accurate as possible. It is important to mention that this research has a defined architecture but did not culminate.

In the Article Integration of Heterogeneous Relational Data bases: RDF Mapping Approach [15] it is reported that the mapping process is important in data integration and knowledge of multiple heterogeneous relational data bases. The mapping from a relational database to a description that uses the RDF shows promising results without compromising the semantics of the data. Integration is also important when carrying out multiple simultaneous databases queries. Mapping is an important process to provide homogeneous points of view for the user. Mapping a relational database (or diagram entity relationship) to an RDF schema for data representation is a solution to overcome the problems of semantic and syntactic heterogeneities. The RDF is a common data format for the Semantic Web. It is a language that describes different kinds of objects (resources), their properties and the relationships between them by using statements.

VI. PROPOSED SOLUTION

The developed software component is responsible for generating RDF files from objects. This process can be part of an information system, website, RSS (Really Simple Syndication), web service, etc. It depends on the context programmer and the method by which ontologies are published.

An object is anything with an independent existence from the universe of discourse being modeled.

The object model allows to integrate within a single modeling procedure both dynamics data and the various aspects of the data structure. An object model is a series of objects. Objects can be considered as extended entities, entities with a certain behavior.

A class is the abstraction of the common characteristics of an objects group. Objects can have common attributes, relationships or methods, object models usually do not indicate objects but their classes. An attribute is a property of a class. A relationship is a connection between classes. A method establishes a behavior of a class.

In object modeling a relationship can be of three kinds:

Association. An associative relationship establishes a connection between instances of classes, and are defined by their cardinality and optionality. Therefore, the association relationships are common for those of structural models.

Generalization. This relationship establishes levels of abstraction between object classes. Generalization is the extraction process of the corresponding common features of a group of object classes and the suppression of the differences between them.

Aggregation. This relationship serves to gather a collection of different classes into a unit or aggregation. An aggregation relationship occurs between whole and its parts. An aggregation is an abstraction in which a relationship between objects is considered as another top-level object.

When a database that represents a particular environment is designed, entities are represented as tables. Similarly, in OOP, each entity in the problem domain is coded as a class[1]

A correspondence between tables and classes of an application is called object-relational mapping. An example of Object-Relational Mapping is shown in Figure 1. The persistence API encapsulates the access to the database and allows the programmer to work on an object-oriented way. Persistence is the ability of objects to survive beyond the execution of the created routine or program.

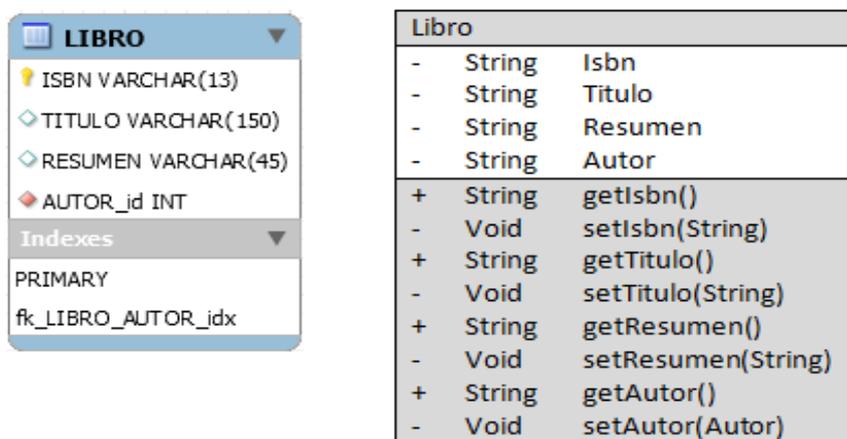


Figure 1 Object-relational mapping, table and class for the Book entity.

One class is created in the application per each table in the database, by each column in each table and a variable is added to the corresponding class. What class corresponds to which table and which variable to which column are specified using annotations. It is precisely these classes the ones meant to be published as RDF files as each one represents a particular concept.

In order to publish these objects as RDF files, the programmer places annotations in the classes whose objects are meant to be published. These annotations are a form of metadata that provide information about a program without being part of the program itself. The annotations have no direct effect on the operation of the code [16].

In order to mark the classes and publishing fields, there were designed: A set of four RDF annotations (Table 1) and a set of 15 Dublin Core Annotations (Table 2).

Table 2 Developed annotations

Annotation	Definition
@RDFSujeto(url, extension):	This annotation is placed in the classes that will be converted to RDF files. In this part the repository URL where the files will be placed in and its extension are specified. E.g. "www.library.com/books"
@RDFIdentifier	It is placed on the method that returns a string whose value is used as the metadata file name. This identifier must not produce duplicated values. It is equivalent to the primary key in the database. E.g. "book32165", "the little Prince";
@RDFPropiedad (url, prefix, nombre)	It is placed to alphanumeric variables whose values will be inserted as nodes in the RDF file. It is necessary to specify the prefix, tag, and the name space of the node to generate. It is used to define custom fields in the RDF file.
@RDFResource(url, prefix, nombre)	It is placed on variables that refer to other classes. In the RDF file a hyperlink to another RDF file is added. The hyperlink is built depending on the referenced class annotations.

The RDF annotation *Property* is used to specify custom fields in the RDF file. Fields Dublin Core standard are contained in the set of additional annotations shown below in Table 3.

Annotation	Dublin Core annotation
@RDFDublinCoreTitle	Title
@RDFDublinCoreSubject	Subject
@RDFDublinCoreDescription	Description
@RDFDublinCoreSource	Source
@RDFDublinCoreLenguaje	Language
@RDFDublinCoreRelation	Relation
@RDFDublinCoreCoverage	Coverage
@RDFDublinCoreCreator	Creator

@RDFDublinCorePublisher	Publisher
@RDFDublinCoreContributor	Contributor
@RDFDublinCoreRights	Rights
@RDFDublinCoreDate	Date
@RDFDublinCoreType	Resource Type
@RDFDublinCoreFormat	Format
@RDFDublinCoreIdentifier	Identifier

An class example (diagram) with annotations is shown below in Table 4:

Table 4 Annotations diagram .

@RDFSujeto(url="http://www.uaemex.com/cursos", extensión="rdf")
Curso
Private id:Integer
Private nombre:String
Private descripción:String
Private fechaInicio:Date
Private profesor:Profesor
@Transient
@RDFIdentifier
Public getIdRdf:String
@RDFDublinCoreTitle
Public getNombre:String
@RDFDublinCoreDescription
Public getDescripcion:String
@Transient
@RDFPropiedad(url="http://www.uaemex.com/cursos", prefix="uaem", nombre="fechaCurso")
Public getFechaCurso:String
@Transient
@RDFDublinCoreCreator
Public getCreador:String
@RDFResource(url="http://www.uaemex.com/profesores", prefix="uaem", nombre="profesor")
Public getProfesor:Profesor
Otros getters & setters

We can observe that cover all standard dublin core labels depends on the information available in the database. Otherwise the missing fields must be added, although some derived attributes can be treated as object-oriented using additional methods that concatenate variables. In the previous example @Transient marked getters are methods whose value is not exactly equal to that offered by the attributes and are not in the interest of persistence api, as can be numeric fields and dates that must be converted into text strings.

Annotated classes are processed using the Java API REFLECTION. This suite contained in the JDK allows to inspect objects and classes, and obtain information from their variables and methods. The reflection API also allows invoking the execution of methods by name. Annotations and the values obtained from the inspection of objects generates a temporary database structure which is the basis for generating the RDF content as text string eventually.

The result may, as an instance, be published in a public section of the application server and the rest of the website will refer to them using hyperlinks. Or an RSS news archive can be generated to announce the current courses, etc.

VII. IMPLEMENTATION OF THE PROPOSAL

To implement this proposal a system developed with the architecture MVC (Model, View, Controller) was developed, where the presentation layer user interface is shown, the business logic layer receives user requests and responses are sent after the process, and data resides in the data layer. Implementing the MVC pattern allows to easily maintenance web applications due to a clear separation among the presentation layer, business logic and data access. For the development of this research a module is created in the business logic layer for RDF; so that later on the information is statically published on a public website and in parallel the RDF is stored in a repository; in Figure 2, the architecture of this proposal is shown.

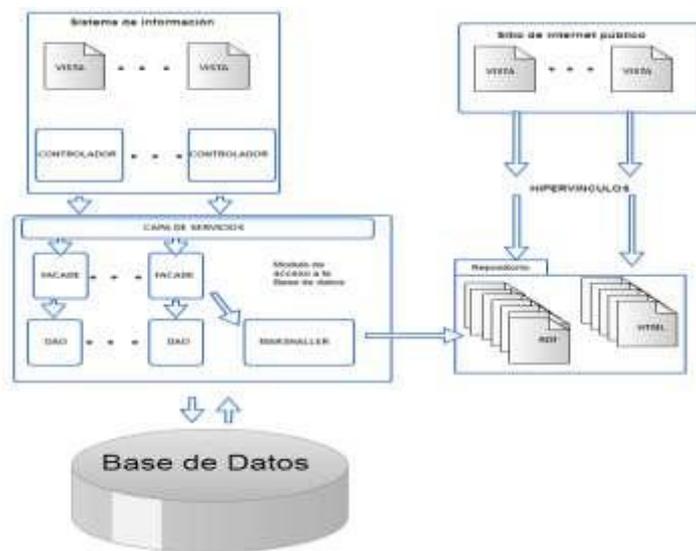


Illustration 2 Proposed Architecture.

This system was developed to manage information on courses offered by teachers at a university. This application requires to be a dynamic web application in which some of your information needs to be released as open content, in a static and semantic format for external users, referenceable by a unique URL. The web application was developed with 2.1 JSF technology (Java Server Faces) implemented with the Primefaces 5.2 component library; web application development framework Spring 4 and Hibernate 4 persistence API were used. The information was stored in a MySQL relational database. All this, into a Tomcat 7 server on JVM 1.7 configured to enable writing on the server. The descriptive classes of objects were developed through the Reflection API, to do so, the necessary notations for the application were defined. JAXB was used to obtain the HTML and XML pages that publish the desired data. The web application is modified so that when a CRUD (Create -Read -Update -Delete) operation is run, it can be performed on any item in the catalogues, [17]. The modifications also allow to generate the respective static HTML files that are saved in a public folder in the Tomcat server.

The web application allows to following up the courses administration. Figure 3 shows the new course registration interface.

The screenshot shows a web form titled 'Curso' for registering a new course. The form contains the following fields:

- Nombre: BASES DE DATOS
- Descripcion: BASICO
- Horario: LUNES A VIERNES 9 A 12
- Profesor: ANGELA RIOS LUNA
- Ubicacion: edificio=1, salon=1

At the bottom of the form, there are two buttons: 'Guardar' (Save) and 'Regresar' (Return).

Figure 3 View of the registration of a new course

With software developed in this work it is accomplished to automatically generate the RDF for each course, and this is stored in a public folder in the Tomcat server; below, the generated RDF is shown in Figure 4.

