



**Universidad de las Ciencias Informáticas**

**FACULTAD 10**

**Título:**

**Módulo de Inferencia para un Repositorio Semántico**

**Trabajo de Diploma para optar por el Título de  
Ingeniero en Ciencias Informáticas**

**Autores**

**Susana Zalazar Bravo**

**Maikel Aparicio Reytor**

**Tutor**

**MSc. David Leyva Leyva**

**Ciudad de La Habana**

**Junio del 2009**





*No basta saber, se debe también aplicar.  
No es suficiente querer, se debe también hacer.*

*Johann Wolfgang Goethe*

## **Declaración de autoría**

Declaramos que somos los únicos autores del trabajo titulado:

Módulo de Inferencia para un Repositorio Semántico

y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Susana Zalazar Bravo

\_\_\_\_\_  
Maikel Aparicio Reytor

\_\_\_\_\_  
MSc. David Leyva Leyva

## **Datos de Contacto**

Nombre y Apellidos del **Tutor**: MSc. David Leyva Leyva

Email: davidl@uci.cu

Teléfono: (53) (07) 837 2511

Currículum: Jefe de Departamento de Técnicas de Programación, Facultad 10, Universidad de las Ciencias Informáticas. Licenciado en Cibernética-Matemática, Universidad de Las Villas (UCLV), 1988. Máster en Computación Aplicada, Universidad de Las Villas (UCLV), 1995.

Profesor de la Universidad de Holguín desde 1990. Jefe de Departamento de Informática (2003-2005). Participó en un Proyecto de Educación a Distancia en la Universidad del 2001 hasta el 2005, año en el que comenzó a trabajar en la Universidad de Ciencias Informáticas. Ha participado en un gran número de eventos nacionales e internacionales. Le fueron concedidas sendas becas Intercampus en La Universidad de Castilla-La Mancha (1998) y La Universidad Autónoma de Madrid (2000). Trabajó como profesor invitado en la Universidad de Belice (Belice, de 2001 a 2003) y de la Universidad Nacional de Ingeniería (Managua, Nicaragua, 2005).

Actualmente es miembro de un grupo de investigación y desarrollo orientado a la extensión de la plataforma Moodle y el desarrollo de otras herramientas para la teleformación.

## **Agradecimientos**

***De Susana:***

***Quisiera agradecer en primer lugar a Dios por ayudarme en todo momento y darme fuerzas para afrontar los momentos difíciles.***

***A mi madre, por su amor sin límites, por los sacrificios, preocupaciones, por ser mi guía, mi consejera y mi mejor amiga.***

***A mi padre quien siempre ha mantenido la confianza en mí y se ha preocupado siempre por mis estudios.***

***A mis abuelos por apoyarme y preocuparse tanto por mí. En especial a mi abuela Milagros por ser mi paño de lágrimas y por darme sabios consejos.***

***A mis hermanas Laurita, Madelín, y Yaniris Blanco.***

***A Mercedes pues ha sido como una madre para mí.***

***A Pulido quien siempre me ha dado muy buenos consejos.***

***En fin agradezco a toda mi familia porque todos de una forma u otra han aportado un granito de arena durante mi carrera para llegar a ser lo que hoy soy.***

***Un lugar muy especial a mi novio quien me ha soportado en mis momentos de angustias.***

***A mi compañero de tesis, amigo por ser paciente conmigo.***

***A todas mis amistades del aula que también me han auxiliado en el presente trabajo, en especial a Mairelis, Yorlenis, Lily y Yadira.***

***Al tutor por ayudarnos a realizar esta difícil tarea.***

***A Roxana quien también ha contribuido en gran parte con la realización del presente trabajo.***

***De Maikel:***

***Les agradezco a mis hermanos Ariel, Ernesto, Leover, Yosbany y Ludiel por estar siempre a mi lado tanto en los momentos malos como en los buenos***

***A mi compañera de tesis Susana Zalazar por estar siempre pendiente del desarrollo del trabajo.***

## **Dedicatoria**

***Susana***

***A mi madre por su amor, dedicación y sacrificio.***

***Maikel***

***A mi abuela Eloina, a quien le agradezco todo lo bueno que he hecho.***

## **Resumen**

Unido al desarrollo de las tecnologías de la comunicación se ha mejorado significativamente el acceso a Internet, convirtiéndose en un instrumento de uso cotidiano en la sociedad actual. La Web actual a pesar de haber alcanzado un gran éxito presenta algunas limitaciones, como son: la existencia de mucha información en la web y la falta de organización de la misma. Una posible solución a estos problemas es la Web Semántica, esta permite el procesamiento automático y organización de la información en la web.

En la Universidad de Ciencias Informáticas se creó un Repositorio de Objetos de Aprendizaje que permite el almacenamiento y búsqueda de Objetos de Aprendizaje (OA), pero el número de OA crece y las búsquedas se hacen más difíciles, por lo que para mejorar el sistema de búsquedas y el acceso a la información que hay en este, y organizar los OA, se utilizarán las tecnologías que se necesitan para la implementación de la web semántica, tales como ontologías para la organización de la meta-información que poseen los OA y la creación de un módulo de inferencia para la gestión del conocimiento en las ontologías.

Como resultado de la investigación se crearon ontologías para modelar la organización de la meta-información de los OA y un módulo de inferencia para la gestión de información en las mismas, formando con esto la base para hacer una extensión semántica del ROA, extensión que contara con otros módulos.

Palabras claves: e-Learning, Gestión del Conocimiento, Metadatos, Objetos de Aprendizaje, Ontologías, Web Semántica.

## Tabla de Contenido

Introducción .....	1
Capítulo 1. Fundamentos Teóricos.....	8
1.1 Introducción al Capítulo. ....	8
1.2 La Web actual .....	8
1.3 La Web Semántica .....	10
1.3.1 Gestión del conocimiento.....	12
1.3.2 Metadatos .....	15
1.3.3 Tecnologías que sustentan la Web Semántica. ....	16
1.3.3.1 Lenguajes que sustentan la Web Semántica.....	17
1.3.3.1.1 Extensible Markup Language (XML) y XML Schema. ....	17
1.3.3.1.2 RDF y RDF-Schema. ....	18
1.3.3.1.3 Ontologías para la Web Semántica.....	21
1.3.3.1.4 Lenguajes para escribir Ontologías.....	24
<b>Simple HTML Ontology Extensions (SHOE).</b> ....	24
<b>DAML-OIL (DARPA's Agent Markup Language + Ontology Inference Layer).</b> ....	24
<b>OWL (Ontology Web Language) y Fuzzy OWL.</b> .....	25
1.3.3.5 Lenguajes de reglas.....	28
<b>RuleML (Rule Markup Language) y Fuzzy-RuleML.</b> .....	28
<b>SWRL (Semantic Web Rule Language) y F-SWRL (Fuzzy Semantic Web Rule Language).</b> ...	29
1.3.3.6 Lenguajes de consulta.....	31
<b>XQuery</b> .....	31



---

---

<b>SPARQL (Simple Protocol and RDF Query Language)</b> .....	32
<b>RDQL (RDF Data Query Language)</b> .....	33
<b>SeRQL (Sesame RDF Query Language)</b> .....	34
1.3.3.2 Herramientas de la Web Semántica .....	35
1.3.3.2.1 Razonadores.....	35
<b>Pellet</b> .....	35
<b>RACER (Renamed ABox and Concept Expression Reasoner)</b> .....	36
<b>Cerebra</b> .....	37
1.3.3.2.2 Editores ontológicos. ....	37
<b>Protégé</b> .....	37
<b>Swoop</b> .....	38
<b>WebOnto</b> .....	40
<b>OntoEdit</b> .....	41
<b>OilEd</b> .....	42
1.3.3.2.3 Herramientas para el trabajo complementario.....	42
<b>eXist (Open Source Native XML Database)</b> .....	43
<b>Jena</b> .....	43
<b>Sesame</b> .....	44
<b>KAON Tool</b> .....	44
<b>KAON2</b> .....	45
<b>JAVA, IDE NetBeans 6.5</b> .....	46
1.3.4 Buscadores Semánticos .....	46
1.4 E-Learning y Repositorios .....	48
1.4.1 Objetos de Aprendizaje (OA).....	49

## Tabla de Contenido

---

---

1.4.2 Repositorios de OA .....	50
1.4.3 Repositorio Semántico .....	51
1.4.4 Repositorio Semántico de OA.....	52
1.5. Repositorio Semántico de OA en la UCI. ....	53
Capítulo 2. Propuesta del Repositorio Semántico.....	55
2.1 Objetivo del repositorio semántico.....	55
2.2 Descripción interna del Repositorio Semántico. ....	56
2.3 Especificación de los procesos del Repositorio Semántico. ....	62
Flujo de procesos .....	62
2.4 Creación de la ontología. ....	67
2.4.1 Proceso de creación de la ontología OntoLOM. ....	67
2.5 Proceso de Inferencia en ontologías .....	72
Capítulo 3. Pruebas.....	75
3.1 Casos de Prueba .....	75
3.1.1 Prueba de caja blanca.....	75
3.1.1.1 Pruebas de unidad. ....	78
3.1.2 Pruebas de rendimiento. ....	81
3.1.3 Pruebas a la Ontología.....	87
Conclusiones Generales.....	89
Recomendaciones .....	90
Referencias Bibliográficas.....	91
Bibliografía.....	99
Anexos .....	100
Glosario de Términos.....	102

## **Índice de Tablas**

Tabla 1: Comparativa entre las diferentes metodologías propuestas en cada trabajo para la creación de ontologías.....	15
Tabla 2: Comparativa entre lenguajes de especificación de ontología.s.....	27
Tabla 3: Dominios correspondientes a propiedades.....	69
Tabla 4: Suma de propiedades para obtener el concepto R_Cost.....	69
Tabla 5: Suma de propiedades para obtener el concepto R_Copyrightandotherrestrictions.....	69
Tabla 6: Suma de propiedades para obtener el concepto Rights.....	70
Tabla 7: Caso de prueba # 1 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.....	77
Tabla 8: Caso de prueba # 2 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.....	77
Tabla 9: Caso de prueba # 3 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.....	78
Tabla 10: Caso de prueba # 4 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.....	78
Tabla 11: Rendimiento del proceso de llenado del Modelo Ontológico en dependencia de la cantidad de manifiestos.....	82
Tabla 12: Validación de la consistencia de los conceptos.....	87
Tabla 13: Validación de la taxonomía.....	87
Tabla 14: Cálculo clases equivalentes.....	88
Tabla 15: Cálculo de tipos inferidos.....	88

## **Índice de Figuras**

Figura 1: Esquema del Repositorio Semántico.....	55
Figura 1: Esquema del Repositorio Semántico.....	55
Figura 2: Persistencia de datos en eXist. ....	58
Figura 3: Flujo de mensajes entre los paquetes.....	60
Figura 4: Interacción entre los paquetes del Módulo Semántico. ....	61
Figura 5: Flujo de proceso Crear Modelo. ....	62
Figura 6: Flujo de proceso Almacenar Modelo. ....	63
Figura 7: Flujo de proceso Cargar los metadatos de los paquetes SCORM. ....	64
Figura 8: Flujo de proceso Actualizar. ....	65
Figura 9: Flujo de proceso Realizar proceso de recuperación de información.....	66
Figura 10: Codificación XML de los metadatos en la categoría Rights. ....	68
Figura 11: Reglas restrictivas por propiedad.....	70
Figura 12: Modelo Ontológico.....	71
Figura 13: Conceptos más generales del Modelo Ontológico.....	71
Figura 14: Desglose conceptual de las nueve categorías de LOM conceptualizadas. ....	72
Figura 15: Arquitectura de Pellet.....	73
Figura 16: Porción de Código.....	76
Figura 17: Grafo de flujo.....	76
Figura 18: Prueba de unidad realizada a la clase Add_to_Model. ....	79
Figura 19: Prueba de unidad realizada a la clase LlenarOntología.....	80
Figura 20: Prueba de unidad realizada a la clase XMLReaderSample. ....	81

Figura 21: Pruebas de rendimiento Call Tree y Hot Spots realizadas a la clase LlenarOntología.....	83
Figura 22: Pruebas Call Tree y Hot Spots realizadas al proyecto completo.....	85
Figura 23: Pruebas de Telemetría a la Máquina Virtual.....	85
Figura 24: Prueba Memory Heap realizada al proyecto completo.....	86
Figura 25: Ejemplo de Tesauro.....	100
Figura 26: Resultados Obtenidos con un buscador normal.....	101
Figura 27: Resultados Obtenidos con un buscador semántico.....	101

### Introducción

Unido al desarrollo de las tecnologías de la comunicación se ha mejorado significativamente el acceso a Internet, convirtiéndose en un instrumento de uso cotidiano en la sociedad actual, comparable a otros medios tan importantes como la radio, la televisión o el teléfono, a los que aventaja en muchos aspectos. Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web<sup>1</sup> (WWW, o "la Web"), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto que no es más que *“una tecnología que organiza una base de información en bloques distintos de contenidos, conectados a través de una serie de enlaces cuya activación o selección provoca la recuperación de información”* según (Bianchini, 2000).

La web es hoy un medio extraordinariamente flexible, económico para la comunicación, el comercio, los negocios, ocio, entretenimiento, acceso a información, servicios, difusión de cultura, etc.

La Web actual a pesar de haber alcanzado un gran éxito presenta algunas limitaciones, las cuales, si se pudiesen mejorar permitirían un mayor grado de aceptación por parte de los usuarios. La existencia de mucha información en la web y la falta de organización de la misma dificulta el trabajo a la hora de realizar alguna búsqueda sobre un tema determinado, pues en ocasiones se busca algo específico sobre un tema y se obtienen documentos cuyo contenido no guardan relación alguna con lo que se quiere encontrar o al contrario, se encuentra lo que se buscaba sólo después de visitar varias páginas.

Muchas veces ocurre que el término utilizado para realizar nuestra búsqueda posee disímiles significados, arrojando resultados ajenos al deseado, pues se obtienen páginas web relacionadas con sus otras acepciones.

---

<sup>1</sup>(Telaraña de Cobertura Mundial, World Wide Web): Sistema de información distribuido, basado en hipertexto, creado a principios de los años 90 por Tim Berners Lee, Suiza. (PSI, 2008)

Una posible solución a estos problemas que presenta la web actual es la Web Semántica (WS), o web del futuro, la cual no es más que *“Una visión de una futura web en la cual la información que en la versión actual de la web es comprensible solamente por los seres humanos también esté disponible de una manera formal para sistemas inteligentes”*. (Sánchez, y otros, 2005)

Esta web posibilita que los investigadores de cualquier disciplina puedan compartir sus conocimientos, permite el procesamiento automático de la información y la organización de la información en la web, la búsqueda de información de manera eficaz y precisa, factores que contribuirían a un mejor funcionamiento de los sistemas para la *gestión del conocimiento* (GC) que no es más que *“Proceso sistemático de buscar, organizar, filtrar y presentar la información con el objetivo de mejorar la comprensión de las personas en un área específica de interés.”* según Figueroa (Figueroa, y otros, 2006), tema que será abordado más adelante.

La WS es una web extendida, dotada de significado, en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma rápida y sencilla gracias a una información mejor definida. Al dotar a la web de más significado, es decir, 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. Esta web extendida y basada en la descripción de la información, haciendo uso de metadatos<sup>2</sup> y ontologías (acerca de las cuales se abordará más adelante), se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante; aunque hoy en día ya existen sitios que poseen semántica utilizando dicha tecnología.

La WS se sustenta de las tecnologías basadas en Extensible Markup Lenguaje (XML), XML Schema, *Resource Description Framework* (RDF)/RDF-Schema y Ontology Web Lenguaje (OWL), este último se considera el lenguaje clave para el modelado de ontologías<sup>3</sup> las mismas *“son especificaciones formales de cómo representar los objetos, conceptos y otras entidades que se*

---

<sup>2</sup> “los metadatos son datos altamente estructurados que describen información, el contenido, la calidad, la condición y otras características de los datos. Es información sobre información o datos sobre los datos”.

<sup>3</sup> Será explicado más adelante. (Metadatos, 2007)

*asume que existen en un área de interés, así como las relaciones que se mantienen entre sí*", según Abián (Abián, 2005), las cuales le dan el fundamento semántico a esta extensión de la web actual.

El concepto de ontología ha estado presente desde hace mucho tiempo en la filosofía, sin embargo recientemente se utiliza en Informática para definir vocabularios que las máquinas puedan entender y que sean especificados con la suficiente precisión como para permitir diferenciar términos y referenciarlos de manera precisa. Las ontologías modelan la información de un dominio (un dominio es simplemente un área de temática específica o un área de conocimiento, tales como medicina, fabricación de herramientas [...], etc.) y a dicha información se le pueden realizar procesos de inferencia, es decir, es necesario que las máquinas actuales sean capaces de mediante procesos lógico-matemáticos inferir, deducir conclusiones a partir de datos, descripciones, conceptos e instancias. (Coss, 2006)

Haciendo uso de las tecnologías que dan soporte a la WS se han creado varios "Buscadores Semánticos" (tales como: SWoogle<sup>4</sup>, NaturalFlinder, Swootti, Ideas Afines, Askwiki, Powerset, Hakia, Lexxe, etc.) (Javier), que no son más que *"aquellos que realizan el rastreo atendiendo al significado del grupo de palabras que se escriben (Web Semántica "WS") y no basándose en las actuales etiquetas. En pocas palabras, un buscador inteligente* (Javier), los cuales son más precisos que los buscadores<sup>5</sup> tradicionales (Casal, 2008) (como el de Google, Yahoo, Altavista, Excite, HotBot, Infoseek, etc.) al brindar los resultados de una búsqueda.

Las mismas dificultades por las que atraviesan los buscadores tradicionales al realizar las búsquedas sucede con los contenidos almacenados en lo que se conoce como "almacenes de contenidos"<sup>6</sup>, y estos contenidos constituyen una parte fundamental de una propuesta de formación que contempla su implementación predominantemente mediante internet, haciendo uso de los servicios y herramientas que esta tecnología provee, conocida como e-Learning, definido por Red

---

<sup>4</sup> <http://swoogle.umbc.edu>.

<sup>5</sup> Un **buscador** es, en realidad, una herramienta que gestiona bases de datos de URLs con distintos contenidos.

<sup>6</sup> Repositorio de contenido.



TTnet como *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes.”* (Red TTnet, 2005)

Una de las herramientas en las que se apoya el e-Learning es en los conocidos “repositorios” los cuales no son más que contenedores de colecciones de recursos de cualquier tipo, pero cuando se habla de e-Learning y repositorios, se deben citar los Repositorios de Objetos de Aprendizaje (ROA), que son los que agrupan y almacenan los llamados Objetos de Aprendizaje (OA) los cuales no son más que *“cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-Learning”*. (López, 2005)

Los repositorios actuales de objetos de aprendizaje, tales como MERLOT<sup>7</sup> (Multimedia Educational Resource for Learning and Online Teaching) o CAREO<sup>8</sup> (Campus Alberta Repository of Educational Objects), no siempre describen los distintos OA existentes en la web, almacenando registros de metadatos asociados a los objetos descritos; pero con la implementación de un “Repositorio Semántico” (RS), los cuales básicamente consisten en contener un conjunto de ontologías de diversos dominios, entonces serán descritos los OA garantizando una búsqueda mucho más estructurada del conocimiento existente. No obstante, la búsqueda no es la única ventaja de la existencia de estos repositorios semánticos, pues permiten revisiones cooperativas de los OA, de modo que la calidad del contenido es revisada y cuestionada por los distintos participantes que acceden al repositorio a través de Internet; pero no solo los humanos podrán consultar y buscar información sino también agentes software o sistemas LMS (Learning Management Systems) externos. Un ejemplo de estos repositorios semánticos para OA es el llamado SLOR<sup>9</sup> (Semantic Learning Objects Repository).

La Universidad de las Ciencias Informáticas (UCI) como centro de estudio dedicado a la investigación, producción de software y servicios informáticos para la sociedad cubana y para el

---

<sup>7</sup> <http://www.merlot.org>

<sup>8</sup> <http://www.careo.org>

<sup>9</sup> Del proyecto en SourceForge: <http://slor.sourceforge.net>

mundo, posee un Polo de Teleformación que dentro de sus líneas de investigación está centrado en el estudio, desarrollo y soporte de herramientas para la Teleformación o e-Learning; y desde hace algunos años se ha venido estudiando esta línea y muestra de ello es la creación de herramientas y el desarrollo de trabajos de diploma relativos al e-Learning, entre los cuales podemos encontrar: Propuesta metodológica para la gestión de conocimiento basada en ontologías (Cabrera Pupo, y otros, 2007), Mecanismos Semiautomáticos y Automáticos para la Generación de Ontologías de Dominio (Sardiñas Suárez, y otros, 2008), entre otros.

El primer trabajo de los anteriormente mencionados (Cabrera Pupo, y otros, 2007) propone una metodología para la gestión del conocimiento basada en ontologías; posterior a este se desarrolló un segundo trabajo (Sardiñas Suárez, y otros, 2008), el cual propone una metodología para la creación de ontologías de forma automática y semi-automática.

Dentro de las herramientas implementadas en el Polo de Teleformación se encuentran: una herramienta de autor nombrada ROXS la cual permite diseñar OA siguiendo el estándar SCORM (Sharable Content Object Reference Model) y un ROA que permite el almacenamiento y búsqueda de OA, este último será puesto a disposición de la comunidad universitaria, lo cual presupone un aumento de OA, lo que pudiera traer consigo que las búsquedas se harían menos eficientes, debido a ello se quiere mejorar el sistema de búsquedas y el acceso a la información que hay en el mismo, y organizar los OA del repositorio de forma óptima, basándose en los metadatos que los describen.

Por todo lo anteriormente expuesto, se plantea como **problema científico**:

¿Cómo gestionar el conocimiento contenido en el Repositorio de Objetos de Aprendizaje de la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio** Ingeniería Ontológica y enmarcando como **campo de acción** la Inferencia Semántica de Información en Repositorios.

Es por ello que el **objetivo general** de este trabajo, es desarrollar un módulo de inferencia que se interconecte mediante un bus de servicios con el Repositorio de Objetos de Aprendizaje de la UCI.

Se plantean los siguientes **objetivos específicos**:

- Identificar las tecnologías que se implementan para el desarrollo de la WS, necesarias para la creación del módulo.
- Crear un módulo de inferencia para el ROA de la UCI.
- Interconectar el módulo de inferencia con el ROA de la UCI mediante un bus de servicios.

### Idea a defender

La creación de un módulo de inferencia para la gestión del conocimiento contenido en los Objetos de Aprendizaje almacenados en el Repositorio de Objetos de Aprendizaje de la UCI, garantizará una recuperación efectiva del conocimiento existente.

En correspondencia con el objetivo y la idea a defender se establecieron las siguientes **Tareas de Investigación**:

- Analizar el estado del arte de la Ingeniería Ontológica y las tecnologías incluidas en ésta.
- Analizar el estado del arte de los ROA y RS.
- Estudio de estándares para OA, enfocándose en SCORM.
- Crear Ontologías para el dominio e-Learning.
- Crear un conjunto de reglas para realizar inferencias en las ontologías previamente creadas.
- Realizar inferencias sobre Ontologías creadas.
- Generar servicios para el ROA de la UCI.
- Realizar pruebas a la ontología.

Para el desarrollo de la investigación es necesaria la combinación de los métodos teóricos y empíricos. Dentro de los teóricos el **Análisis y Síntesis** pues dicho método permite el análisis de teorías de la Ingeniería Ontológica y la documentación referente a la WS, OA, ROA y RS, llevando a cabo una síntesis de los elementos más importantes de lo antes analizado de forma tal que se

obtenga la convergencia entre dichos elementos para así dar paso a la solución de una parte del problema.

Se utiliza también el método de **Modelación** para la elaboración de los diagramas especificados por la metodología; además se hace necesario estudiar la evolución de varias tecnologías necesarias para la implementación de la WS utilizando el método **Histórico-Lógico**.

Dentro de los métodos empíricos se utiliza el de la **Observación** al probar diferentes herramientas, como los editores ontológicos y razonadores, para ayudar a decidir cuáles serían finalmente utilizados para el desarrollo del módulo.

El presente trabajo consta de una introducción, tres capítulos, conclusiones generales, referencias bibliográficas utilizadas durante el desarrollo del trabajo, anexos que complementan el cuerpo del trabajo y por último el glosario de términos, donde se encuentra la relación de los términos no comunes con su respectiva descripción.

**Capítulo 1 Fundamentación teórica:** Se tratan de forma general los principales conceptos que cimientan la investigación, entre los que se encuentran las tecnologías de la WS, los repositorios, los ROA y los RS; así como las ventajas y desventajas que traen consigo los diferentes tipos de repositorios y a qué conclusiones se arriba.

**Capítulo 2 Propuesta del Repositorio Semántico:** Se describe la propuesta de diseño del RS, enfocándose en el diseño de las ontologías y el módulo de inferencia.

**Capítulo 3 Prueba:** en este capítulo se realizan y analizan las pruebas al sistema.

## **Capítulo 1. Fundamentos Teóricos**

### **1.1 Introducción al Capítulo.**

En este capítulo se tratan de forma general los principales conceptos que cimientan la investigación, entre los que se encuentran las tecnologías de la WS, los repositorios, los ROA y los RS; así como las ventajas y desventajas que traen consigo los diferentes tipos de repositorios y a qué conclusiones se arriba.

### **1.2 La Web actual**

La web actualmente es utilizada a escala mundial por miles de personas e instituciones en disímiles actividades tales como búsqueda y publicación de información, para la comercialización o para realizar tareas que forman parte del quehacer diario, todo ello ha sido posible gracias al crecimiento que ha tenido la WWW, la misma surge en 1989 cuando Tim Berners-Lee su creador, presenta este proyecto<sup>10</sup> en el Instituto Europeo de Investigación de la Física de Partículas (CERN) en Ginebra, Suiza, aunque no fue hasta 1990 que recibe su nombre definitivo.

En sus inicios su objetivo era el de compartir los trabajos que realizaban los físicos de este instituto a través de un entorno de hipertexto<sup>11</sup> (Bianchini, 2000) (HTML) y basado en arquitectura cliente-servidor (HTTP) (SWWW, 1996). En 1993 con el lanzamiento de Mosaic, el primer navegador de dominio público, compatible con Unix, Windows, y Macintosh, por el National Center for Supercomputing Applications (NCSA), marca el momento en que la WWW se da a conocer al mundo, desplegándose primero en universidades y laboratorios, y en cuestión de meses al público en general, iniciando así lo que sería su vertiginoso crecimiento. Los usuarios acogieron con entusiasmo la facilidad con que se podían integrar texto y gráficos y saltar de un punto a otro del mundo en una misma interfaz y la extrema sencillez para aportar contenidos a una web mundial. En 1994 algunos de los miembros que desarrollaron Mosaic desarrollan Netscape, otro navegador

---

<sup>10</sup> [http://bvs.sld.cu/revistas/aci/vol13\\_6\\_05/aci030605.htm](http://bvs.sld.cu/revistas/aci/vol13_6_05/aci030605.htm)

<sup>11</sup> “El hipertexto es una tecnología que organiza una base de información en bloques distintos de contenidos, conectados a través de una serie de enlaces cuya activación o selección provoca la recuperación de información.”

con algunas mejoras que impulsan la programación en la web, en ese mismo año se constituye el consorcio W3C el cual es presidido por Tim Berners-Lee. Con la aparición de lenguajes como JavaScript y otros, las páginas web contienen programas enteros, permitiendo una mayor eficiencia, capacidad dinámica y capacidad de interacción entre las páginas. (Castells)

Según el grupo de trabajo del W3C (W3C XML Protocol Working Group) el uso actual de la Web se puede resumir en este texto:

Hoy, el principal uso de la web es para acceso interactivo a documentos y aplicaciones. En casi todos los casos, este acceso es hecho por usuarios humanos, que típicamente trabajan con navegadores web, reproductores de audio otros sistemas interactivos en el lado del usuario. La Web puede crecer significativamente en poder y alcance si se extiende para permitir comunicaciones entre aplicaciones, de un programa a otro.

Si bien la Web actual constituye un avance tecnológico impresionante, posee como carencia crucial la existencia de mucha información y la misma se encuentra desorganizada, ello provoca que a muchos usuarios se les dificulte la búsqueda de información, pues en muchas ocasiones suele suceder que obtienen como resultado páginas web o documentos que no guardan relación alguna con los términos introducidos. Por ejemplo si se introduce la palabra “Ketchup” para buscar información sobre el grupo de música del mismo nombre, obtendremos enlaces a restaurantes, recetas, fabricantes, distribuidores y clubes de aficionados al condimento, y finalmente lo que buscábamos (posiblemente ni siquiera esto si el grupo fuese menos popular). Si se busca un “artículo sobre García Márquez”, se encontrará decenas de artículos de García Márquez, pero ninguno que trate sobre este autor. Si se pregunta sobre estándares XML para la enseñanza (“XML education”), la mayor parte de los resultados se referirán a la enseñanza de XML.

