



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja



ESCUELA DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

TEMA:

“Ontología para la recomendación de recursos educativos
almacenados en el Repositorio de Objetos de Aprendizaje
(ROA) DSpace”

*Memoria de Tesis previa a la obtención del
Título de Ingeniero en Sistemas
Informáticos y Computación.*

AUTOR: Manuel Alberto Cartuche Flores
DIRECTOR: Dunia Inés Jara Roa
CODIRECTOR: Ing. Segundo Benítez Raimundo

Loja – Ecuador

2009



CERTIFICACIÓN

Mgs. Inés Jara Roa

Ing. Segundo Benítez Hurtado

DIRECTORA DE TESIS

CERTIFICA:

Que el presente trabajo de investigación, previo a la obtención del título de INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN, ha sido dirigido, supervisado y revisado en todas sus partes, por lo tanto, cumple con los requisitos legales exigidos por la Universidad Técnica Particular de Loja, quedando autorizada su presentación.

Loja, febrero 2009

Inés Jara Roa

Ing. Segundo Benítez Hurtado



Cesión de derechos

Yo, **Manuel Alberto Cartuche Flores**, declaro ser autor del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la universidad".

Manuel Alberto Cartuche Flores



Autoría

El presente proyecto de tesis previa a la obtención de Título de Ingeniero en Sistemas Informáticos y Computación; sus conceptos, análisis, conclusiones y recomendaciones emitidas, es de absoluta responsabilidad del autor.

Se indica además que la información de otros autores empleada en este trabajo está debidamente especificada en fuentes de referencia y apartados bibliográficos.

Manuel Alberto Cartuche Flores



Agradecimiento

Agradezco a todas las personas que han contribuido con mi formación personal y profesional, mis primeros profesores: mis padres y hermanos, mis amigos y demás familiares, en especial a la Directora de Tesis Inés Jara, quien con su paciencia, disciplina y motivación, hizo posible que culmine el presente proyecto.

Manuel Alberto Cartuche Flores



Dedicatoria

Dedico la culminación del presente proyecto a:

A Dios, nuestro Padre celestial, quien me guía y protege todos los días de mi vida.

A mi padre Julio César Cartuche, y a mi madre María Augusta Flores Cabrera, quienes han sido fuente de inspiración, ejemplo y de apoyo constante a pesar de las dificultades del diario vivir. A mis hermanos y hermana quienes me han transmitido sus enseñanzas y valores.

A todos mis profesores por brindarme una puerta al conocimiento a través de sus experiencias, también a mis amigos y compañeros con quienes he tenido el gusto de compartir su tiempo.

Manuel Alberto Cartuche Flores



Tabla de contenidos

CERTIFICACIÓN.....	i
Cesión de derechos.....	ii
Autoría.....	iii
Agradecimiento.....	iv
Dedicatoria.....	v
RESUMEN.....	6
1. Generalidades de las ontologías.....	9
1.1. Introducción.....	9
1.2. Web Semántica.....	9
1.3. Papel de las ontologías en la Web Semántica.....	9
1.4. Ontologías.....	10
1.5. Clasificación de las ontologías.....	11
1.6. Lenguaje para la representación de ontologías.....	13
1.7. Metodologías de desarrollo de ontologías.....	14
1.7.1. Metodología Cyc.....	14
1.7.2. Metodología de Uschold y King.....	14
1.7.3. Metodología de Grüninger y Fox.....	15
1.7.4. Metodología Kactus.....	15
1.7.5. Metodología Methontology.....	15
1.7.6. Metodología On-to-Knowledge.....	16
1.8. Análisis comparativo de las metodologías investigadas.....	16
1.9. Metadatos.....	18
1.10. Estándares.....	18
1.10.1. DUBLIN CORE (DC).....	18
1.10.2. LEARNING OBJECT METADATA (LOM).....	20
Figura 1.2: Estándar LOM.....	21
1.10.3. INSTRUCTION MANAGEMENT SYSTEM (IMS).....	22
1.11. Comparaciones de metadatos entre los estándares: Dublin Core, LOM, e IMS.....	24



2.	Diseño de la ontología usando la metodología Methontology	27
2.1.	Introducción.....	27
2.2.	Fases de Methontology.....	27
2.2.1.	Especificación.....	27
2.2.2.	Conceptualización	30
2.2.2.3.	Glosario de términos ontología UTPL.....	29
2.2.2.4.	Glosario de términos ontología DSpace-LOM.....	30
2.2.2.5.	Definición de la taxonomía de la ontología UTPL	32
2.2.2.6.	Definición de la taxonomía de la ontología DSpace-LOM	34
2.2.2.7.	Diagramas de relaciones binarias de la ontología UTPL.....	35
2.2.2.8.	Diagramas de relaciones binarias de la ontología DSpace-LOM.....	35
2.2.2.9.	Diagrama de conceptos de la ontología UTPL.....	36
2.2.2.10.	Diagrama de conceptos de la ontología DSpace-LOM.....	37
2.2.2.11.	Detalle de los atributos de las instancias de la ontología UTPL	38
2.2.2.12.	Detalle de los atributos de las instancias de la ontología DSpace	38
2.2.2.13.	Detalle de relaciones binarias de la ontología UTPL.....	39
2.2.2.14.	Detalle de relaciones binarias de la ontología DSpace-LOM.....	39
2.2.2.15.	Definición de reglas de inferencia	39
2.2.2.16.	Definición de constantes	40
2.2.2.17.	Definición de axiomas formales para la ontología UTPL.....	40
2.2.2.18.	Definición de axiomas formales para la ontología DSpace-LOM	41
2.2.2.20.	Instanciamiento de las ontologías	42
2.2.2.2.1	Modelo lógico de ontología UTPL	42
2.2.2.2.2	Instanciación de modelo lógico de ontología DSpace-LOM.....	45
2.2.3.	Implementación de las ontologías	47
2.2.3.1.	Introducción.....	47
2.2.4.	Evaluación y validación.....	48
2.2.4.1.	Evaluación.....	48
2.3.	Validación por medio de métricas.....	51
2.3.1.	Validación a nivel estructural.....	53
2.3.1.1.	Métricas de cohesión	54



2.3.2.	Validación de la ontología a nivel funcional	55
2.3.2.1.	Lógica Descriptiva	56
2.3.2.2.	Lógica Funcional.....	56
2.3.3.	Validación de la ontología a nivel del perfil de usabilidad.....	57
3.	Búsqueda semántica	61
3.1.	Introducción.....	61
3.2.	Contexto actual.....	61
3.3.	Definición	63
3.4.	Modelo lógico del buscador semántico.....	64
3.5.	Diseño Arquitectónico	66
3.6.	Implementación del sistema recomendador	66
3.7.	Pruebas para componentes que hacen uso de la ontología.....	67
3.8.	Validación	75
	Discusión.....	78
	Conclusiones	80
	Recomendaciones	82
	Bibliografía.....	83
	ANEXOS	87
	ANEXO I.....	88
1.	Implementación de la Ontología.....	89
1.1.	Protégé	89
1.2.	Creación de la ontología	89
1.3.	Representación de los conceptos en Protégé	91
1.4.	Propiedades: Object, datatype y annotation	93
1.4.1.	Representación de las relaciones mediante propiedades Object	94
1.4.2.	Representación de propiedades datatype	95
1.5.	Creación de restricciones	96
1.5.1.	Restricciones con cuantificadores	96
1.5.2.	Restricciones de cardinalidad	98
1.5.3.	Clases Enumeradas.....	100
1.6.	Instanciamiento de ontologías.....	101



1.6.1.	Ontología UTPL.....	102
	UTPL: Área.....	104
1.6.2.	Ontología DSpace-LOM.....	105
	DSpace: General>>Estructura	107
	ANEXO II.....	108
1.	Arquitectura de la aplicación.....	109
1.1.	Capa lógica	109
1.1.2.	Añadir la librería Jena a un proyecto Java.....	113
1.2.	Capa de servicios Web	114
1.2.1.	Creación del servicio Web.....	114
1.3.	Capa interfaz de presentación	116
1.3.1.	Buscador	116
1.3.2.	Cliente EVA	117
	ANEXO III.....	1266
1.	Formato usado para las pruebas de aceptación de usuario.....	127
2.	Formato de anotaciones semánticas.....	127



RESUMEN

La Universidad Técnica Particular de Loja (UTPL) en sus dos modalidades de estudio, Presencial y Abierta y a Distancia, hace uso del Repositorio de Objetos de Aprendizaje (ROA) denominado DSpace que fue adaptado para que funcione bajo el estándar LOM (estándar de marcaje de materiales destinados a la educación).

Con el fin de que los Objetos de Aprendizaje (OA) almacenados en el ROA, ayuden en el proceso de aprendizaje nace la idea de realizar una ontología de tal manera que los profesionales en formación de la UTPL, cuentan con OA relacionados con el contenido de cada una de las asignaturas en las que está matriculado.

En una ontología desaparece el concepto de clase – objeto, se orienta al uso de un lenguaje natural, por consiguiente, une el lenguaje natural con los objetos abstraídos del mundo real. En una ontología se realizan dos tipos de operaciones: de unión e intersección. En una ontología se deben definir: clases, propiedades y acciones.

Al analizar el dominio de la ontología, surgió la necesidad de trabajar con dos ontologías, a las que denominamos: a) Ontología UTPL que abarca área, carreras, asignaturas y contenido, instanciándose manualmente la asignatura de Metodología de Estudio, por ser común a todas las carreras de Modalidad Abierta y a Distancia de la UTPL; y, b) Ontología DSpace-LOM, en dónde se encuentran almacenados los OA usados en la UTPL.

Para el diseño de la ontología se utilizó la metodología “Methontology” que abarca las siguientes fases: a) especificación, b) conceptualización, c) implementación, d) evaluación y validación. Methontology no propone criterios específicos para realizar la validación de una ontología, en este caso se utilizó métricas que permiten evaluar la ontología a nivel estructural (cohesión), funcional (precisión), y usabilidad (anotaciones semánticas).

Las ontologías fueron creadas y editadas en Protégé v3.2, para la comunicación con la ontología fue necesario trabajar con un framework semántico denominado Jena v2.4 (razonador y motor de inferencia), el lenguaje usado para el desarrollo del software que usa la ontología es Java v1.6 junto con el IDE Netbeans v6.1.

Con el objeto de realizar búsquedas más precisas y de explotar o consumir las ontologías desarrolladas, es necesario elaborar un nuevo componente que lo hemos denominado “SER” **S**earch of **E**ducational **R**esources (**B**úsqueda de **R**ecursos **E**ducativos), es un buscador en su primera versión que permite comparar cada una de las palabras ingresadas en la ontología DSpace-LOM.



SER, es una aplicación Web que se encuentra desarrollado en JSP mediante el IDE Netbeans, además fue necesario construir un servicio Web que permita comunicación entre el servidor y el cliente (en este caso EVA), el protocolo del servicio Web es SOAP v1.2 y las librerías usadas para su construcción fueron JAX-WS 2.1.

Dentro del EVA se desarrolló un módulo que permite presentar los objetos de aprendizaje que tiene relación con las asignaturas que toma un(a) estudiante, en este caso Metodología de Estudio.

Para la validación de todos los componentes software desarrollados, esto es: a) módulo del EVA, b) SER (aplicación Web independiente) se invitó los estudiantes de la carrera de informática, matriculados en la asignatura de Metodología de Estudio de la Modalidad Abierta y a Distancia, del período de estudios Octubre 2008 –Febrero 2009, a que sean partícipes en la evaluación del software que hace uso de la ontología, a este llamado respondieron ciento cincuenta personas quienes nos ayudaron a determinar las mejoras que se podían dar al trabajo concluido.



CAPITULO I



1. Generalidades de las ontologías

1.1. *Introducción*

En el presente capítulo se realiza una breve descripción de la Web Semántica y uno de sus componentes más importantes “las ontologías”, se presentan diferentes clasificaciones de acuerdo a algunos autores; así mismo, se mencionan las metodologías y los lenguajes existentes para su creación, y que han surgido como un conjunto de buenas prácticas para el desarrollo de ontologías.

1.2. *Web Semántica*

La Web constituye el mayor centro de almacenamiento de información, y se ha convertido en un medio por el cual las personas se comunican, los negocios se desarrollan, el conocimiento se difunde y comparte. Cada día el contenido de la Web se incrementa; sin embargo, la organización actual de la Web presenta algunos inconvenientes, siendo el principal la falta de precisión al momento de realizar búsquedas con los motores disponibles como Google, Yahoo, etc., este problema surge debido a que: “el significado del contenido de la web no es accesible por las máquinas” (Grigoris Antoniou, 2004.); es decir, la información es presentada en un formato que es accesible y entendible por los usuarios, mas no puede ser interpretada por las máquinas.