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/terms"
xmlns:uaem="http://www.uaemex.mx/cursos">
<rdf:Description rdf:about="http://www.uaemex.mx/cursos/Curso1.html">
<dc:title>matemáticas 1</dc:title>
<dc:description>algebra</dc:description>
<dc:creator>Juan_Perez_Hernandez</dc:creator>
<uaem:fecha_de_inicio>Mon Oct 24 00:03:59 CDT 2016</uaem:fecha_de_inicio>
<uaem:profesor
rdf:resource="http://www.uaemex.mx/profesores/Profesor_Juan_Perez_Hernandez.html"></uaem:profesor
or>
<uaem:horario>L-V 7:00 a 9:00</uaem:horario>
</rdf:Description>
</rdf:RDF>
</xml>

```

Figure 4. The generated RDF

The Generated information by the RDF can be displayed as a graph (see Figure 5).

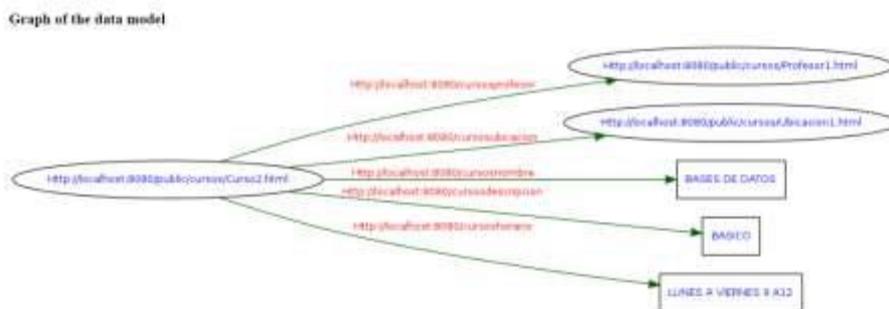


Figure 5 Representation of a course as a graph

IX. CONCLUSION AND FUTURE WORK

Unlike other options, the generated software component can be incorporated into existing databases and applications, and is the programmer who defines the architecture that allows publishing the metadata files.

The programmer can establish a strategy that will allow the information into the invoked metadata files to be updated whenever data is updated in the database through the application, keeping them synchronized.

It will also be possible to develop general purpose applications for end users who are not programmers, and work with specific requirements and generate semantic content. One advantage of RDF annotations is that the reuse of persistence classes without conflicting with the already developed systems, thus reducing the required effort to be implemented and the risk of changing the system operation.

The incorporation of metadata allows the interaction between websites and the creation of new communication and commerce channels for businesses and institutions. Semantic sites position institutions as sources of reliable and referential content, increasing its presence on the Internet.

The Dublin Core metadata definition standard is internationally recognized. It is flexible, interoperable, semantic and in constant growing and evolving through a formal institution consortium by DCMI, it is noteworthy that Dublin Core has bonds with Linked Data in order to achieve standardized files header metadata, this would ease automatic management and improve the effectiveness of search engines.

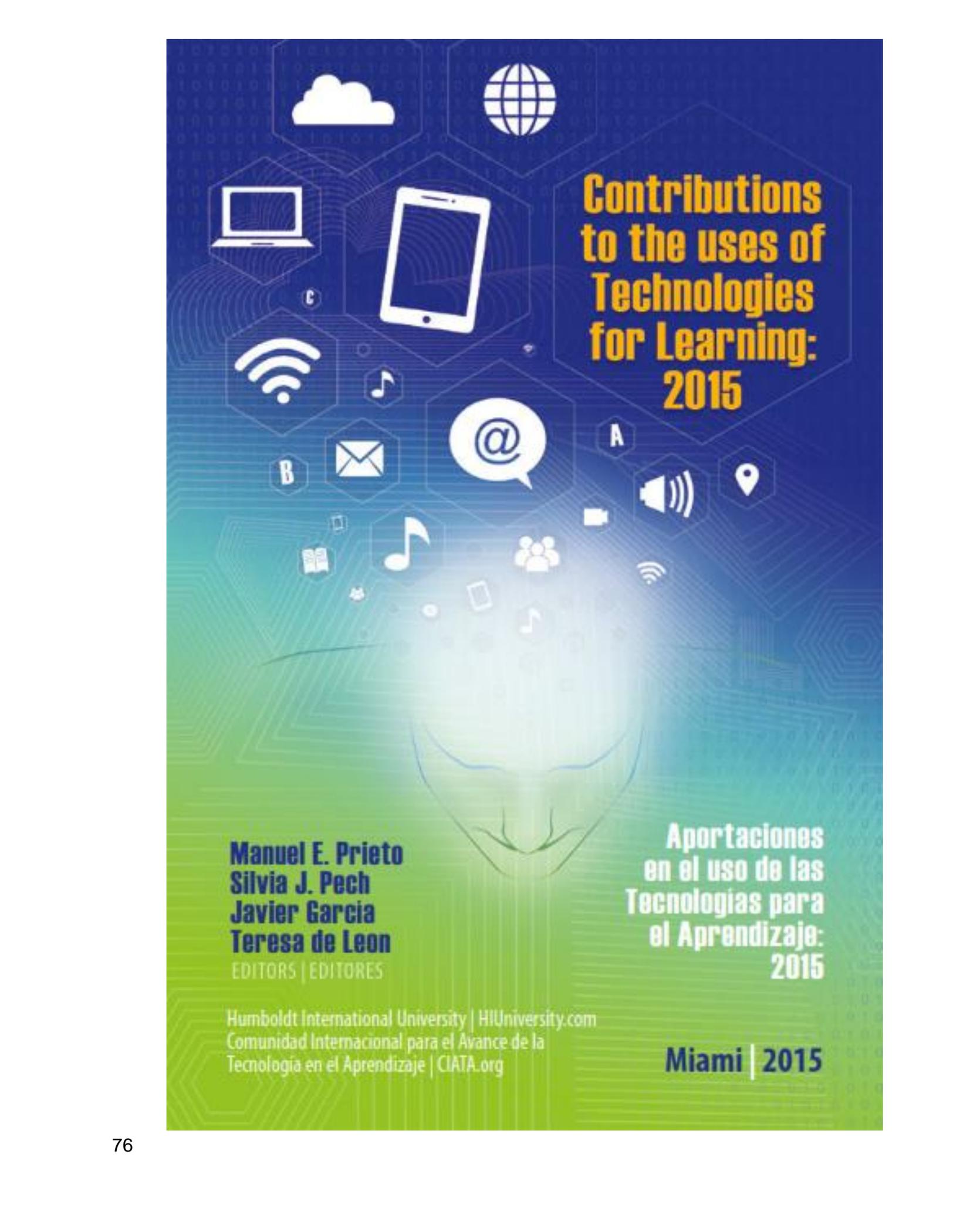
Adding metadata to an information system increases its presence in search engines. The implementation of structured vocabularies to control the content of these metadata and to organize information allows achieving a more effective organization of the collections that will lead to a recovery of more relevant information.

The development of meta information structures should be combined with the power of computer processing.

Both tendencies would converge in the approach of the Semantic Web, in which an intelligent search that takes advantage of the structured knowledge and the added human value embedded into knowledge structures is pursued.

REFERENCES

- [1]. **Davies, Paul Beynon.** Sistemas de informacion. Barcelona, España : Reverte, 2014, pág. 636.
- [2]. **w3c.** Estandares de la web semantica. [En línea] <http://www.w3.org/standards/semanticweb/>.
- [3]. **Heath, Tom.** Linked Data. [En línea] [Citado el: 27 de 03 de 2016.] <http://linkeddata.org/>.
- [4]. Descripción de los recursos de información en Internet: formato Dublin Core. **Fournier, Isabel Daudinot.** 2006, SciELO.
- [5]. RQ-RDF-3X: Going Beyond Triplestores. **Jyoti, Leeka.** Chicago IL : IEEE, 2014. 5th International Workshop on Data Engineering Meets the Semantic Web. pág. 6.
- [6]. Actualización incremental de grafos RDF a partir de bases de datos relacionales. **Álvarez, Liudmila Reyes.** Cádiz : Javier Tuya, Mercedes Ruiz y Nuria Hurtado, 2014. Actas de las XIX Jornadas de Ingeniería del Software y Bases de Datos. pág. 6.
- [7]. Desarrollo Orientado a la Semántica. **Pacheco, Ebenezer Hasdaí.** 2015, Develop Network No. 6.
- [8]. Vocabularios estructurados , Web Semántica y Linked Data : oportunidades y retos para los profesionales de la documentación. **Castro, Carmen Caro.** 2012.
- [9]. Diez años construyendo una web semántica. **Schorlemmer, Marco.** 2011, Fundación General CSIC. [10]. **Priya.L.** Improving E-learning System using Ontology Web Language. International Journal of Modern Engineering Research (IJMER). Jan-Feb de 2012, pág. 5.
- [11]. RDF Object Type and Reification in the Database. **Ravada, Nicole Alexander and Siva.** 2006 , Proceedings of the 22nd International Conference on Data Engineering (ICDE'06) 8-7695-2570-9/06 \$20.00 © IEEE.
- [12]. **w3c.** RDF concepts. [En línea] 25 de febrero de 2014. <http://www.w3.org/TR/rdf11-concepts/>.
- [13]. **Initiative, Dublin Core Metadata.** Dublin Core Metadata Initiative. dublicore. [En línea] 23 de septiembre de 2016. [Citado el: 20 de octubre de 2016.] <http://dublincore.org/metadata-basics/>.
- [14]. **López, Sergio Fernández.** TRIOO, estudio e implementación de modelos de datos RDF en lenguajes orientados a objetos. 2010.
- [15]. Integration of Heterogeneous Relational Databases: RDF Mapping Approach. **Ismail, Maizatul Akmar.** 2008, IEEE, pág. 7.
- [16]. **Oracle java.** Annotations Basics. [En línea] [Citado el: 2016 de 10 de 23.] <https://docs.oracle.com/javase/tutorial/java/annotations/basics.html>.
- [17]. **Dittawit, Kornchnok.** A Linked Data Model for E-books. 2012 IIAI International Conference on Advanced Applied Informatics . s.l. : IEEE, 2012, pág. 5.



Contributions to the uses of Technologies for Learning: 2015

Manuel E. Prieto
Silvia J. Pech
Javier Garcia
Teresa de Leon
EDITORS | EDITORES

**Aportaciones
en el uso de las
Tecnologías para
el Aprendizaje:
2015**

Humboldt International University | HIUniversity.com
Comunidad Internacional para el Avance de la
Tecnología en el Aprendizaje | CIATA.org

Miami | 2015

Manuel E. Prieto • Silvia J. Pech • Javier Garcia • Teresa de Leon

EDITORS | EDITORES

Contributions to the uses of Technologies for Learning: 2015

**Aportaciones en el uso de las Tecnologías
para el Aprendizaje: 2015**



Humboldt
International
University



Humboldt International University | HIUniversity.com

Comunidad Internacional para el Avance de la Tecnología en el Aprendizaje | CIATA.org

Miami, Florida, Estados Unidos de América | 2015

Contributions to the uses of Technologies for Learning: 2015
Aportaciones en el uso de las Tecnologías para el Aprendizaje: 2015
ISBN: 978-0-9915776-2-0

Editors | Editores: Manuel E. Prieto, Silvia J. Pech, Javier García, Teresa de Leon

D.R.© 2015, HUMBOLDT INTERNATIONAL UNIVERSITY
D.R.© 2015, MANUEL E. PRIETO
D.R.© 2015, COMUNIDAD INTERNACIONAL PARA EL AVANCE DE LA
TECNOLOGÍA EN EL APRENDIZAJE
Obra con derechos reservados, prohibida su reproducción
parcial o total sin el permiso de los editores.

HUMBOLDT INTERNATIONAL UNIVERSITY
4000 West Flagler Street,
Miami, Florida, 33134, United States
Tel.: +1 305-448-7454
Fax: +1 305-476-8430
Correo: info@hiuniversity.com
www.hiuniversity.com/

Diseño de portada y maquetación | Cover design and layout: Suelen Y. Torres Mota

Impreso en Miami, Florida, Estados Unidos de América.
Printed in Miami, Florida, United States.

Caracterización de estudiantes en Sistemas Tutores Inteligentes mediante patrones borrosos.	333
<i>José A. Olivas, Francisco P. Romero, M. Rosario Vázquez, Jorge Ruiz-Villanoye, Jesús Serrano-Guerrero</i>	
Sección III	
Comunicaciones Extracted Papers	341
Software livre como ferramenta pedagógica no gerenciamento e mediação do processo de ensino em laboratórios de informática	343
<i>Anderson Luiz Nogueira Vieira, José Luiz Kessler, Douglas Machado Silva, Luciano Amorim, Yago dos Santos</i>	
Edición de Patrones de aprendizaje colaborativo estandarizados	351
<i>Jorge E. Gil Mateos, Víctor Manuel Martín Lloró, Delysa González Martínez</i>	
Uso de Redes Sociales en la Educación Superior en España e Iberoamérica	359
<i>Mario Martínez García, José Sánchez Rodríguez, Julio Ruiz-Palmero</i>	
Situación actual de la web semántica en aplicaciones educativas	367
<i>Irene Aguilar Juárez, Ángela Rivas Luna</i>	
Uso de la tecnología para el aprendizaje en la Educación Básica (Secundaria)	376
<i>Yolanda Juárez López, Jacqueline Sánchez Espinoza, Cozobí García Herrera</i>	
Semejanzas en la programación de actividades en el nivel preescolar y en el nivel superior, considerando el empleo de las TIC	383
<i>Carla Badillo Ortega, Cozobí García Herrera</i>	
Plataforma MOOC para fomentar la cultura emprendedora en estudiantes universitarios	391
<i>Antonio Pérez De la Cruz, Angélica Mata Cárdenas, Francisco José Medina Ocampo, Edda Verónica Martínez Mesa</i>	
Objetos de Aprendizaje para Móviles en la Educación Superior. Caso: Facultad de Contaduría y Administración, Universidad Veracruzana, México.	398
<i>Alma D. Otero Escobar, Elsa Suárez Jasso, Enriqueta Sarabia Román</i>	
Cloud Computing en creación de contenidos para e-learning	404
<i>Pedro Cardona, Angel Muñoz, Jaime Muñoz, Francisco Alvarez, César Velázquez</i>	
Pruebas de Usabilidad de un Chatbot con Niños de Quinto Grado de Primaria	410
<i>Deymi Oj-Tec, Carlos Miranda-Palosa</i>	
Fase de evaluación del software educativo CHEMYSYS centrado en el aprendizaje de estudiantes del nivel medio básico y medio superior	416
<i>Adriana Bastamante Almaraz</i>	
Análisis del impacto de los equipos móviles en el ambiente formativo de la Licenciatura de Informática y Contaduría	424
<i>Patricia Delgado G., Blanca G. Cuevas G., Esperanza Cotea R., Laura C. Méndez G.</i>	
Determinación de estrategias didácticas de acuerdo a los canales de aprendizaje en la unidad de aprendizaje: programación avanzada de la licenciatura en ingeniería en computación	432
<i>Adriana Bastamante Almaraz</i>	

Situación actual de la web semántica en aplicaciones educativas

Irene Aguilar Juárez ^a, Angela Ríos Luna^b,

^a Centro Universitario UAEM Texcoco, Av Jardín Zumpango s/n, Fraccionamiento El Tejocote, Texcoco, Estado de México, México

ireneico@gmail.com, iaguilarj@uaemex.mx

^b Centro Universitario UAEM Texcoco, Av Jardín Zumpango s/n, Fraccionamiento El Tejocote, Texcoco, Estado de México, México

angelariosluna@gmail.com