Todos estos ejemplos son el síntoma de una causa común: la falta de capacidad de las representaciones en que se basa la web actual para expresar significados. Los contenidos y servicios en la web se presentan en formatos (HTML) e interfaces (formularios) comprensibles por personas, pero no por máquinas.

Como una posible solución a estos problemas se están realizando varios estudios para lograr crear la llamada WS o Web del futuro como la llamara en 1999 Tim Berners-Lee Lee, el creador de la web actual y el pensador de esta futura web.

### 1.3 La Web Semántica

Cuando se habla de WS el propio Tim Berners-Lee la ha definido como:

*“El primer paso es colocar los datos en la Web de un modo en que las máquinas puedan entenderlos naturalmente convertirlos a esa forma. Esto crea lo que yo llamo una Web Semántica: una red de datos que pueden ser procesados directa o indirectamente por máquinas.”* (BERNERS-LEE, y otros, 1999)

*“La Web semántica es una extensión de la actual web en la cual la información se da mediante un significado bien definido, lo que facilita que los ordenadores y la gente trabajen en cooperación.”* (BERNERS-LEE, y otros, 2001)

*“La Web Semántica proporcionará estructura al contenido importante de las páginas web, creando un entorno donde los agentes de software que viajan de página en página pueden enseguida llevar a cabo complicadas tareas para los usuarios. Así, un agente que venga de la página web de la clínica no sólo sabrá que la página tiene palabras como "tratamiento, medicina, médico, terapia", sino también que el Dr. Hartman trabaja en la clínica los lunes, miércoles y viernes, y que el script [programa no compilado realizado en un lenguaje de programación sencillo] toma un rango de datos en el formato año-mes-día y devuelve fechas y horas de consulta.”* (BERNERS-LEE, y otros, 2001)

La WS<sup>12</sup> (BERNERS-LEE, y otros, 2001) propone dar solución a las limitaciones de la web actual mediante la introducción de descripciones explícitas del significado, la estructura interna y la estructura global de los contenidos y servicios disponibles en la WWW. Frente a la semántica implícita, el crecimiento caótico de recursos, y la ausencia de una organización clara de la web

---

<sup>12</sup> <http://www.semanticweb.org/>, <http://www.ontology.org/>

actual, la WS aboga por clasificar, dotar de estructura y anotar los recursos con semántica explícita procesable por máquinas.

Dicha web mantiene los principios que han hecho un éxito de la web actual, como son los principios de descentralización, compartición, compatibilidad, máxima facilidad de acceso y contribución, o la apertura al crecimiento y uso no previstos de antemano.

Según Stojanovic *“la WS es la nueva generación de la Web mundial, basada en un formalismo de representación semántica del conocimiento de la red, que permite que la información esté empaquetada en forma de declaraciones del objeto-atributo-valor, denominadas triadas. Se asume que los términos usados en estas declaraciones están basados en ontologías formalmente especificadas (para la comunidad del interés), estas triadas se pueden procesar semánticamente por los agentes de manera automática.”* (Stojanovic, y otros, 2002)

La WS tiene una visión: La idea es tener datos en la red definidos y unidos de manera que puedan ser usados por computadoras, no solo con el propósito de visualización, sino para la automatización, integración y uso repetido de los datos en múltiples aplicaciones. Promete mejorar nuestra habilidad de descubrir, ordenar, y clasificar la información de manera radical en las tareas que consumen mucho tiempo, ya sea ON-Line u OFF Line. (PBCWS, 2001)

Luego de analizadas las definiciones de la WS brindadas por varias personalidades e instituciones se toma la dada por el precursor de la idea, Tim Berners-Lee, quien intentó desde el principio incluir información semántica en su creación, la World Wide Web, pero por diferentes causas no fue posible. Por ese motivo introdujo el concepto de semántica con la intención de recuperar dicha omisión.

El futuro de la tecnología de la web se basará en la capacidad de la red para servir no sólo a las personas sino también a la información de agentes al servicio de los humanos, gracias a la organización de la información que esta propone. Por lo que debido a esto y a la flexibilidad de las tecnologías en la que está soportada es que se usa en la creación de soluciones para la GC.



### 1.3.1 Gestión del conocimiento

El conocimiento, tal como se entiende hoy día, es un recurso que no tan sólo nos permite interpretar nuestro entorno, sino que nos da la posibilidad de actuar. Es un recurso que se halla en las personas y en los objetos –físicos o no– que estas personas utilizan, pero también en las organizaciones a las que pertenecen, en los procesos y en los contextos de dichas organizaciones. La gestión del conocimiento consiste en optimizar la utilización de este recurso mediante la creación de las condiciones necesarias para que los flujos de conocimiento circulen mejor. (Canals, 2003)

Según Figueroa La GC “es el proceso sistemático de buscar, organizar, filtrar y presentar la información con el objetivo de mejorar la comprensión de las personas en un área específica de interés”. (Figueroa, y otros, 2006)

Cuando se hace referencia a la GC no se está hablando de gestión de la información (GI). Ésta última tiene como objetivo básico organizar y poner en uso los recursos de información de la organización (tanto de origen externo como interno) para permitir operar, aprender y adaptarse a los cambios del ambiente. (Figueroa, y otros, 2006)

Los procesos principales de la gestión de información: la identificación de las necesidades de información, la adquisición de las fuentes informativas, su organización y almacenamiento, el desarrollo de productos y servicios, su distribución y uso, que son también la base de la creación del conocimiento durante la existencia productiva de la organización y, por tanto, fundamento de la fase inicial de la gestión del conocimiento. (Rojas Mesa, 2006)

Resulta importante aclarar que existen diferencias entre conocimiento, información y datos. El conocimiento es un paso adelante, es identificar, estructurar y sobre todo utilizar la información para obtener un resultado. Requiere aplicar la intuición y la sabiduría, propios de la persona a la información. Recopilar y organizar datos, es algo que puede hacer un software informático. Se le denomina información al conjunto de datos organizados y analizados en un contexto determinado.

Los datos son una representación simbólica<sup>13</sup> que por sí solo no tiene un valor semántico y constituyen la base de la pirámide del conocimiento. La capacidad de interpretar esos datos es lo que provoca que la información se convierta en conocimiento. (Figueroa, y otros, 2006)

La WS contribuye a que se logre una buena gestión del conocimiento pues al dotar a la web 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. Esta web extendida y basada en la descripción de la información, o sea metadatos.

En el Polo de Teleformación se han desarrollado dos tesis referentes al tema de la gestión del conocimiento haciendo uso de las ontologías, y estas plantean una metodología para la creación de ontologías de manera manual y la otra de forma automática y semi- automática. A continuación se muestra una tabla (Tabla 1) que compara los diferentes métodos propuestos en cada trabajo para la creación de ontologías, donde Trabajo 1 corresponde a “Propuesta metodológica para la gestión de conocimiento basada en ontologías” (Cabrera Pupo, y otros, 2007) y Trabajo 2 a “Mecanismos Semiautomáticos y Automáticos para la Generación de Ontologías de Dominio”. (Sardiñas Suárez, y otros, 2008)

El primer trabajo de los anteriormente mencionados propone una metodología para la gestión del conocimiento basada en ontologías, la cual consta las siguientes fases (Cabrera Pupo, y otros, 2007):

1. Proceso de Construcción de la Ontología: en esta fase se definen una serie de pasos fundamentales para la creación de ontologías.
  - Determinar el dominio de la ontología.
  - Buscar ontologías que guarden relación con la ontología a construir.
  - Definir una lista en la que estén incluidos los términos más formulados por el usuario y que estén relacionados con el dominio.

---

<sup>13</sup> Cuando se menciona una representación simbólica se refiere que pueden ser números, letras, etc.

- Estructurar la información que estará presente en las Ontologías mediante relaciones entre clases y subclases.
  - Definir las propiedades de las clases y sus relaciones.
  - Crear reglas de inferencia
2. Inferencia: en esta fase se hace uso de un motor de razonamiento para inferir conocimiento a través de las reglas de inferencia y la información contenida en la ontología.

Posterior a esta tesis se desarrolló un segundo trabajo (Sardiñas Suárez, y otros, 2008), el cual propone una metodología para la creación de ontologías de forma automática y semi-automática; dicha metodología consta de 2 fases fundamentales: {1} Fase de Etiquetado (en esta fase se propone el utilizar los analizadores lexicológicos), {2} Fase de Búsqueda de Conceptos (encontrar expresiones lingüísticas que representen conceptos).

Metodologías	Trabajo 1	Trabajo 2
<b>Crear Ontología de forma manual</b>	Si	Si
Determinar el dominio de la ontología.	Si	Si
Buscar ontologías que guarden relación con la ontología a construir.	Si	Si
Definir una lista en la que estén incluidos los términos más formulados por el usuario y que estén relacionados con el dominio.	Si	Si
Estructurar la información que estará presente en las Ontologías mediante relaciones entre clases y subclases.	Si (Owl)	Si(Owl)
Crear reglas de inferencia	Si (Swrl)	Si
<b>Crear Ontología de forma automática y semiautomática</b>	No	Si
Fase de Etiquetado	No	Si(analizadores lexicológicos)
Fase de Búsqueda de Conceptos	No	Si
Fase de Inferencia	No	Si (rdr y mcrdr)
<b>Inferencia</b>	Si	Si

**Tabla 1:** Comparativa entre las diferentes metodologías propuestas en cada trabajo para la creación de ontologías.

### 1.3.2 Metadatos

El término “metadatos” no tiene una definición única. Según la definición más difundida de metadatos es que son “datos sobre datos”. También hay muchas declaraciones como “informaciones sobre datos” (Sheldon, 2001), “datos sobre informaciones” (Steinacker, y otros, 2001) e “informaciones sobre informaciones” (W3C, 2002)

Otra clase de definiciones trata de precisar el término como “descripciones estructuradas y opcionales que están disponibles de forma pública para ayudar a localizar objetos” o “datos estructurados y codificadas que describen características de instancias conteniendo informaciones para ayudar a identificar, descubrir, valorar y administrar las instancias descritas”. Esta clase de definiciones hace mayor hincapié en los metadatos en relación con la recuperación de información. (Cid, y otros, 2007)

Los metadatos pueden describir colecciones de objetos y también los procesos en los que están involucrados, describiendo cada uno de los eventos, sus componentes y cada una de las restricciones que se les aplican. Los metadatos definen las relaciones entre los objetos, como las tuplas<sup>14</sup> en una base de datos o clases en orientación a objetos, generando estructuras. (Cid, y otros, 2007)

Los beneficios derivados de la utilización de metadatos son diversos, tales como (Cid, y otros, 2007):

- Añadir contenido, contexto y estructura a los objetos de información, asistiendo de esta forma al proceso de recuperación de conocimiento desde colecciones de objetos.

---

<sup>14</sup> En la teoría de bases de datos, una tupla se define como una función finita que mapea (asocia unívocamente) los nombres con algunos valores. Ejemplo: (jugador : "Luis", puntuación 25)

- Permiten el intercambio de la información sin la necesidad de que implique el intercambio de los propios recursos.
- En cada proceso productivo, o en cada etapa del ciclo de vida de un objeto de información, se van generando metadatos para describirlos y metadatos para describir dichos metadatos (manual o automáticamente). generando de esta forma valor añadido a los recursos
- Los metadatos permiten un acceso a los recursos en forma controlada ya que se conoce con precisión el objeto descrito.
- Los metadatos son esenciales para sostener un crecimiento de una Web a mayor escala, permitiendo búsquedas, integración y recuperación del conocimiento desde un mayor número de fuentes heterogéneas.

El esqueleto de la WS se compondrá de metadatos asociados a los documentos y archivos de la Web actual. Sin embargo, las verdaderas capacidades de la Web sólo se conseguirán aplicando lógica y confianza a los metadatos. La lógica permitirá extraer automáticamente conclusiones de la información almacenada y hacer búsquedas de información muy especializadas. La confianza hará que los usuarios de la Web obtengan respuestas fiables a sus consultas y que no desconfíen de los agentes inteligentes y de sus acciones (Abián, 2007). Todo ello es posible mediante el uso de las diferentes tecnologías que sustentan la WS.

### 1.3.3 Tecnologías que sustentan la Web Semántica.

La WS, como bien indica su nombre, se basa en dotar de semántica a la web ya existente y para eso se utilizan una serie de metalenguajes, como son: el XML, DAML-OIL<sup>15</sup>, etc. Pero como ya se ha explicado anteriormente, la WS también organiza el conocimiento existente en la web y para esto utiliza las bases de conocimientos (pues éstas organizan y le adicionan semántica al contenido) como lo son las ontologías.

---

<sup>15</sup> DAML-OIL (DARPA's Agent Markup Language + Ontology Inference Layer) se puede encontrar en [www.daml.org](http://www.daml.org).

Y con el uso de estos lenguajes y tecnologías se han creado una serie de herramientas que facilitan el trabajo para la implementación de la WS, y acerca de estos lenguajes y herramientas se abordará a continuación.

### **1.3.3.1 Lenguajes que sustentan la Web Semántica.**

#### ***1.3.3.1.1 Extensible Markup Language (XML) y XML Schema.***

XML, es el estándar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados. (Barceló, y otros, 2006)

XML fue creado para permitir la interoperabilidad. Puesto que todo el mundo puede crear sus propios vocabularios XML, se produciría una terrible confusión si diferentes desarrolladores optasen por los mismos nombres de elementos para representar diferentes entidades. Los "espacios de nombres" (NameSpaces) fueron introducidos en XML para resolver conflictos de nombres entre elementos en un documento XML cuando los elementos se derivan de diferentes fuentes, permitiendo el uso de múltiples vocabularios en un mismo documento. (Ortuño Díaz, 2003)

Un "schema XML" según Reino (Reino, 2000), es algo similar a un DTD (Document Type Definition), es decir, que define qué elementos puede contener un documento XML, cómo están organizados, y que atributos y de qué tipo pueden tener sus elementos.

La ventaja de los schemas con respecto a los DTDs es:

- Usan sintaxis de XML, al contrario que los DTDs.
- Permiten especificar los tipos de datos.
- Son extensibles.

Por ejemplo, un schema permite definir el tipo del contenido de un elemento o de un atributo, y especificar si debe ser un número entero, o una cadena de texto, o una fecha, etc. Los DTDs no nos permiten hacer estas cosas. (Reino, 2000)

Aunque XML se incluye aquí como un lenguaje de especificación de ontologías, en realidad no lo es, no está orientado a cumplir ese objetivo sino el intercambio de datos. Este lenguaje permite únicamente extraer datos, pero no su contenido semántico. Desde este punto de vista, XML presenta una serie de problemas por basar su entendimiento en la estructura de los documentos, ya que cuando un esquema XML cambia, por ejemplo, introduce nuevos elementos intermedios que pueden invalidar alguna consulta anterior que haya sido basada en la estructura.

Además, también falla en un objetivo bastante importante, no mencionado anteriormente, la escalabilidad (Samper, 2005):

- El orden en el cual los elementos aparecen en el documento XML es significativo y muchas veces necesario. Esto es altamente antinatural en el mundo de los metadatos. Da igual el orden con el cual se presenten los distintos atributos de una incidencia de tráfico.
- Además, mantener el orden correcto de los elementos de datos es costoso y difícil.

### **1.3.3.1.2 RDF y RDF-Schema.**

RDF (Resource Description Framework, marco de descripción de recursos) es un lenguaje para representar información sobre recursos de la Web (metadatos). Según el W3C, el objetivo de RDF radica en “especificar semántica para los datos basados en XML, de una manera interoperable y estandarizada”. Debido a su carácter general, también puede representar datos.

¿Por qué se recurre a RDF para describir recursos, y no a XML? El motivo estriba en que XML no es apropiado para incluir semántica, tal como se vio en el apartado anterior. El modelo de un dominio puede representarse con varias DTD o varios esquemas XML, y una DTD o un esquema

pueden corresponder a muchos modelos distintos. Por así decirlo, XML no está orientado a objetos. (Abián, 2005)

Si bien suele decirse que RDF es un lenguaje, lo cierto es que resulta más exacto describirlo como un modelo de datos para las instancias de metadatos o, por abreviar, como un modelo de metadatos. En RDF, la construcción básica es la tripleta (sujeto, propiedad, objeto). Toda tripleta RDF corresponde a una sentencia RDF: la parte que identifica a qué se refiere la sentencia es el sujeto; la parte que identifica la característica del sujeto a la que se refiere la sentencia es la propiedad o el predicado; la parte que identifica el valor de la propiedad es el objeto.

RDF no se asocia a ningún dominio en particular: se puede emplear en cualquier campo. Cada persona u organización puede definir su propia terminología o vocabulario mediante lo que se conoce como RDFS (RDF Schema, esquema RDF), que se define en función de sentencias RDF. Además, RDFS permite especificar las entidades a las que pueden aplicarse los atributos del vocabulario.

¿Resulta verdaderamente necesario definir un esquema para RDF? Sí: un esquema RDF permite comprobar si un conjunto de tripletas RDF (un conjunto de metadatos, en definitiva) resulta válido o no para ese esquema, RDFS permite controlar la validez de los valores y restringir las entidades a las cuales pueden aplicarse ciertas propiedades.

El modelo de metadatos que RDFS al igual que los lenguajes de programación orientada a objetos (Smalltalk, Java, C#) permite expresar clases e instancias.

En resumen, puede aceptarse que RDFS<sup>16</sup> proporciona:

- Clases.
- Jerarquías de clases.
- Propiedades.
- Jerarquías de propiedades.

---

<sup>16</sup> El vocabulario completo de RDFS se define en el URI <http://www.w3.org/2000/01/rdf-schema>.



- Restricciones sobre los dominios y los rangos.

Por eficacia computacional, RDF y RDFS tienen también sus semánticas expresadas mediante tripletas (sujeto, propiedad, objeto). En las semánticas se incluye un sistema de inferencia robusto y completo.

Habiendo llegado hasta aquí, puede surgir la pregunta de que si la WS no sería posible sólo con XML/XML-S y RDF/RDF-S: XML proporcionaría la interoperabilidad sintáctica; RDF/RDF-S, la interoperabilidad semántica. La respuesta es no. RDF/RDF-S tiene muchas carencias (Abián, 2005):

- No se pueden declarar restricciones de rango (`rdfs:range`) que sean válidas sólo para algunas clases. `Rdfs:range` define el rango de una propiedad para todas las clases. Así como por ejemplo RDF/RDF-S no permite expresar que los televisores sólo funcionan con corriente alterna, mientras que otros aparatos pueden funcionar con corriente alterna o continua.
- No se pueden representar algunas características de las propiedades. En concreto, no se puede declarar que una propiedad es transitiva (como `menorQue`), simétrica (como `tocaA`), inversa (como `raizCuadradaDe` y `CuadradoDe`) o única (como `AbueloMaternoDe`).
- No se puede reflejar que unas determinadas clases son disjuntas. Por ejemplo, la clase `Persona` tiene asociada dos clases disjuntas (`Hombre` y `Mujer`). En RDF/RDF-S, puede declararse que `Hombre` y `Mujer` son subclases de `Persona`, pero no que son disjuntas. En otras palabras: resulta imposible especificar que ninguna persona puede ser a la vez hombre y mujer).
- No permite expresar restricciones de cardinalidad. Por lo tanto, una propiedad no está restringida en cuanto al número de valores que puede tomar. Así, no se puede expresar que una cuenta bancaria no puede tener más de seis titulares o que un hijo siempre tiene un padre y una madre.
- Existen algunas expresiones cuya semántica no es estándar (es decir, no pueden expresarse mediante la lógica de primer orden). Esta característica es sumamente

indeseable, pues causa que haya sentencias indecibles (sentencias de las que, dado un sistema de axiomas o premisas, no se puede afirmar o negar nada).

Anteriormente se analizaron una serie de metalenguajes que poseen semántica, pero esta no es suficiente para dotar a la web de conocimiento, por lo que basándose en estos lenguajes se escribieron otros que otorgan sentido a la web, tales como los lenguajes que se utilizan para escribir ontologías.

### **1.3.3.1.3 Ontologías para la Web Semántica.**

Aunque el concepto de ontología ha estado presente desde hace mucho tiempo en la filosofía, recientemente se utiliza en Informática para definir vocabularios que las máquinas puedan entender y que sean especificados con la suficiente precisión como para permitir diferenciar términos y referenciarlos de manera precisa.

Dentro de los muchos conceptos relacionados con las ontologías se puede citar los siguientes:

*“Las Ontologías son para la Inteligencia Artificial recursos construidos que permiten representar el conocimiento compartido y común sobre algo” (Arano, 2005). En la rama de la Informática: “Una Ontología define los términos a utilizar para describir y representar un área de conocimiento. Las Ontologías son utilizadas por las personas, las bases de datos, y las aplicaciones que necesitan compartir un dominio de información (un dominio es simplemente un área de temática específica o un área de conocimiento, tales como medicina, fabricación de herramientas [...], etc.). Las Ontologías incluyen definiciones de conceptos básicos del dominio, y las relaciones entre ellos, que son útiles para los ordenadores”. (W3C, 2004)*

Las ontologías se convertirán en una pieza clave en la construcción de la WS, ya que permiten adicionarle semántica al contenido de la web. Se debe realizar un gran esfuerzo en su creación, en

proporcionar infraestructuras adecuadas para el desarrollo, gestión y mapeo de las mismas y en el apropiado control de la evolución.

La primera aproximación de una Ontología es el tesoro. Un tesoro<sup>17</sup> es *“un tipo de lenguaje documental que representa la estructuración conceptual de un determinado campo del conocimiento”* (Ver Anexo 1). *“El tesoro proporciona una organización semántica principalmente a través de la explicitación de las relaciones establecidas entre dichos conceptos y, eventualmente, a través de un significado restringido de los términos que los representan”*. (Arano, 2005)

Las Ontologías resultan muy útiles para facilitar el razonamiento automático, es decir, sin intervención humana, usando para ello reglas de inferencia. Un motor de razonamiento<sup>18</sup> (Bosch, 2000) puede usar los datos de las Ontologías para inferir conclusiones de ellos.

Los beneficios de utilizar Ontologías se pueden resumir de la siguiente forma (Agm, 2006):

- Proporcionan una forma de representar y compartir el conocimiento utilizando un vocabulario común.
- Permiten usar un formato de intercambio de conocimiento.
- Proporcionan un protocolo específico de comunicación.
- Permiten una reutilización del conocimiento.

Las ontologías pueden o no tener reglas, a las ontologías que no tienen reglas se les llaman ontologías ligeras. Al tipo más sencillo de ontologías ligeras se les llama “taxonomías”. Una taxonomía es un conjunto organizado de términos utilizado para organizar información y pensado principalmente para poder navegar por él. En otras palabras, una taxonomía simplemente exige que sus componentes estén organizados de manera que se puedan "recorrer". Su rasgo

---

<sup>17</sup> Un tesoro es un vocabulario controlado y estructurado formalmente, formado por términos que guardan entre sí relaciones semánticas y genéricas: de equivalencia, jerárquicas y asociativas. Se trata de un instrumento de control terminológico que permite convertir el lenguaje natural de los documentos en un lenguaje controlado, ya que representa, de manera unívoca, el contenido de estos, con el fin de servir tanto para la indización, como para la recuperación de los documentos. (Lamarca, 2008)

<sup>18</sup> Un sistema cognoscitivo representa una conjunto de conocimientos, ¿pero cómo operan ellos? Esto es lo que se denomina modo de razonamiento o motor de razonamiento u operatividad del sistema.

fundamental y definatorio es, por tanto, su finalidad de exploración. “Son usadas, por ejemplo, para la categorización y la organización de la información en sitios Web” (Centelles, 2005) y están formadas por una jerarquía de clases que representan los conceptos relevantes en el dominio, de forma que los conceptos más generales se encuentren en los niveles superiores de la jerarquía y los más particulares en los inferiores.

La decisión de utilizar o no reglas deben tomarse cuidadosamente, ya que las reglas nos proporcionan una gran capacidad expresiva pero también una mayor complejidad al razonar con la ontología, lo que puede ocasionar problemas de escalabilidad.

### Componentes de una Ontología

Los componentes presentes en las ontologías que servirán para representar el conocimiento de algún dominio son (Gruber, 2002):

**Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.

**Relaciones:** representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.

**Funciones:** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la Ontología. Por ejemplo, pueden aparecer funciones como categorizar-clase, asignar- fecha, etc.

**Instancias:** se utilizan para representar objetos determinados de un concepto.

**Axiomas:** son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la Ontología. Por ejemplo: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición C1, A es B", etc. (Lozano, 2001)

Las ontologías surgen como una especialización de las redes semánticas en donde los **conceptos** e **instancias** van a formar los nodos de esta red, mientras que las **relaciones** serán los arcos y los **axiomas** los pesos de los arcos o etiquetas definidas para las relaciones.

### **1.3.3.1.4 Lenguajes para escribir Ontologías.**

#### ***Simple HTML Ontology Extensions (SHOE).***

Fue el primer lenguaje de etiquetado para diseñar Ontologías en la Web. Este lenguaje nació antes de que se ideara la WS. Las Ontologías y las etiquetas se incrustaban en archivos HTML. Permite definir clases y reglas de inferencia, pero no negaciones o disyunciones. A su albur se desarrollaron muchos editores, buscadores, APIs, etc.; el proyecto fue abandonado a medida que se desarrollaron OIL y DAML; aunque también existe una socialización de este lenguaje en XML. (Marquez, 2008)

Según Samper es un “Lenguaje desarrollado para ampliar HTML incorporando semántica en los documentos en este formato para que las máquinas los procesasen. Consiste en definir una ontología que describa una clasificación válida de objetos y las relaciones existentes entre ellos, para de esta forma añadir información descriptiva. Las ontologías en SHOE son una jerarquía de clases junto un conjunto de relaciones entre ellas y otro conjunto de reglas de inferencia en forma de cláusulas de Horn, por lo que nos permite realizar afirmaciones sobre la información que las páginas Web poseen. Se trata de un lenguaje de especificación de ontologías que pretende potenciar y mejorar los motores de búsqueda de los buscadores de Internet.” (Samper, 2005)

#### ***DAML-OIL (DARPA's Agent Markup Language + Ontology Inference Layer).***

Nació fruto de la cooperación entre OIL y DARPA y unifica los lenguajes DAML y OIL. Se basa ya en estándares del W3C. El lenguaje DAML se desarrolló como una extensión del lenguaje XML y de RDF y para extender el nivel de expresividad de RDFS DAML- OIL hereda muchas de las características de OIL, pero se aleja del modelo basado en clases (frames) y potencia la lógica descriptiva. Es más potente que RDFS para expresar ontologías. En la última revisión del lenguaje

(DAML+OIL)<sup>19</sup> ofrece ya un rico conjunto de elementos con los cuales se pueden crear ontologías y marcar la información para que sea legible y comprensible por máquina. También funciona como formato de intercambio. Presenta algunas carencias debido a su complejidad conceptual y de uso, complejidad que se intentó solventar con el desarrollo de OWL. No obstante, se desarrollaron muchas aplicaciones que utilizan DAML-OIL y también existen herramientas para convertir DAML a OWL<sup>20</sup>. (Marquez, 2008)

Según Samper, “*DAML+OIL es un lenguaje que nos ofrece más expresiones sofisticadas para las descripciones de clasificaciones y propiedades de los recursos que las que ofrecía RDF y RDFS. DAML+OIL es la evolución de RDFS, en el que se redefinen muchas de sus descripciones y se añaden muchas otras para mejorar el lenguaje y aportar propiedades y mecanismos para que el lenguaje defina ontologías que después pueden ser empleadas por sistemas de razonamiento para poder inferir sobre la información.*” (Samper, 2005)

### ***OWL (Ontology Web Language) y Fuzzy OWL.***

Lenguaje de Ontologías para la Web es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la web. Se trata de una recomendación del W3C, y puede usarse para representar ontologías de forma explícita, es decir, permite definir el significado de términos en vocabularios y las relaciones entre aquellos términos. OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que éste. Diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos. OWL surge como una revisión al lenguaje DAML-OIL y es mucho más potente que este. Al igual que OIL, OWL se estructura en capas que difieren en la complejidad y puede ser adaptado a las necesidades de cada usuario, al nivel de expresividad que

---

<sup>19</sup> <http://www.daml.org/2001/03/daml+oil-index>

<sup>20</sup> <http://www.mindswap.org/2002/owl.shtml>

se precise y a los distintos tipos de aplicaciones existentes como motores de búsqueda, agentes, etc. (Marquez, 2008)

El OWL está diseñado para el uso de aplicaciones que necesitan procesar el contenido de la información en lugar de limitarse a presentar la información a los seres humanos. OWL facilita una mayor interoperabilidad de contenido Web entre las máquinas que no sea apoyada por XML, RDF y RDF-S proveyendo un vocabulario adicional solo con la semántica formal. OWL tiene tres sublenguajes (Samper, 2005): OWL Lite, OWL DL, y OWL Full:

- ❖ OWL-Lite: es utilizado por usuarios que necesitan principalmente una clasificación jerárquica y restricciones simples.
- ❖ OWL-FULL: dirigido a usuarios que quieren máxima expresividad y libertad de sintáctica de RDF sin garantías computacionales.
- ❖ OWL-DL: para usuarios que quieran la máxima expresividad conservando plena compatibilidad (garantizar que todas las soluciones sean compatibles) y resolución (todos los cálculos se resolverán en un tiempo finito).