Debido a éste y otros inconvenientes que tiene consigo la Web actual, Tim Berners Lee (creador de la Web) plantea el proyecto de Web Semántica, denominado así por la característica de dotar de significado o sentido, a la información y recursos disponibles en torno a la Web. En la actualidad se han realizado varios esfuerzos para alcanzar el objetivo de la Web Semántica tales como: descripción de los datos -metadatos-, lenguaje base que sea interoperable -XML-, método o técnica que permita representar el conocimiento -ontologías-, y software que permita comunicar e interpretar la información disponible en la Web -agentes-. Todos estos elementos juegan un papel importante dentro de la Web Semántica.

1.3. *Papel de las ontologías en la Web Semántica*

Uno de los mecanismos mediante el cual la Web semántica pretende alcanzar sus objetivos es con el uso de ontologías, las que permiten a desarrolladores -o cualquier persona- especificar: conceptos, relaciones, propiedades, taxonomías, reglas de inferencias, en torno a un dominio de interés; es decir, a través de dichos elementos es posible inferir conocimiento y obtener razonamiento automático. Sin embargo, las ontologías no se convierten en el único elemento que interviene en la obtención de conocimiento dentro de toda la Web, pues aquí también se ponen de manifiesto elementos como la inteligencia artificial (agentes inteligentes), buscadores



semánticos, sitios Web semánticos acompañados de una interfaz de usuario adecuada, entre otros elementos.

Una de las principales ventajas que permiten esta sinergia de elementos es la disminución del costo de obtención de datos y la creación del conocimiento, y por otro lado el aumento del valor computacional en conjunto.

Según Mills (2008), en una encuesta con el fin de establecer mercados que se ven afectados por la ola de la Web Semántica, a 14 sectores de la industria y 150 casos de estudio, entre ellas el sector público, se evidencia que se está empezando a emplear metodologías y procesos que reflejen de alguna forma la transición de la Web 2.0 a la Web 3.0 o Web Semántica, y por consiguiente el uso inherente de ontologías, se están enfocando sobre:

- El uso de arquitecturas orientadas a los servicios semánticos (SSOA).
- Modelamiento de medio-ambientes IT.
- Administración de procesos.
- Interoperabilidad de información geoespacial.
- Análisis predictivo.
- Análisis gramático de documentos y extracción de entidades.
- Mapeo de redes biológicas y biomarcadores moleculares.
- Ontologías médicas unificadas.
- Soporte al cuidado clínico.
- Servicios de datos con base en el Internet.
- Navegación a través del conocimiento.
- Velocidad de ordenamiento.
- Redes anidadas.
- Aumento en el acceso educacional y entrenamiento, entre otras.

1.4. Ontologías

Las ontologías hacen su aparición como una rama dentro de la Filosofía, las cuales se ocupan de la naturaleza y organización de la realidad. Alrededor de los años 90's, aparecen en la Inteligencia Artificial para la representación del conocimiento, pero han logrado tener su mayor impulso a partir de la aparición de la Web Semántica propuesta por Berners Lee en el año 2001, para transformar la Web de un espacio de información a un espacio de conocimiento.

Las ontologías son una conceptualización de un dominio, es decir, un modelo abstracto de un área en particular, en el que, se describen conceptos y relaciones entre los mismos, por lo que son comúnmente vistas como sistemas de representación del conocimiento. El concepto más difundido es el Gruber (1993) quien hace referencia a las ontologías como: "Una ontología es



la especificación de una conceptualización”, es decir, un modelo conceptual donde se especifican los conceptos y sus relaciones.

Las ontologías son una taxonomía relacional de conceptos y un conjunto de reglas lógicas a las que se las denomina axiomas formales o reglas de inferencia.

Existen muchas ventajas que traen consigo la utilización de ontologías entre las que tenemos:

- Constituyen una herramienta para la adquisición de conocimiento.
- Permiten compartir y reutilizar el conocimiento.
- Crear una red de relaciones entre conceptos.

Las ontologías han sido utilizadas para la representación del conocimiento, permiten definir un dominio de la información no sólo entendible por el ser humano sino también dominios deducibles e interpretables por las máquinas, debido a que poseen un nivel más amplio para la descripción del vocabulario, en contraste con otros medios para la representación de conocimiento tales como: los tesauros.

Su característica fundamental es la capacidad de razonamiento con las que se encuentran ligadas, es decir, el uso de reglas dentro de la ontología permite realizar deducciones sobre conceptos e instancias que conforman la ontología, haciendo posible de esta forma, simular la inteligencia artificial que la Web Semántica pretende lograr.

Sin embargo, en otras investigaciones como (Hurtado, 2006) se mencionan algunas ventajas del por qué usar ontologías, tales como por ejemplo, 1) las ontologías permiten un desarrollo semántico a mayor profundidad, debido a los diferentes niveles jerárquicos que se presentan al momento de formalizar una ontología, sus relaciones y capacidad para formular reglas de inferencia, 2) facilitan la reutilización y normalización de la representación del conocimiento, y 3) potencian la capacidad de comunicación entre máquinas y seres humanos.

En cambio, (Happel Hans-Jörg) hace énfasis en que las ontologías son adecuadas para la evolución del conocimiento de un dominio, y para la especificación de requisitos, además de la validación y chequeo automatizado.

1.5. Clasificación de las ontologías

Existen diversas clasificaciones realizados por varios autores, tales como: Mizoguchi (BARCHINI Graciela, 2006), quien clasifica las ontologías según el grado de generalidad o nivel de dependencia de una tarea. Entre otras clasificaciones tenemos:



a) Ontologías de alto nivel, son aquellas que describen conceptos muy generales tales como: tiempo, espacio, evento, acción, etc. b) Ontologías de dominio, describen el vocabulario con respecto a un dominio genérico (por ejemplo: medicina, arquitectura, etc.), c) Ontologías de tarea, describen una tarea o actividad en específico, tales como: venta, diagnóstico. d) Ontologías de aplicación, aquí los conceptos dependen tanto de un dominio como de una tarea en particular.

En cambio la clasificación propuesta por Van Heist (BARCHINI Graciela, 2006), las divide en: a) Ontologías terminológicas, en donde se especifican los términos para la representación del conocimiento. b) Ontologías de información, en las cuales se detalla la estructura de almacenamiento de la base de datos. c) Ontologías de modelado del conocimiento, se ajustan al uso particular del conocimiento que se describe.

Parafraseando lo que señala Steve (1998), en cuanto a la clasificación de las ontologías y mencionado por Pérez, podemos decir que se dividen de acuerdo a:

El ámbito al que pertenecen:

- Generales, en donde se describen conceptos de forma general;
- De dominio, en el cual se especifica un vocabulario para un dominio; es decir, se tiene, términos, relaciones y reglas;
- Específicas, son ontologías especializadas, ya que describen conceptos de un área de conocimiento.

El tipo de agente al que va destinada:

- Lingüísticas, vinculados a aspectos gramáticos y semánticos;
- No lingüísticas, a diferencia de las lingüísticas que están destinadas para el entendimiento del ser humano, esta clase se orienta hacia los robots y agentes inteligentes;
- Mixtas, combinan características de las dos anteriores.

El grado de abstracción o nivel de razonamiento:

- Descriptivas, llamadas así debido a que permiten relaciones entre conceptos.
- Lógicas, a diferencia de las descriptivas permiten realizar inferencias lógicas, mediante el uso de reglas o axiomas.



1.6. *Lenguaje para la representación de ontologías*

Uno de los primeros lenguajes para la representación de ontologías fue SHOE, el cual es una pequeña extensión de HTML, que permite a los autores de páginas Web, colocar sus anotaciones dentro de ellas (capacidad lectura-máquina) (HELFIN J).

En un inicio se trabajaba con XML (eXtensible Markup Language) que tenía la característica de ser independiente de la plataforma, proporcionando una estructura sintáctica con la que los computadores pueden interpretar los datos.

Luego apareció XMLS (XML Schema) que permite la definición de gramáticas y etiquetas significativas para los documentos mediante namespaces (nombre de espacios), con XMLs la semántica es aparente para los humanos pero no para las máquinas.

DAML+OIL (DARPA Agent Markup Language y Ontologic Inference Layer, respectivamente), se combina las características de ambos lenguajes, para crear ontologías y marcar la información para que sea entendible por las máquinas (Jim, 2004).

Más tarde apareció RDF (Resource Description Framework) sucesor de DAML+OIL, que se basa en una tripleta (sujeto, predicado, objeto) (Wikipedia, Resource Description Framework), en donde:

- El sujeto, representa el recurso en sí que se describe.
- El predicado, muestra propiedades o características acerca del recurso.
- Los objetos son los valores que se asignan a las propiedades y mediante las cuales se establece una relación.

Actualmente, el lenguaje que predomina en el mundo de las ontologías es OWL (Ontology Web Language), que permite publicar y compartir datos mediante el uso de ontologías (OWL). OWL usa RDF, pero tiene un vocabulario más amplio para expresar relaciones complejas.

OWL posee tres variantes que son:

- OWL Lite, permite una clasificación jerárquica, y la agregación de restricciones simples. Una de sus limitantes es que permite únicamente una cardinalidad de 0 a 1.
- OWL DL, denominada así por su correspondencia con las descripciones lógicas, fue diseñado para proveer la máxima expresividad sin perder la disponibilidad de algoritmos de razonamiento, ni la completitud computacional.



- OWL Full, fue diseñada para proveer la máxima expresividad manteniendo la compatibilidad con RDF, también permite aumentar el significado del vocabulario definido inicialmente.

1.7. Metodologías de desarrollo de ontologías

Actualmente existe una gran cantidad de metodologías para la construcción de ontologías, las más usadas son:

1.7.1. Metodología Cyc

Nace como un proyecto de inteligencia artificial que busca la construcción de una ontología comprensible para habilitar el razonamiento humano.

Los pasos para la construcción de una ontología, usando esta metodología son:

- Extracción manual del conocimiento común (de diversas fuentes).
- Utilización de herramientas de procesamiento de lenguaje natural o aprendizaje natural para la adquisición de nuevo conocimiento en la ontología.

El proyecto Cyc surgió en el año de 1984, por parte de la Corporación de Tecnología en Computación y Microelectrónica. La base de conocimiento de Cyc es propietaria, aunque una pequeña versión de esta base fue liberada y está disponible como OpenCyc, la que busca definir un vocabulario común para el conocimiento automatizado (Wikipedia, Wikipedia-Cyc).

Actualmente Cyc cuenta con más de un millón de aserciones en su base de conocimiento, que han sido definidas por seres humanos mediante el lenguaje CycL (lenguaje de programación parecido a Lisp).

1.7.2. Metodología de Uschold y King

Permite la creación de ontologías en base a otras ya existentes. Se recomienda los siguientes pasos (Corcho, 2001):

- Identificar el propósito para la construcción de la ontología.
- Capturar los conceptos y sus relaciones.
- Codificar la ontología.
- Evaluar la ontología.
- Documentar la ontología.



1.7.3. Metodología de Grüninger y Fox

En esta metodología se proponen los siguientes pasos:

- Definición de los escenarios motivadores; es decir, identificación de las posibles aplicaciones en las que la ontología será usada.
- Formulación de preguntas en lenguaje natural, a las que se les denomina cuestiones de competencia, esto con la finalidad de determinar el ámbito de la ontología.
- Especificación de la terminología, en base a las preguntas realizadas en el paso anterior, se define conceptos principales, relaciones, propiedades, etc.
- Formalización de las interrogantes.
- Especificación de axiomas formales; es decir, la representación de un aspecto de la realidad.
- Verificación de la ontología.

Dentro de esta metodología las cuestiones de competencia hacen referencia a consultas a las cuales la ontología debería responder.

1.7.4. Metodología Kactus

Esta metodología centra la construcción de la ontología sobre una base de conocimiento, mediante un proceso de abstracción (Corcho, 2001), para ello hace uso de:

- La especificación del contexto de la aplicación y el punto de vista de modelado. Lo primero hace referencia a la descripción del dominio que tendrá la aplicación, así como también, los objetos de interés y tareas que realizará la ontología; mientras que el punto de vista de modelado se refiere a definir el modelado que se va a realizar: dinámico-estático, funcional-causal.
- Realizar un diseño preparatorio en base a una ontología existente, lo que implica realizar un estudio de ontologías que se hayan construido (mapeo).
- Refinamiento y estructuración de la ontología.
- Finalmente, la documentación y reutilización de la ontología.