Resumen. Desde hace aproximadamente dos décadas se han generado múltiples sitios educativos y Entornos Virtuales de Aprendizaje (EVA), los cuales han sido todo un éxito y son la base de la difusión de la educación a distancia y en los que se apoyan actualmente muchos cursos presenciales. Sin embargo ante la tendencia cada vez mayor de agregar significado a la información publicada en la web, surgen nuevos retos para los EVA. En este trabajo se analizan los aspectos técnicos de los EVA tradicionales, los sitios que implementan la Web semántica y los esfuerzos que se realizan actualmente para implementar los avances logrados de la Web semántica en los sitios educativos basados en web.

Palabras Clave: web semántica, sistemas tutoriales inteligentes, ontologías, LMS.

1 Introducción

La Web, tal y como la conocemos hoy, fue concebida para humanos, contando con nuestra capacidad que entender el lenguaje natural, la interpretación de imágenes y de sonidos. Es por eso que las búsquedas realizadas por los agentes de software en la actualidad entregan vínculos hacia múltiples documentos que pueden resultar irrelevantes o innecesarios, esta situación implica que los usuarios deban invertir mucho de su tiempo en filtrar la información relevante para su búsqueda. A pesar de esta circunstancia la publicación de documentos digitales no deja de crecer; tan solo en el 2013 se reportaba que existen 634 millones de sitios web y 87.8 millones de blogs [1]. YouTube reporta en su página que el número de horas de visualización al mes aumenta un 50% año tras año y que se suben a YouTube 300 horas de vídeo por minuto [2]. Además la mayoría de los sistemas web de apoyo educativo funcionan basados en la web tradicional.

Ante este panorama se presenta la necesidad de adaptar la arquitectura de los sitios actuales y del contenido de la Web para ser usado no solo por los humanos sino también por los agentes de software y así lograr un acceso más eficiente a tanta información disponible; es decir se requieren sitios cuya estructura facilite la búsqueda en términos

semánticos, capaces de ser adaptables a los intereses del usuario. En esta revisión se analiza la tecnología actual de los LMS, las tecnologías necesarias para implementar la web semántica y se revisan los logros de diferentes casos de aplicación, la finalidad es que se identifiquen las oportunidades de desarrollo en la promoción de una nueva generación de sitios educativos y de Espacios Virtuales de Aprendizaje orientados a la adaptabilidad de docentes y alumnos. El trabajo se compone de dos apartados, en el primero se describen los aspectos técnicos de las plataformas educativas actuales, de los sitios semánticos y las herramientas actualmente disponibles para lograrlo, en el segundo apartado se describen los resultados de diferentes proyectos que actualmente exploran la integración de estas tecnologías en la búsqueda de nuevos y mejores servicios educativos.

2 Aspectos técnicos de los LMS y la Web semántica

El Entorno Virtual de Aprendizaje o LMS (Learning Management System) es el software que permitir la creación y gestión de cursos completos para internet sin que sean necesarios conocimientos profundos de programación [3]. La arquitectura de estos sistemas es compleja, pues las plataformas educativas deben integrar módulos que ofrecen múltiples servicios a administradores, docentes y alumnos a través de la web, los cuales de acuerdo a Díaz [3] pueden agruparse en las siguientes categorías, todas ellas basadas en la web tradicional:

- Herramientas de gestión de contenidos o CMS (Content Management System)
- Herramientas de comunicación y colaboración (foros, blogs, chats, correo electrónico)
- Herramientas de seguimiento y evaluación
- Herramientas de administración y asignación de permisos
- Herramientas complementarias, como portafolio, bloc de notas y sistemas de búsquedas de contenidos y materiales educativos.

Según Medved [4], los 8 LMS de código abierto de mayor uso son: 1. Moodle 2. CourseSites de BlackBoard 3. Sakai 4. LatitudLearning 5. Dokeos 6. EFront 7. Schoology 8. Ilias.

Como ejemplo de la estructura interna de los LMS se tiene a Moodle que funciona con un núcleo del sistema (conocido como el Moodle core) rodeado de una gran cantidad de plugins que proveen de funcionalidad específica a la plataforma. Los plugins en Moodle pueden ser de varios tipos: autenticación, bloques, matriculación, temas, actividades, etc, todos los módulos comparten funcionalidad específica entre sí y el core [5]. La arquitectura de Moodle incluye más de 30 módulos, en opinión de Campos [5]. Moodle no usa la orientación a objetos, ni utiliza ningún sistema de encapsulación de datos, ni separa la vista de la lógica de la aplicación; a pesar de ello da servicio a más de 40 millones de usuarios.

Un aspecto fundamental para el éxito de los LMS es el soporte a estándares internacionales para gestionar los recursos y las actividades de los docentes y alumnos.

Actualmente existe una tendencia unificada e integradora entre los principales desarrolladores de especificaciones para e-learning agrupadas bajo la iniciativas IMS, IEEE, LTSC (Learning Technology Standards Comitee), e ISO/IEC que puede ayudar a simplificar y generalizar la creación y adopción de un único estándar educativo que sirva como referencia. Actualmente LOM (IEEE Learning Object Meta-Data) es el estándar de e-learning formalmente aprobado (IEEE 1484.12.1-2002) y ha sido adoptado en las especificaciones IMS Learning Resource metadata. El objetivo de LOM es la creación de descripciones estructuradas de recursos educativos. Su modelo de datos especifica qué aspectos de un objeto de aprendizaje deberían ser descritos y qué vocabularios se pueden utilizar en dicha descripción. Esta es una descripción jerárquica con nueve apartados principales que agrupan el resto del campo expresados en archivos XML. Dado que es el único estándar aprobado de la industria del aprendizaje mediado por tecnología es necesario para las plataformas o sistemas LMS soportar dicho estándar

La evolución de los LMS se observa, sobre todo en la arquitectura de sus componentes o módulos, los primeros LMS surgieron con el paradigma de desarrollo estructurado y con lenguajes de programación sin soporte al Paradigma Orientado a Objetos, sin embargo con la evolución de los entornos de desarrollo y de los lenguajes de programación ahora se generan nuevos LMS con arquitecturas organizadas en capas, y basada en servicios web. Otro aspecto de mejora se presenta respecto a la integración nuevas tecnologías como los entornos de realidad virtual o de realidad aumentada, estas tecnologías están forzando nuevos cambios en los LMS, pues ahora cada vez es más frecuente la necesidad de integrar estas prestaciones a las plataformas educativas. Bernal y Marin [6] abordan esta problemática y proponen nuevas actividades que integran los servicios de los LMS y los MV3D (Mundos Virtuales de 3D).

La web semántica a diferencia de la Web actual se basa en el “qué” de lo publicado no en “cómo” es publicado, es decir su funcionamiento tiene base en el significado de los textos e imágenes que vemos en las pantallas no en “cómo” vemos esa información, esto permite el procesamiento de las publicaciones por agentes de software cuya tarea es filtrar las páginas publicadas de acuerdo a las necesidades y preferencias de los usuarios. Actualmente el W3C (World Wide Web Consortium) trabaja desde hace años en establecer estándares y recomendaciones de buen uso para lograr el crecimiento y la potenciación de la web a largo plazo. El papel del W3C en la elaboración de las recomendaciones es atraer la atención de los desarrolladores sobre las especificaciones y promover así un amplio despliegue de las mismas. Todo ello para mejorar la funcionalidad y la interoperabilidad de la Web [7]

Peredo [8] identifica que para la generación de sitios semánticos se requiere la integración de varios componentes, los cuales son: declaraciones e Identificador de Recurso Uniforme (URI), lenguaje de consultas, ontologías, instancias de datos, motores de reglas, razonadores y marcos de trabajo o framework para el desarrollo web.

Para la representación de las declaraciones y de las URI en la Web semántica se puede utilizar RDF (Resource Description Framework) ya que el lenguaje es una estructura

definida para los metadatos en la World Wide Web y es reconocida por el W3C. RDF tiene un mecanismo que permite describir los recursos. Existe el RDF/XML que contiene un URI, sus propiedades como por ejemplo mailbox o fullName (en forma abreviada) y sus respectivos valores.

Como HTML, este RDF/XML es procesable por la máquina y mediante los URI, se puede enlazar la información en la Web. Sin embargo, al contrario que el hipertexto convencional, los URI de RDF pueden hacer referencia a cualquier cosa identificable, incluyendo cosas que pueden no ser directamente recuperables en la Web, en RDF tanto en los sujetos, como las propiedades y los objetos, son recursos. [9]

La utilización de cualquier modelo de metadatos es viable y su reutilización por parte de buscadores es posible gracias a su codificación en XML y la utilización de los lenguajes de consulta adecuados. La forma de expresar conjuntos de metadatos específicos es mediante RDFSchemas, también posee implícitamente la posibilidad de combinarlos de forma que se adapten a una aplicación concreta sin que estos pierdan su significado original [10]. Esto representa una ventaja para la integración de sitios semánticos con sitios ya publicados, pues los sitios y plataformas educativos hacen uso de metadatos específicos como el SCORM (Sharable Content Resource Management), el LOM (Learning Object Metadata), el Dublin Core o el IMS (Instruccional Management System). La posibilidad de una interpretación de los documentos XML de los metadatos existentes permite la interacción entre sitios tradicionales y los motores de búsqueda semánticos. La estructura de la sintaxis de RDF es un conjunto de tres elementos llamadas tripletas y está compuesto de un “sujeto, predicado y un objeto” [7], a este conjunto se llama grafo RDF, donde un grafo RDF se puede visualizar como nodo del grafo y arco del grafo, en el que cada predicado o triple se representa como un nodo de enlace.