La tabla 1 muestra una comparativa entre los lenguajes de ontologías basados en Web más actuales y empleados.

	XML DTD	XML SCHEMA	RDF(S)	DAML+OIL	RDF(S) 2002	OWL
Listas Limitadas				X	X	X
Restricciones cardinalidad	X	X		X		X
Expresiones de clases				X		X
Datatypes		X		X	X	X
Clases definidas				X		X
Enumeraciones	X	X		X		X
Equivalencias				X		X
Extensibilidad			X	X	X	X
Semántica formal				X	X	X
Herencia			X	X	X	X
Inferencia				X		X
Restricciones locales				X		X
Restricciones cualificadas				X		
Reificación			X	X	X	X

**Tabla 2:** Comparativa entre lenguajes de especificación de ontologías (Samper, 2005)

Se denomina lenguaje F-OWL o Fuzzy OWL a una extensión del lenguaje OWL empleada para definir ontologías con elementos de la lógica difusa. También proporciona una descripción de un lenguaje DL con características difusas. Así mismo contiene una extensión para el lenguaje de reglas SWRL. La idea es ampliar los lenguajes que forman la WS para permitir expresar el grado de verdad que tienen los elementos de las ontologías y los resultados de los razonamientos.

Debido a las ventajas que posee OWL (tales como son: poseer mayor poder expresivo, y soporte de una gran variedad de herramientas y Apis para el desarrollo de aplicaciones haciendo uso de este lenguaje) sobre el resto de los lenguajes de etiquetado, anteriormente analizados, se decide utilizar el mismo para la creación de ontologías.



### 1.3.3.5 Lenguajes de reglas.

Otro de los objetivos de la WS es la definición de un lenguaje de reglas semánticas. Las reglas en la WS pueden ser de utilidad a los razonadores como métodos para inferir sobre el conocimiento, obtener nueva información, y que además, puedan ser empleadas por la WS para especificar tareas.

Como lenguajes de reglas destacan dos, RuleML y SWRL. RuleML ha sido el primer lenguaje empleado para definir reglas en aplicaciones basadas en WS, aunque recientemente se ha propuesto el uso de SWRL (del cual se abordará más adelante).

#### ***RuleML (Rule Markup Language) y Fuzzy-RuleML.***

RuleML es una iniciativa de diferentes instituciones y grupos investigadores que tenían como objetivo definir un lenguaje de hechos y reglas semánticas que pudiese ser utilizado en los diferentes sistemas de inferencia comerciales de reglas. La primera especificación estable de este lenguaje es del 2001, y en ella se quiso dar una sintaxis para todas las formas de reglas existentes en la lógica, pero al ser demasiado extenso se decidió proporcionar al lenguaje una sintaxis que permitiese especificar reglas de implicación y hechos.

El lenguaje que se describió debía de poseer ciertas características para poder escribir las reglas, pero la principal fue la capacidad de expresarlas. Para ello, se siguió la notación siguiente:

Consecuente: Antecedente1 ^ Antecedente2 ^...^ Antecedentek

La sintaxis del lenguaje para definir reglas fue escrita primero en una DTD<sup>21</sup>, y a partir de esta, más tarde se desarrolló la misma sintaxis pero escrita en XML Schema<sup>22</sup>, gracias a que este ofrecía más facilidades que las DTD. (Samper, 2005)

---

<sup>21</sup> <http://www.ruleml.org/indtd0.85.html>

Según Sánchez, RuleML es un lenguaje XML para la definición de reglas y orientado a su aplicación en la WS. RuleML contiene aspectos de programación lógica, programación funcional y orientación a objetos. Los impulsores de RuleML han enviado una propuesta del lenguaje llamado SWRL (Semantic Web Rule Language) para su consideración como estándar del W3C. SWRL es una combinación de OWL-DL (subconjunto de OWL) y un subconjunto de RuleML. (Sánchez, y otros, 2005)

Se ha creado una extensión de RuleML con lógica difusa, llamado Fuzzy RuleML, donde los hechos sobre el mundo pueden incluir una especificación de un "grado" (de verdad, un valor entre 0 y 1) de la confianza con la que se puede afirmar una combinación de hechos. Por ejemplo, la siguiente regla borrosa afirma que ser saludable es más importante que ser rico para determinar si uno es feliz:

Rico (?P) 0,5 ^ Saludable (?P) 0,9 -> Feliz (?P)

Donde rico, saludable y feliz son los predicados unario, y 0,5 y 0,9 son los pesos de los átomos Ricos (?p) y saludables (?p), respectivamente. Los predicados rico, saludable y feliz son representados por predicados difusos, ya que el grado en que alguien es rico es a la vez subjetivo y varía de acuerdo con el nivel que uno es rico. La misma propiedad se sostiene para los predicados "Feliz" y "Saludable", donde no se puede asignar grados absolutos de los miembros de un individuo a tales predicados. Por lo tanto, sería mejor para alguien poder asignar un grado a cada individuo en particular que pertenezca a los predicados anteriores (los conceptos). (Stamou)

### ***SWRL (Semantic Web Rule Language) y F-SWRL (Fuzzy Semantic Web Rule Language).***

SWRL<sup>23</sup> posee las siguientes características principales (Samper, 2005):

- Basado en OWL DL, OWL Lite y RuleML.
- Incluye sintaxis para expresiones de "cláusulas de Horn"<sup>24</sup>. (Fernández, 2004)

---

<sup>22</sup> <http://www.ruleml.org/xsd/0.85/>

<sup>23</sup> <http://www.w3.org/Submission/SWRL/> (Fernández, 2004)

- Se aporta un modelo teórico-semántico para incluir reglas en ontologías OWL
- Aporta sintaxis para emplearse junto OWL-XML y OWL-RDF

La propuesta de SWRL es la de permitir definir reglas de Horn en una base de conocimiento OWL. Para ello extiende el conjunto de axiomas de OWL, es decir, añade los axiomas adecuados para que el lenguaje de especificación de ontologías OWL permita definir las reglas.

Las reglas propuestas para este lenguaje tienen forma de implicación, con un antecedente (body) y un consecuente (head), igual a las definidas en RuleML. El significado que se quiere dar con estas reglas es el siguiente: si las condiciones especificadas en el antecedente son ciertas, entonces las condiciones del consecuente también lo son.

SWRL se ha convertido en el lenguaje de reglas recomendado por la comunidad de WS. Por ejemplo en los lenguajes de descripción semántica de servicios, es utilizado para expresar las condiciones necesarias en la especificación de diferentes aspectos, pues SWRL hace que las ontologías creadas para OWL-Service<sup>25</sup> sean más potentes, ya que utiliza el poder expresivo de las reglas.

Actualmente se puede encontrar un editor de este lenguaje como plugin del editor de ontologías Protégé. (Samper, 2005)

La experiencia en la utilización de ontologías y las normas en las aplicaciones ha demostrado que en muchos casos sería mejor ampliar su capacidad de representación y razonamiento para hacer frente a los conocimientos vagos o imprecisos para lo cual se ha desarrollado el lenguaje F-SWRL, creado sobre las bases de F-OWL y F-RuleML. En esta extensión los hechos OWL (axiomas que relacionan a los individuos) pueden incluir una especificación del "grado" (la verdad un valor entre 0

---

<sup>24</sup> Una cláusula de Horn es una regla de inferencia lógica con una serie de premisas (cero, una o más), y un único consecuente. Las cláusulas de Horn son las instrucciones básicas del lenguaje de programación PROLOG, de paradigma declarativo.

<sup>25</sup> OWL-S OWL es un servicio Web basado en la ontología, que abastece de proveedores de servicios web con un conjunto básico de construcciones de lenguaje de marcas para describir las propiedades y capacidades de sus servicios en la web sin ambigüedades. (DAML, 2006)

y 1) de confianza con los que afirman que un individuo (o par de personas) es una instancia de una determinada clase (o propiedad), y los átomos SWRL puede incluir un "peso" (la verdad un valor entre 0 y 1) que representa la "importancia" del átomo en una regla. La sintaxis de la extiende la existente sintaxis abstracta para SWRL, mientras que la semántica se basa en la teoría de conjuntos difusos. (Stamou, 2005)

Para la escritura de reglas se decide hacer uso de SWRL, debido a las características antes expuestas y que existen varias herramientas para la edición de estas reglas.

### **1.3.3.6 Lenguajes de consulta.**

#### ***XQuery***

Según Barceló es un “*Lenguaje de Consulta XML (XML Query)*. Es un lenguaje que ofrece la posibilidad de realizar consultas en infinidad de tipos diferentes de documentos como son documentos estructurados, colecciones de documentos, bases de datos, estructuras DOM, catálogos, etc., para extraer datos en la Web.” (Barceló, y otros, 2006)

Gutiérrez plantea:

“*De manera rápida podemos definir XQuery con un símil en el que XQuery es a XML lo mismo que SQL es a las bases de datos relacionales.*” (Gutiérrez, y otros, 2005)

XQuery es un lenguaje de consulta diseñado para escribir consultas sobre colecciones de datos expresadas en XML. Abarca desde archivos XML hasta bases de datos relacionales con funciones de conversión de registros a XML. Su principal función es extraer información de un conjunto de datos organizados como un árbol n-ario de etiquetas XML. En este sentido XQuery es independiente del origen de los datos.

XQuery es un lenguaje funcional, lo que significa que, en vez de ejecutar una lista de comandos como un lenguaje procedimental clásico, cada consulta es una expresión que es evaluada y devuelve un resultado, al igual que en SQL. Diversas expresiones pueden combinarse de una manera muy flexible con otras expresiones para crear nuevas expresiones más complejas y de mayor potencia semántica.

XQuery está llamado a ser el futuro estándar de consultas sobre documentos XML. Actualmente, XQuery es un conjunto de borradores en el que trabaja el grupo W3C. Sin embargo, a pesar de no tener una redacción definitiva ya existen o están en proceso numerosas implementaciones de motores y herramientas que lo soportan. (Gutiérrez, y otros, 2005)

### ***SPARQL (Simple Protocol and RDF Query Language)***

SPARQL (pronunciado "sparkle") es un lenguaje de recuperación basado en RDF; su nombre es un acrónimo recursivo del inglés SPARQL Protocol and RDF Query Language. Se trata de una recomendación para crear un lenguaje de consulta dentro de la WS que está ya implementada en muchos lenguajes y bases de datos. Desde 2005 está en proceso de estandarización por el RDF Data Access Working Group (DAWG) del W3C; en abril del 2006 se anunció el paso de su especificación a Candidate Recommendation, aunque volvió al estado de "Working Draft"<sup>26</sup> en octubre de 2006.

Con SPARQL los desarrolladores y usuarios finales pueden representar y utilizar los resultados obtenidos en las búsquedas a través de una gran variedad de información como son datos personales, redes sociales y metadatos sobre recursos digitales como música e imágenes. Es de utilidad para la recuperación y organización de información. (Wiki3, 2007 )

Samper (Samper, 2005) se refiere a SPARQL como: *"un lenguaje de consultas para grafos RDF propuesto recientemente por W3C. Ofrece a los desarrolladores y usuarios finales un camino para*

---

<sup>26</sup> Proyecto de trabajo: <http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004/>

presentar y utilizar los resultados de búsquedas a través de una gran variedad de información como puede ser datos personales, redes sociales y metadatos sobre recursos digitales como música e imágenes. SPARQL también proporciona un camino de integración sobre recursos diferentes. Permite:

- *Extraer información en diversas formas, incluyendo URIs.*
- *Extraer subgrafos RDF.*
- *Construir nuevos grafos RDF basados en la información de los grafos consultados.”*

### ***RDQL (RDF Data Query Language).***

RDQL (RDF Data Query Language) fue desarrollado por Hewlett Packard (HP) para que fuese el lenguaje de consulta para RDF en los modelos de Jena<sup>27</sup>, con la idea de convertirse en un modelo de consulta orientado a datos por ser una aproximación más declarativa. Debido a esto, solo se pueden hacer consultas sobre la información que hay en el modelo, por lo que la inferencia o razonamiento no es posible.

RDQL deriva de SquishQL que es un lenguaje de consulta para RDF, que a la vez deriva de RDFDB, y son una clase de lenguajes para RDF, no para el RDF-S a no ser que esté explícitamente en el modelo que se maneje.

Como RDF provee una estructura de grafos, donde los nodos son recursos o literales, RDQL permite especificar el patrón de la tripleta <sujeito, predicado, objeto> que se quiera buscar en el grafo del modelo para poder recuperar cualquier parte de la tripleta, devolviendo todas las tripletas que cumplan el patrón que se le pase (muy parecido a como actúa RQL).

Además de la desventaja de no permitir realizar ninguna inferencia, la utilización de filtros para obtener resultados es muy limitada. Las ventajas son la sencillez de manejo, ya que solo hace falta tener claro que tripleta <sujeito, predicado, objeto> se quiere preguntar, y otra de las ventajas son la

---

<sup>27</sup> Será abordado más adelante.

facilidades de integración con Java, debido a que ha sido implementado por los mismos desarrolladores de Jena. (Samper, 2005)

### ***SeRQL (Sesame RDF Query Language).***

SeRQL (Sesame RDF Query Language, pronunciado como "circle") es un lenguaje de recuperación para RDF/RDF-S desarrollado por Aduna como parte del software Sesame. Combina características de otros lenguajes (principalmente RQL, RDQL, N-Triples y N3) y añade otras propias.

Se asocia al programa Sesame. Sesame es un framework para RDF de código libre, con soporte para inferencias en RDF Schema y recuperación mediante consultas. Originalmente desarrollado por Aduna<sup>28</sup>, ahora es mantenido en cooperación con NLnet Foundation, y un grupo de desarrolladores voluntarios.

El diseño de SeRQL tuvo el objetivo de unificar algunas de las ventajas que ofrecían otros lenguajes, como parte de un lenguaje de recuperación ligero pero potente. Algunas de las características más importantes de SeRQL para la recuperación y organización de la información son (Wiki3, 2007):

- Transformación de grafos.
- Soporte de RDF Schema.
- Soporte de los tipos de datos de XML Schema.
- Emparejado de caminos opcionales.

Según Samper:

*“SeRQL es un lenguaje de consulta desarrollado por los administradores de Sesame, combinando las características de RQL, RDQL, N-Triples y N3, añadiendo alguna más propia. Este lenguaje de consultas posee tres componentes importantes: URIs, literales y variables.*

---

<sup>28</sup> <http://www.aduna-software.com/home/overview.view>

*Una de las diferencias de este lenguaje respecto al resto es la de presentar dos formas diferentes de hacer la consulta, la devolución de tablas con los posibles valores que pueden tomar las variables en nuestra consulta (“Select”, común al resto de lenguajes), y devolviendo el resultado en la forma de subgrafo que nosotros le sugerimos, almacenando la información del resultado en un grafo RDF (son consultas conocidas como “Construct”).” (Samper, 2005)*

### **1.3.3.2 Herramientas de la Web Semántica**

#### **1.3.3.2.1 Razonadores.**

##### ***Pellet***

Pellet es un razonador de OWL-DL basado en Java. Puede ser utilizado conjuntamente con bibliotecas del API de Jena o del OWL. Mediante su uso es posible validar, comprobar la consistencia de ontologías, clasificar la taxonomía y contestar a un subconjunto de consultas RDQL (conocido como consultas a ABox en terminología del DL). Se trata de un razonador DL basado en los algoritmos tableaux desarrollados para DL expresiva. Soporta todas las construcciones del OWL DL incluyendo las relacionadas con los nominales, es decir, owl:oneOf y owl:hasValue. (30)

Entre las características más relevantes de Pellet se destaca que es capaz de [1] analizar y reparar ontologías, incorporando un conjunto de heurísticas para detectar y corregir automáticamente las ontologías OWL para asegurar que son OWL DL; [2] razonar con tipos de datos definidos en el estándar XML-S; [3] validar “especies” para garantizar que los individuos se clasifican de modo correcto en la taxonomía de conceptos de la ontología; y [4] razonamiento con múltiples ontologías. (Lama, y otros, 2008)



Se arriba a la conclusión de emplear este razonador debido a las características mencionadas, por ser de código abierto, por poder interpretar las reglas escritas en SWRL y estar escritas en java que es el lenguaje base para la implementación del módulo de inferencia.

### ***RACER (Renamed ABox and Concept Expression Reasoner).***

RACER fue desarrollado por Ralf Möller y Volker Haarslev en 1999, pero que ha sido renovado periódicamente hasta la fecha. Es un razonador diseñado para la WS. Permite la inferencia tanto en conceptos como en instancias, soporta ontologías escritas en RDF/RDF-S/DAML/OWL (apenas tiene restricciones con estos lenguajes) y posee un lenguaje de consulta sencillo para la inferencia de instancias. Puede ser utilizado tanto por OilEd como por Protégé tanto para comprobar la consistencia de la ontología, como para hacer consultas sobre el conocimiento. (Samper, 2005)

El sistema RACER es el razonador basado en lógica de descripciones más representativo. En la actualidad es uno de los sistemas más completo. No sólo implementa mecanismos de razonamiento sobre la Tbox<sup>29</sup>, sino que además proporciona un lenguaje de consultas, el nRQL (new Racer Query Language) que permite realizar consultas conjuntivas. Además, también implementa, en su última versión, mecanismos de razonamiento sobre la Abox<sup>30</sup>. Sin embargo, el sistema no ofrece persistencia y los mecanismos de razonamiento se implementan mediante complejas técnicas de reducción. Esto implica que ontologías con grandes cantidades de instancias no puedan ser cargadas. Además, cada vez que se inicia el sistema debemos cargar de nuevo la ontología y procesarla. Por otro lado, RACER es en la actualidad una herramienta comercial, y por tanto, el razonador PELLET está ganado poco a poco más popularidad, ya que es una herramienta de libre disposición. Pellet ofrece la misma funcionalidad que RACER, y presenta exactamente sus mismos problemas. (Roldán, y otros, 2007)

---

<sup>29</sup> TBox: contiene el conocimiento normativo (terminológico) en forma de una terminología y es construido a través de declaraciones que describen propiedades generales de conceptos (definiciones intencionales de conceptos y roles). En otras palabras, contienen las definiciones de los conceptos y roles. (Samper, 2005)

<sup>30</sup> ABox: contiene conocimiento extensional o factual (aserciones sobre objetos), específico para los individuales del dominio de discurso, o sea contienen las definiciones de los individuales (instancias). (Samper, 2005)

### ***Cerebra.***

Es un motor de inferencia desarrollado por Network Inference. Parecido al razonador de lógica descriptiva FaCT<sup>31</sup> (Samper, 2005), solo que en este caso si que aporta soporte para la inferencia sobre instancias y datatypes. Soporta tanto ontologías RDF, como DAML + OIL y OWL, y como lenguaje de consulta para razonar emplea XQuery. Tiene versiones para trabajar conjuntamente con los editores OilEd y Protégé. (Samper, 2005)

Los razonadores basados en lógica descriptiva deben de poseer la suficiente expresividad para poder permitir que el conjunto de constructores que forman parte de los lenguajes de ontologías sean soportados, y poder permitir ambos tipos de inferencia, tanto de clasificación como de chequeo de instancias. (Samper, 2005)

### ***1.3.3.2 Editores ontológicos.***

#### ***Protégé.***

Protégé es un editor de ontologías escrito en Java, gratuito y de código abierto. Tras él hay una gran comunidad de desarrolladores y de usuarios universitarios, empresariales y gubernamentales. Actualmente, Protégé permite trabajar con RDFS (las ontologías se pueden exportar a RDF-S) y dispone de una extensión para OWL. Su sencillez y su buena documentación lo hacen ideal para los neófitos en ontologías. (Abián, 2005)

Este editor puede exportar ontologías a una variedad de formatos, incluyendo RDF (S), OWL y XML-S, para esto cuenta con una de las dos principales plataforma de modelado: Protégé-OWL<sup>32</sup> y

---

<sup>31</sup> Fast Classification of Terminologies (FaCT): razonador empleado por defecto por el editor de ontologías OilEd para clasificar los conceptos en una jerarquía según las descripciones que tengan.

<sup>32</sup> <http://protege.stanford.edu/overview/protege-owl.html>

la otra es Protégé-Frames<sup>33</sup>, la cual se usa para el diseño gráfico mediante el uso de una serie de frames .

Este editor trae embebido al SPARQL, SWRL, y a los razonadores DIG y Pellet; aparte de que hay una gran variedad de plugings (tales como: Jambalaya,SWRL-Tab,Xml-Tab, entre otros<sup>34</sup>) que facilitan una gran cantidad de actividades, ofrece a los desarrolladores y usuarios finales un camino para presentar y utilizar los resultados de búsquedas a través de una gran variedad de datos, los cuales pueden ser personales, redes sociales y metadatos sobre recursos digitales como música e imágenes, siendo útil para recuperar y organizar la información.

Se emplea Protégé para la edición de ontologías a desarrollar, debido a todas sus utilidades antes expuestas y la existencia de una gran variedad de plugings que le permiten realizar trabajos colaborativos con otras herramientas.

### **Swoop.**

Swoop es una herramienta para la creación, edición y depuración de ontologías OWL. Fue producido en el laboratorio de la Universidad de Maryland, College Park, pero ahora es un proyecto de código abierto con los contribuyentes de todo el mundo. (Google code)

La mayoría de las herramientas de desarrollo de ontologías proporcionan un entorno integrado para crear y editar dichas ontologías, la verificación de errores e incoherencias (con un razonador), navegar por múltiples ontologías, y compartir y reutilizar los datos existentes mediante el establecimiento de asignaciones entre las diferentes entidades ontológicas. Sin embargo, su diseño de la interfaz de usuario (look & feel) y el estilo de uso están inspirados por los paradigmas tradicionales basados en los razonadores de conocimiento (Knowledge Reasoner).

Características de Swoop (W3Cmember, 2004):

---

<sup>33</sup><http://protege.stanford.edu/overview/protege-frames.html>

<sup>34</sup><http://protege.stanford.edu/overview/protege-owl.html>

Navegador web como look & feel: (ver capturas de pantalla por encima de la muestra)

- Una barra de direcciones donde la URI de la ontología (o la clase / la propiedad / persona) se puede introducir directamente para la carga; hipervínculo basado en la navegación a través de entidades ontológicas (la barra de direcciones URL cambia en consecuencia); historia de botones (Atrás, Siguiente, etc) para la transversal; y marcadores que se pueden guardar para su posterior referencia.

Edición en línea:

Todas las ontologías editadas en Swoop se realizan en línea con el HTML Renderer, utilizando diferentes códigos de colores y estilos de fuente para destacar los cambios en la ontología, por ejemplo diferentes representaciones de axiomas añadidos vs axiomas suprimidos vs axiomas inferidos. Deshacer/rehacer opciones disponen de una ontología de registro de cambio y una opción de retroceso.

Diseñado para Media OWL.

- Especies de validación
  - Utilización Manchester OWL API
- Presentación sintaxis pestañas
  - Ver Formato conciso
  - Resumen de sintaxis (de OWL API)
  - RDF / XML (de HyperDAML BBN)
  - Turtle/N3
  - (plug-in de la arquitectura para las nuevas pestañas)
- Soporte de múltiples Ontologías
  - La navegación, mapeo, Comparación

Funciones avanzadas:

- Ejecutar un "buen y completo" conjunto de consultas ABox (escritas en RDQL) en una ontología utilizando Pellet.

- Partición de Ontologías automáticamente para su transformación en una E-relación.
- El modo de depuración: Al exponer el interior de un flujo de trabajo descripción de los razonadores lógicos de Tableaux (pellets) de un modo comprensible y legible, las explicaciones se ofrecen para ayudar a los usuarios a comprender la causa (y quitar) incoherencias detectadas en ontologías.
- Anotación Colaborativa de Apoyo: SWOOP tiene un Annotea plugin que permite a los usuarios escribir y compartir anotaciones sobre cualquier entidad ontológica (clase / propiedad / persona). Adicionalmente, los cambios que establece la ontología se pueden unir a la anotación para el intercambio de mensajes.

### **WebOnto.**

El proyecto WebOnto se inició en 1997 con el objetivo de desarrollar una web basada en un editor de ontología, proporcionando una navegación más sofisticada, una visualización y edición de las instalaciones mejor que las disponibles a partir de las soluciones existentes, como el servidor Ontolingua.

WebOnto fue diseñado para apoyar la navegación colaborativa, la creación y edición de ontologías, sin sufrir los problemas relacionados con los editores estándar basados en HTML, tales como apoyo limitado para la manipulación directa de interfaces, la incapacidad para manejar la comunicación asincrónica (por ejemplo, incapacidad para enviar avisos recordando al usuario que guarde los datos), y un apoyo limitado para interfaces gráficas.

Para evitar estos problemas WebOnto se implementa como un applet de Java, y ambos incluyen una interfaz gráfica de usuario, para la navegación de grano grueso, de grano fino y de inspector de las ventanas, hasta el inicio en detalle sobre los datos. WebOnto también proporciona un rico conjunto de opciones para personalizar la presentación de la información, lo que lo hace fácil la

navegación por grandes ontologías. El lenguaje OCML<sup>35</sup> proporciona la representación subrayada para modelos WebOnto.

Por último WebOnto también proporciona una API de cliente, que puede utilizarse para recuperar información de las ontologías WebOnto y para la ejecución de aplicaciones construidas usando WebOnto. (Domingo, y otros, 1997)

Según Samper la WebOnto<sup>36</sup> fue desarrollada por el Knowledge Media Institute (KMI) en 1997 como editor de ontologías OCML, cuya característica principal es que permite la participación de varios usuarios en el desarrollo de la ontología. (Samper, 2005)

### ***OntoEdit.***

OntoEdit es una herramienta para el desarrollo y mantenimiento de ontologías utilizando medios gráficos. OntoEdit construye las ontologías encima de una poderosa ontología modelo, interna de la herramienta. El modelo interno de la ontología se puede serializar usando XML, que apoya el manejo de archivos internos.

Este paradigma apoya el modelado de representación-lenguaje neutral en la medida de lo posible de los conceptos, relaciones y axiomas. Contiene varias vistas gráficas de la estructura del modelado de la ontología como apoyo a las diferentes fases del ciclo de ingeniería de la ontología.

OntoEdit v0.6 se apoya en F-Logic, RDF-Schema y OIL. En la versión actual OntoEdit posee una interfaz para el Karlsruher Inferencia Lógica F-Engine (la columna vertebral de OntoBroker), en la próxima versión FaCT del sistema será accesible desde OntoEdit. Además, las ontologías pueden ser exportadas a la base de datos de objeto-relacional esquemas y DTD. Se puede realizar un mapeo de los conceptos a las palabras (por ejemplo, para la conexión a un sistema de procesamiento de lenguaje natural) se define mediante el dominio de vista léxico OntoEdit. (AIFB)

---

<sup>35</sup> Operational Conceptual Modeling Language(OCML)

<sup>36</sup> <http://webonto.open.ac.uk/webonto/>

### ***OilEd.***

OilEd es un sencillo editor de ontología desarrollado por Sean Bechhofer en la Universidad de Manchester. OilEd permite al usuario:

- construir ontologías;
- utilizar el razonador FaCT para comprobar la consistencia de las ontologías y añadir de forma implícita una subClassOf;
- exportar ontologías en una serie de formatos, incluyendo ambos recursos OIL-RDF y DAML-RDF.