1.7.5. Metodología Methontology

Es una metodología creada en el Laboratorio de Inteligencia Artificial de la Universidad Técnica de Madrid. La creación de la ontología puede empezar desde cero o en base a la reutilización de otras existentes. Methontology incluye la identificación del proceso de desarrollo de la ontología (calendario, control, aseguramiento de calidad, adquisición de conocimiento), un ciclo de vida basado en la evolución de prototipos, para lo cual se sigue



los pasos definidos en el estándar IEEE 1074 de desarrollo de software (Corcho, 2001), que son:

- Especificación.- Definir el alcance y granularidad de la ontología.
- Conceptualización.- Permite organizar y estructurar el conocimiento adquirido mediante tablas, lenguaje UML, jerarquías etc.
- Implementación.- Representa la formalización de la ontología; es decir, pasar la conceptualización de la ontología a un lenguaje como RDF, OWL, etc.
- Evaluación.- Comprobar el funcionamiento de la ontología.

1.7.6. Metodología On-to-Knowledge

Es un proyecto de la IST (Tecnologías de la Sociedad de la Información) (Fensel D. Van Hermelen F.) y (Fensel D, 1999), en el que se han desarrollado herramientas y métodos que soporten la administración de conocimiento, apoyado en una ontología compartible y usable. Esta metodología aplica ontologías a la información electrónica con la finalidad de mejorar la administración de conocimiento. Incluye los siguientes aspectos:

- Identificación de metas, las cuales deberán ser cumplidas por herramientas de gestión de conocimiento.
- Evaluación de la ontología a partir de casos de estudio.

1.8. *Análisis comparativo de las metodologías investigadas*

Todas las metodologías investigadas anteriormente, definen un proceso inicial en base a requisitos o necesidades por los cuales se construye la ontología, aunque con diferentes nombres o desde diversos puntos de vista. Es así que los diversos autores de las metodologías definen los pasos iniciales como: especificación, determinación del ámbito de la ontología, etc., refiriéndose a una formulación inicial del área que representará la ontología.

Las metodologías se diferencian principalmente en las fases a seguir para la construcción del modelo ontológico, en la granularidad que recae en los pasos posteriores e incluso en la utilización de herramientas de apoyo para la construcción de una ontología. En el siguiente cuadro se resume las características de las metodologías revisadas.



Metodologías/ Características	Cyc	Uschold y King	Grüninger y Fox	Kactus	Methontology	On-to Knowledge
Uso de herramientas de gestión de conocimiento						X
Basado en estándares					X	
Reutilización de ontologías		X		X	X	
Identificación de metas y propósito de la ontología		X	X			X
Identificar conceptos, relaciones y propiedades	X	X		X	X	
Documentación de la ontología				X	X	

Tabla 1.1: Comparación entre metodologías para la construcción de ontologías

La metodología utilizada en el desarrollo de la ontología para el ROA (Repositorio de Objetos de Aprendizaje) es: Methontology, debido a las siguientes características:

- Es la metodología más difundida para el desarrollo de ontologías.
- Provee un proceso de desarrollo basado en el ciclo de vida del software; es decir, se basa en el estándar IEEE 1074¹, el mismo que permite que el software evolucione de acuerdo a un ciclo de actividades tales como: análisis de requerimientos, diseño, implementación, pruebas, entrega y mantenimiento de software, mediante un proceso de construcción en espiral.
- Es una metodología que utiliza un proceso iterativo, permitiendo la entrega de software en pequeños incrementos que a su vez permite la mejora del mismo.
- Permite realizar tareas como: calendarios, control, aseguramiento de calidad, adquisición de conocimiento, etc.

Cabe mencionar que el desarrollo de una ontología no es precisamente similar al desarrollo de software Orientado a Objetos (OO), debido a que en aplicaciones orientadas a objetos los programadores se centran en los métodos que tiene una clase, es decir, en características operacionales, mientras que en una ontología las decisiones se basan en características

¹ "Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software" IEEE 1074
Obtenido de: <http://alarcos.infcr.uclm.es/doc/ISOFTWAREI/Tema03.pdf>



estructurales, esto es la definición y especificación correcta de un dominio o área de conocimiento.

Otro aspecto importante para el desarrollo de la presente tesis es la comprensión de los denominados metadatos que permiten describir cada uno de los recursos provenientes del (ROA), estos metadatos a su vez poseen diversos estándares para su manejo, y que se describen a continuación.

1.9. Metadatos

Son muchas las definiciones acerca de éste término, entre ellas la definición de “datos sobre datos” (Metadato), es decir, los metadatos son datos que describen a otros datos; son datos estructurados que pueden describir colecciones de objetos, así como también, las relaciones entre los mismos, describen el contenido, la calidad, la condición de los datos etc. A nivel mundial se llevan proyectos para la construcción de repositorios de materiales educativos, los cuales permiten que las aplicaciones actuales y futuras puedan extraer y transportar contenido, con el fin de reutilizarlos y modificarlos, siendo de gran aplicabilidad en el campo del e-learning, éstos se han convertido en un soporte para el intercambio de contenidos entre sistemas, es así que los metadatos juegan un rol muy importante ya que son un mecanismo para clasificar, describir y, catalogar los recursos que se encuentran en la World Wide Web, todo esto con la finalidad de poder realizar una búsqueda y recuperar la información de manera óptima.

Los metadatos determinan una estructura sintáctica, en cambio, las ontologías permiten una estructuración semántica en base a los metadatos.

1.10. Estándares

En la actualidad existen varios estándares para manejar los metadatos entre los más destacados se encuentran Dublin Core, LOM, IMS.

1.10.1. DUBLIN CORE (DC)

Es un modelo de metadatos fomentado y auspiciado por la Dublin Core Metadata Initiative (DCMI), con la finalidad de promover un vocabulario extenso y familiarizado. DC, es un sistema de quince descriptores, diseñado específicamente para proporcionar un vocabulario de características "base", capaces de otorgar la información descriptiva básica sobre cualquier recurso bibliográfico, sin que importe el formato de origen, el área de especialización, o el origen cultural.

Mediante DC se puede establecer 2 niveles: simple y calificado (Dublin Core Metadata Initiative-Using Dublin Core - Dublin Core Qualifiers, 2005), el primero constituye 15 elementos (véase la Figura 1.1), mientras que en el segundo se añaden 3 elementos al nivel anterior.

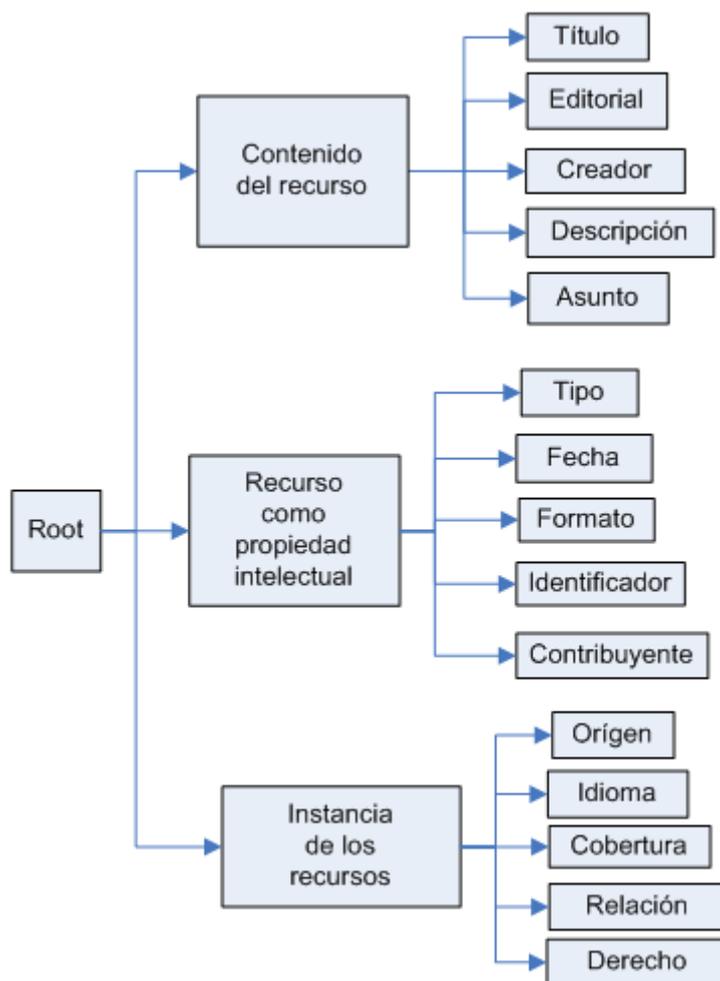


Figura 1.1: Representación del nivel simple de DC

La clasificación simple incluye tres divisiones las cuales son:

Contenido del recurso

- Título.
- Editorial.
- Creador.
- Descripción.
- Asunto.

Recurso visto como una propiedad intelectual

- Tipo.
- Fecha.
- Formato.
- Identificador.
- Contribuyente.

Instancias de los recursos

- Origen.



- Idioma.
- Cobertura.
- Relación.
- Derecho.

Y los tres elementos añadidos al nivel calificado son:

- Audiencia.
- Procedencia.
- Derechos de titular.

La DCMI establece dos tipos de calificadores: los de refinación de elementos y el esquema de codificación. Mediante los primeros se logra obtener un significado más específico de los elementos, mientras que el esquema de codificación permite identificar esquemas mediante los cuales se puede interpretar los valores de los elementos; estos dos calificadores actualmente han sido incluidos en la especificación DCMI Metadata Terms (Dublin Core Metadata Initiative-DCMI Metadata Terms, 2008). Dublin Core es el estándar más extendido a nivel internacional, debido a que permite adaptarlo según las necesidades de una aplicación concreta, y además describir los datos a través de RDF.

1.10.2. LEARNING OBJECT METADATA (LOM)

Learning Object Metadata (Metadatos de Objetos de Aprendizaje), es un modelo de datos que se encuentra generalmente codificado en XML, usado principalmente para el aprendizaje en línea.

Sus características principales son la reutilización de los objetos de aprendizaje, y facilitar la interoperabilidad. En el esquema base LOM v1.0, los elementos se agrupan en nueve categorías (Wikipedia, Wikipedia-Learning object metadata):

- General.
- Ciclo de Vida.
- Metadatos.
- Técnica.
- Educativa.
- Derechos.
- Relaciones.
- Anotaciones.
- Clasificación