En el sujeto se describe de quien se está hablando, en el predicado que también se le conoce como propiedad se representa una conexión entre un sujeto y un objeto, normalmente una relación del sujeto con algo o un atributo del sujeto, en el objeto el valor de lo declarado. A continuación se muestra un ejemplo de una tripleta:

- Sujeto: `Http://ejemplo.com/pepe`
- Predicado: `Http://ejemplo.com/nacimineto`
- Objeto: `“14/08/1988”`

El lenguaje para expresar consultas que permiten interrogar diversas fuentes de datos, si los datos se almacenan de forma nativa como RDF o son definidos mediante vistas RDF a través de algún sistema middleware es SPARQL (Protocol and RDF Query Language). Este lenguaje contiene las capacidades para la consulta de los patrones obligatorios y opcionales del grafo, junto con sus conjunciones y disyunciones. SPARQL también soporta la ampliación o restricción del ámbito de las consultas indicando los grafos sobre los que se opera. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF. [7]. Con el uso de RDF es posible crear, gestionar o buscar datos en la web de manera entendible por las computadoras, promoviendo la

transformación de la web en un repositorio de información procesable por las computadoras.

Las ontologías y la instancia de datos tiene la finalidad de modelar el dominio de la aplicación definiendo los conceptos y sus relaciones; las ontologías buscan capturar y representar el conocimiento de forma consensuada, para que puede ser reutilizado y compartido tanto por aplicaciones software como por grupos de personas [12]. Para la implementación de las ontologías puede usarse el OWL (Web Ontology Language) un lenguaje orientado a la especificación de ontologías, con OWL también es posible definir las instancias de datos que son datos concretos sobre el sitio y sus documentos. Los razonadores y los motores de reglas son software que permite inferir nueva información a partir de la definida en las ontologías y los marcos de trabajo son los programas de software que nos permiten integrar todos los elementos antes mencionados. Los frameworks permiten crear y gestionar documentos en RDF, ontologías expresadas en OWL y consultas mediante SPARQL, también permiten usar razonadores como Pellet reasoner, entre varios. Uno de los frameworks más difundidos es JENA una API para desarrollar con java sitios semánticos.

3 Aplicaciones de Web Semantica en la Tecnología Educativa

En la actualidad se hacen esfuerzos por aplicar la web semántica en los sitios educativos, a continuación se explican algunos casos de aplicación y los logros alcanzados.

Gestión de documentos digitales: La gestión de documentos digitales es un proceso necesario en todo sitio web, las búsquedas pueden ser más eficiente si se agregan anotaciones semánticas, pues estas etiquetas permiten encontrar documentos digitales relacionados por la temática tratada o por las propiedades comunes con el criterio de búsqueda. Un ejemplo de aplicación de etiquetado semántico que ha probado la eficacia en la gestión es el sistema SABIOS, el cual permite mejorar los procesos de inserción, catalogación y recuperación de documentos digitales a través de uso de anotaciones semánticas combinado con sistemas Multiagentes [13]

Otro ejemplo de la implementación gestores inteligentes es DBpedia, quien se originó mediante un esfuerzo de la comunidad para extraer información estructurada de Wikipedia y poner esta información nuevamente disponible en la Web. En el caso de dbpedia.org, el proceso de extracción produce 100 millones de tripletas RDF a partir de la versión para el español de la wikipedia. En el SPARQL endpoint (punto final de un servicio web) están disponibles todas estas tripletas y pueden ser usadas por los usuarios. La información completa sobre la sección del idioma español de la DBpedia como el SPARQL endpoint, los datos y la información para desarrolladores se pueden encontrar en el Wiki. Cabe señalar que para hacer posible a DBpedia es necesaria la tecnología Linked Data la cual permite acceder a un conjunto de datos relacionados entre sí, mediante ella una persona o una maquina pueden explorar la red de datos, con datos vinculados, se pueden encontrar nuevos datos relacionados. [14]

Tesauros documentales: Los tesauros surgen con el objetivo de normalizar terminos con base en la relación de conceptos sinonimos y antonimos, estos diccionarios facilitan el procesamiento automatizado por software que permite búsquedas más efectivas de los documentos. En [15] se presenta una propuesta básica de automatización y utilización de tesauros documentales en entornos distribuidos para recuperación de información mediante servicios web basados en RDF.

Visualización de información: La web semántica y sus características permiten adaptar la visualización de la información para que tome ventaja de las propiedades que ofrecen las ontologías [16], [17]. Es así que se habla del reto de visualizar la información basada en ontologías de una forma práctica para el usuario [18]. Una vez que se tienen los archivos en RDF (conceptos y sus relaciones, asociadas a los conceptos), se puede presentar al usuario los resultados de dichas relaciones de una manera apropiada con servicios de visualización tipo grafo; esto con la finalidad de tener la opción de inferir más información sobre dichos documentos y visualizar los resultados tanto de las notaciones semánticas como de las consultas inferidas. Actualmente la implementación de este tipo de visualización se puede realizar mediante el Ontograph en Protegé, utilizando el RDF-Schema [13].

Espacios Virtuales de Aprendizaje

Recientemente, algunas iniciativas surgidas del área de la Inteligencia Artificial y de la comunidad de Educación pretenden desarrollar e implementar los Sistemas Educativos basados en la Web semántica (SWBES por sus siglas en inglés). Los SWBES son sistemas de información que tiene el objetivo de facilitar el aprendizaje en cualquier momento, en cualquier lugar y por cualquier persona, mediante la gran cantidad de recursos y objetos de aprendizaje reusables disponibles en la Web, proporcionados por agentes pedagógicos inteligentes [19]. En opinión de Uribe [20] las aplicaciones semánticas tienen potencialidad para la gestión de la información y del conocimiento, al facilitar la localización, la selección, la recuperación, la organización, la evaluación, la producción y la divulgación en forma adecuada y eficiente de la información.

Bittencourt [19] define una arquitectura base para los SEWS, en ella se identifica la necesidad de integrar varias tecnologías: ontologías, estándares de recursos educativos, servicios de web semántica y agentes inteligentes. En su propuesta se identifican varios roles de trabajo necesarios en la interacción con estos sistemas, los roles identificados son: rol del alumno, el rol del docente, el rol de grupo (para actividades de aprendizaje), el rol del autor de contenidos, el rol de desarrollador, el rol del administrador y el rol del auditor del sistema.

En la arquitectura de los SWEBS es posible apoyarse de Sistemas Tutores Inteligentes (STI), los cuales son sistemas que son utilizados como tutores o compañeros de estudio, asistiendo en los procesos de aprendizaje a seres humanos en una variedad de dominios, tienen la característica de ser activos pero no dominantes, actuando como colaboradores o competidores de los aprendices. Pueden integrarse como módulos de los SWEBS o módulos complementarios a los LMS tradicionales, su función principal es modelar tanto a los alumnos como a los docentes para soportar sitios adaptativos tanto en los perfiles de los usuarios como en las propiedades de los contenidos.

Un ejemplo de aplicación de estos STI es el proyecto documentado por Caviedes [21], en su proyecto se trabaja para lograr el aprovechamiento de la web semántica mediante la modelación del estudiante y del contenido; el modelo representa el perfil personal, el perfil cognitivo y el perfil del conocimiento. Los autores también desarrollan el modelo de adaptación que acondiciona el entorno al estudiante de acuerdo con el estilo y el conocimiento de cada uno. En la investigación se desarrolla un módulo pedagógico, un módulo de interfase, un módulo de estudiante y un módulo de dominio. Para la implementación se desarrollan varios agentes: un agente pedagógico, un agente de

presentación, un agente de usuario y un agente WSI (interfaz de servicio web). Como resultado en esta investigación se ofrecer un sistema tutorial adaptativo en entornos web.

Otro caso de aplicación que sigue la misma línea es el proyecto realizado por Peredo [8] en el cual se desarrolló un módulo multiagente que tiene la función de trabajar con la información del LMS (Learning Management System) para generar una base de conocimientos, su proyecto también cuenta con un sistema asistencial de apoyo al profesor. La interacción entre los módulos es posible por la capa de metadatos, dicha capa usa tres tipos de metadatos: el SCORM para conservar la compatibilidad con los estándares internacionales y reusar los objetos de aprendizaje ya disponibles, usa archivos de configuración de componentes IRLCOO (Intelligent Reusable Learning Components Object Oriented) y usa archivos RDF en los que se define el perfil de docentes alumnos y contenidos. Los agentes se desarrollan usando la API JADE y JADEx y usan la ontología Friend of a Friend (FOAF). La plataforma multiagente se desarrolla con el framework JENA, se usa para parsear y serializar los archivos RDF. Mediante la ontología y los datos extraídos del LMS se modela el perfil de las personas, las actividades y sus relaciones.

Silva [22] coincide en esta línea de investigación y trabaja un estudio sobre la construcción real de los Sistemas Inteligentes de Tutoría insertados como parte de los entornos virtuales de aprendizaje para la educación a distancia o en modalidad presencial; teniendo en cuenta sobre todo las características de desarrollo basados en la reutilización y en el desarrollo a gran escala. En su propuesta se recomienda considerar: los diferentes perfiles de usuario, el hecho de que estos usuarios no tienen conocimientos avanzados en relación a la informática y la heterogeneidad del conocimiento del dominio. El objetivo de estos sistemas es encargarse del constante cambio en el alumno y de la evolución del conocimiento.