La intención detrás de OilEd es proporcionar una forma simple, editor freeware que demuestra su uso, y estimula el interés en OIL. OilEd no está destinado a ser un completo desarrollador - no va a apoyar activamente el desarrollo de ontologías a gran escala-, la migración y la integración de ontologías, las versiones, argumentos y muchas otras actividades que se dedican a la construcción de la ontología. Por el contrario, es el "Bloc de notas" de los editores de ontologías, ofreciendo sólo lo suficiente funcionalidad para permitir a los usuarios construir ontologías y para demostrar cómo el razonador FaCT se puede utilizar para revisar y enriquecer las ontologías. (Horrocks, 2000)

OilEd es la herramienta bautizada como el "notepad" de los editores de ontologías. Basado inicialmente para el desarrollo de ontologías OIL y DAML + OIL se han ido realizando numerosas actualizaciones para que acepte la mayoría de los lenguajes de especificación actuales. Es un editor bastante utilizado por los investigadores porque aporta la posibilidad de interactuar con un razonador como FaCT o RACER que permiten comprobar la consistencia de una ontología. Una de las desventajas que presenta este editor es la carencia de recursos para soportar ontologías grandes, migración e integración de otras ontologías y diferenciación de versiones. (Samper, 2005)

### ***1.3.3.2.3 Herramientas para el trabajo complementario.***

**eXist (Open Source Native XML Database):** eXist-db es una base de datos de código abierto del sistema de gestión totalmente basado en tecnología XML. Almacena los datos XML de acuerdo con el modelo de datos XML y características eficientes, basado en el índice de procesamiento de XQuery.

eXist-db se apoya en muchos estándares de tecnología(web) por lo que es una excelente plataforma de aplicación:

- \* XQuery 1.0 / XPath 2.0.

- \* XSLT 1.0 (usando Apache Xalan) o XSLT 2.0 (opcional usando Saxon).

- \* HTTP interfaces REST, WebDAV, SOAP, XMLRPC, Atom Editorial Protocolo.

- \* Base de datos XML específico: XMLDB, XQJ/JSR-225 (en desarrollo), XUpdate, XQuery actualización de las extensiones (que debe armonizarse con el nuevo mecanismo de Actualización de XQuery 1.0 eXist-db es altamente compatible con el estándar XQuery (actual XQTS resultado es 99,4%). El motor de búsqueda es altamente extensible y cuenta con una gran colección de módulos de función XQuery.

eXist-db proporciona un potente entorno para el desarrollo de aplicaciones web basadas en XQuery y estándares relacionados. Todas las aplicaciones web pueden ser escritas en XQuery, utilizando XSLT, XHTML, CSS y Javascript (AJAX para la funcionalidad). Las páginas del servidor XQuery se pueden ejecutar desde el archivo del sistema o almacenarlas en la base de datos. EXist-db es liberado al público bajo los términos de la licencia GNU LGPL. (Meier, 2000)

Se hace uso de esta base de datos debido a las facilidades antes mencionadas y que esta escrita en Java, lenguaje base del modulo, y esta basada en XML, lo que permite el almacenamiento de las ontologías.

**Jena:** Jena 2 es colección de herramientas desarrollado por Hewlett Packard para la WS. En esta colección, hay un analizador sintáctico para RDF, un API, e lenguaje de consulta RDQL, soporte para ontologías RDFS, DAML + OIL y OWL y un sistema de almacenamiento basado en bases de datos BerkeleyDB. (Samper, 2005)



**Sesame:** es un repositorio para RDF-S desarrollado por AIdministrador Nederland bv. Tiene funciones para añadir y eliminar información escrita en RDF en los repositorios, para ser almacenada en cualquier tipo de base de datos (MySQL, Oracle etc.). Soporta los lenguajes de consulta RQL, RDQL y SeRQL, para acceder a conocimiento. (Samper, 2005)

**KAON Tool:** desarrollado por la infraestructura KAON (Karlsruhe Ontology) Semantic Web, implementa un interfaz independiente del sistema en el que se almacenarán las ontologías, ya sean cualquier base de datos o un fichero texto. Implementa un API para leer las descripciones de los recursos, emplea RQL para realizar consultas y soporta tanto ontologías DAML + OIL como RDF. (Samper, 2005)

Según la FZI WIM<sup>37</sup> y AIFB LS3<sup>38</sup>, KAON es una infraestructura de gestión de ontologías de código abierto, específica para las aplicaciones empresariales. Incluye una colección de herramientas comprensivas que facilitan la creación de ontologías y la gestión y proporciona un marco para la creación de aplicaciones basadas en ontologías. Un enfoque importante de Kaon es que es escalable y eficiente razonando con ontologías.

Información básica (FZI WIM and AIFB LS3, 2005 ):

Desarrollado por: El Instituto de Karlsruhe<sup>39</sup> y AIFB<sup>40</sup>

Versión actual: 1.2.7

Plataforma: Java J2SE 1.4.0\_01<sup>41</sup>

Opcional: JBoss 3.0 <sup>42</sup> y Apache Tomcat 4.0.3<sup>43</sup>

Base de datos de soporte: Cualquier SQL2 compatible, MS SQL Server, PostgreSQL, IBM DB2, Oracle.

---

<sup>37</sup> <http://www.fzi.de/wim>

<sup>38</sup> <http://www.aifb.uni-karlsruhe.de/>

<sup>39</sup> <http://www.fzi.de/>

<sup>40</sup> <http://www.aifb.uni-karlsruhe.de/WBS/>

<sup>41</sup> <http://java.sun.com/j2se/1.4/index.html>

<sup>42</sup> <http://www.jboss.org/>

<sup>43</sup> <http://jakarta.apache.org/tomcat/>

Lenguaje de Ontología: Basado en RDF(S)<sup>44</sup> con extensiones para características algebraicas (simétrica, transitiva y inversa), cardinalidad, modularización, meta-modelado, la representación explícita de la información léxica.

Licencia: GNU Lesser General Public Licence (LGPL)<sup>45</sup>.

**KAON2:** es una infraestructura para la gestión de OWL-DL, SWRL, y ontologías F-Logic. Fue producido por el esfuerzo conjunto de las siguientes instituciones:

- Ingeniería de Procesos de Información (IPE)<sup>46</sup> en el Centro de Investigación en Tecnologías de la Información (Instituto de Karlsruhe) <sup>47</sup>
- Instituto de Informática Aplicada y Método Descriptivo Formal (AIFB)<sup>48</sup> de la Universidad de Karlsruhe <sup>49</sup>
- Grupo de Gestión de la Información (BIM)<sup>50</sup> en la Universidad de Manchester<sup>51</sup>

KAON2 es un sucesor del proyecto KAON (a menudo denominado KAON1). La principal diferencia a KAON1 es el lenguaje de ontologías en el que se apoya: KAON1 utilizó una extensión propietaria de RDFS, mientras que KAON2 es basado en OWL-DL y F-Logic. Es necesario tener en cuenta que KAON2 es un sistema completamente nuevo, y no es compatible hacia atrás con KAON1.

KAON2 proporciona las siguientes características (Motik):

- Un API para la gestión de OWL-DL, SWRL, y ontologías F-Logic,
- Un servidor independiente que da acceso a las ontologías de forma distribuida utilizando RMI<sup>52</sup>,
- Un motor de inferencia para responder a las preguntas conjuntivas (expresado mediante sintaxis SPARQL),

---

<sup>44</sup> <http://www.w3.org/TR/rdf-schema/>

<sup>45</sup> <http://www.gnu.org/copyleft/lesser.txt>

<sup>46</sup> <http://www.fzi.de/ipe/eng/index.html>

<sup>47</sup> <http://www.fzi.de/eng/index.php>

<sup>48</sup> <http://www.aifb.uni-karlsruhe.de/english>

<sup>49</sup> [http://www.uni-karlsruhe.de/index\\_en.html](http://www.uni-karlsruhe.de/index_en.html)

<sup>50</sup> <http://img.cs.manchester.ac.uk/>

<sup>51</sup> <http://www.manchester.ac.uk/>

<sup>52</sup> RMI(Java Remote Method Invocation) <http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>

- Una interfaz DIG<sup>53</sup>, lo que permite el acceso desde herramientas como Protégé,
- Un módulo para la extracción de instancias de ontología de las bases de datos relacionales.

### **JAVA54, IDE NetBeans 6.5<sup>55</sup>:**

Java es una plataforma de desarrollo orientada a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Es un lenguaje multiplataforma, de código abierto que consta con una gran cantidad de Apis y frameworks para agilizar el trabajo con la misma.

En los últimos años la Sun ha apoyado al desarrollo de la plataforma Java utilizando el IDE NetBeans 6.5 debido a que éste es de código abierto, pues se distribuye bajo licencia GPL e implementa las plataformas de desarrollo: Java ME, Java SE y Java EE. Se ha seleccionado este IDE para el desarrollo de la herramienta debido a que este cuenta con una gran cantidad de pluggins que apoyan el desarrollo usando las tecnologías de la WS.

### **1.3.4 Buscadores Semánticos**

Los buscadores semánticos son *“aquellos que realizan el rastreo atendiendo al significado del grupo de palabras que se escriben (Web Semántica “WS”) y no basándose en las actuales etiquetas. En pocas palabras, un buscador inteligente.”* (Javier)

Al incorporarle semántica a la Web los resultados de la búsqueda serían exactos, utilizando un buscador semántico. Estos resultados ofrecerán al usuario la información exacta que estaba buscando. La ubicación geográfica desde la que el usuario envía su pregunta es detectada de

---

<sup>53</sup> <http://dig.sourceforge.net/>

<sup>54</sup> <http://java.sun.com>

<sup>55</sup> <http://netbeans.org>

forma automática sin necesidad de especificar el punto de partida y el resultado final sería la obtención de forma rápida y sencilla de la información requerida. (Ver Anexo 2 y 3)

La forma en la que se procesará esta información no sólo será en términos de entrada y salida de parámetros sino en términos de su **SEMÁNTICA**. La WS como infraestructura basada en metadatos aporta un camino para razonar en la Web, extendiendo así sus capacidades. (W3C)

Los buscadores semánticos mejoran la capacidad de comprensión de las preguntas formuladas por el usuario mediante sistemas PLN (Procesamiento de Lenguaje Natural) y proporcionan enlaces de calidad, ya que, son capaces de aplicar lógica descriptiva, es por esto que se debe utilizar OWL y no RDF. (Criado, 2008)

A diferencia de buscadores tradicionales los buscadores semánticos devuelven solamente la información solicitada (Castells, y otros, 2003). Se hace necesario que los sistemas informáticos vayan más allá de una simple lectura de las letras que componen las palabras y que intenten en cierto modo descifrarlas, factor que puede ser aprovechado para la gestión del conocimiento. (Pagola, y otros, 1999)

En cambio los buscadores actuales presentan determinados problemas a la hora de mostrar sus resultados debido a que se buscan las palabras claves en los documentos existentes en la web sin atender a la sinonimia o la polisemia, esto representa una molestia para el cliente que en muchas ocasiones tendrá que consultar decenas de páginas para encontrar la información deseada y en algunos casos puede que no encuentre nada de lo que buscaba. (Abián, 2005)

La polisemia se puede apreciar cuando “realizamos una consulta en el buscador introduciendo un término y encontramos páginas web que contienen tal término pero con un significado distinto de aquello que nos interesa”, la sinonimia cuando “realizamos una consulta en el buscador introduciendo un término y no encontramos páginas web relacionadas con aquello que nos interesa porque tales páginas web contienen un sinónimo del término (pero no el propio término)” y la última barrera de la Web es el idioma. A esta se le conoce como Multilingüismo y se puede

apreciar siempre que “realizamos una consulta en el buscador introduciendo términos en inglés y nos encontramos páginas Web relacionadas con lo que nos interesa pero escritas en español (o cualquier otro lenguaje)”. (Sánchez, y otros, 2005)

Todos estos problemas serán resueltos con la WS, y por consecuencia con la utilización de buscadores Semánticos entre otras tecnologías que trae consigo dicha web.

### 1.4 E-Learning y Repositorios

La constante capacitación es un requerimiento de las sociedades actuales. Con todas las presiones del día a día, que los diferentes usuarios utilicen Internet para capacitarse, no es una moda, sino una herramienta que facilitará esta constante actualización.

Desde la revolución industrial, la educación a distancia surge para facilitar la especialización del personal. Y así con el avanzar de los tiempos, la tecnología siempre ha presentado diferentes opciones para compartir información, desde material impreso, radio-conferencias, teleconferencias, videos e incluso CD-ROMs apoyados por correspondencia postal y llamadas telefónicas. Con Internet se ha dado un nuevo paso en el tema de la educación a distancia convirtiéndola en una experiencia virtual. E-Learning es la forma de designar a este tipo de educación que se brinda a través de Internet. (Henst, 2002)

De forma general RedTTent define el e-Learning como *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes.”* (Red TTnet, 2005)

Entre las principales ventajas que ofrece la educación virtual se encuentran (Henst, 2002):

- La reducción de costos para dar cursos a más número de participantes que lo tradicional en un aula de clases.
- Ahorro en seminarios y capacitación de empresas muy descentralizadas como los bancos y la flexibilidad de horarios, factor de suma importancia pues permite al estudiante calendarizar el curso de la mejor forma posible.

- La interacción que generan los cursos, despertando el interés del estudiante y ayudando a aquellos tímidos a ser de los más activos en clases por medio de foros de discusión y otros medios de participación.

Una de las herramientas en las que se apoya el E-Learning es en los llamados “repositorios o repositorios digitales”.

Un repositorio es un lugar donde se almacena información, la cual puede o no estar estructurada. Cuando un usuario busca información en un repositorio, describe de manera breve el recurso que quiere obtener como respuesta. Los buscadores son la herramienta primaria para realizar búsquedas en repositorios. (Cabrera Pupo, y otros, 2007)

Según la definición de Sarasa *“un repositorio digital es un almacén de recursos digitales a los que se puede acceder sin que sea necesario un conocimiento previo de la organización o la estructura de dicho almacén. Además de los componentes recopilados, se contempla un almacenamiento de metadatos que aporten información sobre dichos componentes y que son el elemento principal que permiten la recuperación de los objetos. Existen diferentes especificaciones para la creación de repositorios digitales de contenidos y la interoperabilidad entre ellos.”* (Sarasa)

Mas cuando se habla del almacenamiento de recursos digitales, con carácter educativo, que son utilizados por el e-Learning, hay que aclarar que estos recursos son conocidos como OA y los repositorios donde se almacenan estos se les llama ROA, temas que serán tratados a continuación.

### **1.4.1 Objetos de Aprendizaje (OA)**

Un OA, abstrayendo las distintas definiciones e implementaciones que existen, se puede conceptualizar como una entidad digital que consta de tres elementos: unos contenidos o recursos, unas descripciones del comportamiento del objeto, y un conjunto de metadatos que hacen referencia al objeto. Los OA sirven para crear recursos educativos digitales de una forma simple, mediante el ensamblaje de recursos ya

existentes localizados en el propio sistema o en otros sistemas interconectados mediante una red, o bien contruidos para plataformas heterogéneas. (Sarasa)

López lo define como *“cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-Learning”*. (López, 2005) El presente trabajo se basa en la esta definición.

### 1.4.2 Repositorios de OA

El e-Learning necesita un sistema que le facilite la recopilación, el acceso y el compartir recursos educativos, en el que, apegándose a las necesidades específicas del sector, se tenga un sistema de almacenamiento de contenidos que se integre y comunique fácilmente con los otros sistemas que operan en los ambientes de aprendizaje en línea y es aquí en donde los Repositorios de Objetos de Aprendizaje tienen su origen.

Se ha insistido en que la reutilización debe ser una de sus principales bondades de ROA, es entonces casi intrínseco que se tenga un lugar destinado para su almacenamiento y clasificación para facilitar posteriormente su mantenimiento, localización y, posiblemente, también compartir estos OA con otros sistemas para aplicaciones diversas. Este almacén es lo que se conoce como Repositorio de Objetos de Aprendizaje. Los sistemas de repositorios son la infraestructura clave para el desarrollo, almacenamiento, administración, localización y recuperación de todo tipo de contenido digital. (ADL, 2002)

Tratando el tema de los repositorios vistos como almacenes de OA y extrapolados al contexto educativo, el programa (CANARIE, 2001) dice que los ROA *“son un catálogo electrónico/digital que facilita las búsquedas en Internet de objetos digitales para el aprendizaje”*.

Según Daniel, a partir de los términos *“repositorio digital”, “objeto de aprendizaje”* y *“metadato”* dice que *“los repositorios de objetos de aprendizaje son bases de datos con búsquedas que alojan recursos digitales y/o metadatos que pueden ser utilizados para el aprendizaje mediado”*. (Daniel, 2004)

El JORUM+ Project (JORUM+ Project, 2004) adopta la siguiente definición: “Un ROA es una colección de OA que tienen información (metadatos) detallada que es accesible vía Internet. Además de alojar los OA los ROA pueden almacenar las ubicaciones de aquellos objetos almacenados en otros sitios, tanto en línea como en ubicaciones locales”. (López, 2005)

El principal problema de estos repositorios es el de carecer de un modelo conceptual que establezca qué es un OA y qué descriptores de metadatos hay asociados a cada una de las diferentes conceptualizaciones. Sin acuerdo universal sobre el modelo de metadatos a utilizar, ni certeza sobre la compleción de los mismos por parte del creador del registro de metadatos, surge una grave carencia que dificulta la automatización en estos repositorios. Actualmente, la calidad de los registros de metadatos depende, entre otros, de los siguientes factores:

- La información proporcionada en los metadatos depende de la bondad del creador del registro y del tiempo necesario para añadir dicha información.
- Las capacidades de edición o herramientas proporcionadas por el repositorio.
- El nivel de conocimiento del creador del registro sobre los estándares de metadatos de OA.
- El modelo conceptual del repositorio: qué entiende el creador del registro que es un objeto de aprendizaje, y qué estructura de información de metadatos debe tener.

Al carecer estos metadatos de la calidad requerida se dificultan varios procesos, tales como el almacenamiento, la búsqueda y la recuperación desde repositorios distribuidos, así como la composición de nuevos materiales didácticos (entre otros).

Estos problemas se verían resueltos con la implementación de los llamados RS.

### **1.4.3 Repositorio Semántico**

Los RS son aquellos que utilizan tecnologías de la WS para optimizar varios procesos, que se realizan de forma ineficiente en los repositorios tradicionales, tales como la recuperación/búsqueda, la organización y el almacenamiento de la información contenida en los mismos, y para esto se utilizan esencialmente las ontologías, pues se pueden crear varias ontologías de diversos dominios para la organización de la



información por áreas, y los razonadores, para mejorar los métodos de recuperación de la información mediante las inferencias que estos permiten realizar.

Las ontologías podrían constituir un almacén (repositorio) de vocabulario para resolver los problemas de léxico, semánticos, de sinónimos así como los de las suposiciones efectuadas.

Resumiendo, la utilización extendida y estandarizada de ontologías resolvería muchos problemas. Existiría un entendimiento compartido que unificaría diferentes puntos de vista y que sería útil para:

- Compartir conocimiento común sobre la estructura de la información, bien entre personas bien entre agentes software.
- Permitir la reutilización del conocimiento sobre el dominio.
- Hacer explícitas las suposiciones sobre el dominio.
- Separar el conocimiento operacional del conocimiento sobre el dominio.
- Analizar el conocimiento sobre el dominio.
- Crear una red de relaciones entre conceptos.
- La reutilización de conocimiento.
- Crear una red de relaciones entre conceptos.
- La reutilización de conocimiento.

### **1.4.4 Repositorio Semántico de OA.**

Con el uso de las tecnologías de la WS en los ROA, para añadirle significado y mejorar los procesos que se llevan a cabo en éstos, hasta convertirlos en Repositorio Semánticos de Objetos de Aprendizaje (RSOA). Se crean ontologías para añadirle semántica a los ROA de forma tal que se puede diseñar una ontología donde cualquier concepto enlazado a la representación de una actividad de aprendizaje es considerado un objeto de aprendizaje. Este nuevo modelo debe proporcionar una serie de funcionalidades adaptadas a cada conceptualización particular de “objeto de aprendizaje” y no necesariamente restringidas a una única definición. La flexibilidad de este nuevo esquema, nos permitirá almacenar en un repositorio cualquier tipo de información sobre un objeto de aprendizaje, normalizada o sin normalizar, y si

se desea (depende de la concepción particular del creador) con la especificación del propósito educativo del mismo.

La implementación de estos RSOA le permite a los clientes (usuario finales, agentes software y sistemas de administrador de contenidos didácticos), poder, entre otras funcionalidades, añadir, recuperar, modificar y buscar objetos de aprendizaje a pesar de la conceptualización utilizada por el creador. Por ejemplo, un cliente software utilizando un modelo basado en LOM<sup>56</sup>, podría recuperar objetos de tales repositorios, incluso aquellos objetos que estén albergados en otro tipo de sistemas con otras especificaciones de ejecución (como por ejemplo sistemas de administración de contenidos de aprendizaje basados en SCORM).

### 1.5. Repositorio Semántico de OA en la UCI.

El primer intento realizado en Cuba para construir un Repositorio de Objetos de Aprendizaje fue en la Universidad de Holguín, en donde se “propuso desarrollar una plataforma Web dinámica para el e-Learning que posibilitara la configuración y administración de la misma, además de la creación, actualización y oferta de cursos de una forma flexible y agradable, sustentada sobre un sistema código abierto”. (Tamayo, y otros, 2005)

Basado en la experiencia de la Universidad de Holguín, en la Universidad de las Ciencias Informáticas se implementó un Repositorio de Objetos de Aprendizaje (ROA). “Este sistema permite disponer de un lugar común donde profesores y estudiantes almacenen los recursos empleados en el proceso de enseñanza aprendizaje” (Leyva, y otros, 2006). Actualmente ROA cuenta con las 4 funcionalidades básicas de cualquier repositorio: subir, actualizar, explorar paquetes y buscar recursos. (Leyva, y otros, 2006)

Con el desarrollo de esta herramienta y la puesta en marcha en la Universidad, se espera que su principal problema ocurra al crecer el número de OA, lo que traerá consigo que las búsquedas se harán ineficientes, lo que traerá consigo buscar un modo de mejorar el sistema de búsquedas y a la vez el

---

<sup>56</sup> LOM (LEARNING OBJECT METADATA) <http://ltsc.ieee.org>

acceso a la información del mismo, y conjuntamente organizar los OA del repositorio teniendo en cuenta las descripciones de sus datos.

## Capítulo 2. Propuesta del Repositorio Semántico

### 2.1 Objetivo del repositorio semántico.

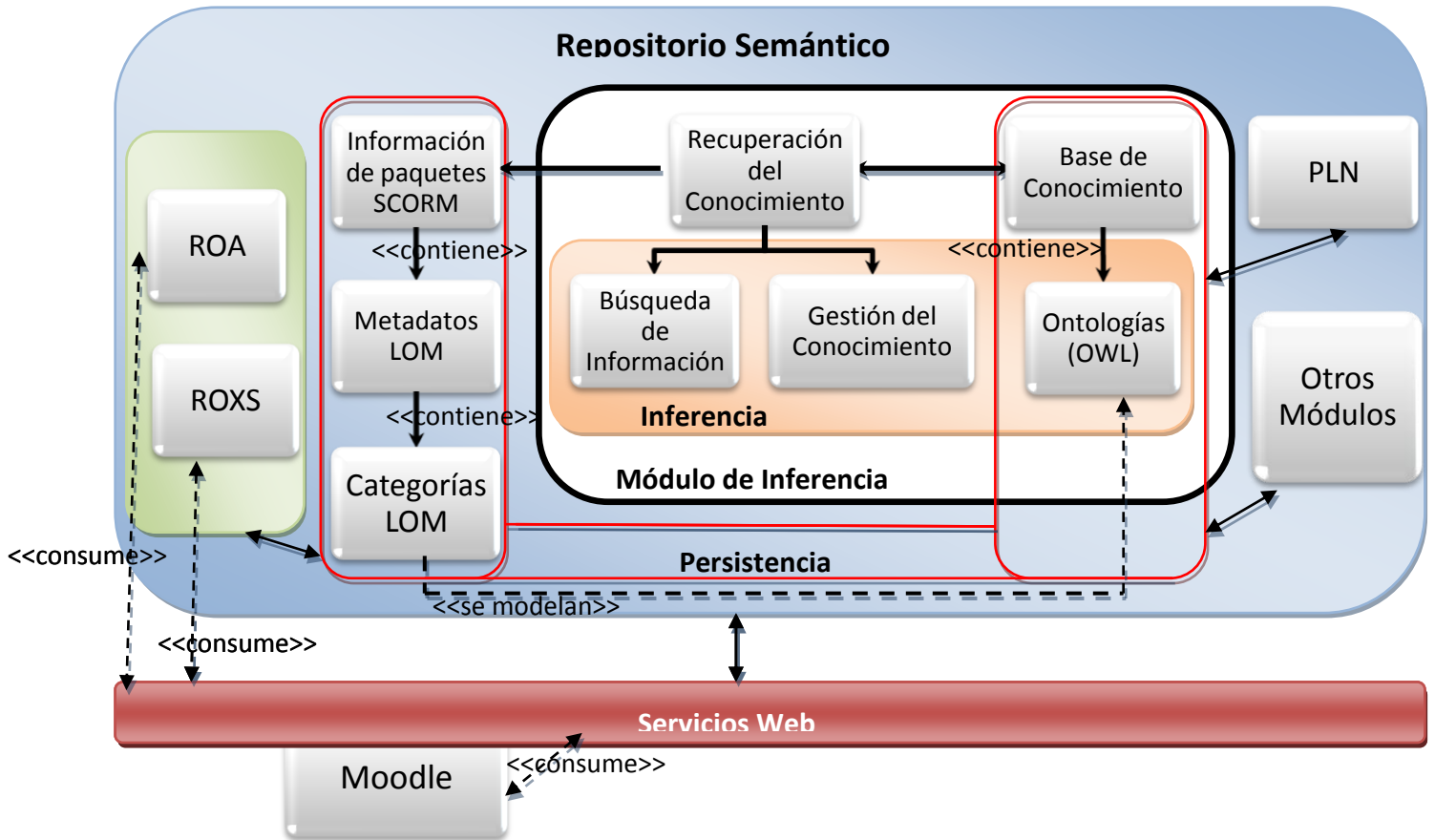


Figura 2: Esquema del Repositorio Semántico.

El Repositorio Semántico toma las funcionalidades que están encapsuladas en las herramientas ROA y ROXS, y las optimiza haciendo uso de la gestión de la información y brinda nuevos servicios a las herramientas anteriormente mencionadas. Las funcionalidades desarrolladas por el RS y las herramientas ROA y ROXS le brindan valor agregado al LMS Moodle.

### **2.2 Descripción interna del Repositorio Semántico.**

El ROA almacena los metadatos de los paquetes SCORM en la base de datos de XML, eXist, mientras que los recursos que están contenidos en éstos paquetes se almacenan en una base de datos PostgreSQL.

Para la implementación del Repositorio Semántico se obtiene la información de los paquetes SCORM, los mismos contienen metadatos bajo el estándar LOM y éstos a su vez poseen categorías LOM, las cuales están modeladas en una ontología (OntoLOM) que forma parte de la base de conocimientos del RS, esta base de conocimientos también contiene ontologías para modelar otros conceptos que serán necesarios para los módulos que conforman el sistema, como es el módulo de Procesamiento de Lenguaje Natural (PLN).

El módulo principal de nuestro sistema es el de Inferencia, este módulo será el encargado de realizar los procesos de recuperación del conocimiento, tales como son: la búsqueda de información, las inferencias y la gestión del conocimiento; para realizar estos procesos se necesita tener la información en base de conocimiento, dicha base de conocimiento estrá formada por ontologías.

El RS contiene un conjunto de ontologías bajo el dominio e-Learning, en las cuales se modelan los conceptos manejados por el estándar SCORM específicamente LOM, para la gestión del conocimiento de los paquetes (SCORM) que se encuentra en ROA (ver figura 2). Las ontologías y toda la información que almacena la misma se encuentran guardadas en una base de datos eXist. La capa intermedia Servicios Web está creada para permitir que el ROA pueda consumir los servicios de gestión del conocimiento que se brindan en el Repositorio Semántico.

## *Capítulo 2. Propuesta del Repositorio Semántico*

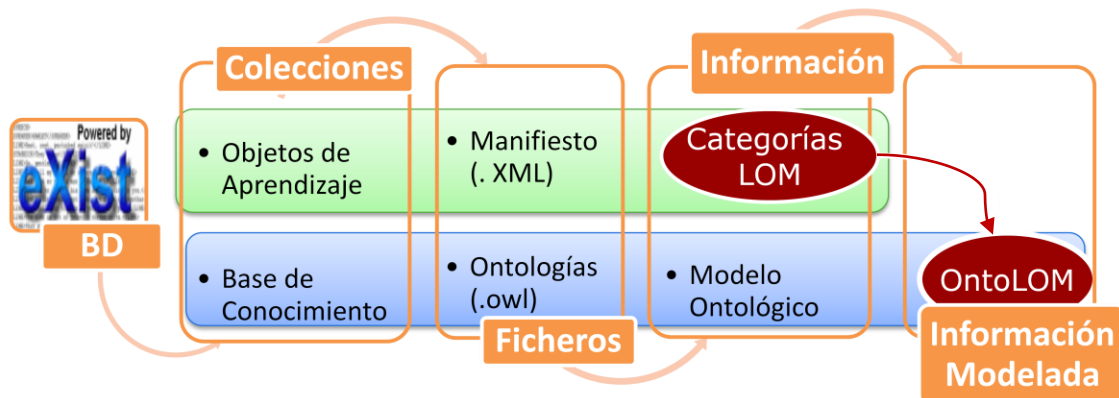
---

---

Básicamente el Repositorio Semántico se encarga de tomar la información de los metadatos de los paquetes SCORM contenidos en una base de datos eXist y cargarla en una ontología (OntoLOM) que modela esa información para así realizar una gestión del conocimiento en el modelo ontológico dando respuesta a los servicios solicitados por ROA, ROXS o Moodle a través de la Capa de Servicios Web.