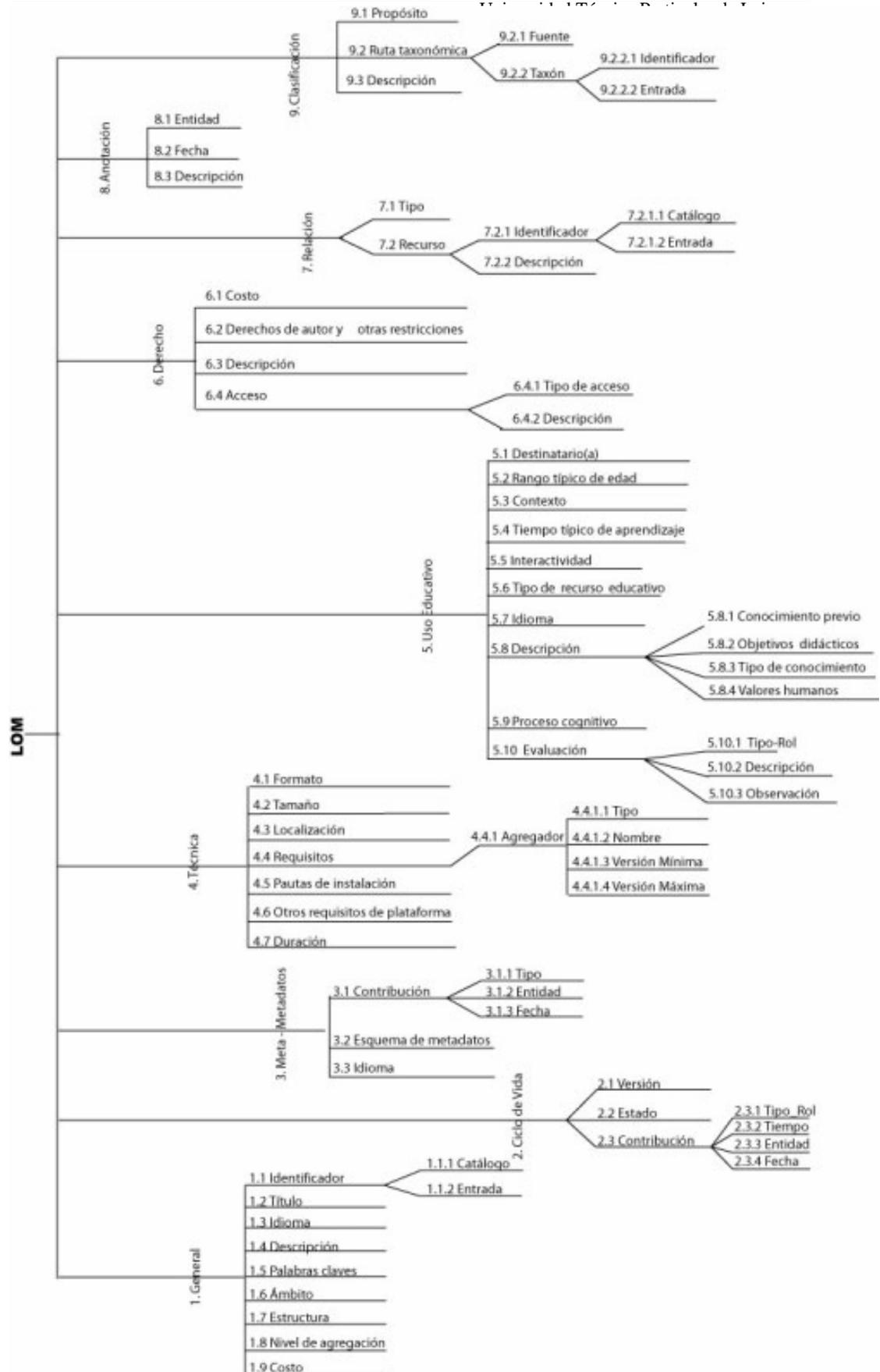


Figura 1.2: Estándar LOM



1.10.3. INSTRUCTION MANAGEMENT SYSTEM (IMS)

El IMS Global Learning Consortium, conocido usualmente como IMS tiene como objetivo crear un conjunto de descripciones para la educación e investigación científica. Su organización se la puede representar mediante un árbol (Figura 1.3). El IMS tiene ciertas similitudes con el modelo LOM en cuanto a categorías.

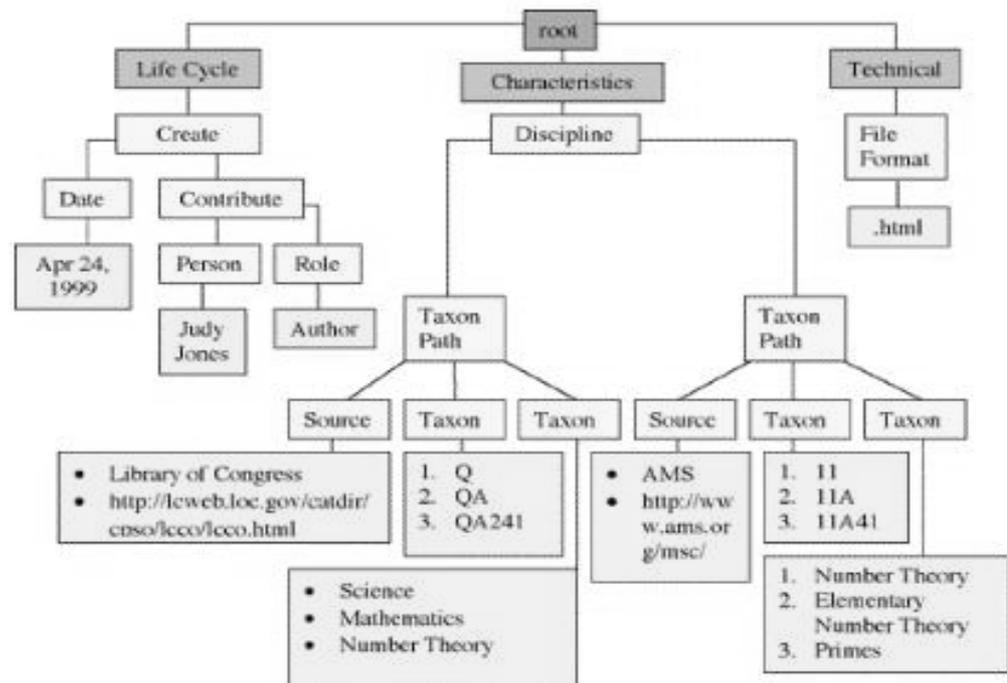


Figura 1.3: Árbol de metadatos de IMS [Robby Robson R.]

La raíz del árbol representa al objeto que se va a describir, de él se desprenden categorías, las cuales constituyen una propiedad en particular del mismo, mientras más se adentre en el árbol mayor será la especialización del objeto descrito. Cada nodo representa un único camino mediante el cual se puede tener una interpretación de una pieza de información.

Generalizando los descriptores dentro de IMS se encuentran catalogados en nueve categorías las cuales son:

- Anotaciones.
- Características.
- Uso dependiente educativo.
- Ciclo de vida.
- General.
- Meta-Meta-Data.
- Relaciones.



- Gestión de derechos.
- Técnica.

El modelo IMS posee varias divisiones tales como:

IMS-MD (MetaData).- Realiza una prestación más óptima de los recursos de aprendizaje para un determinado aprendiz en una situación dada, mediante la descripción de cada uno de los recursos de aprendizaje.

IMS-RDCEO (Reusable Definition of Competence or Educational Objective). - Provee referencias a descripciones de competencias y objetivos para ser incluidos en los medios de información; es decir, establece acuerdos comunes acerca de aptitudes que deben ser incluidas en un plan de aprendizaje o de carrera.

IMS-LD (Learning Design).- Permite la combinación de recursos educativos con las actividades pedagógicas, además de la interacción de personas con diferentes roles, (resolviendo de esta manera uno de los problemas que se presentaban en UoL, término acuñado por la Open University of the Netherlands, para definir tres elementos en el aprendizaje: recursos educativos, múltiples personas actuando en varios roles, y actividades pedagógicas), y finalmente ofrece soporte para múltiples alumnos, contemplando la comunicación entre ellos.

IMS-LIP (Learning Information Package).- Almacena información acerca de los estudiantes, incluyendo sus progresos, fechas en las que han realizado peticiones, etc. con la finalidad de transferir esta información entre aplicaciones de software.

IMS-CP (Content Packaging).- Provee la funcionalidad para la describir y empaquetar materiales de aprendizaje, ya sea como cursos individuales o como una colección de cursos, con la finalidad de que los paquetes sean interoperables y distribuibles.



1.11. Comparaciones de metadatos entre los estándares: Dublin Core, LOM, e IMS

Metadato	Definición	Dublin Core		LOM		IMS	
		Nivel	Elem. Correspondencia	Nivel	Elem. Correspondencia	Nivel	Elem. Correspondencia
Título	Un nombre dado al Recurso	1	Title	2	General.title	2	General.Title
Creador o Autor	Una entidad responsable del recurso (persona, organización o servicio)	1	Creator	3	Life cycle.contribute.entity	5	Life cycle.create.contribute.role. Author
Asunto	El tema del recurso	1	Subject	2	General.keywords	2	General.keywords
Descripción	Descripción del recurso	1	Description	2	General.description	2	General.description
Fecha	Fecha con la que se registra el recurso	1	Date	3	Lifecycle.contribute.date	4	Life cycle.create.date
Tipo	La naturaleza o género del recurso	1	Type	2	Educational.learningresourcetype	2	Educational.learningresourcetype
Origen	Origen con el cual se relaciona el recurso	1	Source	2	Relational.resource	2	Relational.resource
Idioma	Derechos sobre el recurso	1	Language	2	General.language	2	General.language
Derecho	Idioma del recurso	1	Rights	2	Rights.description	2	Rights.description
Relaciones	Relaciones entre recursos	1	Relation	2	Relation.kind	2	Relation.kind

Tabla 1.2: Comparación entre estándares (DC, LOM e IMS)



Los estándares de metadatos anteriormente comparados, fueron diseñados para un dominio en específico, es así que el estándar DC fue creado para describir a través de metadatos recursos bibliográficos, el estándar LOM se orienta a recursos educativos y el IMS es un estándar para la definición de requisitos educativos, así como también para el desarrollo de productos y servicios relacionados con la educación.



CAPITULO II



2. Diseño de la ontología usando la metodología Methontology

2.1. Introducción

El desarrollo de ontologías a través de la metodología Methontology implica una serie de pasos y actividades que deben realizarse, las mismas que permiten representar en detalle todos los componentes y conceptos que conforman el dominio que se desea modelar.

En el presente capítulo, se describe las ontologías que fueron creadas para las recomendaciones de objetos de aprendizaje provenientes del repositorio DSpace, a través de los pasos que recomienda la metodología denominada Methontology.

2.2. Fases de Methontology

2.2.1. Especificación

2.2.1.1. Alcance

El desarrollo de la ontología servirá de base semántica para la extracción de información proveniente del repositorio DSpace hacia una capa superior de presentación; es decir, una página personalizable, en donde el usuario puede realizar búsquedas de recursos digitales disponibles en la Universidad Técnica Particular de Loja (UTPL).

2.2.1.2. Objetivos generales

- Desarrollar la ontología que se utilizará como base semántica para la búsqueda de recursos digitales almacenados en el ROA en la UTPL.
- Presentar recursos digitales, teniendo como fuente de información el repositorio DSpace.

2.2.1.3. Objetivos específicos

- Investigar y seleccionar la metodología a seguir para el desarrollo e implementación de la ontología aplicada al ROA.
- Definir el modelo conceptual ontológico.
- Definir el modelo interno de la ontología; es decir, la fuente de información de la que se extrae los datos.
- Formalización e implementación de la ontología mediante la herramienta Protégé.
- Conectar el repositorio y la ontología para la extracción de datos, y su posterior envío a la capa de presentación.



2.2.1.4. Definición de requisitos

Dado el dominio de la aplicación, se ha creído conveniente definir dos subdominios, por consiguiente el desarrollo de dos ontologías. La primera ontología, permitirá representar el dominio que baraca las modalidades de estudio que se brinda en la universidad, mientras que en la segunda, se modelará el dominio del repositorio DSpace.

2.2.1.4.1. Requisitos funcionales

Nombre de requisito	REQ001: Extracción de información.
Descripción	Su función es la extracción de recursos digitales de cualquier tipo (.doc, .pdf, .html, videos, etc.) desde el repositorio DSpace.
Entradas	Las entradas para este requerimiento la constituyen los criterios de búsquedas realizadas por los usuarios, sean estos internos (estudiante/profesores, de acuerdo a las materias que cursan o dictan) o externos.

Tabla 2.1: Requisito REQ001

Nombre de requisito	REQ002: Presentación de OA
Descripción	Las recomendaciones son OA que se presentarán a los usuarios de acuerdo a: <ul style="list-style-type: none">● Materias (estudiantes UTPL); es decir, tomando únicamente el concepto de "materias", teniendo como base el criterio de "materias comunes" entre carreras (libre configuración).● Carreras-Materias (estudiante UTPL), relación existente entre las diversas materias, pertenecientes a una carrera.



Entradas	<ul style="list-style-type: none">● Materias (caso de estudiantes de la UTPL)● Preferencias de búsqueda de usuarios
-----------------	--

Tabla 2.2: Requisito REQ002

Nombre de requisito	REQ003: Comunicación intérprete–interfaz de usuario.
Descripción	La comunicación intérprete – interfaz de usuario se realizará a través de servicios web, mediante el protocolo SOAP, manteniendo la arquitectura SOA (Arquitectura Orientada a Servicios), planteada en la capa de presentación.
Entradas	La entrada para éste requisito lo constituyen los datos provenientes de la ontología, como resultado de: enlaces de recomendaciones o de búsqueda de los usuarios. Esto con la finalidad de construir la capa de servicios web, y brindar comunicación a la capa de presentación.

Tabla 2.3: Requisito REQ003

2.2.1.4.2. *Requisitos no funcionales*

○ Tiempo de respuesta

El tiempo de respuesta de búsqueda ya sea por: criterios de usuarios, o perfiles académicos (estudiantes UTPL), deberá fluctuar entre 1 seg y 5 seg.

○ Lenguajes y herramientas

Se utilizará el lenguaje Java v1.6 para la creación de clases que permitan extraer la información contenida en los archivos OWL (ontologías).

Se utilizará la herramienta Protégé versión 3.3.2 para el modelado y formalización de la ontología.

El lenguaje de representación de ontologías será OWL.



- Disponibilidad
Concepto 24/7; es decir, que la aplicación sea accesible en cualquier instante.

- Fiabilidad
La fiabilidad de la ontología dependerá de la fiabilidad de las reglas de inferencia, así como de los demás componentes del sistema.