Gestión de referencias bibliográficas: el proyecto de Sánchez[23] obtuvo como resultado el desarrollo de un gestor bibliográfico que facilita el tratamiento de referencias y ofrece la posibilidad de traducir entre varios formatos de publicación, centrándose en tecnologías de la web semántica.

Buscadores semánticos: los buscadores semánticos han dejado los laboratorios de investigación para convertirse en aplicaciones funcionales. Uno de los primeros buscadores semánticos fue SWOOGLE [24] actualmente el buscador está todavía activo y realiza búsquedas sobre más de 10000 ontologías. SWOOGLE extrae metadatos a partir de documentos RDF y OWL, también es capaz de encontrar las relaciones existentes entre dichos documentos. En el 2006 surge SemSearch, un buscador semántico orientado a usuarios comunes que no están familiarizados con las tecnologías de la web semántica o con el dominio específico de los datos semánticos. Otro buscador semántico es Watson presentado por D'Aquin [25], es un buscador que proporciona un conjunto de API que contienen funciones de alto nivel para encontrar, explorar y consultar datos semánticos y ontologías que han sido publicadas en línea.

Bibliotecas digitales: Los repositorios digitales especializados son una fuente de información valiosa, en la actualidad es posible consultar bibliotecas digitales que

interactúan con otras bibliotecas digitales gracias al estándar OAI-PMH (Open Archives Initiative - Protocol for Metadata Harvesting), sin embargo aún no se generaliza el uso de las etiquetas semánticas para el acceso a la información. Un primer caso de aplicación es la Biblioteca Estudiantil Digital (BED), está basada en los estándares de la web semántica, implementa la nueva forma de entender la web y las búsquedas que en ella se hacen, se basan en relacionar los significados y no solamente los textos. Por eso, cualquier usuario puede comprobar que los resultados obtenidos estarán acompañados de información relevante y útil. Dentro de cada sección, se dispone de un completo buscador que permite filtrar las características que se crean más destacadas con el objetivo de hacer búsquedas más rápidas y satisfactorias. Además, todos los contenidos podrán ser comentados, compartidos y valorados [26]

Desde un punto de vista organizativo, la BED está estructurada en bloques de contenido que facilitan la búsqueda de información: objetos Educativos, centro de conocimiento, buenas prácticas, recursos, actualidad y debates.

Conclusiones

Gracias a la Web Semántica y a la utilización de lenguajes de metadatos como el RDF es posible mejorar la usabilidad y la personalización de la Web; los sitios o módulos semánticos requieren de la integración un conjunto de componentes. Los documentos etiquetados con información semántica son usados para facilitar el trabajo a los robots (agentes de software). Se pretende que esta información sea interpretada por el ordenador con una capacidad comparable a la del lector humano. El etiquetado puede incluir metadatos descriptivos de otros aspectos documentales o protocolarios. Los vocabularios comunes de metadatos (ontologías) y mapas entre vocabularios permiten a quienes elaboran los documentos disponer de nociones claras de cómo deben etiquetarlos para que los agentes automáticos puedan usar la información contenida en los metadatos. Una parte importante de la web semántica son los servicios web que proporcionan información a los agentes.

La implementación de la web semántica facilita la construcción de materiales adaptativos, las trayectorias de los alumnos podrán personalizarse de acuerdo de los aprendizajes adquiridos por ellos, de igual forma es posible desarrollar módulos asistenciales para el docente, estos módulos podrán identificar patrones de comportamiento que podrán ser aprovechados en la redirección de las trayectorias de los estudiantes. Aunque el desarrollo de este tipo de software está en etapa de desarrollo vale la pena continuar sobre la línea para integrar los logros ya alcanzados en aplicaciones disponibles en múltiples sistemas y plataformas.

Referencia

- [1] Pingdom, "Tech blog," enero 2013. [Online]. Available: <http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/>. [Accessed abril 2015].
- [2] YouTube, "YouTube.com," abril 2015. [Online]. Available: <https://www.youtube.com/yt/press/es-419/statistics.html>.

- [3] Díaz B. S, 2009, "Plataformas Educativas, un entorno para profesores y alumnos", Revista Temas para la Educación, número 2 mayo 2009, Federación de Enseñanza de C.C. OO de Andalucía, ISSN: 1989-4023, consultado en diciembre 2013, [Online]. Available: <http://www.youblisher.com/p/37864-Please-Add-a-Title/>
- [4] Medved J.P, 2013, "The top 8 Free/Open source LMS", Capterra, consultado en diciembre de 2013, [Online]. Available: <http://blog.capterra.com/top-8-freeopen-source-lmss/>
- [5] Campos Oscar, (2011), Escribiendo módulos para Moodle: Introducción, disponible, consultado en Diciembre 2013, disponible On-Line <http://www.genbetadev.com/desarrolloweb/escribiendo-modulos-para-moodle-introduccion>
- [6] Bernal G. D. H, y Marín L. A, 2012, "Propuesta de integración de plataformas LMS y mundos virtuales 3D utilizando una arquitectura orientada a servicios", Revista Politécnica ISSN 1900-2351, Año 8, número 14 pp. 39-48, consultado en diciembre de 2013, disponible [Online]. Available en http://www.politecnicojic.edu.co/images/stories/medios/revista_politecnica/revista_14.pdf
- [7] W3C, "RDF concepts," 25 febrero 2014. [Online]. Available: <http://www.w3.org/TR/rdf11-concepts/>.
- [8] R. Peredo Valderrama and I. Peredo Valderrama, "Aplicación Web Semántica educativa incorporando," in *Avances de la Tecnologías de la Informática y de la Comunicación*, México, ALFA-OMEGA, 2014, pp. 199-205.
- [9] J. C. Tello, "La Web Semántica y el lenguaje RDF," 2006.
- [10] D. F. d. V. Gastaminza, D. R. Sánchez Jiménez and R. D. J. Pérez Agüera, "Descripción e identificación de recursos en internet: Metadata," 2005.
- [11] W3C, "SPARQL Lenguaje de consultas para RDF," 15 Enero 2008. [Online]. Available: <http://skos.um.es/TR/rdf-sparql-query/>. [Accessed 23 Abril 2015].
- [12] M. S. Choque Callisaya, "Técnicas de la Inteligencia Artificial Aplicadas a la Educación," *Revistas Bolivarianas*, vol. vol 11, no. num 33, pp. pag 34-36, 2010.
- [13] J. Guzmán Luna, D. Torres Pardo and D. A. Ovalle, "Una aplicación de la web semántica para la gestión de documentos digitales," 2007.
- [14] w. S. W. Linked, "Linked Data," [Online]. Available: <http://www.w3.org/standards/semanticweb/data>.
- [15] J. R. P. Perez Agüera, "Automatización de tesauros y su utilización en la web semántica," 2004.
- [16] Chen, "Information Visualization Versus the Semantic Web. En Visualizing the Semantic Web," 2002.
- [17] R. Georgieva, "Ontology-Based Information Representation," p. 16, 2005.
- [18] C. Fluit, "Supporting User Tasks through Visualization of Light-weight Ontologies," p. 20, 2003.
- [19] I. I. Bittencourt, E. Costa, S. Isotani, R. Mizoguchi and I. M. Bittencourt, "Towards a Reference Model to Semantic Web-based".
- [20] Uribe, "La web semántica y sus posibles aplicaciones en las universidades," 2010.
- [21] D. Caviedes Pulido, V. H. Medina García and O. García Palencia, "Diseño de un sistema tutor inteligente basado en estilos," 2010.
- [22] A. P. Silva, E. Costa and I. I. Bettencourt, "Uma Linha de Produto de Software baseada na Web," *Revista Brasileira De Informática Na Educação*, 2012.
- [23] J. M. G. Sánchez, *Aplicación Web Semántica Para La Gestión De Referencias Bibliográficas*, 2010.
- [24] Ding and F., "Swoogle: A Semantic Web Search and Metadata Engine," 2004.
- [25] D'Aquin and Motta, "More Than a Semantic Web Search Engine. Semantic Web," 2011.
- [26] Centro Internacional de Tecnologías Avanzadas de la Fundación Germán Sánchez Ruipérez, "Biblioteca Escolar Digital (BED)," 2014. [Online]. Available: <http://bibliotecaescolardigital.es/comunidad/BibliotecaEscolarDigital/acerca-de>.
- [27] DBpedia, "DBpedia," 2014. [Online]. Available: <http://dbpedia.org/>.
- [28] V. J. J. I. Carlos Carrascosa Casamayo, Estudio del buscador semántico Swoogle, 2006.
- [29] M. Shaw, "Writing good software engineering Research papers," in *Proceedings of the 25th International Conference on Software Engineering*, 2003.
- [30] V. U. a. E. M. Yuanguí Lei, "SemSearch: A Search Engine for the Semantic web," 2007.



Grigori Sidorov <computacion-y-sistemas@cic.ipn.mx>

para mí ▾

Dear Angela Rios Luna,

Thank you for submitting the manuscript, "Publicación de contenido semántico a partir de una BD relacional en un Sistema de Información" to Computación y Sistemas. With the online journal management system that we are using, you will be able to track its progress through the editorial process by logging in to the journal web site:

Manuscript URL:

<http://www.cys.cic.ipn.mx/ojs/index.php/CyS/author/submission/2388>

Username: angelarios

If you have any questions, please contact me. Thank you for considering this journal as a venue for your work.

Sincerely,

Grigori Sidorov
Computación y Sistemas

Publicación de contenido semántico a partir de una BD relacional en un Sistema de Información

Resumen. El World Wide Web fue originalmente diseñado para presentar información mediante una apariencia sencilla con sitios que no tenían una complicación significativa, sin embargo a medida que fue creciendo, la enorme complejidad de las aplicaciones web han generado la necesidad de contar con herramientas que discriminen y promuevan automáticamente sus contenidos. Una Web carente de semántica, con una eficiencia cuestionable en los servicios que proveen, puede convertir el acceso a la información en una tarea difícil y ardua para el usuario final. La web 3.0 plantea las directrices que le permiten a los sitios web interactuar entre sí para enriquecer la información presentada. En este trabajo se plantea una estrategia de evolución mediante la publicación automática de contenidos digitales a partir de información contenida en una base de datos relacional de acuerdo a las directrices de la web 3.0. Para ello fue necesario el desarrollo de una API que genera archivos de metadatos RDF a partir de objetos con anotaciones.

Palabras claves: web semántica, RDF, bases de datos, tripleta, anotaciones.