En la base de datos eXist, mencionada anteriormente se encuentra toda la información organizada por colecciones, contando con la colección “Paquetes SCORM” (la cual almacena los paquetes SCORM) y la colección “Base de Conocimiento” (donde se encuentran un conjunto de ontologías que modelan los conceptos fundamentales para el funcionamiento del repositorio). En la siguiente imagen se modela conceptualmente como está organizada la información en la base de datos y como los ROA trabajan directo con la información contenida en la primera colección mencionada anteriormente, mientras que el RS lleva esta información a modelos semánticos que se encuentran en la segunda colección mencionada, para que el módulo de Inferencia realice el proceso de recuperación de la información.

Además del Módulo de Inferencia será necesario la creación de otros módulos para darle más funcionalidades al RS, entre estos módulos se encuentra identificado el de PLN, el cual será el encargado de hacer resúmenes acerca del tema que tratan los recursos contenidos en los paquetes para posibilitar su reutilización en otros y permitir la creación de nuevos recursos a partir de la unión de conocimiento de varios de éstos; todas las funcionalidades que implementan estos módulos serán brindadas como servicios a través de una Capa de Servicios Web, la cual le facilitará su trabajo a herramientas tales como ROXS, ROA y Moodle. Todos los elementos abordados anteriormente se encuentran representados en las figuras 1 y 2.



**Figura 3:** Persistencia de datos en eXist.

### Leyenda



Información que manipula ROA



Meta-modelado de la Información que manipula ROA

En la figura 3 se representan los paquetes que conforman el repositorio, los cuales son LOM, Interface\_U, dbExist, OWL, System\_Owl, Inference, Web\_Services, Persistence y OWL.

- **LOM:** se encarga de toda la gestión de los ficheros xml extraídos de la base de datos donde ROA almacena los metadatos y los paquetes SCORM.
- **Interface\_U:** interfaz de administración de sistema.
- **dbExist:** gestiona con la base de datos de xml (eXist) las ontologías almacenadas en ésta.
- **System\_Owl:** controla el envío de los mensajes en el resto de los paquetes.

## Capítulo 2. Propuesta del Repositorio Semántico

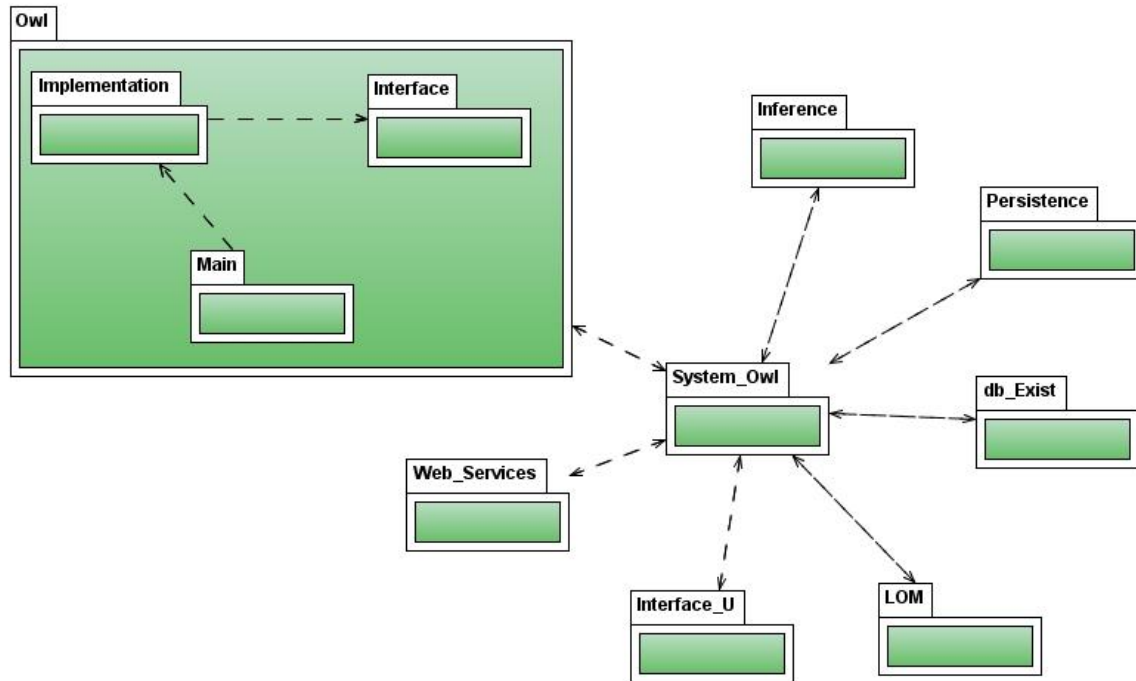
---

---

- **Inference:** realiza los procesos de inferencia sobre el modelo ontológico.
- **Web\_Services:** es el encargado de brindar los Servicios Web.
- **Persistence:** trabaja directamente con los ficheros que se extraen y se guardan en la base de datos dbExist.
- **OWL:** este paquete contiene 3 sub-paquetes (Interface, Implementation, Main).
  - **Interface:** contiene las interfaces de los conceptos que se modelan en la ontología bajo el dominio e-Learning.
  - **Implementation:** implementa las interfaces contenidas en el paquete Interface.
  - **Main:** permite la gestión de los componentes modelados en la ontología.



En la imagen siguiente (figura 3) se muestra el flujo de mensajes entre los paquetes anteriormente mencionados.

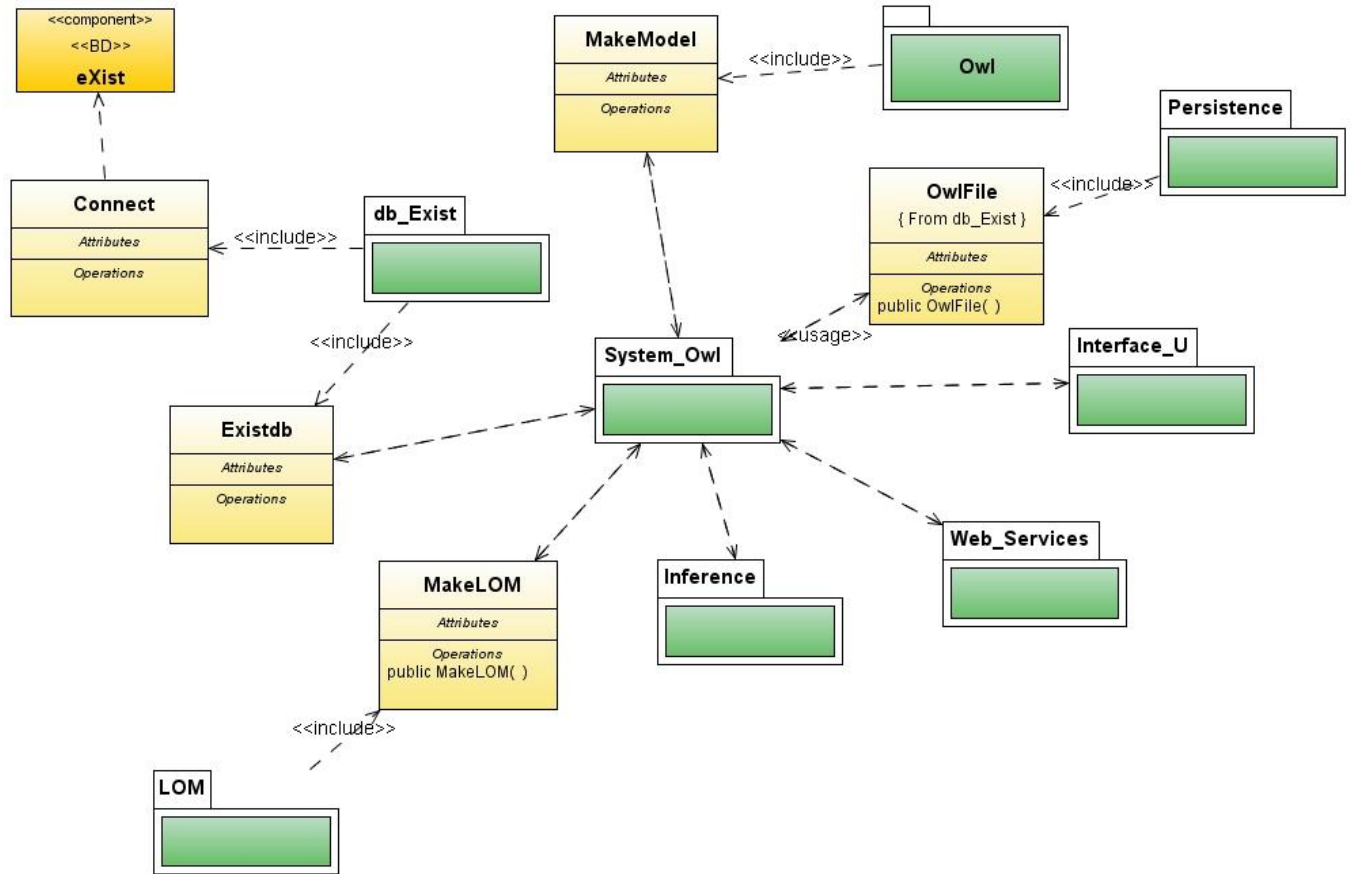


**Figura 4:** Flujo de mensajes entre los paquetes.

Los pedidos son recepcionados por el paquete Web\_Services y este lo envía al paquete System\_Owl (constituye el núcleo del sistema, es el encargado de controlar el flujo de mensajes y hacer los pedidos al resto de los paquetes haciendo uso de clases que sirven de interfaces de salida en el resto de los paquetes), éste solicita a db\_Exist la extracción de la ontología de la base de datos eXist y solicita también la extracción de información de los paquetes SCORM (que está en la colección ROA de la base de datos de eXist), luego dbExist pasa la información obtenida a System\_Owl y éste la lleva al paquete Persistence, el mismo envía los ficheros a System\_Owl, una vez allí los ficheros son enviados a OWL donde se crea el modelo y se le añade la información obtenida por db\_Exist, modelo que es llevado por System\_Owl hasta Inference donde se extrae el conocimiento solicitado a través de Web\_Services, se genera una respuesta y System\_Owl la lleva

a Web\_Services dando un servicio de respuesta. El paquete Interface\_U es el que construye la interfaz visual, la que va a permitir la administración del sistema.

Este flujo de información que relaciona a los paquetes del repositorio se encuentra representado a continuación (figura 4).



**Figura 5:** Interacción entre los paquetes del Módulo Semántico.

## 2.3 Especificación de los procesos del Repositorio Semántico.

### Flujo de procesos

El flujo “Crear Modelo” se desarrolla mediante una conexión a la base de datos eXist, en la cual se extrae la ontología (ficheros en extensión \*.owl), luego carga la información que contiene la ontología (taxonomía, propiedades, instancias, etc.) transformándola en un Modelo Ontológico. Este flujo se muestra en la figura 5 y la leyenda a continuación se aplica para todos los diagramas que representan flujos.

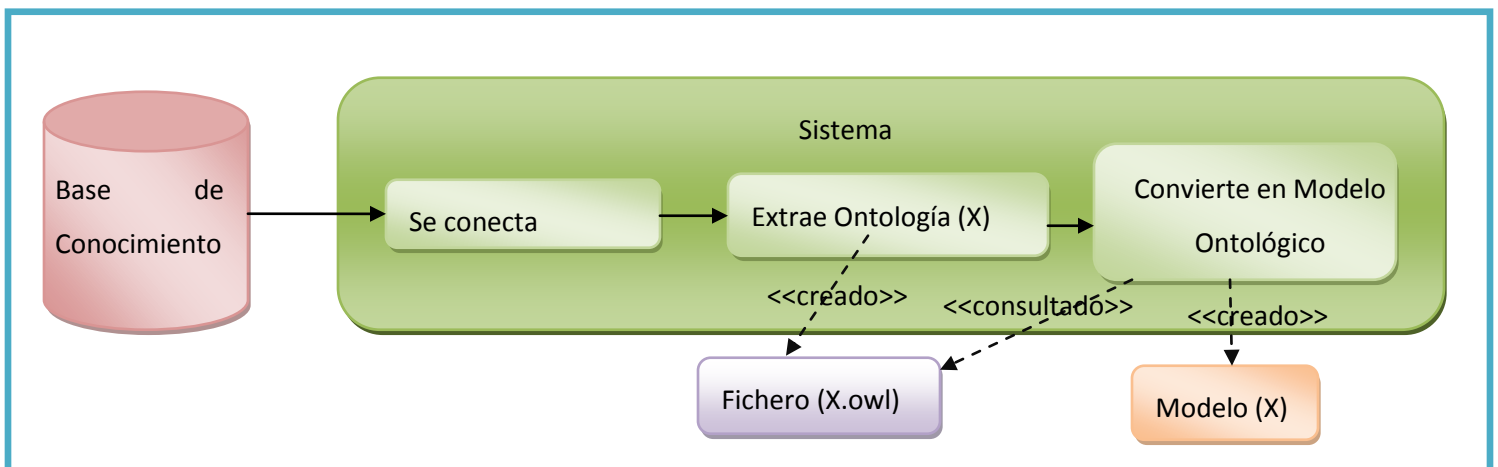
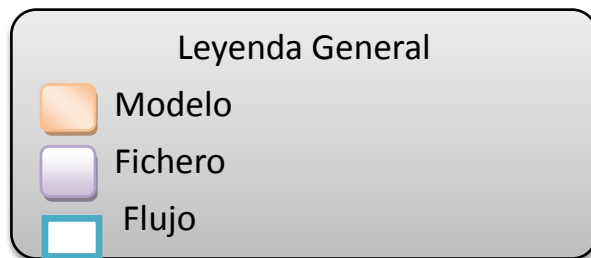
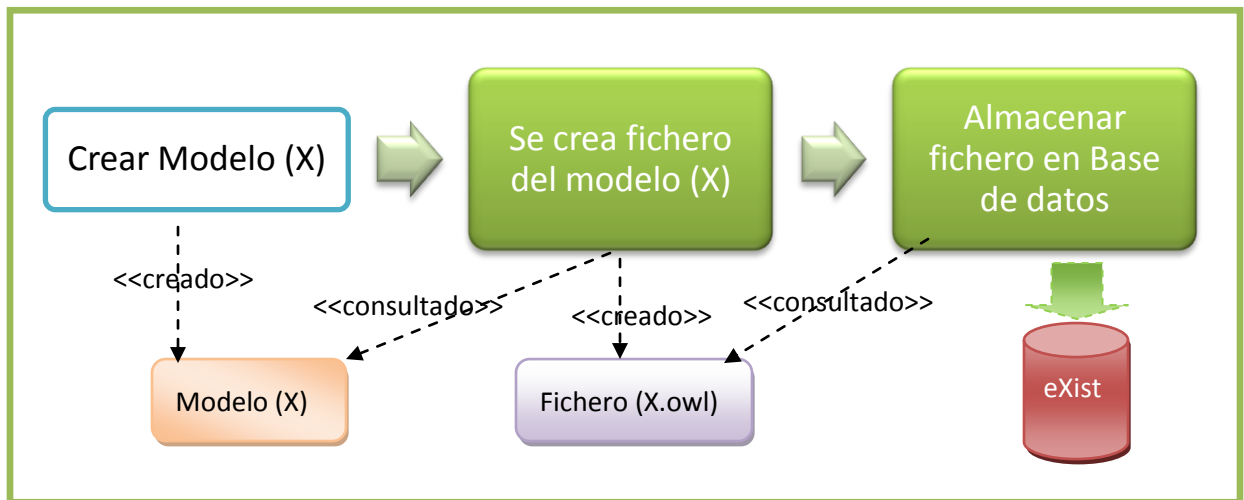


Figura 6: Flujo de proceso Crear Modelo.

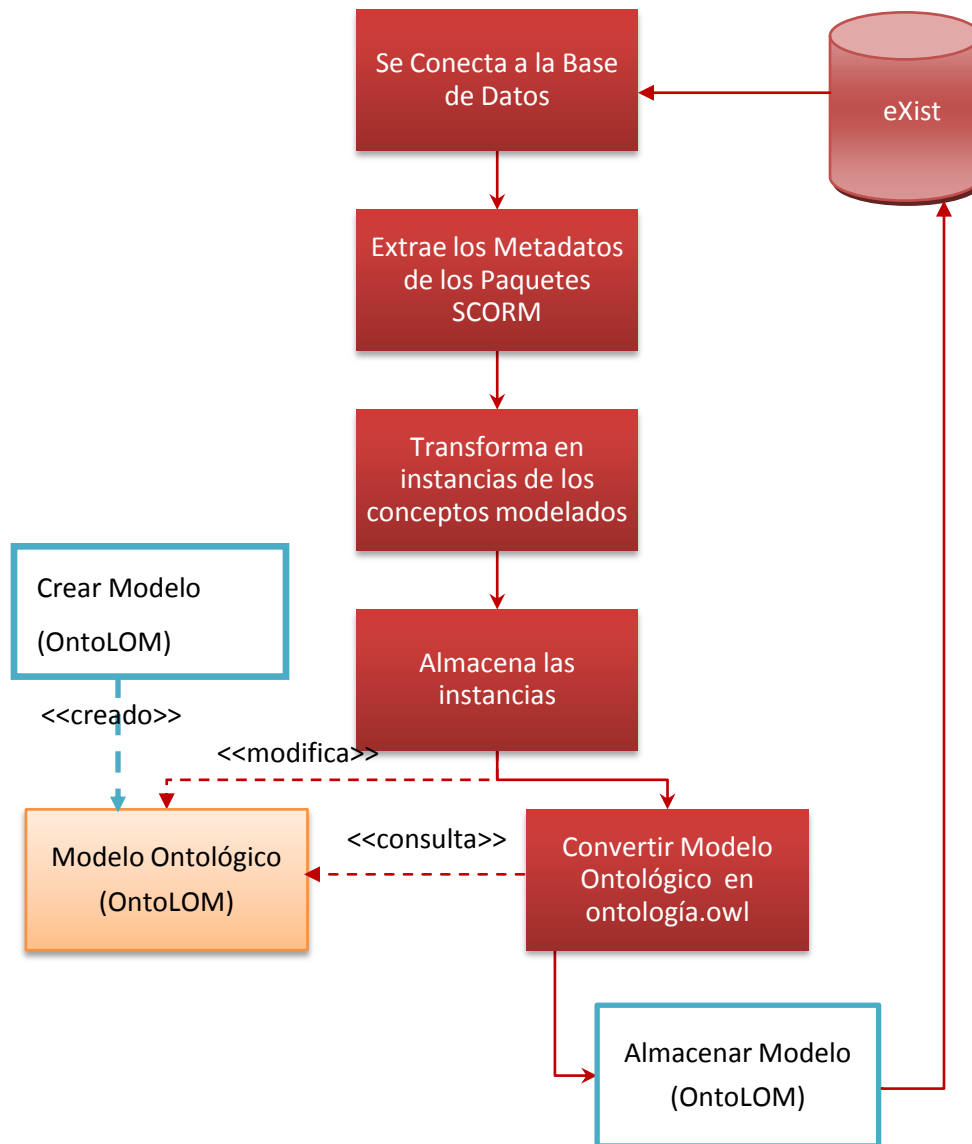
El flujo “Almacenar Modelo” comienza cuando está creado con anterioridad el modelo, mediante el flujo “Crear Modelo”, luego toma la información del modelo convirtiéndola en el fichero (con extensión \*.owl) y posterior a ello el sistema realiza una consulta a este fichero y la almacena en la base de datos eXist. A continuación se representa dicho flujo de procesos.



**Figura 7:** Flujo de proceso Almacenar Modelo.

El flujo “Cargar los metadatos de los paquetes SCORM” comienza cuando el sistema se conecta a la base de datos eXist, de la cual extrae los metadatos de los paquetes SCORM, luego se transforma en instancias los conceptos modelados por los metadatos SCORM, es decir, toma los valores de las 9 categorías del estándar LOM contenidas en los manifiestos del SCORM y las convierte en instancias de las 9 categorías conceptualizadas en el modelo ontológico de la ontología OntoLOM, dicho modelo se crea mediante el flujo anteriormente explicado “Crear Modelo”. Estas instancias creadas se almacenan en el modelo ontológico para luego convertir este

modelo en una ontología la cual se almacena en la base de datos. A continuación se muestra la figura de dicho flujo de procesos.



**Figura 8:** Flujo de proceso Cargar los metadatos de los paquetes SCORM.

El flujo “Actualizar” se inicia cuando se crea un modelo de la última ontología OntoLOM guardada y es creado el modelo Historial (ambos creados mediante el flujo Crear Modelo), paralelo a ello se cargan los metadatos de los paquetes SCORM (se realiza el flujo anteriormente explicado Cargar los metadatos de los paquetes SCORM, pero hasta el paso donde se crea el modelo ontológico), luego se comparan los modelos de OntoLOM, el último guardado y el actual, teniendo en cuenta varios parámetros tales como cantidad de paquetes, paquetes actualizados, paquetes eliminados y dependiendo de los resultados, si se decide mantener el último modelo se detiene el flujo porque ya está guardado el modelo, sin embargo si se decide guardar el modelo actual se realiza mediante el flujo Almacenar Modelo para luego instanciar el modelo (último guardado) en el modelo Historial el cual luego será almacenado en la base de datos mediante el flujo Almacenar Modelo (Ver figura 8).

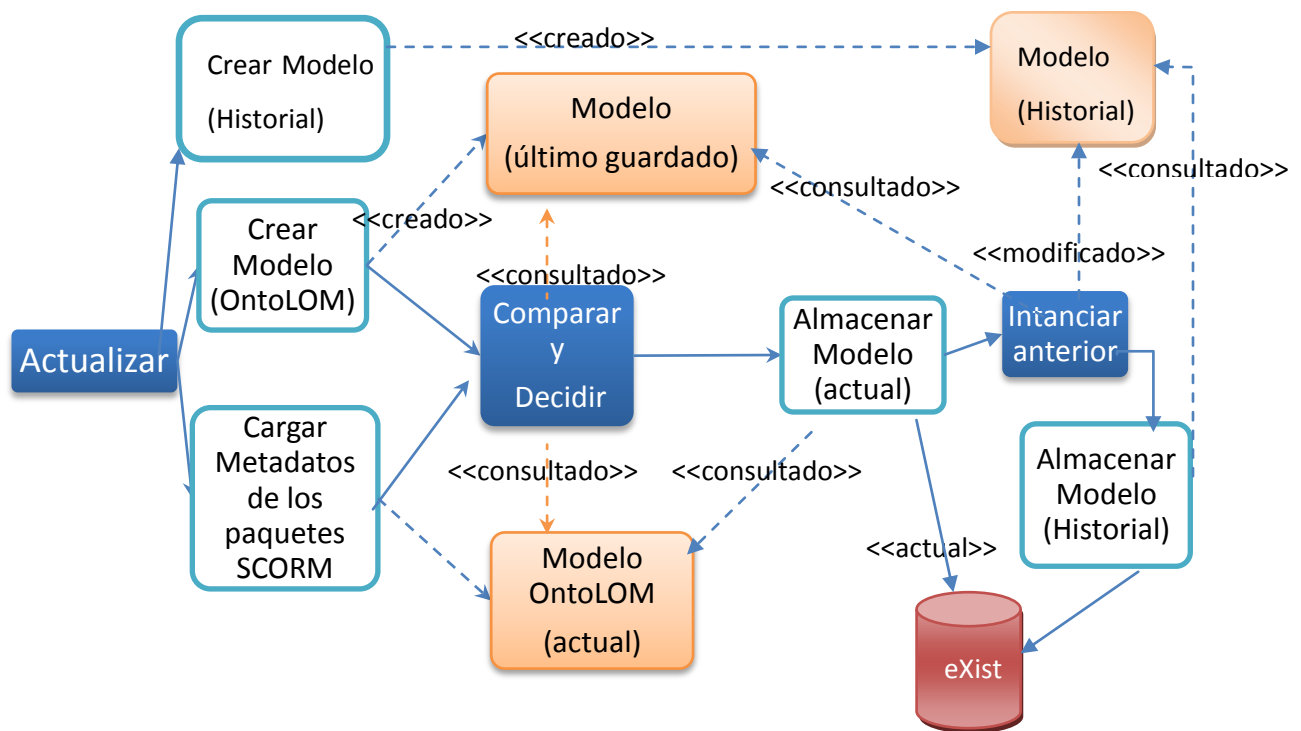
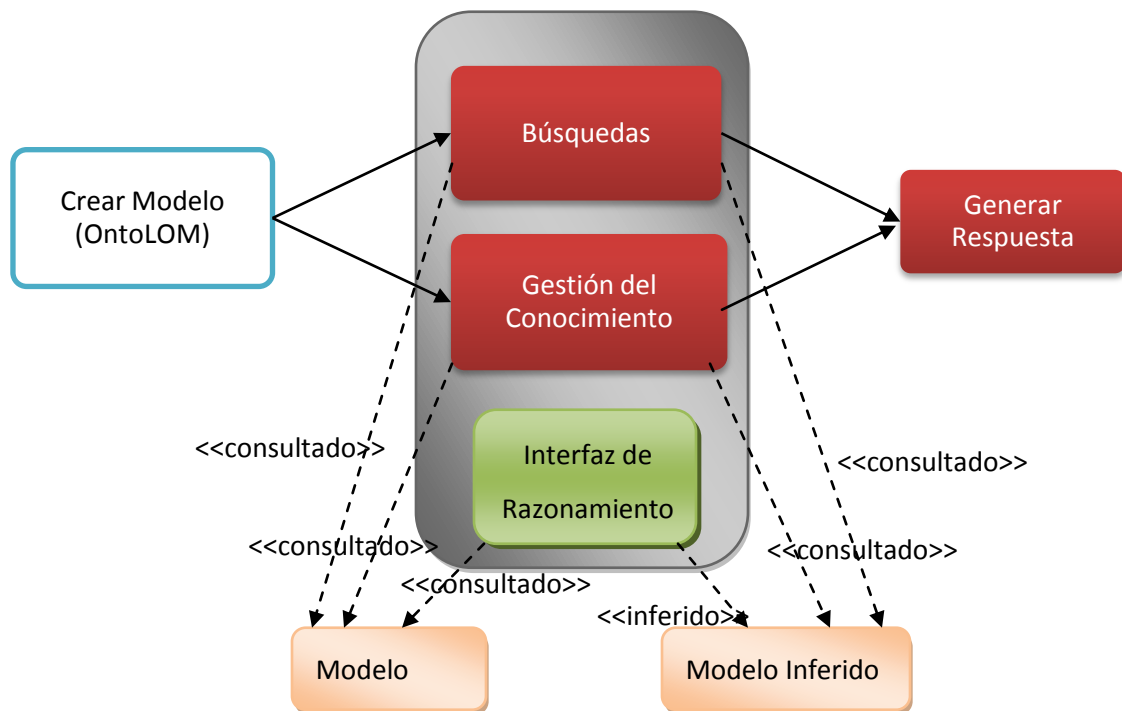


Figura 9: Flujo de proceso Actualizar.

El diagrama que se muestra a continuación (Figura 9) representa el flujo “Realizar proceso de recuperación de información” que se inicia cuando se crea el modelo de la ontología OntoLOM a través del flujo Crear Modelo, para luego poder realizar procesos de búsqueda y de gestión del conocimiento (buscar, organizar, filtrar y presentar la información). Estos dos procesos mencionados anteriormente se pueden apoyar en las inferencias sobre el modelo para generar la respuesta. Cuando se realizan las inferencias la interfaz de razonamiento infiere un modelo a partir del modelo anteriormente creado sobre el cual se realizarán entonces la gestión del conocimiento. La interfaz de razonamiento es la que se encarga de conectarse con el razonador Pellet 2.0 rc-5 a través de los servicios web que éste brinda para así validar el modelo e inferir el nuevo modelo.