2.2.2. Conceptualización

Dentro de la Unidad de Virtualización de la UTPL, se desarrollaron dos proyectos de ontologías en conjunto, la primera de ellas destinada a representar el dominio de bibliotecas virtuales que posee la universidad para los estudiantes, mientras que la presente tesis toma como fuente de información el repositorio DSpace para la presentación de OA relacionados con el diseño instruccional de la asignatura Metodología de estudio.

Para ambos equipos fue conveniente definir un marco común de trabajo, es decir, la selección y desarrollo de una metodología que sirva de guía para el proceso de creación de ontologías. Resultado de esto se definió una ontología denominada UTPL, la misma que representa parte de la oferta académica de la universidad. Luego de esto se desarrollaron las soluciones independientemente, de esta forma se concibe un dominio adicional denominado DSpace para la representación de OA.

2.2.2.1. Modelo conceptual de la ontología UTPL

El siguiente gráfico, muestra el modelo conceptual de la primera ontología, denominada ontología UTPL.

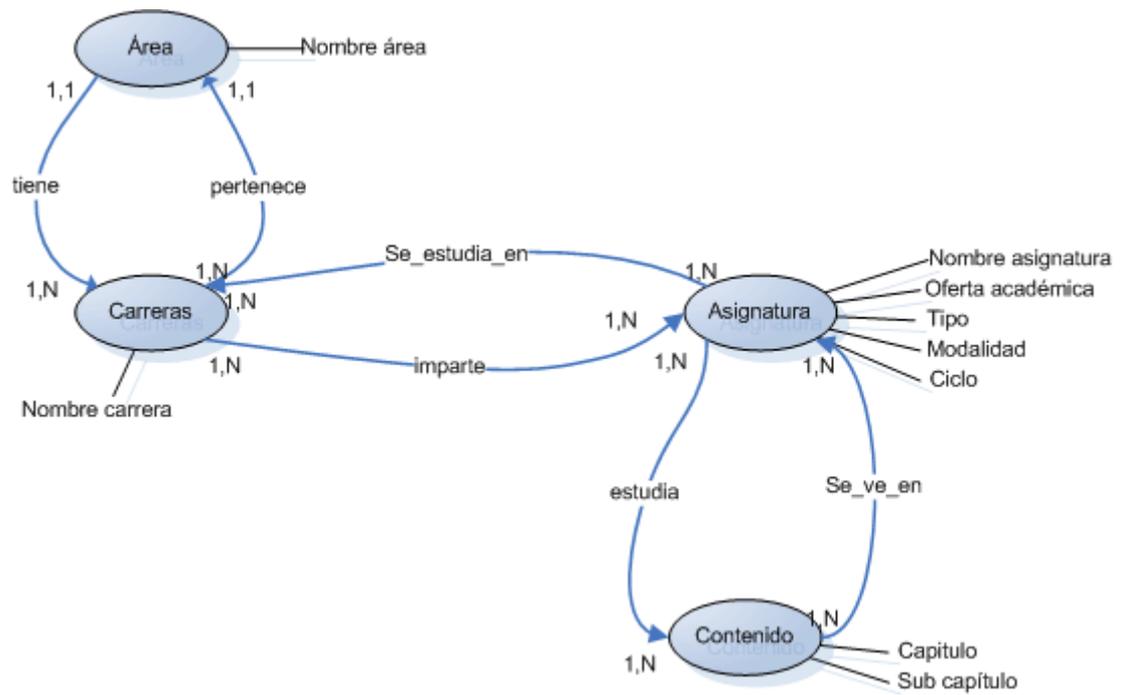
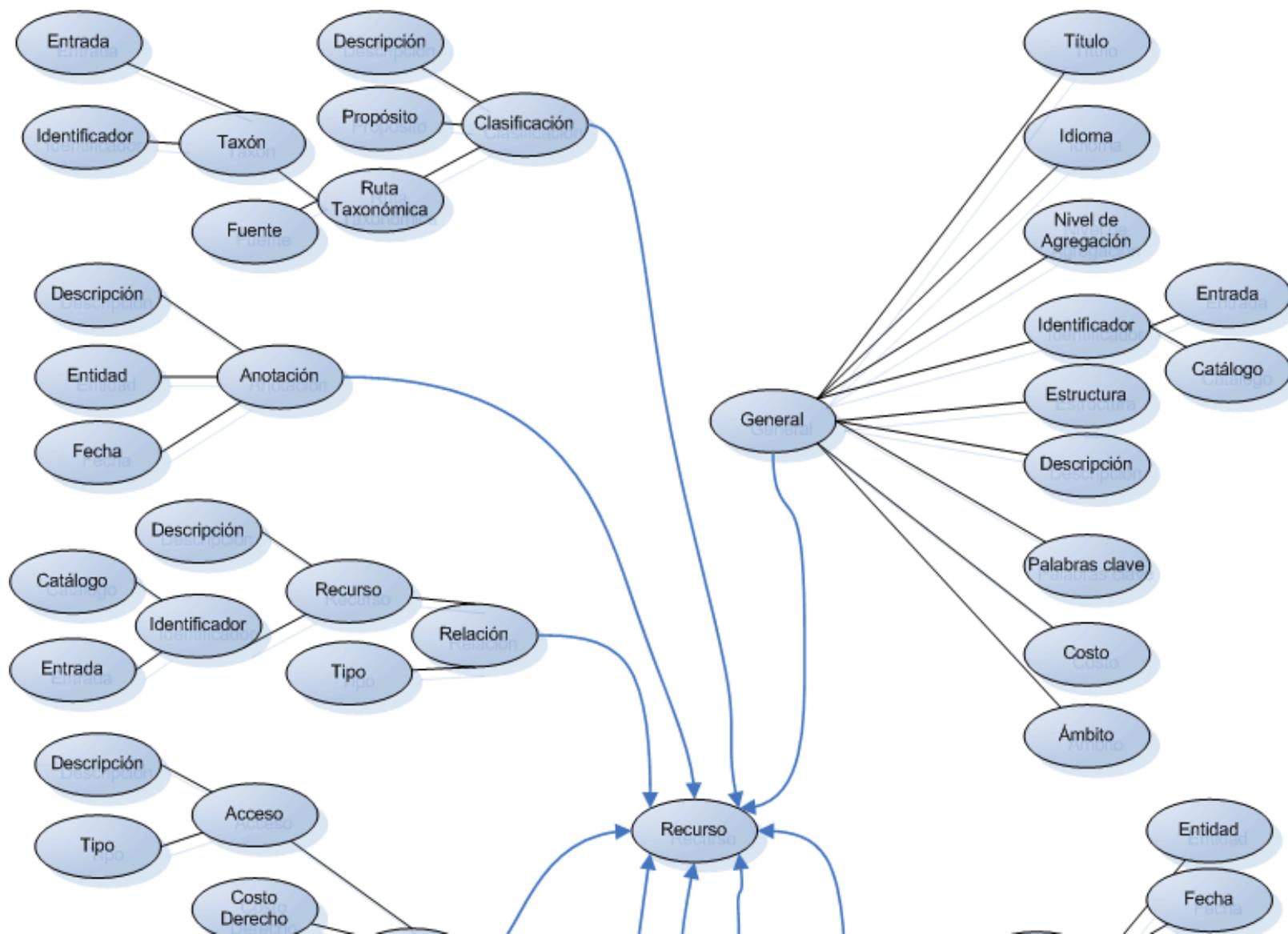


Figura 2.1: Modelo conceptual de la ontología UTPL

2.2.2.2. Modelo conceptual de la ontología DSpace-LOM

La siguiente figura, muestra el modelo conceptual de la segunda ontología denominada DSpace, la misma que permite representar OA en base a los metadatos y relaciones existentes entre los mismos.



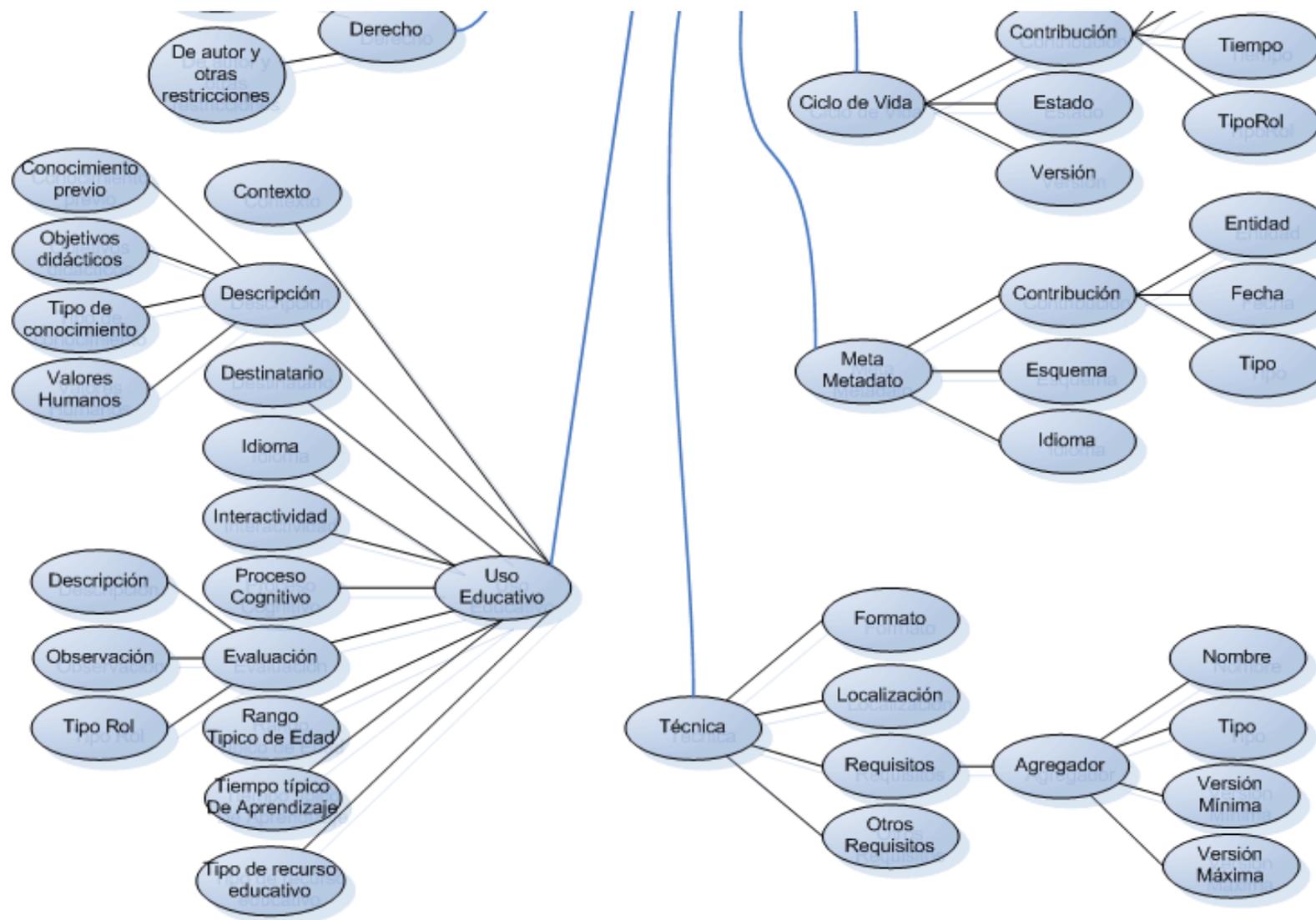


Figura 2.2: Modelo conceptual de la ontología DSpace



2.2.2.3. Glosario de términos ontología UTPL

El glosario de términos, hace referencia a un conjunto de conceptos que son de interés del dominio de la ontología, sus descripciones, sinónimos y acrónimos, en el glosario se definen los términos que forman parte de la ontología UTPL. A continuación se presenta parte de éstos términos.

Ontología UTPL				
Término	Descripción	Sinónimo	Acrónimo	Tipo
Áreas	Forma en la que tanto las escuelas y los CITTEs se organizan dentro de la UTPL.	Secciones	--	Concepto
Técnica	Está conformada por las escuelas con las que la Universidad abrió sus puertas como: Ingeniería Civil, Arquitectura y cuenta actualmente con tecnología e infraestructura de última generación, que permite potenciar el proceso de enseñanza - aprendizaje.	--	--	Instancia
Carreras	Hace referencia a las diversas especializaciones dentro de cada una de las modalidades	Especialidad	--	Concepto
Administración de Empresas	--	--	AE	Instancia
Biológica	--	--	BL	Instancia
Asignaturas	Representa las asignaturas que se encuentran dentro de una determinada	Materias	AS	Concepto



	carrera.			
Arquitectura de Aplicaciones	--	AAPP	--	Instancia
Base de Datos Avanzadas	---	--	BDA	Instancia
Contenido	Permite representar todos aquellos conceptos que definen a una asignatura	--	--	Concepto
Componente del Computador y buses	--	--	CC	Instancia

Tabla 2.4: Glosario de términos ontología UTPL

2.2.2.4. Glosario de términos ontología DSpace-LOM

El siguiente cuadro muestra algunas de las categorías pertenecientes al estándar LOM que han sido representadas en esta ontología.