Publication of semantic content from a Relational Database in an Information System

Abstract. The World Wide Web was originally designed to present information through a simple appearance with sites that did not have a significant complication, however as it grew the enormous complexity of web applications have generated the need for tools that discriminate and automatically promote its contents. A Web devoid of semantics with questionable efficiency in the services they provide, can make access to information in a difficult and hard task for the end user. Web 3.0 presents the guidelines that allow websites to interact with each other to enrich the

information presented. In this paper is proposed a strategy of evolution by automatically publishing digital content from information contained in a relational database according to the guidelines of the web 3.0. This focus have required the development of an API that generates RDF metadata files from objects with annotations.

Keywords. semantic web, RDF, data base, tripleta, annotations.

1 Introducción

Los contenidos de Internet se clasifican en estáticos y dinámicos: En los sitios estáticos el contenido es almacenado como archivos HTML identificables por un identificador *Uniform Resource Identifier (URI)* relacionados entre sí por hipervínculos. En los sitios dinámicos el contenido HTML es generado a tiempo de ejecución a partir de información, generalmente almacenada en una base de datos; en estos sitios el contenido presentado a cada usuario depende de sus privilegios y del momento en que hace la consulta.

Si bien, las organizaciones almacenan en sus bases de datos información operativa que es confidencial como es el caso de cuentas bancarias que se muestran en los portales, también se puede almacenar información pública que debe difundirse, como la que muestra el Instituto Nacional de Estadística y Geografía (INEGI). Es deseable que sólo a partir de publicación de información autorizada de una organización, los proveedores de servicios de búsqueda pueden indexar su contenido y sugerirlo al público en general.

Las compañías que ofrecen servicios de búsqueda en Internet, leen el contenido de los sitios web y generan claves por las cuales un sitio aparecerá como resultado de una búsqueda. Los sitios son indexados empezando por la página de

inicio y siguiendo recursivamente los hipervínculos que conectan a los archivos entre sí. Además de poder indexarse por motores de búsqueda, los contenidos estáticos de un sitio web que poseen metadatos pueden ser consultados por otros sitios y aplicaciones. Los metadatos que representan el contenido de un sitio web deben hacerlo en un formato neutro. Sin embargo, la tarea de describir cada contenido supone un esfuerzo extra a la construcción de un sitio web corporativo. Del mismo modo en que la creación de páginas de Internet se apoya en herramientas que agilizan su desarrollo, la generación de archivos de metadatos requiere estrategias que faciliten su creación.

En el año 2007 surgió en el W3C el proyecto Linked Open Data, con el fin de divulgar la web semántica. Cada año se publica un grafo que muestra los enlaces creados entre todas las webs semánticas, se puede apreciar que el crecimiento ha sido exponencial (Heath). En el centro del grafo tenemos a la DBpedia, la versión semántica de la Wikipedia.

En este trabajo se describe una estrategia de generación automática de contenidos semánticos (recurso + metadatos) utilizando el lenguaje de programación Java a partir de información almacenada en una base de datos relacional. La ventaja de esta propuesta es que puede ser implementada en organizaciones que ya cuenten con sistemas de información. El objetivo de este trabajo es buscar una estrategia de evolución de la web tradicional hacia la Web 3.0 para las organizaciones ya publican información a partir de bases de datos relacionales.

El presente trabajo está dividido de la siguiente forma, en la siguiente sección se presenta la *problemática* donde se establece la delimitación del trabajo. A continuación el *trabajo relacionado*, en donde se abordan los trabajos más importantes de acuerdo al objetivo del presente trabajo. Posteriormente la *propuesta*, en donde se describe las generalidades del trabajo. En el desarrollo de la propuesta se presenta la arquitectura, la implementación y los resultados obtenidos. Finalmente se abordan las *conclusiones y el trabajo a futuro*.

2 Problemática

La principal desventaja de los sitios de Internet estáticos es que al no estar su información almacenada en una base de datos no es posible realizar consultas a partir de un publicador de contenido. Mientras que los sitios de Internet dinámicos presentan desventajas al momento de ser indexados, pues los contenidos son generados sobre demanda y se necesitan directrices que marquen los atributos primarios que deben solicitarse para su indexación. Para indexar un sitio dinámico no basta con dar permisos de acceso a la información privada, cada recurso debe ser accesible mediante una URL única. Cabe mencionar que la información algunas veces es utilizada con más de un propósito; por ejemplo, en el sitio de Internet de una agencia de autos, los precios de los automóviles pueden ser utilizados por el departamento de marketing como información pública, o como información privada en el departamento de contabilidad. En los sitios estáticos, al no estar la información almacenada en una base de datos, este tipo de datos no son reverenciados más que como contenido web limitando la utilidad de esta información.

Las instituciones y empresas necesitan por tanto una estrategia que combine contenidos estáticos y dinámicos, una combinación de página de Internet corporativa e Intranet privada. Para agilizar la administración de páginas de Internet corporativas (estáticas) pueden utilizarse publicadores de contenidos tales como wordpress (Mullenweg) o Joomla (Source). Los publicadores de contenido son herramientas que a partir de la captura de información cruda automatizan su publicación con formato en un servidor. Sin embargo, esto supone capturar la información dos veces, una en la Intranet y otra en el publicador de contenidos. La administración de un publicador de contenidos requiere personal con un perfil diferente al que administra las herramientas propietarias, se necesita el doble del personal. Algunos publicadores de contenidos almacenan la información en una base de datos, pero bajo el diseño que la herramienta define y no bajo la arquitectura que la organización necesita. La información dinámica y la estática no son actualizadas al mismo tiempo por lo que

podrían ser diferentes y presentar inconsistencias.

En relación de los metadatos, el formato de datos RDF es la forma común de representar semánticamente información vinculada a datos en la web y se utiliza para realizar consultas en SPARQL en los repositorios de datos RDF. Los últimos años han visto un enorme crecimiento en el tamaño y la variedad de datos RDF con la disponibilidad de datos semánticos aplicado para cualquier disciplina (Jyoti, 2014); La mayoría de los datos en los que se basa la Web actual se almacenan en bases de datos relacionales con mayor frecuencia. Los datos representados en RDF pueden ser interpretados, procesados y razonados por los agentes de software. De ahí que el cambio del modelo relacional al modelo RDF para la manipulación de la información sea uno de los retos de investigación actuales (Álvarez, 2014). Así que, de acuerdo a las directrices de la web 3.0, nuestros contenidos requieren de una parte visual (para las personas) y de metadatos (para ser procesadas automáticamente por máquinas).

3 Trabajos relacionados

Estudios recientes demuestran que es posible trasladar del modelo de las base de datos relacionales al modelo basado en grafos RDF. Por medio de diferentes métodos y herramientas es posible realizar actualizaciones que reducen el costo del traslado de la información de las bases de datos al modelo de datos RDF. La investigación de (Álvarez, 2014) se enfoca a la actualización incremental de grafos RDF a partir de bases de datos relacionales. Cuando se produce un cambio se generan tripletas RDF que representan las actualizaciones, con esta información se actualiza el grafo RDF. Este enfoque reduce el coste en tiempo y recursos computacionales en relación de las técnicas tradicionales. Este trabajo muestra la relevancia de seguir un enfoque incremental, que evita hacer procesos desde cero y aprovecha los resultados existentes antes de los cambios.

Otro trabajo interesante es el de "Generación de Datos Semánticos a partir de una Base de Datos Relacional de una Institución de Educación Superior (León, 2014)". Esta investigación se enfoca en la difícil tarea de obtener datos abiertos RDF, en la propuesta este proceso se divide en cuatro etapas. La primera etapa: *análisis de los Datos*, se usa un gestor de Base de Datos (MySQL), para el proceso de migración y almacenamiento de los nuevos datos. La segunda etapa: *generación de mapeos*, se hace uso de las clases y propiedades definidas por la ontología DBpedia, así como también otros estándares como Dublin Core (DC), Friend of a Friend (FOAF), entre otros. En la tercera etapa: *obtención de datos RDF*, se realizó el traslado de los datos a formato RDF, lo que permite su posterior publicación y reutilización, estos estarían representados en forma de tripletas y contienen toda la información de la RDB original. En la cuarta y última etapa: *visualización de los datos*, se ejecutan consultas por medio de lenguaje SPARQL EndPoint (FrontEnd), que permite a los usuarios acceder y consultar datos RDF mediante el lenguaje SPARQL.

En "TRIOO, estudio e implementación de modelos de datos RDF en lenguajes orientados a objetos (López, 2010)"; se propone una tecnología que permite de una manera sencilla utilizar datos RDF directamente desde lenguajes orientados a objetos, permitiendo que el origen y la forma de los datos no tenga consecuencias para los diseños orientados a objetos, a la vez que la semántica de los datos se refleje de la manera más fiel posible. Es importante mencionar que en esta investigación se tiene una arquitectura bien definida basada en los argumentos. Lamentablemente por falta de tiempo estas pruebas no se realizaron de manera sistemática; por lo tanto estas pruebas no han sido totalmente completadas y documentadas.