**Figura 10:** Flujo de proceso Realizar proceso de recuperación de información.

### 2.4 Creación de la ontología.

#### 2.4.1 Proceso de creación de la ontología OntoLOM.

Para poder realizar los procesos de búsqueda/gestión del conocimiento se hace necesaria la creación de una ontología que modele la información acerca de los OA (fundamentalmente los conceptos manejados por el estándar LOM); para el desarrollo de la misma se estudió la ontología SLOR, la cual modela los 3 conceptos de OA siguientes:

- 1) Los OA pueden ser utilizadas en eventos de aprendizaje.(Ejemplo: un libro)
- 2) Los OA son cualquier cosa digital con o sin propósito educativo que pueden ser utilizadas en eventos de aprendizaje.
- 3) Los OA son objetos digitales que contengan información específica (de metadatos) para su utilización con un carácter educativo.

De estos conceptos se escogió el último pues este se adapta a los OA que se encuentran en el ROA y tiene en cuenta los metadatos que estos poseen, y una parte fundamental a modelar es el estándar LOM.

Para la creación de la ontología se definieron los siguientes pasos:

1. Definir el **dominio de ontología** donde se enumeran los conceptos a modelar, este constituye uno de los pasos imprescindibles ya que se restringe el área de modelado.
2. Definir los **conceptos**, que pertenecen al dominio de ontología definido, a representar en el modelo.
3. Determinar las **propiedades** correspondientes a cada concepto (se elige el tipo de dato y se restringen al dominio de su concepto). Se revisa si existen propiedades que se repiten, en caso de existir se les pone en el dominio de los conceptos en los que se encuentra, sin duplicar propiedades si no es necesario.
4. Se suman las propiedades para obtener los conceptos más simples, y luego se le añaden otros conceptos y/o propiedades a estos conceptos simples, para así obtener otros conceptos más complejos.
5. Se escriben las reglas restrictivas para las propiedades que sean necesarias.
6. Verificar la consistencia de la ontología haciendo uso de un razonador (Pellet).



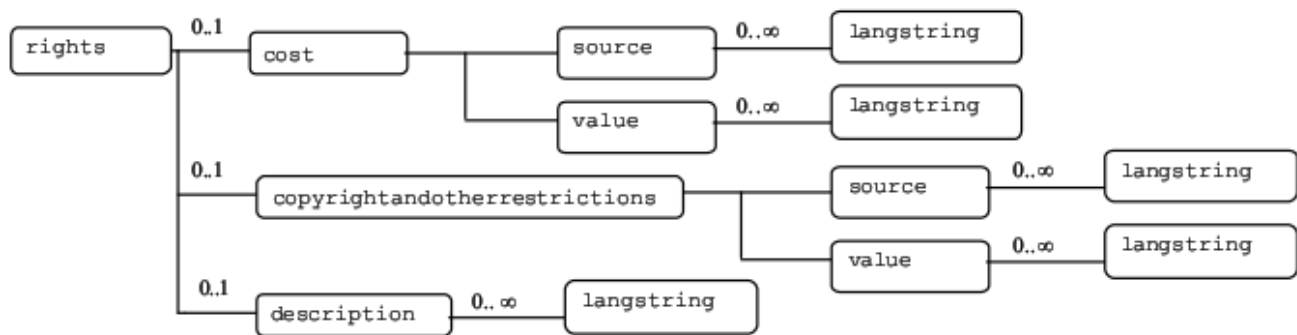
Ejemplo de creación de una ontología:

### Paso 1

Se determina el dominio de la ontología OntoLOM: e-Learning, ya que este engloba los conceptos que se desean modelar.

### Paso 2

Los conceptos a modelar son: el de OA seleccionado de los anteriormente explicados, SCORM, LOM y las 9 categorías de LOM. También se determinaron como conceptos las propiedades de las 9 categorías de LOM, que poseen 2 o más sub-propiedades. Por ejemplo:



**Figura 11:** Codificación XML de los metadatos en la categoría Rights.

La categoría general posee las propiedades “Cost y Copyrightandotherrestrictions” con más de una sub-propiedad, por lo que estas fueron tomadas como conceptos.

### Paso 3

Siguiendo con el ejemplo anterior se tienen los conceptos: Cost, Copyrightandotherrestrictions y Rights.

Propiedades por conceptos:

- Cost: source(String), value(String)
- Copyrightandotherrestrictions: source(String), value(String)

## Capítulo 2. Propuesta del Repositorio Semántico

- Rights: descriptions (String)

Se puede observar que las propiedades **source** y **value** están presentes en los conceptos **cost** y **copyrightandotherrestrictions**, por lo que se deben de modelar una sola vez, pero incluirlas en el dominio de ambos conceptos, quedando entonces:

Propiedades	Dominios
<b>Source</b>	Cost y Copyrightandotherrestrictions
<b>Value</b>	Cost y Copyrightandotherrestrictions
<b>descriptions</b>	Rights

**Tabla 3:** Dominios correspondientes a propiedades.

### Paso 4

		Propiedad o Concepto (P/C)	Dominio de Propiedades
	R_cost_source	P	Cost
+	R_cost_value	P	Cost
=	R_Cost	C	

**Tabla 4:** Suma de propiedades para obtener el concepto R\_Cost.

		Propiedad o Concepto (P/C)	Dominio de Propiedades
	R_Copyrightandotherrestrictions_source	P	Copyrightandot
+	R_Copyrightandotherrestrictions value	P	Copyrightandot
=	R_Copyrightandotherrestrictions	C	

**Tabla 5:** Suma de propiedades para obtener el concepto R\_Copyrightandotherrestrictions.

		Propiedad o Concepto (P/C)	Dominio de Propiedades
+	R_descriptions	P	Rights
	R_Copyrightandotherrestrictions	C	
	R_Cost	C	
=	Rights	C	

Tabla 6: Suma de propiedades para obtener el concepto Rights.

Paso 5

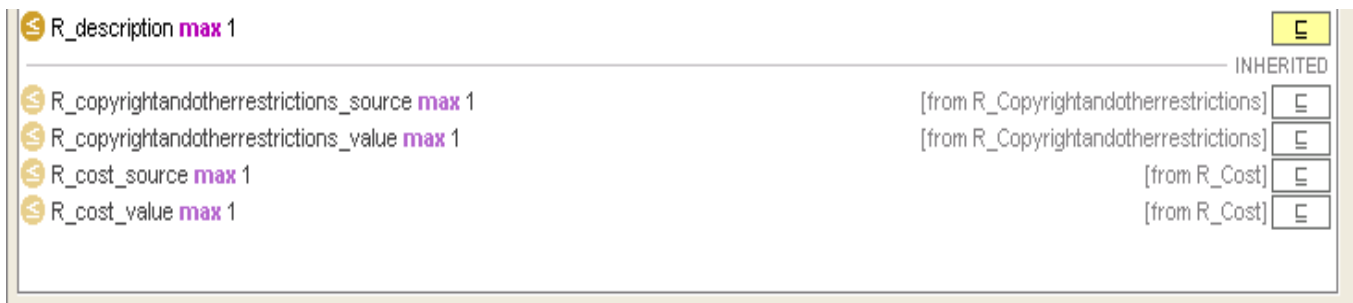


Figura 12: Reglas restrictivas por propiedad.

A continuación se muestran las representaciones esquemáticas de cómo están modelados los conceptos de LOM en la Ontología OntoLOM creada siguiendo los pasos anteriormente expuestos.



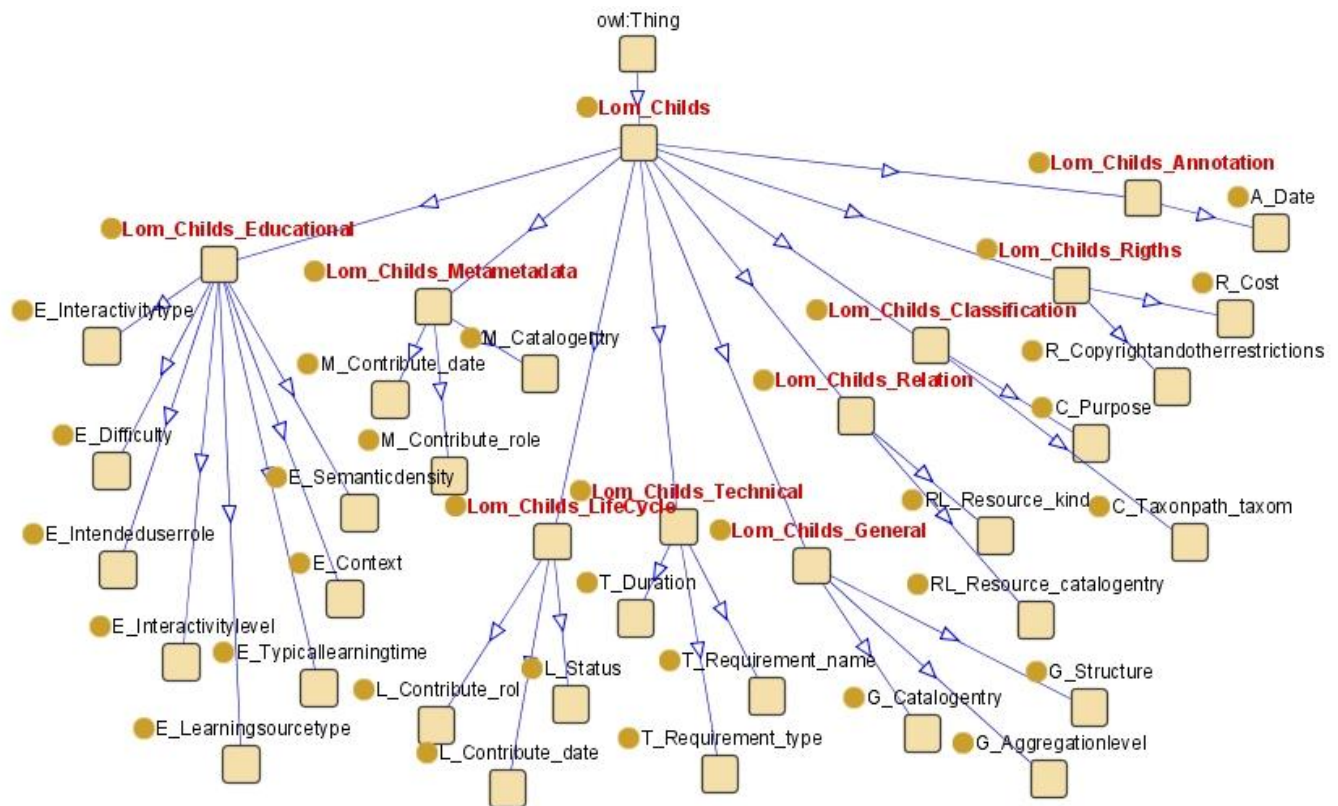


Figura 15: Desglose conceptual de las nueve categorías de LOM conceptualizadas.

### 2.5 Proceso de Inferencia en ontologías

Para realizar los procesos de inferencia en el Repositorio Semántico se implementa el Módulo de Inferencia, el cual apoya el proceso de Recuperación/Búsqueda de Información en las ontologías creadas, haciendo uso de las siguientes Apis: Jena 2.5.6, OWL 2.0 y Pellet 2.0rc5.

Jena y OWL trabajan con los modelos ontológicos, se utilizan para leer y escribir dichos modelos, añadir recursos (conceptos, propiedades, instancias, etc.) y Pellet que es un razonador el cual contiene una interfaz para la entrada de la ontología, la cual luego pasa a una etapa donde se valida y/o se repara la misma, para luego hacer la extracción de la información llevándolos axiomas sobre clases (subclases, clases equivalentes y axiomas desunidos) al componente TBox y las

aserciones individuales (tipos y propiedad de las aserciones) al componente ABox. Para luego haciendo uso de las Apis Jena, Owl o DIG interactuar con la interfaz de la base de conocimiento para así poder realizar las inferencias. Por otra parte, Pellet cuenta con el parser SPARQL el cual le permite realizar búsquedas directas sobre las aserciones individuales (ABox), como se muestra en la siguiente figura.

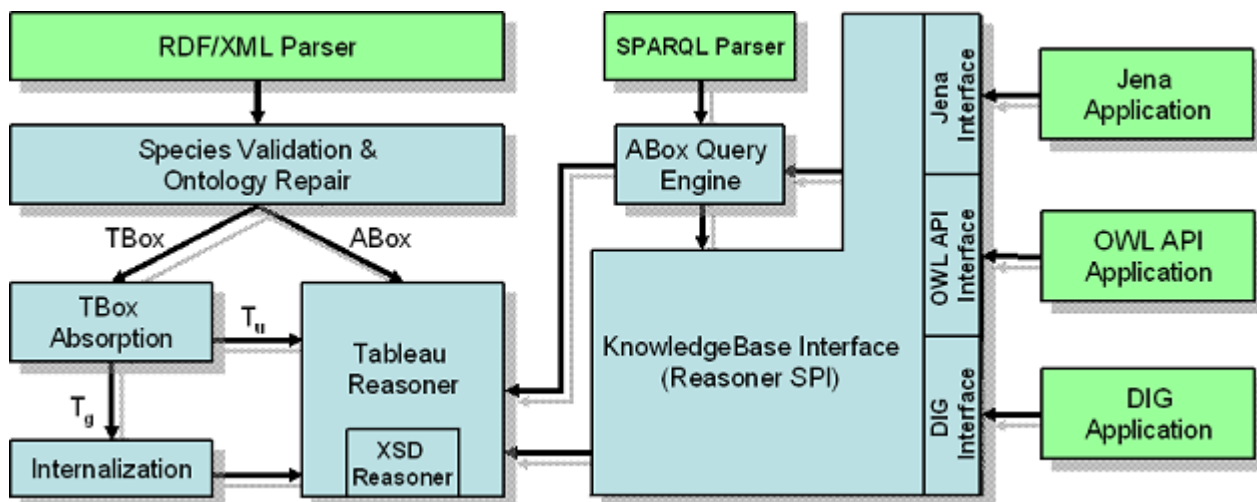


Figura 16: Arquitectura de Pellet.

Las ontologías aceptan varias ligas tales como:

- B es una subclase o instancia de A (es)
- B y A están relacionados en la forma indicada por R
- Conjunción (A & B & C ...)
- Disjunción (A v B v C ...)

En la ontología se cuenta con ligaduras tales como:

$LOM\_Cat\_Clasificación \equiv C\_Purpose \sqcup C\_Taxonpath$  // el concepto LOM es equivalente a la unión del concepto C\_Purpose y el concepto C\_Taxonpath.

La ontología se puede ver dividida en planos. En cada plano se tiene la definición de un concepto, pero estos tienen ligas a otros planos en que hay conceptos relacionados. Es decir que un nodo

## Capítulo 2. Propuesta del Repositorio Semántico

---

---

tiene ligas a nodos del mismo plano que lo definen, pero también a nodos de otros planos que están relacionados, como subclases, superclases, analogías, etc. En cada plano hay un nodo tipo y una serie de nodos “token”, los conceptos representados de esta manera se pueden definir inmediatamente (la definición directa mediante los nodos en el mismo plano) y por concepto completo (todos los nodos y relaciones a las que se pueda llegar en red partiendo de dicho nodo).

En general es posible usar este tipo de estructuras para diferentes tipos de razonamiento:

1. Búsqueda asociativa: Encontrar si están relacionados dos o más conceptos, y su tipo de relación.
2. Reconocimiento: Dada una serie de características (nodos “token”) encontrar el concepto (nodo clase) que mejor las define.
3. Descripción: Expresar un concepto en base a sus componentes y relaciones entre ellas (procesamiento del lenguaje natural).

De estos tipos de razonamiento se hizo uso de la búsqueda asociativa y la de reconocimiento auxiliándose para ello de las reglas transitivas que maneja el razonador. Para un mejor aprovechamiento de estas reglas que implementa el razonador Pellet es necesario crear reglas restrictivas tales como:

$\leq$  A\_description max 1 // el concepto en el que está esta regla solo va a tener como máximo una propiedad A\_description.

= identifierManifest exactly 1 // el concepto en el que está esta regla solo va a tener exactamente una propiedad identifierManifest.

$\geq$  T\_requeriment\_Type\_value min 1 // el concepto en el que está esta regla solo va a tener como mínimo una propiedad T\_requeriment\_Type\_value.

$\exists$  hasAssociatedMetadataRecord some LearningMetadataRecord // el concepto del que esté esta regla va a poseer varios valores del concepto LearningMetadataRecord.

L\_Status\_value (single owl: oneof {“draft”, “final”} ) (max Cardinality 1) // la propiedad L\_Status\_value va a poseer solo un valor draft o final.

## **Capítulo 3. Pruebas**

Probar es imprescindible para verificar la calidad y el adecuado funcionamiento de un software. La prueba es un proceso de ejecución de un programa con la intención de comprobar que el producto satisface los requerimientos y se comporta como se desea. Para que las pruebas tengan éxito es necesario realizar casos de pruebas que tengan probabilidad de descubrir los errores en el sistema y utilizar técnicas que nos guíen el proceso de la prueba.

### **3.1 Casos de Prueba**

Un Caso de Prueba es el conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular como, por ejemplo; ejercitar el camino concreto de un programa o verificar el cumplimiento de un requisito determinado. Para probar el módulo en cuestión se utilizaron las técnicas de caja blanca y se aplicaron pruebas de rendimiento.

#### **3.1.1 Prueba de caja blanca**

Las técnicas de caja blanca o técnicas estructurales se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento. Los pasos a realizar para aplicar esta técnica son:

- Representar el programa en un grafo de flujo.
- Calcular la complejidad ciclomática.
- Determinar el conjunto básico de caminos independientes.
- Derivar los casos de prueba que fuerzan la ejecución de cada camino.
- A continuación se muestra una porción del código programado. Sobre este ejemplo se realizarán pruebas de caja blanca. Este código corresponde al flujo de proceso "Crear



Modelo” específicamente a la clase “MakeModel”. El mismo llena una lista con las instancias (OntClass) del modelo obviando las que poseen una url nula.

```

public ArrayList<OntClass> listOntClasses(){

    ArrayList<OntClass> list = new ArrayList<OntClass>();
    Iterator i = MakeModel.getOntmodelo().listClasses();

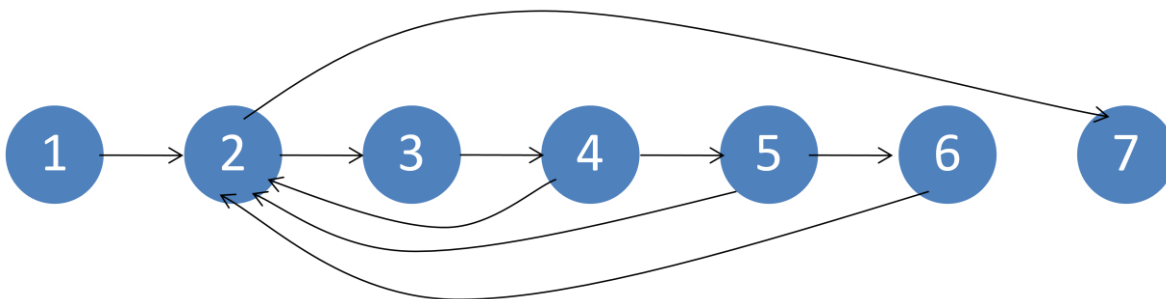
    while (i.hasNext()) {
        OntClass c = (OntClass) i.next();

        if(c != null && c.getURI() != null)
            list.add(c);
    }
    return list;
}

```

**Figura 17:** Porción de Código.

La siguiente figura muestra el grafo de flujo representado para el código anterior:



**Figura 18:** Grafo de flujo.

**Cálculo de la complejidad ciclomática:**

Complejidad ciclomática  $V(G) = \text{Cantidad de Aristas [A]} - \text{Cantidad de nodos [N]} + 2$ .

$$V(G) = A - N + 2$$

$$V(G) = 9 - 7 + 2$$

$$V(G) = 4$$

El resultado de la complejidad ciclomática muestra la cantidad de casos de prueba que son necesarios realizar para el caso de uso. A continuación se muestra el diseño de los casos de pruebas.

Caso de prueba Camino 1(1, 2, 7)

<b>Flujo de Proceso:</b>	<b>“Crear Modelo”</b>
<b>Caso de prueba:</b>	1
<b>Entrada:</b>	Una lista que tiene una sola clase.
<b>Salida:</b>	No se devuelve el resultado que se pide, pues no existe (devuelve una lista vacía).

**Tabla 7:** Caso de prueba # 1 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.

Caso de prueba Camino 2(1, 2, 3, 4, 2, 7)

<b>Caso de Uso:</b>	<b>“Crear Modelo”</b>
<b>Caso de prueba:</b>	2
<b>Entrada:</b>	Obtiene una lista que contiene 2 instancias de OntClass, donde la segunda instancia es de valor nulo.
<b>Salida:</b>	Devuelve una lista vacía.

**Tabla 8:** Caso de prueba # 2 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.

Caso de prueba Camino 3(1, 2, 3, 4, 5, 2, 7)

<b>Caso de Uso:</b>	<b>“Crear Modelo”</b>
<b>Caso de prueba:</b>	3
<b>Entrada:</b>	Obtiene una lista que contiene 2 instancias de OntClass, donde la segunda instancia posee URL con valor nulo.
<b>Salida:</b>	Devuelve una lista vacía.

**Tabla 9:** Caso de prueba # 3 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.

Caso de prueba Camino 4(1, 2, 3, 4, 5, 6, 2, 7)

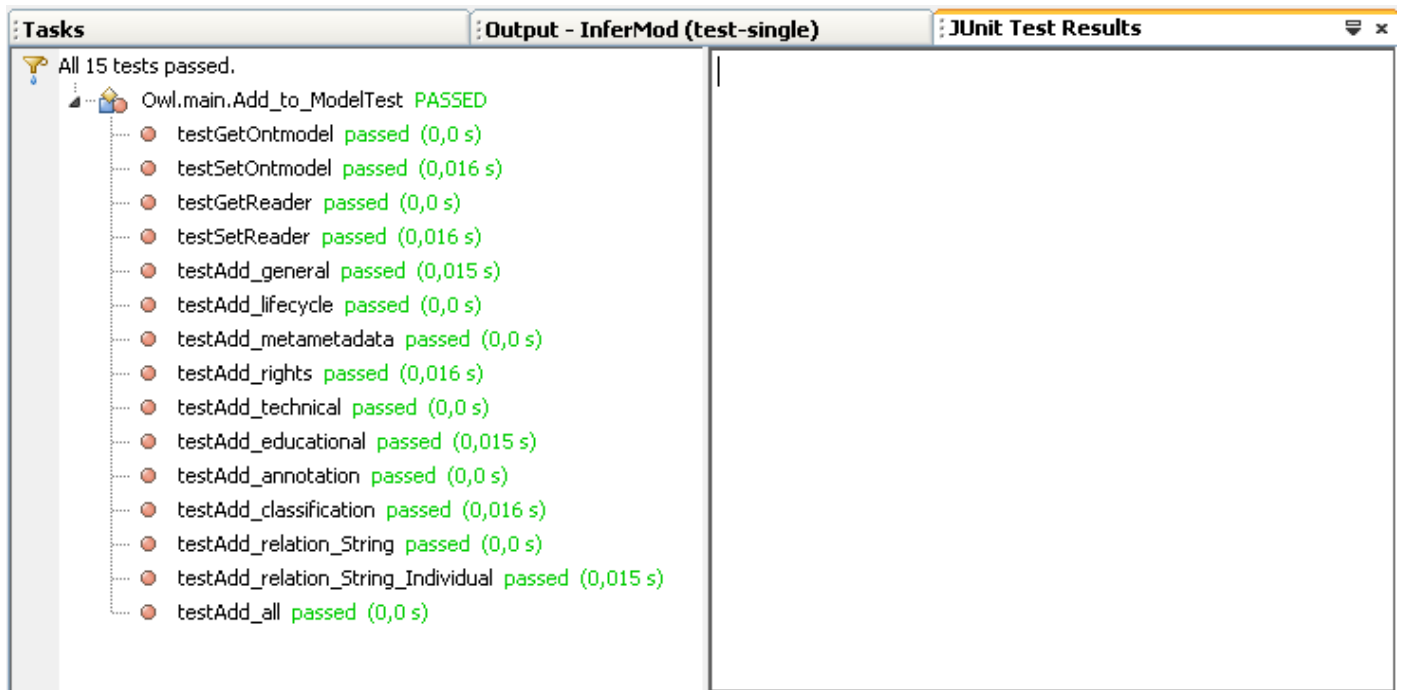
<b>Caso de Uso:</b>	<b>“Crear Modelo”</b>
<b>Caso de prueba:</b>	4
<b>Entrada:</b>	Obtiene una lista que contiene varias instancias de OntClass, que no poseen valor nulo y que tienen la URL diferente del valor nulo.
<b>Salida:</b>	Devuelve una lista con las instancias de OntClass que no poseen URL con valor nulo.

**Tabla 10:** Caso de prueba # 4 utilizando técnicas de caja blanca al flujo de procesos Crear Modelo.

### 3.1.1.1 Pruebas de unidad.

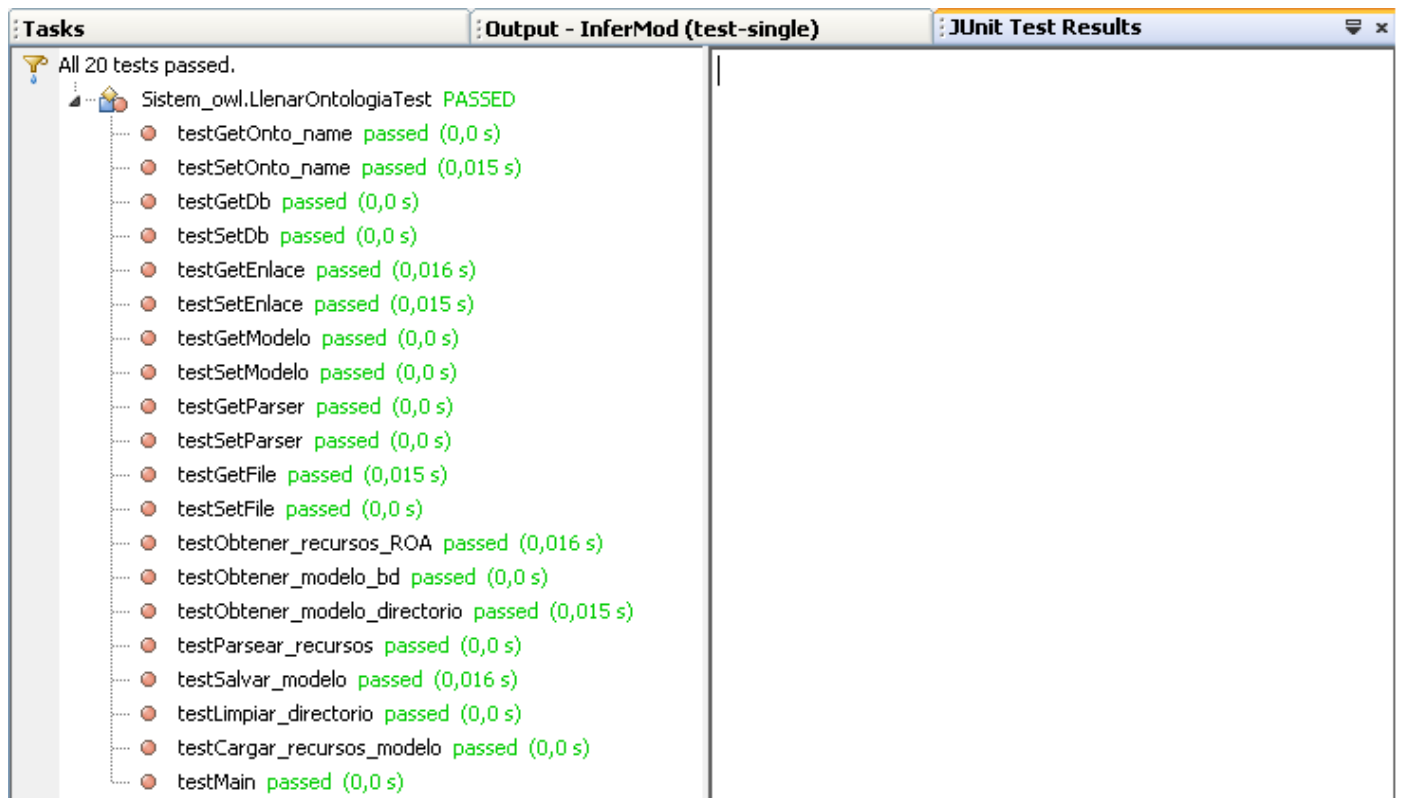
Dentro de las pruebas de caja blanca se encuentra la prueba de unidad, la misma centra el proceso de verificación en la menor unidad del diseño del software: el componente software o módulo, éstas pruebas de unidad se hicieron con el IDE NetBeans 6.5 haciendo uso de su JUnit 4, a continuación

se muestran las figuras obtenidas como resultado de aplicar el proceso de prueba a la clases Add\_to\_Model, LlenarOntología y XMLReaderSample.