Ontología para recursos del DSPACE				
Término	Descripción	Sinónimo	Acrónimo	Tipo
General	Los metadatos de esta categoría representan información general sobre el recurso educativo que describe el mismo como un todo.	--	--	Concepto
Ciclo de vida	Categoría que permite describir la historia y el estado actual del recurso educativo, así como las entidades que han intervenido en su creación y evolución.	--	--	Concepto
Meta -	Esta categoría	--	--	Concepto



Metadato	describe el propio registro de metadatos y como puede ser identificada esta instancia de metadatos, quién lo creó, cómo, cuándo y con qué referencias (en lugar de la información que describe el recurso educativo).			
Uso Educativo	En esta categoría se describe las características educativas o pedagógicas de este recurso educativo. Esta es la información pedagógica esencial para aquellos(as) agentes involucrados(as) en la consecución de una experiencia educativa de calidad.	--	--	Concepto
Uso Educativo. Destinatario	Permite identificar a el(os) usuario(os) principal(es) para el(os) que ha(n) sido diseñado(s) el(os) recurso(s) educativo(s).	--	--	Concepto
Derecho	Esta categoría describe los derechos de propiedad intelectual y las condiciones de uso aplicables a este	--	--	Concepto



	recurso educativo. La intención es reutilizar trabajos procedentes de las comunidades del derecho de la propiedad intelectual y el comercio electrónico.			
Naturaleza y tipos de resúmenes	---	----	---	Instancia
Resumen	----	-----	----	Instancia
Metodología de estudio	---	---	---	Instancia
Analizar	Se aplica este nivel cognitivo cuando el recurso educativo tiene relación con un problema y el(a) aprendiz debe descomponer el problema dado en sus partes y describir las relaciones existentes entre ellas.	----	----	Instancia

Tabla 2.5: Glosario de términos para la ontología de recursos del DSpace

2.2.2.5. Definición de la taxonomía de la ontología UTPL

La taxonomía se refiere a una jerarquía y relación existente entre los conceptos que componen el dominio de la ontología para el manejo de los recursos educativos almacenados en el repositorio DSpace.

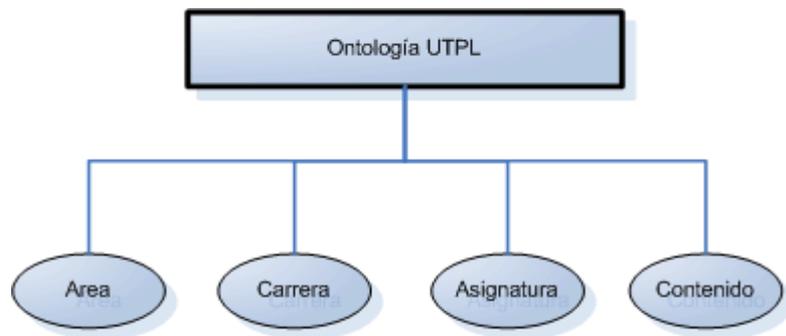


Figura 2.3: Taxonomía del modelo de ontología UTP



2.2.2.6. Definición de la taxonomía de la ontología DSpace-LOM

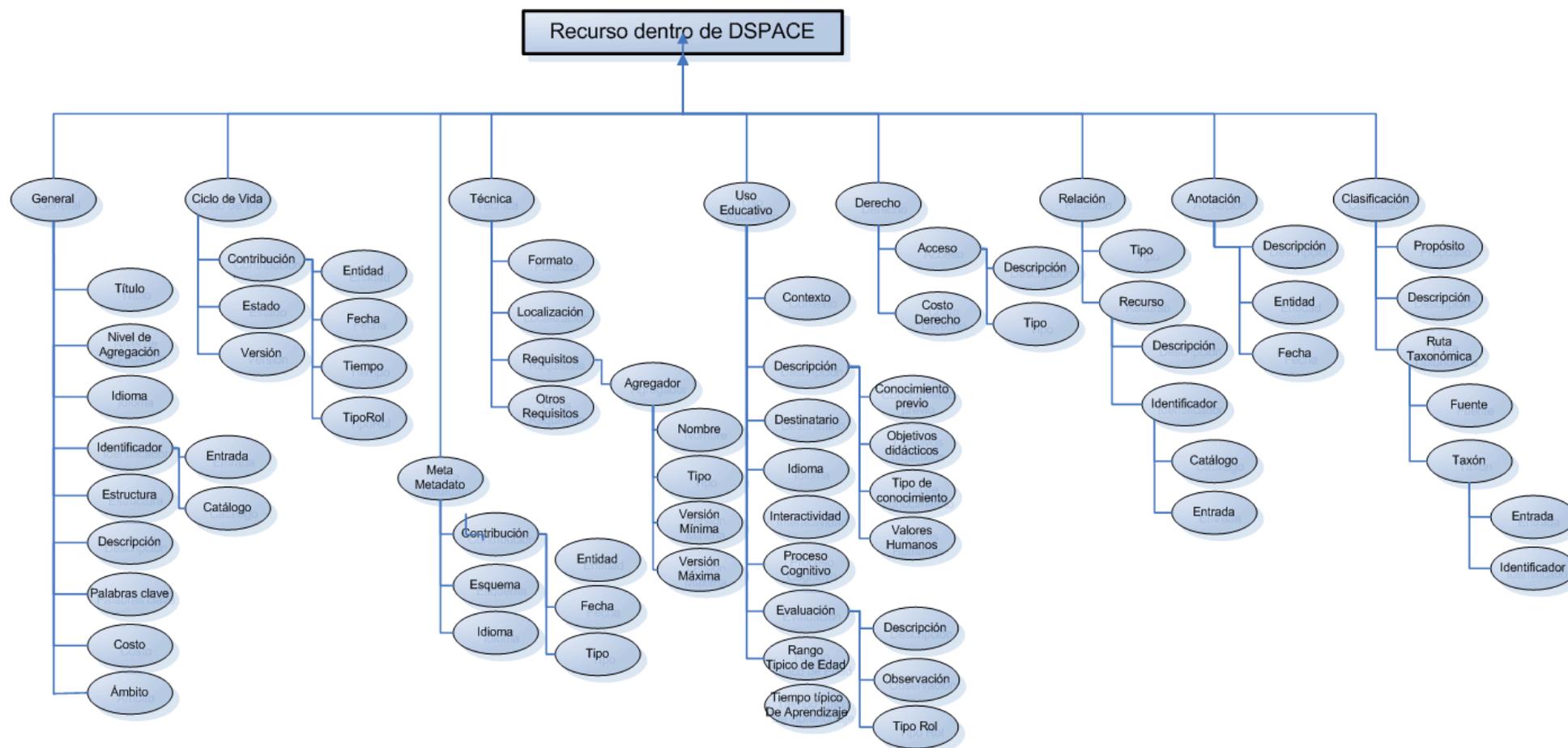


Figura 2.4: Taxonomía del modelo de ontología para recursos del DSpace



2.2.2.7. Diagramas de relaciones binarias de la ontología UTPL

El diagrama de relaciones binarias permite conocer los tipos de relaciones entre conceptos (relaciones inversas, funcionales, transitivas, etc.)

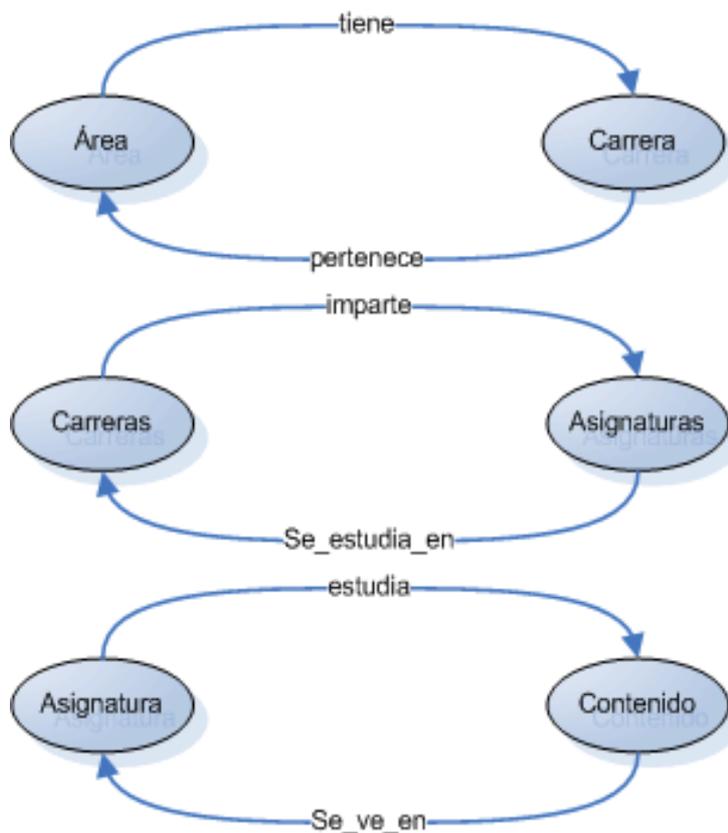


Figura 2.5: Relaciones binarias de ontología UTPL

2.2.2.8. Diagramas de relaciones binarias de la ontología DSpace-LOM

En la siguiente figura, se puede apreciar parte de las relaciones binarias definidas para el modelo DSpace.

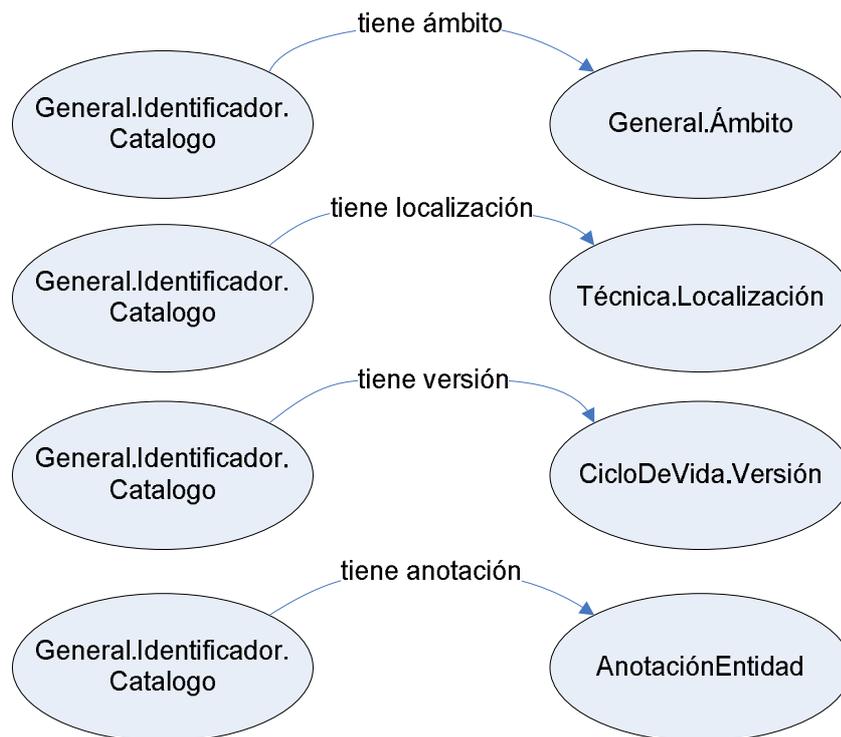


Figura 2.6: Relaciones binarias de ontología DSpace

2.2.2.9. Diagrama de conceptos de la ontología UTPL

Permite definir relaciones, atributos de clases y atributos de instancias.