En el artículo de (Ismail, 2008) se describe que el proceso de mapeo es importante en la integración de datos y conocimiento de múltiples bases de datos relacionales heterogéneas. El mapeo desde una base de datos relacional a una descripción que utiliza el Resource Description Framework (RDF) muestra resultados prometedores sin poner en peligro la semántica

de los datos. La integración es también importante en la realización de consultas simultáneas a múltiples bases de datos. El mapeo es un proceso importante para proporcionar puntos de vista homogéneos para el usuario. El Mapeo de una base de datos relacional (o diagrama ER) a un esquema RDF para una representación de los datos es una solución para superar los problemas de heterogeneidades semánticas y sintácticas. El RDF es un formato común de los datos para la Web Semántica. Es un lenguaje que describe diversas clases de objetos (recursos), sus propiedades y las relaciones entre ellos mediante el uso de declaraciones.

4 Propuesta

Para publicar información semántica a partir de una base de datos relacional se propone usar el lenguaje java; cuando las instrucciones relacionales de una base de datos desean invocarse orientadas a objetos se utiliza un API de persistencia. La persistencia se entiende como la capacidad de los objetos a sobrevivir más allá del tiempo que dura la ejecución de la aplicación (o rutina) en la que se crean. El API de persistencia encapsula el acceso a la base de datos y le permite al programador trabajar orientado a objetos.

Cuando se diseña una base de datos que representa un ámbito determinado, las entidades son representadas como tablas. Del mismo modo, en programación orientada a objetos, cada entidad del dominio de un problema es codificada como una clase.

A la correspondencia entre tablas y clases de una aplicación se le llama mapeo objeto-relacional. Son precisamente estas clases las que interesa publicar como archivos RDF pues cada una de ellas representa un concepto en particular; En la ilustración 1 se muestra un ejemplo del Mapeo Objeto-Relacional.

LIBRO	
ISBN	VARCHAr(13)
TITULO	VARCHAr(190)
RESUMEN	VARCHAr(45)
AUTOR_id	INT
Indexes	
PRIMARY	
fk_LIBRO_AUTOR_idx	

Libro	
-	String isbn
-	String titulo
-	String resumen
-	String autor
+	String getisbn()
-	Void setisbn(String)
+	String getTitulo()
-	Void setTitulo(String)
+	String getResumen()
-	Void setResumen(String)
+	String getAutor()
-	Void setAutor(Autor)

Ilustración 1 Mapeo objeto-relacional, tabla y clase para la entidad libro(elaboración propia).

Por cada tabla de la base de datos se crea una clase en la aplicación y por cada columna de cada tabla se le agrega una variable a cada clase. Mediante anotaciones se especifica cuál clase corresponde a cuál tabla y cuál variable a cuál columna.

Siguiendo este mismo patrón se determinaron 4 que describen las clases y su correspondencia con nodos de archivos RDF. Hay que recordar que la sintaxis RDF define un conjunto de tres elementos llamado tripleta compuesto de un "sujeto, un predicado y un objeto" (w3c, 2014), a este conjunto se le llama grafo RDF. En el sujeto se describe de quien se está hablando; en el predicado que también se le conoce como propiedad se representa una conexión entre un sujeto y un objeto, normalmente una relación del sujeto con algo o un atributo del sujeto; en el objeto se almacena el valor de lo declarado. En la tabla 1 se muestran parte de las anotaciones que se desarrollaron.

Tabla 1 Anotaciones que se desarrollaron

Anotación	Definición
@RDFSujeto(url, extension):	Esta es una anotación que se le coloca a las clases que se van a convertir a archivos RDF. En esta se especifica la URL del repositorio en la que se van a ubicar los archivos y la extensión del mismo. Ej. http://biblioteca.com/libros ,

	http://biblioteca.com/autores
@RDFIdentifier	Se le coloca al método que retorna una cadena de texto cuyo valor es utilizado como nombre del archivo RDF. Es por tanto el identificador y no debe producir valores repetidos. Ej. libro32165.rdf; por lo que lo hace único.
@RDFPropiedad (url, prefix, nombre)	Se le coloca a las variables alfanuméricas cuyo valor se insertará como nodo en el archivo RDF. Es necesario especificar el prefijo, etiqueta y namespace del nodo a generar.
@RDFResource(url, prefix, nombre)	Se le coloca a las variables que hacen referencia a otras clases. En el archivo RDF se agregará un hipervínculo a otro archivo RDF. El hipervínculo se construye dependiendo de las anotaciones de la clase referenciada.

A continuación se muestra el diagrama de clases donde se representa los elementos necesarios para generar la tripleta; donde cada uno de ellos debe contener una URI esta es importante ya que hace referencia a la tripleta única para esto se crea un RDFIdentifier este crea un id único e irrepitible; En la ilustración 1 se muestra.



Ilustración 2 Diagrama de Clases (elaboración propia).

Cabe mencionar que estas anotaciones no afectan el funcionamiento del API de persistencia. A la conversión de objetos en archivos XML se le llama marshalling. En este caso se desarrolló una clase marshaller RDF que inspecciona las propiedades y datos de un objeto utilizando el API "reflection" de Java y que accede a los valores de sus métodos utilizando el método `invoke`. A partir de la obtención de estos valores y de las configuraciones especificadas en las anotaciones se concatenan las secciones del archivo RDF en una cadena de texto.

Los contenidos RDF son almacenados en una sección pública del servidor de aplicaciones y el resto del sitio web hará referencia a ellos mediante hipervínculos.

XML (Xtensible Markup Language) Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. (Bray, 2006).

Para la generación automática de los archivos HTML se puede utilizar el API JAXB (Java Architecture for XML Binding) que permite generar archivos XML también a partir de objetos con anotaciones. A partir de un archivo XML puede obtenerse un archivo HTML si se combina con una hoja de estilos XSL (Extensible Stylesheet Language) para generar un nuevo

XML mediante la filtración de algunas secciones de la información en el archivo XML original.(Medhioub, 2011).

5 Desarrollo de la propuesta

Para la implementación de esta investigación se desarrolló un sistema con el patrón arquitectura de software MVC (Modelo, Visita, Controlador) donde la capa de presentación se muestra la interfaz de usuario, la capa lógica de negocios recibe las peticiones del usuario y se envían las respuestas tras el proceso, y la capa de datos donde residen los datos. Implementar el patrón MVC permite dar un mantenimiento fácil a las aplicaciones web debido a una clara separación entre las capas de presentación, lógica de negocio y acceso a datos. Para el desarrollo de esta investigación se crea un módulo en la capa lógica de negocios para los RDF; para que posteriormente se publique la información de manera estática en un sitio web público y de forma paralela se guarda el RDF en un repositorio; En la figura 3 se muestra la arquitectura de esta propuesta.

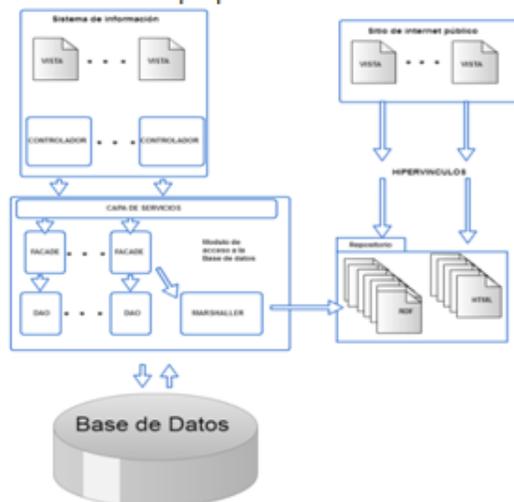


Ilustración 3 Arquitectura propuesta (elaboración propia).

Para la implementación de esta técnica se desarrolló un Sistema de Información que

administra la información de cursos ofertados por profesores en una Universidad. Esta aplicación requiere que sea una aplicación web dinámica en la que cierta parte de su información necesita de su divulgación como contenido abierto, en formato estático y semántico para los usuarios externos, referenciable por una URL única. La aplicación web se desarrolló con la tecnología JSF 2.1 (Java Server Faces) implementada con la biblioteca de componentes de Primefaces 5.2; Se utilizó el framework de desarrollo de aplicaciones web Spring 4 y el API de persistencia Hibernate 4. La información es almacenada en una base de datos relacional MySQL. Todo esto, en un servidor Tomcat 7 sobre JVM 1.7 configurado para permitir la escritura en el servidor.

Se desarrollaron las clases descriptoras de los objetos mediante la API Reflection, para ello se definieron las notaciones necesarias para la aplicación. Se usó JAXB para obtener páginas HTML y los XML que permiten publicar los datos que se deseados. La aplicación web se modificó para que cuando se ejecuten operaciones CRUD (Create -Read -Update -Delete) las operaciones se puede realizar en cualquier elemento en sus catálogos, (Dittawit, 2012). Las modificaciones también permiten generar los respectivos archivos HTML estáticos que son guardados en una carpeta pública del servidor Tomcat.

6 Resultados

La aplicación web permite darle seguimiento a la administración de cursos a continuación se muestra la interfaz de alta de un curso; En la figura 4 se muestra Vista de alta de un nuevo curso.

La aplicación web permite darle seguimiento a la administración de cursos a continuación se muestra la interfaz de alta de un curso; En la figura 4 se muestra Vista de alta de un nuevo curso.

Curso

Curso

Nombre: Bases de Datos

Descripción: BASICO

Horario: LUNES A VIERNES 9 A 12

Profesor: ANGELA ROS LUNA

Ubicación: edificio=1, salon=1

Ilustración 4 Vista de alta de un nuevo curso(elaboración propia).

Para el alta de un curso se deberá tener acceso al sistema ingresando usuario y contraseña. Una vez que el administrador da de alta el curso; se muestra de manera estática para que los usuarios externos que no se loguen al sistema puedan visualizar la lista de cursos que se ofertan; En la figura 5 se muestra Vista Externa de la lista de cursos.

Cursos

Lista de cursos

TEST	BASES DE DATOS
ANGELA ROS LUNA 2016-02-16 12:47:22.0 edificio=1, salon=1 Mas informacion...	ANGELA ROS LUNA 2016-02-22 21:59:16.0 edificio=1, salon=1 Mas informacion...
SISTEMAS DE INFORMACION	
ANGELA ROS LUNA 2016-02-22 22:04:44.0 edificio=1, salon=1 Mas informacion...	

Ilustración 5 Vista Externa de la lista de cursos(elaboración propia).

Con el software desarrollado en este trabajo se logra generar de forma automática el RDF para cada uno de los cursos y este se guarda en una carpeta pública dentro del servidor Tomcat; a continuación se muestra el RDF generado; En la figura 6 se muestra Generación automática del RDF.