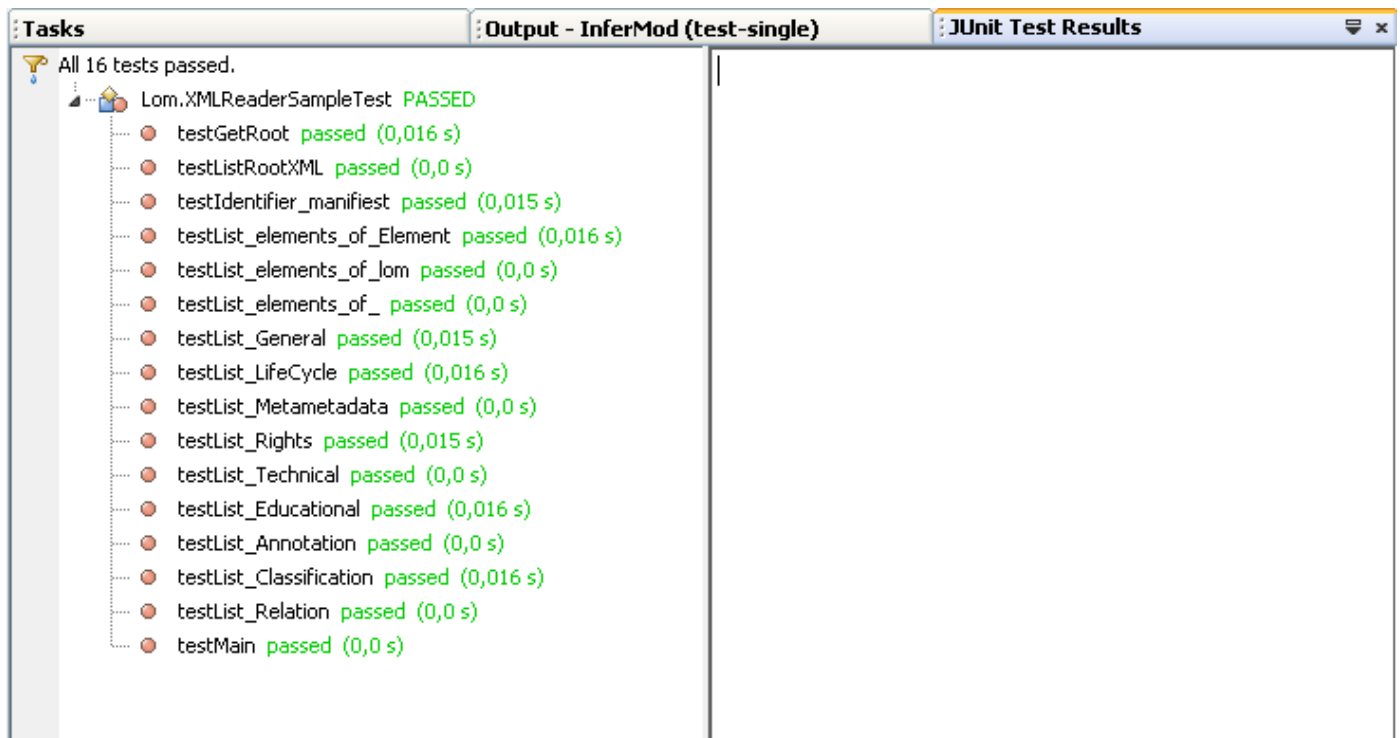
```
compile-test-single:  
Testsuite: Owl.main.Add_to_ModelTest  
Tests run: 15, Failures: 0, Errors: 0, Time elapsed: 0,406
```

**Figura 19:** Prueba de unidad realizada a la clase Add\_to\_Model.



```
compile-test-single:  
Testsuite: Sistem_owl.LlenarOntologiaTest  
Tests run: 20, Failures: 0, Errors: 0, Time elapsed: 0,469 sec
```

**Figura 20:** Prueba de unidad realizada a la clase LlenarOntología.



```

compile-test-single:
  Testsuite: Lom.XMLReaderSampleTest
  Tests run: 16, Failures: 0, Errors: 0, Time elapsed: 0,422 sec
  
```

**Figura 21:** Prueba de unidad realizada a la clase XMLReaderSample.

### 3.1.2 Pruebas de rendimiento.

Todas las pruebas de rendimiento se realizaron en una máquina de 1 GB de RAM, un micro a 3.0 Ghz en el sistema operativo Windows XP SP3, corriendo con JDK<sup>57</sup> 1.6.

A continuación se muestra una tabla que muestra el tiempo que demoró el Modelo Ontológico en llenarse dependiendo de la cantidad de manifiestos que se insertaron en la base de datos.

<sup>57</sup> Java Development Kit (JDK) <http://java.sun.com/javase/downloads/index.jsp>

Cantidad de Manifiestos en la base de datos	Tiempo de llenado del Modelo Ontológico (min)
60	0.35 – 0.37
118	1.12 – 1.15
177	2.5
295	4.44

**Tabla 11:** Rendimiento del proceso de llenado del Modelo Ontológico en dependencia de la cantidad de manifiestos.

El llenado del Modelo consiste en extraer de la Base de datos el modelo o sea extraer la ontología y se almacena como un fichero temporal, luego se extraen uno a uno los manifiestos, se parsean y se convierten en objetos que son insertados como instancias de la ontología.

Haciendo uso del NetBeans 6.5 se le realizaron pruebas a varias clases entre ellas a la clase LlenarOntología, SearchModel, XMLReaderSample, etc. Se les realizaron pruebas tales como Hot Spots, la cual consiste en medir el total de tiempo de ejecución y el número de invocaciones para cada método con independencia del contexto (una única cadena de llamadas a los métodos que conduzcan a la invocación del método), Call Tree muestra el árbol de Contexto que muestra el método de cadena de llamadas y el tiempo / número de invocaciones de la ejecución de los hilos y los métodos, a continuación se muestra la figura que contiene los resultados de estas pruebas realizadas a la clase LlenarOntología.

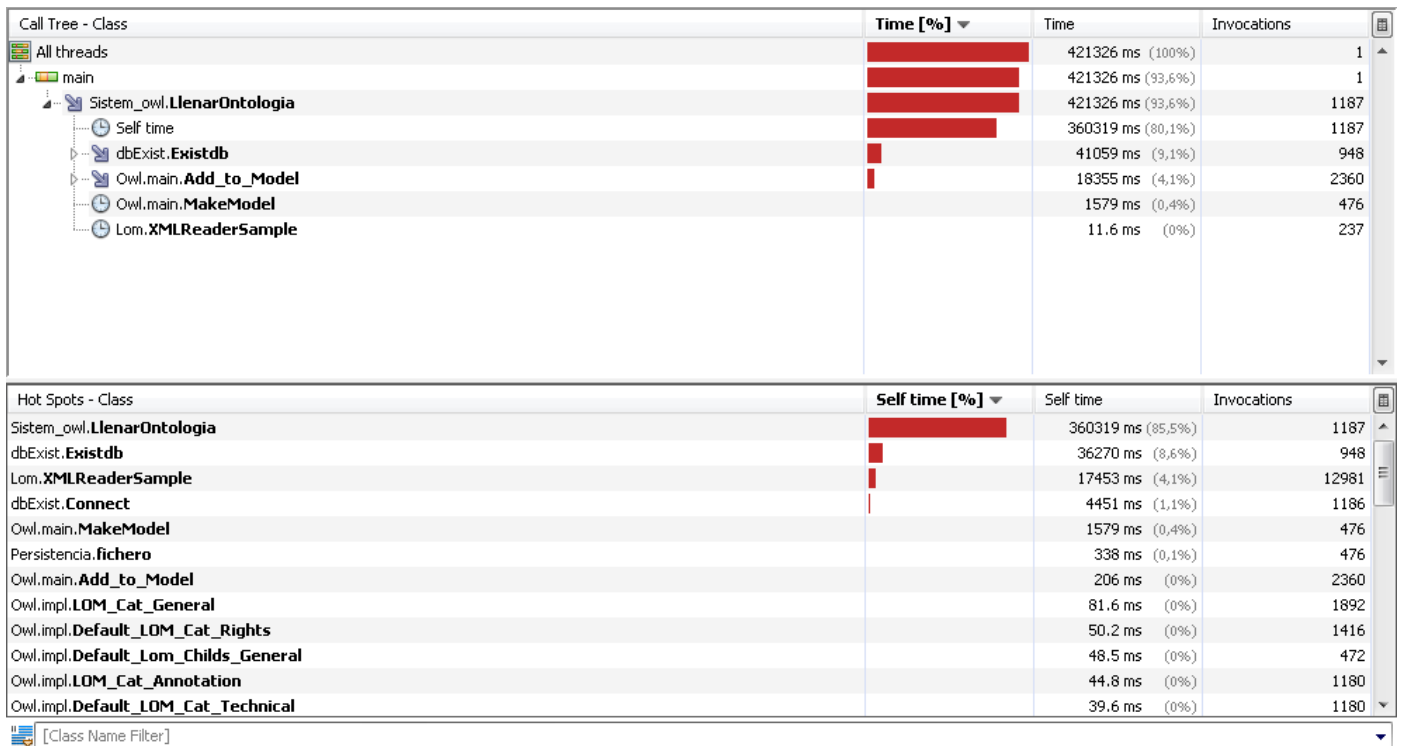


Figura 22: Pruebas de rendimiento Call Tree y Hot Spots realizadas a la clase LlenarOntología.

A continuación se muestra la información que brinda el NetBeans del resultado de las mismas:

Data collected from: lunes, 1 jun 2009, 11:19:21 PM

Snapshot taken at: lunes, 1 jun 2009, 11:27:14 PM

File:

C:\Usuarios\Maikel\WebSemantica\WorkSpace\InferMod\nbproject\private\profiler\snap

hot-1243891634625.nps

File Size: 8.608 B

Settings:



Settings Name: Analyze Performance

Profiling Type: CPU Profiling (Entire Application)

Root Methods: Sistem\_owl.LlenarOntologia.<all>()

CPU Profiling Type: Instrumentation

CPU Timer: Off

Exclude time spent in Thread.sleep() and Object.wait(): Yes

Limit number of profiled threads: 32

Instrumentation Filter: Profile only project classes

Instrumentation Scheme: Total

Instrument Method Invoke: Yes

Instrument New Threads: No

Instrument Getters and Setters: No

Instrument Empty Methods: No

Se aplicaron éstas pruebas de rendimiento, además de las pruebas de Telemetría a la máquina virtual y del uso de Memory Heap (memoria del montón) para el proyecto completo obteniéndose los siguientes resultados:

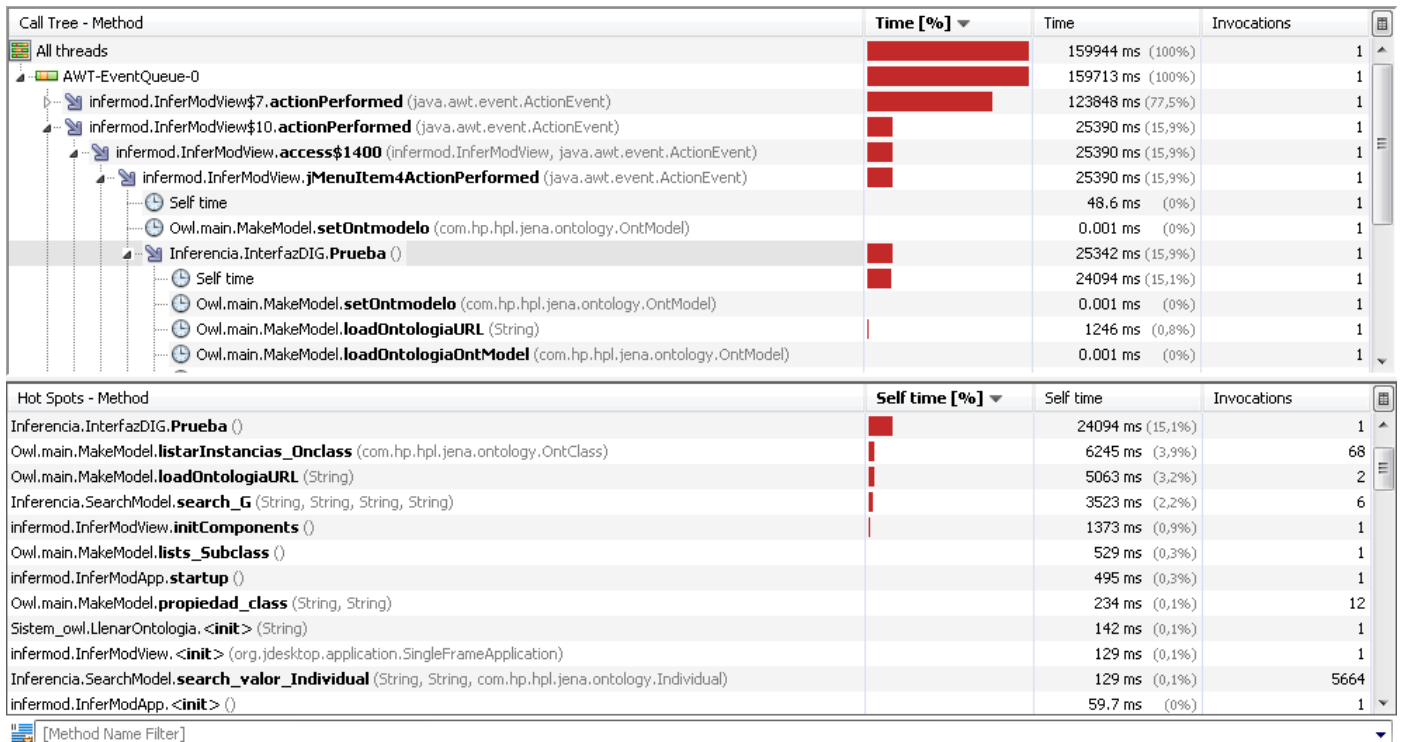


Figura 23: Pruebas Call Tree y Hot Spots realizadas al proyecto completo.

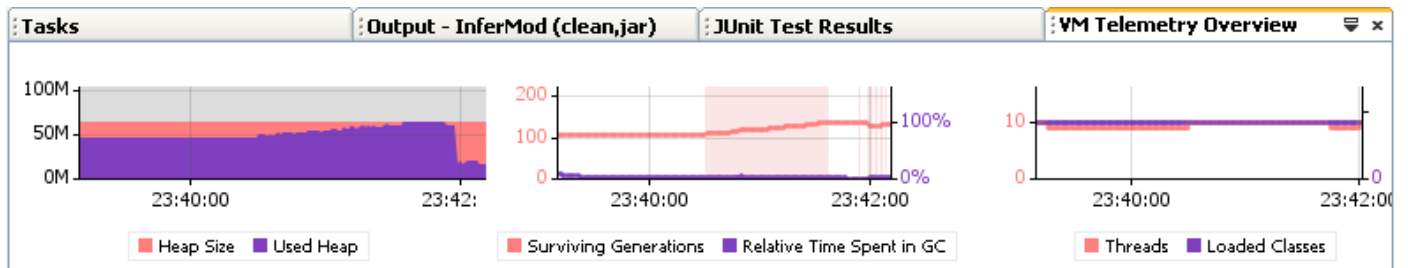
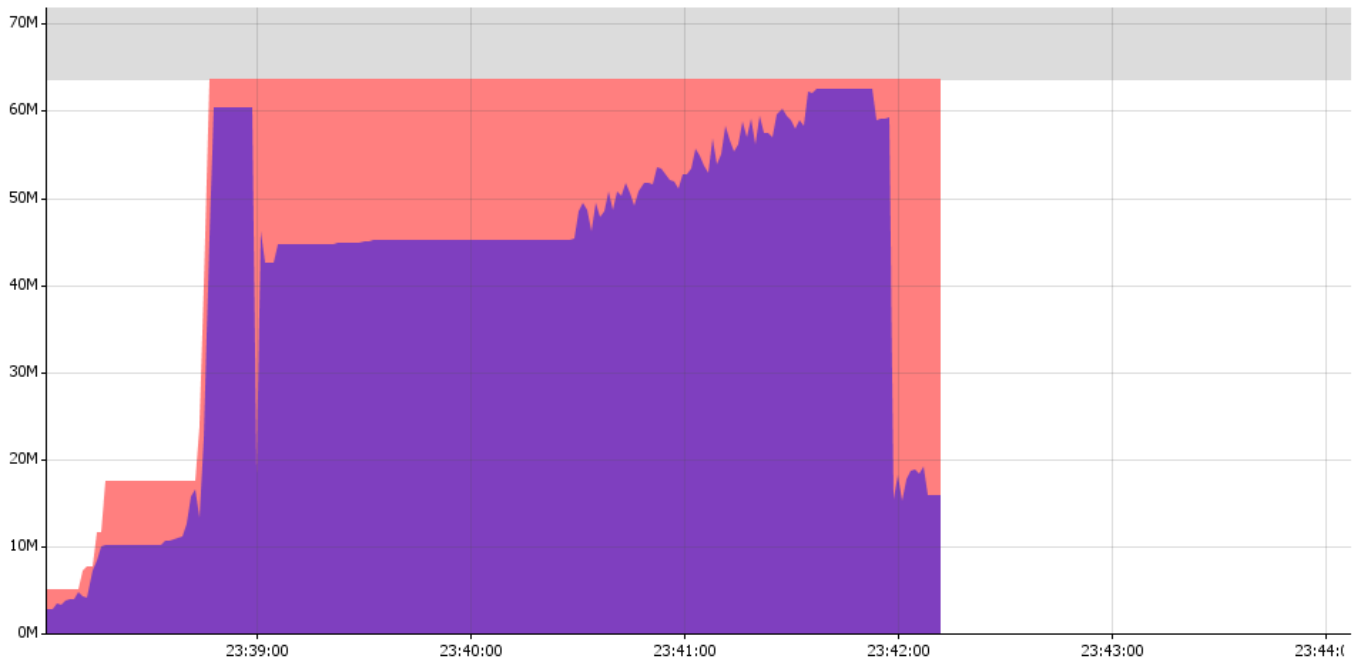


Figura 24: Pruebas de Telemetría a la Máquina Virtual.



**Figura 25:** Prueba Memory Heap realizada al proyecto completo.

En la figura 24, la cual es una ampliación de la grafica de la izquierda de la figura 23, se muestra en color rojo el tamaño de memoria del montón de la Java Virtual Machine(JVM) y la superposición de púrpura indica la cantidad de espacio de almacenamiento dinámico realmente en uso.

El gráfico en el centro de la figura 23 muestra dos importantes estadísticas Heap, porcentaje de tiempo de ejecución por la JVM, representado por la línea azul, y generaciones sobrevivientes (en línea roja) las cuales son el número de edades diferentes de todos los objetos en la pila de la JVM, donde la "edad" se define como el número de colecciones de basura de un objeto que ha sobrevivido; mientras que el gráfico de la derecha muestra el número de hilos activos en la JVM.

### 3.1.3 Pruebas a la Ontología.

Se realizaron pruebas locales a la Ontología haciendo uso del Protégé 3.4 y el razonador Pellet en su versión 2.0 rc 5 y el razonador en línea de Pellet que se encuentra en la siguiente dirección <http://www.mindswap.org/2003/pellet/demo.shtml>.

Con la utilización de Protégé 3.4 y Pellet 2.0 rc 5 se comprobó la taxonomía, la consistencia del modelo así como los tipos inferidos, obteniéndose como resultado los representados en las tablas siguientes:

Leyenda

TCC: Tiempo para construir consultas.

TERR: Tiempo para enviar y recibir desde el razonador.

TA: Tiempo de actualización de Protégé-OWL.

TT: Tiempo total.

TCR: Tiempo de consulta al razonador.

Procesos	Tiempo (Seg)	
	Protégé y Pellet	Mindswap
TCC	> 0.001	
TERR	0.031	
TA	0.031	
TT	0.078	0.054

**Tabla 12:** Validación de la consistencia de los conceptos.

Procesos	Tiempo (Seg)
TCC	> 0.016
TCR	0.203
TA	0.062

**Tabla 13:** Validación de la taxonomía.

Procesos	Tiempo (Seg)
TCC	> 0.001
TCR	0.047
TA	> 0.001

Tabla 14: Cálculo clases equivalentes.

Procesos	Tiempo (Seg)
TCC	> 0.001
TCR	0.031
TA	> 0.001
TT	0.047

Tabla 15: Cálculo de tipos inferidos.

El razonador Pellet en línea muestra los siguientes resultados:

OWL Species: Full

Consistent: Yes

Time 9672 ms (

Loading: 6435 ms

Species Validation: 1736 ms

Consistency: 54 ms

Classification: 864 ms

Realization: 582 ms)

## **Conclusiones Generales**

Luego de concluida toda la investigación que trajo consigo el desarrollo de un módulo de inferencia para un RS que permitió realizar los procesos de recuperación del conocimiento, tales como son: las inferencias y la gestión del conocimiento, se pueden arribar a las siguientes conclusiones:

- ✓ El análisis exhaustivo que se realizó acerca de las diferentes tendencias tecnologías actuales permitió que se escogieran las herramientas adecuadas para el modelado e implementación de la aplicación.
- ✓ Se creó un módulo de inferencia, que realiza procesos de gestión del conocimiento, el cual brinda respuestas al Repositorio de OA de la UCI mediante un bus de servicios.
- ✓ Se ganó experiencia en la utilización de las tecnologías de la Web Semántica para realizar procesos de gestión del conocimiento así como la creación de bases de conocimiento haciendo uso del lenguaje OWL y herramientas tales como: los editores ontológicos (Protege y Swoop), Apis de desarrollo(Jena y OWL-API) y razonadores (Pellet).
- ✓ Con la puesta práctica de la aplicación se logró una eficiente organización de la información de los paquetes mediante la ontología OntoLOM y sobre esta se pudieron realizar búsquedas eficientes y una buena gestión de la información.
- ✓ Se generó la documentación de ayuda para desarrolladores, en la cual se describen las funcionalidades de los paquetes y las clases que estos contienen.
- ✓ Se creó una metodología para la creación de ontologías, la cual se puso en práctica en la creación de las ontologías que se utilizaron en el módulo de inferencia.

## **Recomendaciones**

A partir de los resultados o beneficios que proporciona este trabajo de diploma, se proponen las siguientes recomendaciones:

- ✓ Que se realice la creación de otros módulos para darle más funcionalidades al RS, entre estos módulos se encuentra identificado el de PLN.
- ✓ Que se gestione la información acerca de los recursos.
- ✓ Que sea consultada la bibliografía de la investigación, por parte de todas aquellas personas que se apoyen en este trabajo, para que amplíen así la información obtenida en el mismo.
- ✓ Que se cree un modelo ontológico donde se modele la información que se necesita para la creación de un historial.
- ✓ Agregar que las búsquedas se realicen no sólo por paquetes sino también por los recursos de los paquetes.
- ✓ Que se brinden los servicios a través de una capa de Servicios Web Semánticos para un uso óptimo del sistema.
- ✓ Que la ontología se convierta a OWL-DL, para un proceso de inferencia que asimile una mayor variedad de reglas.
- ✓ Que todos los procesos que se realizan sobre la base de datos eXist sean mediante los servicios web que ésta presta.

## Referencias Bibliográficas

**Abián, Miguel Ángel. 2005.** El futuro de la Web. *javaHispano*. [En línea] 20 de 09 de 2005. [Citado el: 22 de 11 de 2008.] <http://www.javahispano.org/contenidos/es/el futuro de la web/>.

—. **2007.** LA WEB SEMÁNTICA: ADEMÁS DE METADATOS (ONTOLOGÍAS), LÓGICA Y CONFIANZA. *Web Semántica Hoy*. [En línea] 30 de 01 de 2007. [Citado el: 02 de 02 de 2009.] <http://www.wshoy.sidar.org/index.php?2007/01/30/37-la-web-semantica-metadatos-ontologias-logica-y-confianza>.

**ADL. 2002.** Emerging and Enabling Technologies for the design of Learning Object Repositories Report. *Advanced Distributed Learning*. [En línea] 16 de 9 de 2002. [Citado el: 13 de 02 de 2009.] <http://xml.coverpages.org/ADLRepositoryTIR.pdf>.

**Agm. 2006.** web semántica MINERIA DE LA WEB. *mineriaweb*. [En línea] 06 de 06 de 2006. [Citado el: 02 de 02 de 2009.] <http://cs.uns.edu.ar/~agm/mineriaweb/downloads/Outlines/clase15-outline.pdf>.

**AIFB.** OntoEdit. *The OTK Tool Repository*. [En línea] AIFB, University of Karlsruhe. [Citado el: 12 de 02 de 2009.] <http://www.ontoknowledge.org/tools/ontoedit.shtml>.

**Answers Corporation. 2009.** Application programming interface. *Answers.com*. [En línea] 22 de 03 de 2009. [Citado el: 26 de 03 de 2009.] <http://www.answers.com/topic/application-programming-interface>.

**Arano, Silvia. 2005.** Los tesauros y las ontologías en la Biblioteconomía y la Documentación. *hipertext.net*. [En línea] 2005. [Citado el: 02 de 02 de 2009.] <http://www.hiptertext.net/web/pag260.htm> . ISSN 1695-5498.

**Barceló, Mario, Alfonso, Guzmán Gerardo y Pérez, Alonso. 2006.** Revista Ingeniería Informática. *udec*. [En línea] 12 de 04 de 2006. [Citado el: 4 de 02 de 2009.] <http://www.inf.udec.cl/revista>.

**BERNERS-LEE, Tim y Fischetti, Mark. 1999.** *Weaving the Web*. San Francisco : Siglo Editores, 1999. 84-323-1040-9.

**BERNERS-LEE, Tim, HENDLER, J. y LASSILA, O. 2001.** Scientific American. *The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*". [En línea] 05 de 2001. [Citado el: 16 de 01 de 2009.] <http://www.sciam.com>.



**Bianchini, Adelaide. 2000.** Conceptos y definiciones de hipertexto. Depto. de Computación y Tecnología de la Información. [En línea] Universidad Simón Bolívar Caracas – Venezuela, 06 de 2000. [Citado el: 21 de 11 de 2008.] <http://www ldc.usb.ve/~abianc/hipertexto.html#Definiciones>.

**Bosch, Mela. 2000.** Los Sistemas Informáticos y el conocimiento. *Las personas y los contenidos en el medio digital*. [En línea] 2000. [Citado el: 9 de 02 de 2009.] [http://perio.unlp.edu.ar/~tecnologias/conoc\\_hereda/sibila/new\\_page\\_1.htm](http://perio.unlp.edu.ar/~tecnologias/conoc_hereda/sibila/new_page_1.htm).

**Cabrera Pupo, Katuska y Martínez Gamboa, Aylene. 2007.** Propuesta metodológica para la gestión del conocimiento basada en otología. Habana : UCI, 2007.

**Canals, Agustí. 2003.** Acto de presentación del libro Gestión del conocimiento. *"La gestión del conocimiento"*. [En línea] 2003. [Citado el: 29 de 01 de 2009.] <http://www.uoc.edu/dt/20251/index.html> .

**CANARIE. 2001.** White Paper for a Learning Object Repository . [En línea] 2001. [Citado el: 13 de 02 de 2009.] [http://oknl.edu.gov.on.ca/eng/pdf/1\\_3\\_13\\_1.pdf](http://oknl.edu.gov.on.ca/eng/pdf/1_3_13_1.pdf)..

**Casal, Manuel. 2008.** Introducción a Internet. [En línea] 2008. [Citado el: 19 de 11 de 2008.] <http://www.casdeiro.info/internet/def.htm>.

**Castells, Pablo.** La web semántica. [En línea] [Citado el: 10 de 12 de 2008.] <http://www.ii.uam.es/~castells/publications/castells-uclm03.pdf>..

**Castells, Pablo, Bravo, C. y Redondo, M. A. 2003.** La web semántica. *Journals*. [En línea] 2003. [Citado el: 10 de 02 de 2009.] <http://www.ii.uam.es/~castells> . ISBN: 84-8427-352-0.

**Centelles, Miquel. 2005.** Taxonomías para la categorización y la organización de la información en sitios web. *Hipertext.net*. [En línea] 2005. [Citado el: 09 de 02 de 2009.] <http://www.hipertext.net/web/pag264.htm> . ISSN 1695-5498.

**Cid, María Carabaño, y otros. 2007.** INTRODUCCIÓN A LOS METADATOS. *iEspaña*. [En línea] 26 de 04 de 2007. [Citado el: 01 de 02 de 2009.] <http://sistemasavanzadosderecuperaciondeinformacion.iespana.es/>.