Nombre del concepto	Instancias	Atributos de clase	Atributos de instancias	Relaciones
Área	A_01	Nombre área	Administrativa	tiene
	A_02		Socio-Humanística	
	A_03		Biológica	
	A_04		Técnica	
Carrera	CA_AD_01	Nombre carrera	Administración de empresas	Pertenece/imparte
	CA_TE_01		Arquitectura	Pertenece/imparte
Asignatura	AS_SI_01	Nombre asignatura	Fundamentos de electrónica	Estudia/se estudia en
		Modalidad	Presencial	
		Tipo	Genérica	
		Oferta	Pregrado	



		Ciclo	IX	
Contenido	CO_FE_01	Capítulo	Compuertas lógicas y álgebra booleana	Se_ve_en
		Subcapítulo	---	

Tabla 2.6: Conceptos de la ontología UTP

2.2.2.10. Diagrama de conceptos de la ontología DSpace-LOM

Nombre del concepto	Instancias	Atributos de clase	Atributos de instancias	Relaciones
General Identificador Catalogo	AT_SI_ME_01	ISBN	--	tiene General titulo
				tiene Anotación Entidad
				tiene General titulo
Uso Educativo	CTXT_01	Asistencia	Profesor Independiente Compañero Mixta	
		Lugar	Entorno real Domicilio	
Destinatario	Administrador(a)	--	--	--
	Agrupamiento de aprendices			
	Aprendiz			
	Autor(a)			
	Profesor			
General. Titulo	Naturaleza y tipos de resumen	--	--	
Derecho. Autor y otras restricciones	Reconocimiento	----	--	--
	Reconocimiento-Compartir igual	--	--	--
	Reconocimiento-Sin obra derivada-No comercial	--	--	--
	Reconocimiento No comercial-	--	--	-



	Compartir igual			
Relación.				Es parte de
Recurso.				Se basa en
Identificador.	RR_IC_01	--	--	Se relaciona con
Catalogo				

Tabla 2.7: Conceptos de la ontología DSpace-LOM

2.2.2.11. Detalle de los atributos de las instancias de la ontología UTPL

Nombre	Concepto al que pertenecen	Tipo de dato	Cardinalidad
Oferta	Asignatura	String	1:1
Tipo	Asignatura	String	1:1
Modalidad	Asignatura	String	1:1
Ciclo	Asignatura	String	1:1
Nombre asignatura	Asignatura	String	1:1
Nombre carrera	Carrera	String	1:1
Nombre área	Área	String	1:1
Capítulo	Contenido	String	1:1
Subcapítulo	Contenido	String	1:N

Tabla 2.8: Atributos de las instancias de ontología UTPL

2.2.2.12. Detalle de los atributos de las instancias de la ontología DSpace

Nombre	Concepto al que pertenecen	Tipo de dato	Cardinalidad
UE Descripción	Uso Educativo. Evaluación. Descripción	String	1:1
ISBN	General. Identificador. Catálogo	String	1:1
URL	Técnica. Localización	String	1:1
Tema	General. Estructura	String	1:1
Subtema	General. Estructura	String	1:1
Modulo	General. Estructura	String	1:1
Curso	General. Estructura	String	1:1
Directorio	General. Estructura	String	1:1
Cuestionario	General. Estructura	String	1:1

Tabla 2.9: Atributos de las instancias de ontología DSpace-LOM



2.2.2.13. Detalle de relaciones binarias de la ontología UTPL

Nombre de relación	Dominio	Rango	Cardinalidad	Relación inversa
<i>Tiene</i>	Área	Carrera	1:N	<i>pertenece</i>
<i>pertenece</i>	Carrera	Área	1:1	<i>tiene</i>
<i>imparte</i>	Carrera	Asignatura	1:N	<i>se estudia en</i>
<i>se estudia en</i>	Asignatura	Carrera	1:N	<i>imparte</i>
<i>estudia</i>	Asignatura	Contenido	1:N	<i>se ve en</i>
<i>se ve en</i>	Contenido	Asignatura	1:N	<i>estudia</i>

Tabla 2.10: Relaciones binarias de la ontología UTPL

2.2.2.14. Detalle de relaciones binarias de la ontología DSpace-LOM

Nombre de relación	Dominio	Rango	Cardinalidad
<i>tiene General Titulo</i>	General. Identificador. Catálogo	General. Titulo	1:1
<i>tiene General Descripción</i>	General. Identificador. Catálogo	General. Descripción	1:N
<i>tiene General Palabras Clave</i>	General. Identificador. Catálogo	General Palabras Clave	1:N
<i>Tiene Técnica Localización</i>	General. Identificador. Catálogo	Técnica. Localización	1:N

Tabla 2.11: Diagrama de relaciones binarias de la ontología DSpace-LOM

2.2.2.15. Definición de reglas de inferencia

En siguiente cuadro se puede apreciar algunas de las reglas de inferencias representadas en las ontologías.

Definición	Nombre de regla	Variables	Modelo Matemático	Representación lógica
En un área se imparten asignaturas.	Area-Carrera-Asignatura	Área(?x) Carrera(?y) Asignatura(?z)	$X \rightarrow Y$ $Y \rightarrow Z$ $X \rightarrow Z$	$tiene(?x, ?y) \wedge$ $imparte(?y, ?z) \rightarrow$ $imparte(?x, ?z)$
Algunas asignaturas se estudian en un área.	Asignatura-Carrera-Área	Asignatura(?z) Carrera(?y) Área(?x)	$Z \rightarrow Y$ $Y \rightarrow X$ $Z \rightarrow X$	$se_estudia_en(?z,$ $?y) \wedge$ $pertenece(?y, ?x)$ \rightarrow $se_estudia_en(?z,$



				?x)
En una carrera se estudia algunos contenidos de estudio	Carrera-Asignatura-Contenido	Carrera(?x) Asignatura(?y) Contenido(?z)	$X \rightarrow Y$ $\underline{Y \rightarrow Z}$ $X \rightarrow Z$	$\text{imparte}(\text{?x}, \text{?y}) \wedge$ $\text{estudia}(\text{?y}, \text{?z}) \rightarrow$ $\text{estudia}(\text{?x}, \text{?z})$
Un contenido se estudia en una o más carreras	Contenido-Asignatura-Carrera	Contenido(?z) Asignatura(?y) Carrera(?x)	$Z \rightarrow Y$ $\underline{Y \rightarrow X}$ $Z \rightarrow X$	$\text{se_ve_en}(\text{?z}, \text{?y})$ \wedge $\text{se_estudia_en}(\text{?y},$ $\text{?x}) \rightarrow$ $\text{se_estudia_en}(\text{?z},$ $\text{?x})$

Tabla 2.12: Reglas de inferencia ontología UTP

2.2.2.16. Definición de constantes

Se definen mediante el uso de una tabla de constantes, en la que se especifica el nombre, tipo de valor, y unidad de medida. Para el presente dominio no fue necesario definir estas constantes.

2.2.2.17. Definición de axiomas formales para la ontología UTP

Para realizar los axiomas formales dentro de una ontología es necesario el uso de: relaciones, conceptos y atributos, los que se detallaron con anterioridad. En el siguiente cuadro se muestra parte de los axiomas implementados para la ontología UTP.

Nombre del axioma	Descripción	Conceptos	Relaciones	Representación lógica
Área/Carrera	Un área alberga como mínimo una carrera	Área	Tiene	$A = \{x/x \text{ es un área}\}$ $C = \{y/y \text{ es una carrera}\}$ AC = {x/y y>=1}
Asignatura/Contenidos	Una asignatura se encuentra definida por	Asignatura	Estudia	$A = \{x/x \text{ es una asignatura}\}$



	algunos contenidos			$C = \{y/y \text{ es un contenido}\}$ $AC = \{y/y \text{ define una asignatura}\}$
Carreras/ Áreas	Las carreras pertenecen a alguna área en específico	Carrera	Pertenece	$C = \{x/x \text{ es una carrera}\}$ $A = \{y/y \text{ es una área}\}$ $CA = \{x/x \in y\}$

Tabla 2.13: Relaciones binarias de la ontología UTPL

2.2.2.18. Definición de axiomas formales para la ontología DSpace-LOM

Nombre del axioma	Descripción	Conceptos	Relaciones
Catalogo/Titulo	Un OA posee un titulo descriptivo	General. Identificador. Catalogo	General.Titulo
Catalogo/ Palabras Clave	Un OA puede contener palabras claves que permiten identificarlo	General. Identificador. Catalogo	General. PalabrasClave
Catalogo/ Versión	Un OA puede tener diferentes versiones en el transcurso de su ciclo de vida	General. Identificador. Catalogo	CiclodeVida. Versión
Catalogo/ Esquema	Un OA se encuentra detallado por un esquema de metadatos	General. Identificador. Catalogo	MetaMetadato. Esquema

Tabla 2.14: Relaciones binarias de la ontología DSpace-LOM

2.2.2.19. Casos de uso

En el siguiente caso de uso se especifican las acciones que tendrán que realizarse para el proceso de creación de la ontología mediante una herramienta especializada para el caso, esto después de realizar la definición y conceptualización de la ontología

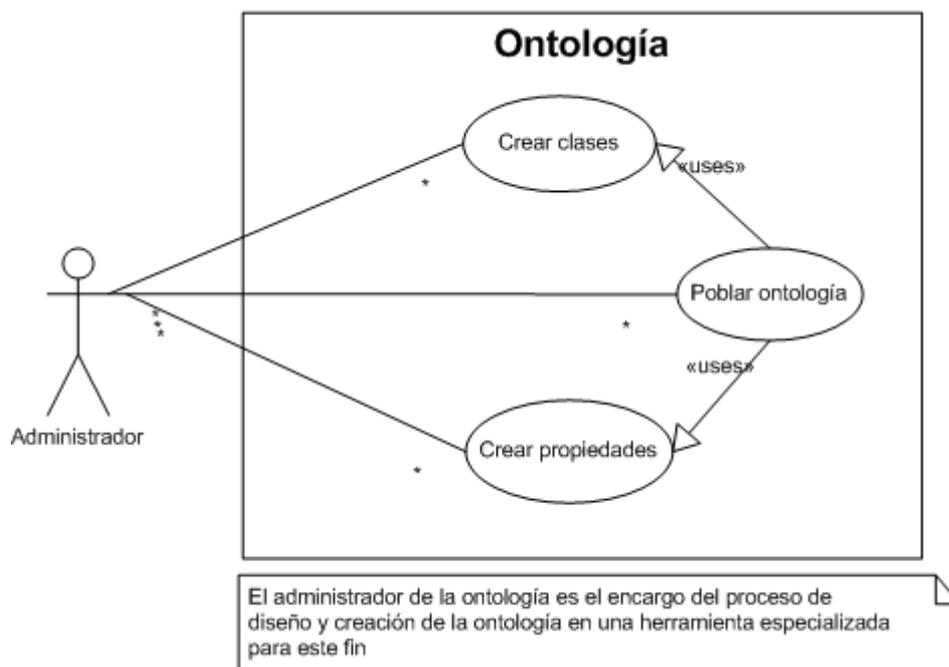


Figura 2.7: Caso de uso para de la construcción de la ontología

2.2.2.20. Instanciamiento de las ontologías

Para observar la creación de instancias mediante la herramienta Protégé v3.2 referirse al manual del administrador Anexo I, inciso 1.6.

El desarrollo del modelo y representación de las ontologías Protégé se realizó de forma separada por lo siguiente:

- El dominio a representar posee dos subdominios: UTPL y dominio DSpace.
- La ontología DSpace-LOM no depende explícitamente del modelo UTPL, por lo que puede ser reutilizada en cualquier otro proyecto semántico.
- Las ontologías pueden ubicarse a futuro en repositorios ontológicos y ser accesibles para otros usuarios en la Web.

2.2.2.2.1 Modelo lógico de ontología UTPL

Las instancias que serán representadas en la ontología UTPL es la materia Metodología de Estudio, la cual es



una materia común de la modalidad abierta y a distancia.

Para el instanciamiento del contenido de las asignaturas en el modelo lógico de la UTP, dentro de la herramienta Protégé, se hace necesaria la definición de los capítulos y subcapítulos como propiedades de tipo datatype; esto permitirá tener una definición más clara y precisa de los contenidos relacionados a cada asignatura.



The screenshot displays the Protege ontology editor interface. It is divided into several panes:

- CLASS BROWSER:** Shows a class hierarchy for the project 'utpl_model'. The classes listed are owl:Thing, Areas (4), Asignaturas (48), Carreras (23), and Contenido (5). 'Contenido' is currently selected.
- INSTANCE BROWSER:** Shows instances for the selected class 'Contenido'. Under the 'Asserted' tab, five instances are listed: MET_EST01, MET_EST02, MET_EST03, MET_EST04, and MET_EST05. MET_EST01 is selected.
- INDIVIDUAL EDITOR:** Shows the editor for the selected instance 'MET_EST01'. It includes a table for properties and values, and sections for defining and describing the instance.