```

<!DOCTYPE html>
<html version="1.0" >
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:uam="http://localhost:8080/curso/"
<rdf:Description rdf:about="http://localhost:8080/publico/curso/Curso01.html">
<uam:profesor rdf:resource="http://localhost:8080/publico/curso/Profesor01.html"/></uam:profesor>
<uam:ubicacion rdf:resource="http://localhost:8080/publico/curso/Ubicacion01.html"/></uam:ubicacion>
<uam:nombre>BASES DE DATOS</uam:nombre>
<uam:descripcion>BASICO</uam:descripcion>
<uam:fecha_de_inicio/></uam:fecha_de_inicio>
<uam:fecha_de_termino/></uam:fecha_de_termino>
<uam:horario>LUNES A VIERNES 9 A 12</uam:horario>
</rdf:Description>
</rdf:RDF>
</html>

```

Ilustración 6 Generación automática del RDF(elaboración propia).

La información generada mediante los RDF permiten que sea visualizada de dos formas distintas, de acuerdo a la necesidad de la empresa, puede visualizarse en formato de tripleta (véase ilustración 7) o en forma de grafo (véase la Ilustración 8).

Triplets of the Data Model

Subject	Predicate	Object
http://localhost:8080/publico/curso/Curso01.html	http://localhost:8080/publico/curso/Profesor01.html	http://localhost:8080/publico/curso/Profesor01.html
http://localhost:8080/publico/curso/Curso01.html	http://localhost:8080/publico/curso/Ubicacion01.html	http://localhost:8080/publico/curso/Ubicacion01.html
http://localhost:8080/publico/curso/Curso01.html	http://localhost:8080/publico/curso/Nombre01.html	BASES DE DATOS
http://localhost:8080/publico/curso/Curso01.html	http://localhost:8080/publico/curso/Descripcion01.html	BASICO
http://localhost:8080/publico/curso/Curso01.html	http://localhost:8080/publico/curso/Horario01.html	LUNES A VIERNES 9 A 12

Ilustración 7 Tripleta de un curso(elaboración propia).



Ilustración 8 Representación de un curso en grafo(elaboración propia).

Conclusiones y trabajos futuros

Se cuenta con una API que facilita la publicación de contenidos semánticos a partir de objetos obtenidos de bases de datos relacionales, y esta sirve para aquellos desarrolladores que requieran convertir su sitio web estático o dinámico a un sitio web 3.0; este representa una herramienta de desarrollo que transforma información almacenada en una base de datos relacional, datos estructurados a contenido semántico que contiene datos semiestructurados; genera nuevos canales de comunicación y comercio para las empresas e instituciones. Los sitios semánticos posicionan a las instituciones como generadoras de contenidos confiables y referenciales, aumentando su presencia en internet.

Este trabajo reporta la primera versión de esta API sobre la generación automática de

contenidos semánticos a partir de sistemas de información de uso general, cabe mencionar que para trabajos a futuro se pretende darle mantenimiento a esta API para que sea más completa, algunos aspectos que no se han desarrollado son:

- Cuando la aplicación puede modificarse.
- Cuando la base de datos ya tiene información.
- Cuando la información debe ser descrita con estándares usando metadatos

También se podrá desarrollar para los usuarios finales que no sean programadores aplicaciones de propósito general que funcione con requerimientos específicos y genere contenido semántico. Una de las ventajas de las anotaciones RDF es que reutilizan las clases de persistencia sin entrar en conflicto con los sistemas ya desarrollados, disminuyendo así el esfuerzo necesario para su implementación y los riesgos de modificar la operación del sistema.

Bibliografía

Álvarez, L. R. (2014). Actualización incremental de grafos RDF a partir de bases de datos relacionales. Actas de las XIX Jornadas de Ingeniería del Software y Bases de Datos (pág. 6). Cádiz: Javier Tuya, Mercedes Ruiz y Nuria Hurtado.

Bray, T. (2006). *Extensible Markup Language (XML)*.

Dittawit, K. (2012). A Linked Data Model for E-books. En *2012 IIAI International Conference on Advanced Applied Informatics* (pág. 5). IEEE.

Heath, T. (s.f.). *Linked Data*. Recuperado el 27 de 03 de 2016, de <http://linkeddata.org/>

Ismail, M. A. (2008). Integration of Heterogeneous Relational Databases: RDF Mapping Approach. *IEEE*, 7.

Jyoti, L. (2014). RQ-RDF-3X: Going Beyond Triplestores. *5th International Workshop on Data Engineering Meets the Semantic Web* (pág. 6). Chicago IL: IEEE.

León, f. T. (2014). Generación de Datos Semánticos a partir de una Base de Datos

Relacional de una Institución de Educación Superior. *10th Latin American and Caribbean Conference for Engineering and Technology*, (pág. 2). Ecuador.

López, S. F. (2010). *TRIOO, estudio e implementación de modelos de datos RDF en lenguajes orientados a objetos*.

Medhioub, H. (2011). Design, Implementation and Evaluation of Virtual Resource Description and Clustering Framework. En *2011 International Conference on Advanced Information Networking and Applications* (pág. 7). IEEE.

Mullenweg, M. C. (s.f.). *WordPress*. Recuperado el 27 de 03 de 2016, de <https://es.wordpress.com/about/>

Source, M. O. (s.f.). *Joomla*. Recuperado el 27 de 03 de 2016, de <https://www.joomla.org/about-joomla.html>

w3c. (25 de febrero de 2014). *RDF concepts*. Obtenido de <http://www.w3.org/TR/rdf11-concepts/>

BIBLIOGRAFÍA

42, M. (s.f.). *mind42*. Obtenido de <http://mind42.com/>

Actualización incremental de grafos RDF a partir de bases de datos relacionales 2014 *Actas de las XIX Jornadas de Ingeniería del Software y Bases de Datos* Cádiz Javier Tuya, Mercedes Ruiz y Nuria Hurtado

Applications, A. F. (2013). Javier Solis.

Bray, T. (2006). *Extensible Markup Language (XML)*.

Builder, S. W. (s.f.). *Semantic Web Builder*. Obtenido de <http://www.semanticwebbuilder.org.mx/>

Carmen. (02 de febrero de 2011). *La Web de Datos Enlazados como espacio global de información*.

Castro, C. C. (2012). Vocabularios estructurados , Web Semántica y Linked Data : oportunidades y retos para los profesionales de la documentación.

Codina, L. (2009). Web 2.0, Web 3.0 o Web Semántica?: El impacto en los sistemas de información de la Web. *I Congreso Internacional de Cyberperiodismo y Web 2.0*. . Universidad Pompeu Fabra .

Cortes, D. F. (2012). *Planificación TFC / Diseño y población semiautomática de ontologías*. .

Davies, P. B. (2014). *Sistemas de informacion*. Barcelona, España: Reverte.

Dittawit, K. (2012). A Linked Data Model for E-books. En *2012 IIAI International Conference on Advanced Applied Informatics* (pág. 5). IEEE.

Fernandez, S. (2010). Keeping the Semantics of Data Safe and Sound into Object-Oriented Software.

Florescu, D. (2000). *Storing and Querying XML Data using an RDMBS*.

Gamma, E. (2009). *Patrones de Diseño Elementos de software orientados a objetos reusable*. Madrid.

García Ruiz, R., & Arroyo Gómez, S. (2008). XML y Web Semántica.

García, G. G. (s.f.). *Instituto de Matemáticas de la UNAM*. Obtenido de http://www.matem.unam.mx/~grecia/semantic_web/rdf.html

Google. (s.f.). *google-refine*. Obtenido de <https://code.google.com/p/google-refine/>

Group, O. W. (11 de 12 de 2012). *OWL*. Obtenido de <http://www.w3.org/>: Web Ontology Language (OWL)

Heath, T. (s.f.). *Linked Data*. Recuperado el 27 de 03 de 2016, de <http://linkeddata.org/>

hp. (s.f.). *Jena*. Obtenido de <https://jena.apache.org/index.html>

HP. (s.f.). *RDF JSON*. Obtenido de <http://jena.apache.org/documentation/io/rdf-json.html>

Ismail, M. A. (2008). Integration of Heterogeneous Relational Databases: RDF Mapping Approach. *IEEE* , 7.

Jyoti, L. (2014). RQ-RDF-3X: Going Beyond Triplestores. *5th International Workshop on Data Engineering Meets the Semantic Web* (pág. 6). Chicago IL: IEEE.

kowari. (s.f.). *Kowari Metastore Banner*. Obtenido de <http://kowari.sourceforge.net/>

León, f. T. (2014). Generación de Datos Semánticos a partir de una Base de Datos Relacional de una Institución de Educación Superior.

López, S. F. (2010). *TRIOO, estudio e implementación de modelos de datos RDF en lenguajes orientados a objetos*.

Matute, J. H. *Un estudio comparativo entre los sistemas gestores RDF*. MADRID: UNIVERSIDAD CARLOS III DE MADRID ESCUELA.

McBride, B. (2014). *W3C Recommendation RDF Schema* .

Medhioub, H. (2011). Design, Implementation and Evaluation of Virtual Resource Description and Clustering Framework. En *2011 International Conference on Advanced Information Networking and Applications* (pág. 7). leee.

Mehta, B. (s.f.). *Oracle Technology Network*. Recuperado el 18 de 2013 de 2016, de ORACLE: <http://www.oracle.com/technetwork/articles/javase/index-140168.html>

Mota, S. A. (2009). *Paradigmas de Modelado de Base de Datos*.

Mullenweg, M. C. (s.f.). *WordPress*. Recuperado el 27 de 03 de 2016, de <https://es.wordpress.com/about/>

Pacheco, E. H. (2015). Desarrollo Orientado a la Semántica. *Develop Network No. 6* .

Prud'hommeaux, E. (03 de 2013). *SPARQL Query Language for RDF*. Obtenido de w3c: <http://www.w3.org/TR/rdf-sparql-query/>

Ramachandran, K. (03 de 08 de 2011). *Creación de aplicaciones Web inteligentes para un mundo más inteligente*. Obtenido de developerWorks: <http://www.ibm.com>

Ramírez, R. (2011). Informe de estado del arte de Tecnologías Web Semántica y Social aplicada a la accesibilidad .

Ravada, N. A. (2006). RDF Object Type and Reification in the Database. *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)* 8-7695-2570-9/06 \$20.00 © IEEE .

Ruiz, R. G. (2008). *XML y Web Semántica*.

Ruiz, R. G. (208). *XML y Web Semántica*.

Saquete, R. (10 de Junio de 2013). El impredecible futuro de la Web semántica.

Schorlemmer, M. (2011). Diez años construyendo una web semántica. *Fundación General CSIC* .

Sciences, U. o. (2014). Actualización incremental de grafos RDF a partir de bases de datos relacionales.

Semantica, W. (s.f.). Obtenido de http://www.semanticwebbuilder.org.mx/es_mx/swb/SWB_Platform

Solis, J. (2013). A Framework for Semantic Web Applications Development.

Source, M. O. (s.f.). *Joomla*. Recuperado el 27 de 03 de 2016, de <https://www.joomla.org/about-joomla.html>

Tapia, L. F. (2014). Generación de Datos Semánticos a partir de una Base de Datos Relacional de una Institución de Educación Superior. *12th Latin American and Caribbean Conference for Engineering and Technology*, (pág. 2). Guayaquil Ecuador.

W3. (s.f.). *Notacion 3*. Obtenido de <http://www.w3.org/DesignIssues/Notation3.html>

W3. (s.f.). *N-TRIPLES*. Obtenido de <http://www.w3.org/TR/n-triples/>

W3. (s.f.). *RDF*. Obtenido de <http://www.w3.org/TR/rdfa-lite/>

w3c. (2001). *Especificación del Modelo y la Sintaxis Resource Description Framework*. Recuperado el 2016, de <http://www.sidar.org/recur/desdi/traduc/es/rdf/rdfesp.htm>

w3c. (s.f.). *Estandares de la web semantica*. Obtenido de <http://www.w3.org/standards/semanticweb/>

w3c. (25 de febrero de 2014). *RDF concepts*. Obtenido de <http://www.w3.org/TR/rdf11-concepts/>

w3c. (s.f.). *Web Semantica*. Obtenido de <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>