**Coss, Juenlis. 2006.** "miMatrix: Web Semántica: La Web del futuro. (I Parte)", [En línea] 2006. [Citado el: 12 de 11 de 2008.] <http://mimatrix.blogspot.com/2006/03/web-semntica-la-web-del-futuro-i-parte.html>..

- Criado, Luis. 2008.** Situación de los buscadores semánticos a principios del 2008. *Web Semántica*. [En línea] 06 de 06 de 2008. [Citado el: 12 de 02 de 2009.] <http://lcriadof.blogspot.com/2008/06/situacin-de-los-buscadores-semnticos.html>. ISBN 7-488-343-251.
- DAML. 2006.** OWL-S 1.0 Release. *DAML Services*. [En línea] 16 de 04 de 2006. [Citado el: 27 de 02 de 2009.] <http://www.daml.org/services/owl-s/1.0/>.
- Daniel. 2004.** WWWtools for Education. *Learning Object Repositories*. [En línea] 2004. [Citado el: 13 de 02 de 2009.] <http://magazines.fasfind.com/wwwtools/m/1030.cfm>..
- Domingo, Dr. JB y Motta, Dr. E. 1997.** WebOnto de la Universidad Abierta. *AKT- Advanced Knowledge Technologies*. [En línea] Universidad Abierta, 1997. [Citado el: 12 de 02 de 2009.] <http://www.aktors.org/technologies/webonto/>.
- DPCBD. 2006.** Diez pasos para conquistar la Biblioteca Digital. [En línea] 2006. <http://biblioteca.unisabana.edu.co/>..
- Fernández, Gregorio. 2004 .** 3.8.6. Cláusulas de Horn. *Representación del conocimiento en sistemas inteligentes*. [En línea] 28 de 11 de 2004 . [Citado el: 18 de 02 de 2009.] <http://www.gsi.dit.upm.es/~gfer/ssii/rcsi/rcsisu39.html>.
- Figuroa, Liliana y Palavecino, Rosa. 2006.** Aproximación a la diferencia entre Gestión de la Información y la Gestión del Conocimiento. [En línea] 2006. [Citado el: 29 de 01 de 2009.] <http://www.cibersociedad.net/congres2006/gts/comunicacio.php?id=618>.
- FZI WIM and AIFB LS3. 2005 .** Welcome to KAON. *KAON - The Karlsruhe Ontology and Semantic Web tool suite*. [En línea] 05 de 10 de 2005 . [Citado el: 27 de 02 de 2009.] <http://kaon.semanticweb.org/>.
- Garrone, Marcelo. 2009.** OpenThesaurus-es - Tesoro en español. *OpenThesaurus-es Principal*. [En línea] 2009. [Citado el: 07 de 02 de 2009.] <http://openthes-es.berlios.de/>. ISO-8859-1.
- Google code.** Inicio Proyecto. *Swoop Editor Web Semántica Ontología*. [En línea] [Citado el: 11 de 02 de 2009.] <http://translate.google.com/cu/translate?hl=es&sl=en&u=http://code.google.com/p/swoop/&ei=Hh6TSb3cB4HwsAOy1t2xCw&sa=X&oi=translate&resnum=2&ct=result&prev=/search%3Fq%3DSwoop%26hl%3Des%26sa%3DG>.

- Gruber, Thomas R. 2002.** Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. [En línea] 24 de 06 de 2002. [Citado el: 09 de 02 de 2009.] <http://scholar.google.com/cu/scholar?q=Principles+for++the+Design+of+Ontologies+Used+for+Knowledge++Sharing+Technical+Report&hl=es&um=1&ie=UTF-8&oi=scholart>.
- Gutiérrez, J. J., y otros. 2005.** XQuery . *XQuery* . Sevilla : Escuela Técnica Superior de Ingeniería Informática Universidad de Sevilla , 02 de 2005. Vol. I. 41012.
- Henst, Christian Van Der. 2002.** E-learning. *maestros del web*. [En línea] 16 de 08 de 2002. [Citado el: 12 de 02 de 2009.] <http://www.maestrosdelweb.com/editorial/elearning/>.
- Herrera, Ernestina. 2003.** Glosario. *Clarín.com Periodismo en Internet*. [En línea] 2003. [Citado el: 26 de 03 de 2009.] <http://www.clarin.com/suplementos/informatica/htm/glosario.htm#basededatos>.
- Horrocks, Ian. 2000.** Announcement: OilEd Ontology Editor. *Cover Pages*. [En línea] 4 de 12 de 2000. [Citado el: 12 de 02 de 2009.] <http://xml.coverpages.org/oilEdANn20001204.html>.
- Hors, Arnaud Le y Sutor, Robert S. 2004.** Glosario. *W3C Recommendation*. [En línea] W3C, 07 de 04 de 2004. [Citado el: 01 de 04 de 2009.] <http://www.w3.org/2005/03/DOM3Core-es/glosario.html>.
- Innovapyme.** GLOSARIO DE TÉRMINOS RELACIONADOS CON LA GESTIÓN DEL CONOCIMIENTO. *Innovapyme*. [En línea] [Citado el: 26 de 03 de 2009.]
- Javier.** Lista de buscadores semánticos. *Ven a ver a Javier*. [En línea] [Citado el: 24 de 11 de 2008.] <http://www.javi.it/semantic.html>.
- JORUM+ Project. 2004.** The JISC Online Repository for [learning and teaching] Materials. [En línea] 2004. [Citado el: 13 de 02 de 2009.] [http://www.jorum.ac.uk/docs/Vol1\\_Fin.pdf](http://www.jorum.ac.uk/docs/Vol1_Fin.pdf).
- Lama, Manuel y Sánchez Vila, Eduardo M. 2008.** Análisis de técnicas de aprendizaje adaptativo con ontologías. *eduforge.org*. [En línea] 22 de 03 de 2008. [Citado el: 9 de 02 de 2009 .] [https://eduforge.org/docman/view.php/230/3231/E4.4\\_5\\_vf.pdf](https://eduforge.org/docman/view.php/230/3231/E4.4_5_vf.pdf).
- Lamarca, María Jesús. 2008.** Hipertexto: El nuevo concepto de documento en la cultura de la imagen. [En línea] 22 de 07 de 2008. [Citado el: 23 de 11 de 2008.] <http://www.hipertexto.info/documentos/tesauros.htm>.

**Leyva, David, y otros. 2006.** HERRAMIENTAS PARA LA CREACIÓN Y GESTIÓN DE OBJETOS DE APRENDIZAJE REUTILIZABLES. *Informática 2009*. [En línea] 06 de 2006. [Citado el: 22 de 01 de 2009.] [http://www.informaticahabana.com/evento\\_virtual/files/SWL08.pdf](http://www.informaticahabana.com/evento_virtual/files/SWL08.pdf).

**López, Clara. 2005.** Objetos de Aprendizaje. *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*. [En línea] Tesina doctoral, Universidad de Salamanca., 2005. [Citado el: 12 de 02 de 2009.] [http://www.biblioweb.dgsca.unam.mx/libros/repositorios/objetos\\_aprendizaje.htm](http://www.biblioweb.dgsca.unam.mx/libros/repositorios/objetos_aprendizaje.htm).

**Lozano, Adolfo. 2001.** Ontologías en la Web Semántica. *Ontologías en la Web Semántica*. [En línea] 19 de 12 de 2001. [Citado el: 09 de 02 de 2009.] <http://www.google.com/cu/search?hl=es&q=http%3A%2F%2Fwww.informandote.com%2FjornadasIngWEB%2Farticulos%2Fjiw02.pdf+&btnG=Buscar&meta=>.

**Marquez, Santiago. 2008.** Lenguajes. *lawebsemantica.com*. [En línea] 2008. [Citado el: 09 de 02 de 2009.] <http://www.lawebsemantica.com/contents/webSemantica/ontologias4.html>.

**Meier, Wolfgang M. 2000.** eXist. *Open Source Native XML Database*. [En línea] 2000. [Citado el: 26 de 02 de 2009.] <http://exist-db.org/>.

**Metadatos. 2007.** ¿Qué son los Metadatos? [En línea] 2007. [Citado el: 08 de 11 de 2008.] <http://antares.inegi.gob.mx/metadatos/metadat1.htm>.

**Motik, Boris.** Introduction. *KAON2 Ontology Management for the Semantic Web*. [En línea] [Citado el: 27 de 02 de 2009.] <http://kaon2.semanticweb.org/>.

**Ortuño Díaz, Pedro Manuel. 2003.** La hermoteca científica en línea. *Redalyc*. [En línea] 2003. [Citado el: 4 de 02 de 2009.] <http://redalyc.uaemex.mx/>. ISSN 1697-7904..

**Pagola, Gracia y Roy, Réjean. 1999.** La gestión del conocimiento y de la información textual en soporte electrónico. *El profesional de la Información*. [En línea] Revista especializada para los profesionales de la información de habla hispana, 06 de 1999. [Citado el: 10 de 02 de 2009.] [http://www.elprofesionaldelainformacion.com/contenidos/1999/julio/la\\_gestion\\_del\\_conocimiento\\_y\\_de\\_la\\_informacion\\_textual\\_en\\_soporte\\_electronico.html](http://www.elprofesionaldelainformacion.com/contenidos/1999/julio/la_gestion_del_conocimiento_y_de_la_informacion_textual_en_soporte_electronico.html). ISSN .

**PBCWS. 2001.** Para buscar concepto de la Web Semántica. [En línea] 2001. [Citado el: 24 de 11 de 2008.] <http://www.w3.org/2001/sw/>.

- PSI. 2008.** Glosario de Términos. *Portal seguridad en Internet / Sistema Nacional e-México*. [En línea] 2008. [Citado el: 21 de 11 de 2008.] [http://www.e-salud.gob.mx/wb2/eMex/eMex\\_Glosario\\_de\\_terminos\\_Seguridad?page=35](http://www.e-salud.gob.mx/wb2/eMex/eMex_Glosario_de_terminos_Seguridad?page=35).
- RAE. 2001.** DICCIONARIO DE LA LENGUA ESPAÑOLA - Vigésima segunda edición. *REAL ACADEMIA ESPAÑOLA*. [En línea] 2001. [Citado el: 18 de 01 de 2009.] [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=sem%C3%A1ntica](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=sem%C3%A1ntica).
- Red TTnet. 2005.** La formación sin distancia. *Estudio realizado por el Grupo de Estudio de e-Learning de la red TTnet*. [En línea] 2005. [Citado el: 22 de 11 de 2008.] [http://www.inem.es/otras/TTnet/pdfs/LIBRO\\_laformacionsindistancia.pdf](http://www.inem.es/otras/TTnet/pdfs/LIBRO_laformacionsindistancia.pdf).
- Reino, Alfredo. 2000.** Índice Tutorial XML. *Universidad de Las Palmas de Gran Canaria*. [En línea] 26 de 01 de 2000. [Citado el: 06 de 02 de 2009.] <http://www.ulpgc.es/otros/tutoriales/xml/>.
- Rojas Mesa, Yuniét. 2006.** De la gestión de información a la gestión del conocimiento. [En línea] 2006. [Citado el: 29 de 01 de 2009.] [http://bvs.sld.cu/revistas/aci/vol14\\_1\\_06/aci02106.htm](http://bvs.sld.cu/revistas/aci/vol14_1_06/aci02106.htm).
- Roldán, María del Mar y Aldana, Jose F. 2007.** DBOWL: Persistencia y Escalabilidad de Consultas y Razonamientos en la Web Semántica. *khaos.uma.es*. [En línea] 09 de 10 de 2007. [Citado el: 09 de 02 de 2009.] <http://khaos.uma.es/dbowl/caepia2007.pdf>.
- Ruiz, Francisco José. 2006.** Glosario de términos específicos de Internet. *Internet como recurso educativo*. [En línea] 19 de 02 de 2006. [Citado el: 11 de 03 de 2009.] <http://internetrecursoeducativo.blogia.com/2006/021901-glosario-de-terminos-especificos-de-internet..php>.
- Samper, José Javier. 2005.** ONTOLOGÍAS PARA SERVICIOS WEB SEMÁNTICOS DE INFORMACIÓN DE TRÁFICO: DESCRIPCIÓN Y HERRAMIENTAS DE EXPLOTACIÓN. Valencia : Universitat de València Servei de Publicacions, 2005. I.S.B.N.:84-370-6270-5.
- Sánchez, Luis y Fernández, Norberto. 2005.** La Web Semántica: fundamentos y breve "estado del arte". *Novática ATI Asociación de técnicos de Informática*. [En línea] 2005. [Citado el: 09 de 02 de 2009.] <http://www.ati.es/novatica/lecturas.html#2005>.

- Sarasa, Antonio.** Repositorios digitales educativos. *Congreso Internet en el Aula*. [En línea] [Citado el: 12 de 02 de 2009.] <http://www.congresointernetenelaula.es/virtual/?q=node/319&espacio=5>.
- Sardiñas Suárez, Yolanda y Calvo de la Paz, César Miguel. 2008.** Mecanismos Semiautomáticos y Automáticos para la Generación de Ontologías de Dominio. Habana : UCI, 2008.
- SDN. 2006.** Code Samples and Apps. *Sun Developer Network*. [En línea] 27 de 01 de 2006. [Citado el: 12 de 03 de 2009.] <http://java.sun.com/applets/>.
- Sheldon, Tom. 2001.** Linktionary.com. [En línea] 2001. [Citado el: 02 de 1 de 2009.] <http://www.linktionary.com/m/metadata.html..>
- Stamou, Giorgos.** Technical Group Charter. *FuzzyRuleML Fuzziness in RuleML*. [En línea] [Citado el: 26 de 02 de 2009.] <http://www.image.ntua.gr/FuzzyRuleML>.
- . **2005.** Una extensión de Fuzzy SWRL. *Microsoft Libra*. [En línea] 2005. [Citado el: 26 de 02 de 2009.] <http://www.google.com/cu/search?hl=es&q=fuzzy+swrl+tutorial&btnG=Buscar&meta=lr%3D>.
- Steinacker, A, Ghavam, A y Steinmetz, R. 2001.** Index of /~garcia/cursos/TEICOPIN. *Departamento de Sistemas de Computación de la Universidad Federal de Campina Grande*. [En línea] 01 de 2001. [Citado el: 1 de 01 de 2009.] <http://www.dsc.ufcg.edu.br/~garcia/cursos/TEICOPIN/metadataWE.pdf..>
- Stojanovic, Nenad y Handschuh, Siegfried. 2002.** A Framework for Knowledge Management on the Semantic Web. [En línea] 2002. [Citado el: 28 de 10 de 2008.] <http://www2002.org/CDROM/poster/130.pdf>.
- SWWW. 1996.** SURGIMIENTO WORLD WIDE WEB. *Departamento de física aplicada I. (E.U.I.T.I. e I.T.T.)*. [En línea] 1996. [Citado el: 27 de 10 de 2008.] <http://www.vc.ehu.es/wuagacaj/manual/web/origen.html..>
- Tamayo, Daymy y Querejeta, Andrey. 2005.** Plataforma para el desarrollo de e-learning en la Universidad de Holguín Oscar Lucero Moya. . *Plataforma para el desarrollo de e-learning en la Universidad de Holguín Oscar Lucero Moya*. . Holguín : Universidad de Holguín Oscar Lucero Moya, 2005.
- W3C.** Guía Breve de Web Semántica. *W3C World Wide Web Consortium Oficina Española*. [En línea] [Citado el: 12 de 02 de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/websemantica#>.
- . **2002.** Metadata Activity Statement. . *W3C Technology and Society domain*. [En línea] 23 de 08 de 2002. [Citado el: 1 de 02 de 2009.] <http://www.w3.org/Metadata/Activity.html..>

—. **2004**. OWL Web Ontology Language Use Cases and Requirements. *W3C Recommendation*. [En línea] 10 de 02 de 2004. [Citado el: 02 de 02 de 2009.] <http://www.w3.org/TR/2004/REC-webont-req-20040210/>.

**W3Cmember. 2004**. SWOOP - A Hypermedia-based Featherweight OWL Ontology Editor. *Mindswap Maryland Information and Network Dynamics Lab Semantic Web Agents Project*. [En línea] 2004. [Citado el: 11 de 02 de 2009.] <http://www.mindswap.org/2004/SWOOP/>. ICBM 38.99358387.

**Wiki3. 2007**. SPARQL. *SeRQL y SPARQL - Recuperación y organización de la información*. [En línea] 01 de 04 de 2007. [Citado el: 10 de 02 de 2009.] <http://serqlsparql.50webs.com/sparql.html>.

**Woodley, Mary S. 2005**. Glosario DCMI. *Dublin Core Metadata Initiative*. [En línea] 07 de 11 de 2005. [Citado el: 17 de 03 de 2009.] [http://74.125.93.132/translate\\_c?hl=es&sl=en&u=http://dublincore.org/documents/usageguide/glossary.shtml&prev=/search%3Fq%3Dhttp://es.dublincore.org%26hl%3Des%26sa%3DG&usg=ALkJrhh-SsksBRbOoR4oYnBiy2UXkApmow](http://74.125.93.132/translate_c?hl=es&sl=en&u=http://dublincore.org/documents/usageguide/glossary.shtml&prev=/search%3Fq%3Dhttp://es.dublincore.org%26hl%3Des%26sa%3DG&usg=ALkJrhh-SsksBRbOoR4oYnBiy2UXkApmow).

## Bibliografía

(AulaDiez ,2000-2008) © 2000-2008 AulaDiez español online, S.L.L. Definición de e-learning. Disponible en: <http://www.auladiez.com/didactica/e-learning-01.html> (Consultado el 07/11/08).

**Casal, Manuel. 2008.** Introducción a Internet. [En línea] 2008. [Citado el: 19 de 11 de 2008.] <http://www.casdeiro.info/internet/def.htm>.

**DPCBD. 2006.** Diez pasos para conquistar la Biblioteca Digital. [En línea] 2006. [http://biblioteca.unisabana.edu.co/..](http://biblioteca.unisabana.edu.co/)

(Invenia, 2006) <http://www.invenia.es/oai:dialnet.unirioja.es:ART0000070763> (Consultado 29/01/09).

**Gruber, Thomas R. 2002.** Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. [En línea] 24 de 06 de 2002. [Citado el: 09 de 02 de 2009.] <http://scholar.google.com/cu/scholar?q=Principles+for++the+Design+of+Ontologies+Used+for+Knowledge++Sharing+Technical+Report&hl=es&um=1&ie=UTF-8&oi=scholart>.

**Metadatos. 2007.** ¿Qué son los Metadatos? [En línea] 2007. [Citado el: 08 de 11 de 2008.] <http://antares.inegi.gob.mx/metadatos/metadat1.htm>.

(Mindswap, 2008) Jim Hendler and Jennifer Golbeck, MINDSWAP Directors. SWOOP - A Hypermedia-based Featherweight OWL Ontology Editor. Disponible en: <http://www.mindswap.org/2004/SWOOP/> . (Consultado 07/11/08).

(Prieto, 2003) Prieto, Isabel María; Revilla, Elena. Estilos de Gestión del Conocimiento. La velocidad de aprendizaje como factor clave. Available at: <http://ideas.repec.org/p/emp/wpaper/wpe03-05.html> [Accessed January 29, 2009].

(REdWS,2009)[http://www.redwebsemantica.es/semweb/home.jsp?content=/sew/viewTerm&\\_origin=%2Fview%2Finstance%2FPerson.jsp&\\_sew\\_instance\\_set=RedTem&\\_sew\\_instance=Tecnolog%EDas+de+Modelado%2C+Procesamiento+y+Gesti%F3n+de+Conocimiento&\\_sew\\_var\\_name=instance](http://www.redwebsemantica.es/semweb/home.jsp?content=/sew/viewTerm&_origin=%2Fview%2Finstance%2FPerson.jsp&_sew_instance_set=RedTem&_sew_instance=Tecnolog%EDas+de+Modelado%2C+Procesamiento+y+Gesti%F3n+de+Conocimiento&_sew_var_name=instance) (Consultado 29/01/09).



## Anexos

## Anexo 1

**OpenThesaurus-es - Tesoro en español**

OpenThesaurus-es es un Tesoro para la lengua española en Código Abierto.

>>Palabra:

Que la contenga

[>>Verificar sinonimia](#)

[>>Usuarios - TOP 15](#)

**Anuncios Google**

[OpenOffice IFilter](#)  
Search OpenOffice documents with Sharepoint and Index Server  
[www.ifiltershop.com](http://www.ifiltershop.com)

[www.elpre.com](#)  
Todo sobre el caballo de PRE Mas de 150 ejemplares a la venta  
[www.elpre.com](http://www.elpre.com)

**Estadística**

2009-02-07 21:44

Palabras: 21.682

Sinonimias: 8.046

Usuarios: 114

Cambios (últimos 7 días): 0

**Enlaces**

- [Lista de Correo OpenThesaurus-es](#)
- [Wikipedia / Wiktionary](#)
- [Sinónimos de la lengua española](#)
- [Diccionario de la lengua española](#)
- [Proyecto OpenThesaurus-es](#)
- [OpenThesaurus: Polaco, Alemán, Eslovaco, Noruego, Portugués](#)

**Noticias**

[>>Archivadas](#)

2007-11-01 Instructivo para instalar en Windows XP. Descargar [AQUI](#) (Por David Hernandez).

2007-09-29 Ya se puede especificar el tipo morfológico (sustantivo, adjetivo, adverbio, etc) de las sinonimias y en sus correspondientes acepciones.

2007-09-15 Actualización del sistema OpenThesaurus-es

**Descargas**

[>>detalles](#)

[OpenOffice.org-2.x-Thesaurus](#) (818 KB, 2009-02-07)

[OpenOffice.org-1.x-Thesaurus](#) (399 KB, 2009-02-07)

[Text-Thesaurus](#) (142 KB, 2009-02-07, para ser usada con [Ding](#))

[KWord-Thesaurus](#) (141 KB, 2009-02-07)

[Mysql Dump](#) (453 KB, 2009-02-07)

**SOURCE (subversion):**

```

* Acceso anónimo via SVN:
  svn checkout svn://svn.berlios.de/openthes-es/trunk
* Acceso anónimo via HTTP:
  svn checkout http://svn.berlios.de/svnroot/repos/openthes-es/trunk
* Navegar con HTTP
* Navegar con ViewCVS
* Navegar con WebSVN

```

Descargas | Dungeon Siege ... | Limpiar

Figura 26: Ejemplo de Tesoro. (Garrone, 2009)

## Anexo 2

**Buscador Actual**

**Resultados de la búsqueda:**

[Toda la magia de Budapest y Praga](#)  
... Suplementos Gran Premio Fórmula 1 en Budapest **para** las salidas del ... con Ferias y/o Congresos en **Praga** del 9 ... Más información de los **vuelos** ...

[LA VANGUARDIA DIGITAL - Praga, testigo de la historia europea](#)  
... Para emergencias el teléfono de la policía es el 150, el de las ambulancias el ... 46) y **Praga** tres días **por** semana. Los **vuelos** salen de Madrid (Tel ...

[Foros sobre Europa República Checa Praga inquietante](#)  
... solo decirte que me llamó la atención tu alias (aunque no me llamo Raula) y que me voy **mañana** mismo **para Praga** ... buscador de **vuelos** ...

[ofertas de espectáculos, viajes y hoteles al mejor precio](#)  
... autoridades que tienen tres copas gigantes **para** entregar a ... **mañana** creo que cogeremos el bus **mañana** ... En Atrápalo puedes también reservar **vuelos** ...

Figura 27: Resultados Obtenidos con un buscador normal. (W3C)

## Anexo 3

**Buscador Semántico**

**Resultados de la búsqueda:**

[viajaconnosotros.com - viajes a Praga](#)  
... todos los **vuelos** a **Praga** desde tu ciudad que saldrán **mañana por la mañana**, ordenados según su hora de salida ...

[viajes a Praga - vuelos disponibles](#)  
... lista de **vuelos**. Horarios de salida y llegada ...

[Ofertas especiales - vuelos a Praga](#)  
... ofertas especiales de **vuelos** a **Praga** ...

Figura 28: Resultados Obtenidos con un buscador semántico. (W3C)

## **Glosario de Términos**

1. **API:** interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. (Answers Corporation, 2009)
2. **Applet:** Un applet es un programa escrito en el lenguaje de programación Java que puede ser incluido en una página HTML, de la misma manera que una imagen se incluye en una página. Cuando se utiliza una tecnología Java es habilitado el navegador para ver una página que contiene un applet, el código del applet se transfiere a su sistema y ejecutado por el navegador de la Java Virtual Machine (JVM). (SDN, 2006)
3. **Base de datos:** Conjunto de datos organizados de modo tal que resulte fácil acceder a ellos, gestionarlos y actualizarlos. (Herrera, 2003)
4. **Buscador, motor de búsqueda:** Es un programa, ubicado en un sitio de Internet, que recibe un pedido de búsqueda, lo compara con las entradas de su base de datos y devuelve el resultado. (Herrera, 2003)
5. **e-Learning:** Es aquella modalidad de formación a distancia no presencial o semipresencial que utiliza una metodología específica basada en las nuevas tecnologías de la información y la comunicación. (Innovapyme)
6. **Entidad digital:** Es un documento de cualquier tipo, ya sea imagen, texto, video, audio etc, que ha sido digitalizado, codificado e integrado con metadatos para su uso y recuperación posterior. (DPCBD, 2006)
7. **Internet:** Red mundial de ordenadores interconectados basada en el protocolo TCP/IP. Ofrece distintos servicios: e-mail, foros, FTP, chat, compartir ficheros, videoconferencia, etc. (Ruiz, 2006)
8. **Intranet:** Red privada, local o no, que utiliza los mismos protocolos de Internet. (Ruiz, 2006)

9. **Interoperabilidad:** La posibilidad de que distintos tipos de ordenadores, redes, sistemas operativos, y aplicaciones trabajen juntos de forma eficaz, sin comunicación previa, de tal forma que puedan intercambiar información de manera útil y con sentido. (Woodley, 2005)
10. **Metadatos:** Información estructurada que describe, explica, localiza, o maneja información especializada acerca de un recurso de información que tenga algún sentido para las máquinas o los humanos. (DPCBD, 2006)
11. **Modelo:** Es la representación real de los datos obtenidos a partir de la información disponible. Ejemplos son el modelo de estructura y el modelo de estilo de que representan la estructura analítica y la información de estilo asociada a un documento. El modelo podría ser un árbol, o un grafo orientado, o cualquier otra. (Hors, y otros, 2004)
12. **Objetos de aprendizaje:** materiales digitales creados como pequeñas piezas de contenido o de información, con la finalidad de maximizar el número de situaciones educativas en que las que el recurso pueda ser utilizado. (DPCBD, 2006)
13. **Ontología:** Estructura jerárquica que define formalmente las relaciones semánticas de un conjunto de conceptos. Se usa para crear vocabularios controlados/estructurados para la recuperación o el intercambio de información. (Woodley, 2005)
14. **Repositorio:** Lugar físico donde se almacenan los documentos en forma digital. (DPCBD, 2006)
15. **Semántica:** Estudio del significado de los signos lingüísticos y de sus combinaciones desde el punto de vista sincrónico en la teoría lingüística generativa, componente de la gramática que interpreta la significación de los enunciados generados por la sintaxis y el léxico. (RAE, 2001)
16. **Schema (esquema):** Define un conjunto de obligaciones estructurales y de valores aplicables a documentos XML. Los esquemas pueden ser expresados en lenguajes de esquema, como XML. (Hors, y otros, 2004)
17. **Tesauro:** Un tesauro es un vocabulario controlado y estructurado formalmente, formado por términos que guardan entre sí relaciones semánticas y genéricas: de

equivalencia, jerárquicas y asociativas. Se trata de un instrumento de control terminológico que permite convertir el lenguaje natural de los documentos en un lenguaje controlado, ya que representa, de manera unívoca, el contenido de estos, con el fin de servir tanto para la indización, como para la recuperación de los documentos. (Lamarca, 2008)

18. **W3C (World Wide Web Consortium):** Consorcio fundado en 1994-para desarrollar estándares comunes para la Web. Inicialmente, el W3C estuvo vinculado al CERN (Centre Européen pour la Recherche Nucléaire, Centro Europeo de Investigación Nuclear), donde se originó la Web, y tuvo el apoyo de DARPAc (Defense Advanced Research Projects Agency, Agencia de Proyectos de Investigación Avanzada para la Defensa) y de la Comisión Europea. (PSI, 2008)
19. **Web Semántica:** Es una extensión de la Web actual que cuenta con información debidamente estructurada, lo que proporciona un significado bien definido. Mejora la forma en la que las máquinas y las personas trabajan en cooperación. (Samper, 2005)
20. **XML:** El lenguaje extensible de marcas es un metalenguaje que nació como forma restringida del SGML (Standard Generalized Markup Language). (Innovapyme)