Property	Value
rdfs:comment	

Define	Nombre	Value	Type
Metodología_de_Estudio		La educación a distancia: Ni...	string

Descripción	Value	Type
	La educación a distancia s...	string
	La expansión de la educa...	string
	La capacidad de innovaci...	string
	Razones para la educación...	string

Figura 2.8: Creación de instancia con la definición de sus atributos correspondientes



2.2.2.2.2 Instanciación de modelo lógico de ontología DSpace-LOM

El instanciamiento de la ontología que representa el modelo del DSpace, se lo realizará inicialmente tomando en consideración los OA con sus respectivos metadatos, que se encuentran almacenados en el repositorio DSpace. Este método de población o instanciamiento de ontologías, al igual que el modelo UTPL, ha sido realizado inicialmente de forma manual.

Para el uso de las ontologías en la interfaz de usuario; es decir, para presentar los OA recomendados de acuerdo al contenido de las asignaturas, en este caso Metodología de Estudio, y el uso del buscador semántico, se utilizó la ontología DSpace-LOM dentro de la ontología UTPL, es decir, se importaron los recursos existentes de una ontología dentro de otra, lo cual no implica necesariamente que las ontologías se fusionen y conformen una sola, sino que mantienen su independencia y continúan representando su dominio. El resultado se puede apreciar en la siguiente figura.

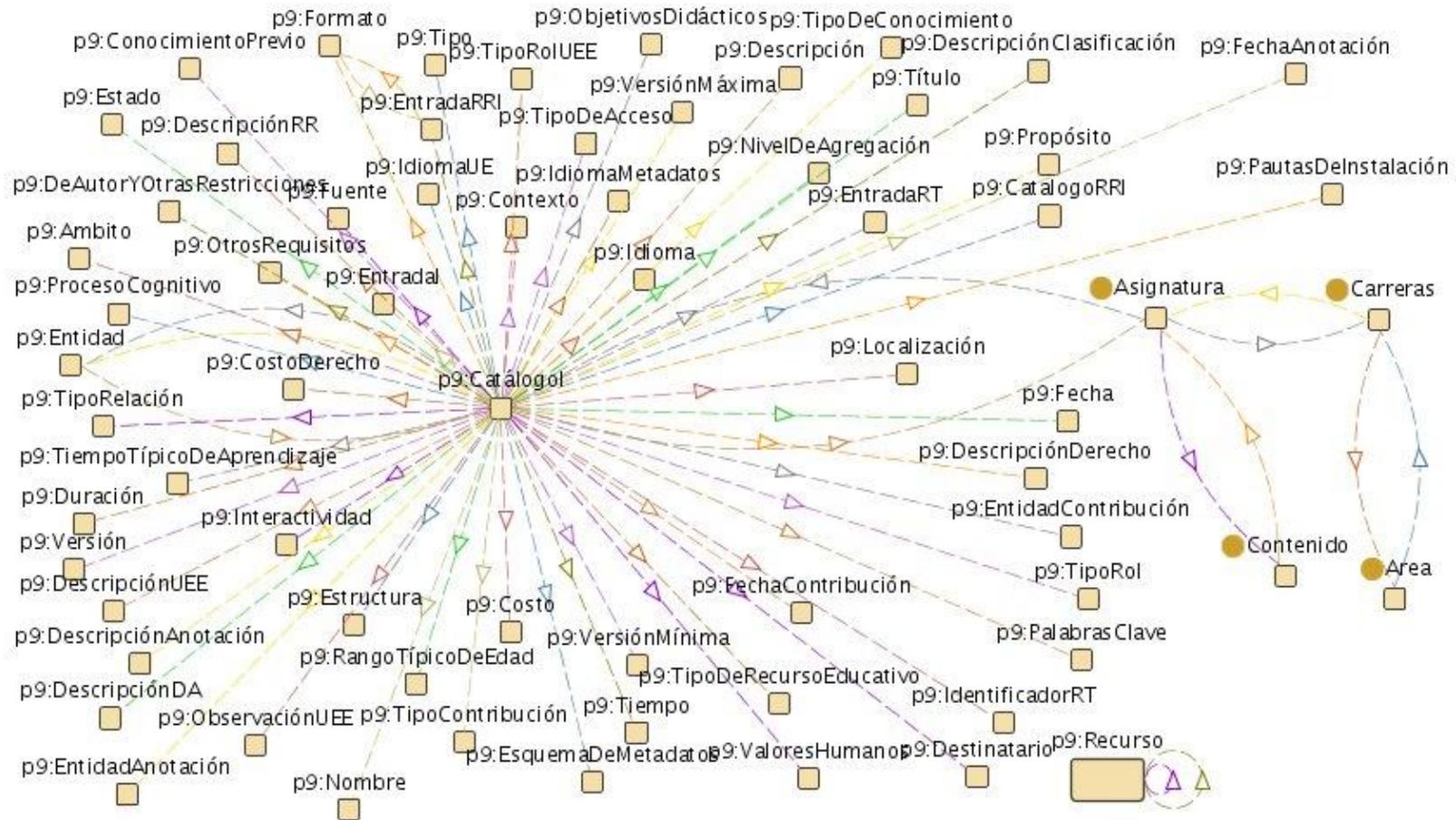


Figura 2.9: Vinculación de modelos: UTPL y DSpace-LOM



2.2.3. Implementación de las ontologías

2.2.3.1. Introducción

La fase de implementación es una tarea que involucra la representación del diseñado lógico de la ontología a un lenguaje formal, como: RDF ó OWL.

El proceso de formalización se realiza a través de una herramienta de edición y manejo de ontologías el que se encuentra descrito en el Manual del Administrador (Anexo I), en donde se explica el procedimiento desarrollado para la construcción de las ontologías: UTP y DSpace, con la representación de sus respectivos conceptos, relaciones, atributos, axiomas o restricciones, etc.

Para la implementación de las ontologías se utilizó el software Protégé v3.2 y el lenguaje OWL DL. La implementación o formalización de una ontología implica una definición previa de: objetivos a cumplir, conceptos, relaciones, jerarquía entre conceptos, instancias, formulaciones lógicas o matemáticas que conformarán la ontología,

El primer paso a considerar en la formalización es crear en la herramienta cada uno de los conceptos (conservando su jerarquía) pertenecientes al modelo; es decir, la elaboración de la taxonomía de conceptos (todo esto a través de: glosario de términos, y la definición taxonómica). El siguiente paso, es la creación de las relaciones existentes, ya sean relaciones directas o inversas y la cardinalidad entre conceptos (relaciones binarias). Luego se crean las restricciones y reglas de inferencia (axiomas formales y reglas de inferencia), para finalmente realizar la creación de instancias pertenecientes a cada uno de los modelos.

Algunos problemas que aparecieron durante la formalización de la ontología tales como: instancias repetidas, errores de taxonomía, etc., se dieron debido a una incorrecta representación e interpretación del dominio del problema y a una falta de versionamiento de los mismos.



2.2.4. Evaluación y validación

2.2.4.1. Evaluación

En la actualidad existen varios criterios que permiten evaluar y analizar una ontología, ya sea esto con la finalidad de integrarla a un nuevo proyecto o de medir la calidad y funcionalidad de la misma. El desarrollo de esta tarea es una actividad que se realiza de manera paralela a cada una de las fases de la metodología usada (Methontology), permite medir la potencialidad de la ontología a construirse, e ir detectando posibles errores en cuanto a prestaciones o funcionalidades, con la finalidad de documentar cada una de las fases de la metodología y continuar el ciclo de vida de la ontología.

La evaluación permitió realizar un refinamiento de tres versiones de ontologías, así:

Versión	1							
	Objeto	Clase	Atributo	Relación	Operativo-Funcional		Filtrado de consultas	
					Si	No	Si	No
Modalidad	X				X	X		
Asignatura	X				X		X	
Carreras	X				X		X	
Asignatura	X				X		X	
Contenido	X				X		X	
Agrupar					X			
Define			X		X			
Define_por			X		X			
Poseen			X		X			
Descripción		X			X			
Oferta		X			X			

Tabla 2.15: Evaluación de la primera versión ontológica



Detalle del problema en la versión del modelo

El principal inconveniente en la primera versión de la ontología, fue el establecimiento de jerarquías entre las clases existentes; es decir, el concepto de herencia, lo que suponía que cada una de las clases hijas heredaban los atributos de las clases padres o superclases, esto a su vez conllevó a dos problemas: definición de instancias en cada uno de los niveles de la ontología y mayor codificación en el filtrado de resultados.

Versión		2					
Objeto	Clase	Atributo	Relación	Operativo-Funcional		Filtrado de consultas	
				Si	No	Si	No
Modalidad		X		X			X
Asignatura	X			X			X
Carreras	X			X			X
Asignatura	X			X			X
Contenido	X			X			X
Agrupar					X		
Define			X		X		
Define_por			X		X		
Poseen			X		X		
Descripción		X		X			
Oferta		X		X			
Comunidad	X				X		X
Metadatos	X				X		X
Papers	X				X		X
Libros	X				X		X
Revistas	X				X		X
Videos	X				X		X
Se_relaciona			X		X		
Conformado_por			X		X		

Tabla 2.16: Evaluación de la segunda versión ontológica



Detalle del problema en la versión del modelo

El inconveniente en la segunda versión de las ontologías fue en la representación de objetos como clases, que permitan definir un OA, además de la inconsistencia al momento de crear las instancias.

Versión	3						
Objeto	Clase	Atributo	Relación	Operativo-Funcional		Filtrado de consultas	
				Si	No	Si	No
Asignatura	X			X			X
Carreras	X			X			X
Asignatura	X			X			X
Contenido	X			X	X		X
General	X			X		X	
Ciclo de Vida	X			X		X	
Técnica	X			X		X	
Relación	X			X		X	
Uso Educativo	X			X		X	
Clasificación	X			X		X	
Anotación	X			X		X	
Meta	X			X		X	
Metadato	X			X		X	
Derechos	X			X		X	
Ciclo		X		X		X	
Descripción		X		X		X	
Modalidad		X		X		X	
Nombre_área		X		X		X	
Nombre asignatura		X		X		X	
Nombre carrera		X		X		X	
Se_asocia_a			X	X		X	
tieneAsociado OA			X	X		X	
URL		X		X		X	

Tabla 2.17: Evaluación de la tercera versión ontológica



Detalle

Se solucionó el inconveniente de instancias repetidas mediante la codificación de las mismas, existía el suficiente nivel de expresividad para la representación de un OA, debido al uso del estándar LOM, además la adición de relaciones inversas en el modelo permite un razonamiento de clases inferiores a superiores y viceversa.

	Número de clases	Número de relaciones directas	Número de relaciones inversas	Número de axiomas	Número de reglas de inferencia
Versión 1	11	9	8	5	0
Versión 2	10	8	16	16	0
Versión 3	88	62	14	14	5

Tabla 2.18: Comparación de las versiones de ontologías

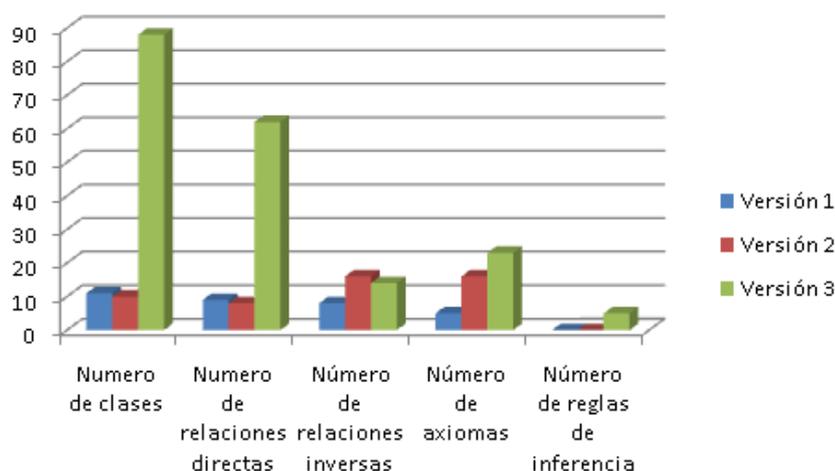


Figura 2.10: Evaluación de versiones de las ontologías

2.3. Validación por medio de métricas

Uno de los objetivos fundamentales de la validación de ontologías por medio de métricas es llegar a **determinar la reusabilidad de una ontología**; es decir, reutilizar ontologías que se encuentran en diversas fuentes como OntolinguaServer, el servidor WebODE (X-WebODE), etc., en el desarrollo de un nuevo sistema de conocimiento. El tema de reutilización de ontologías se aborda en (Lozano Tello, 2002), mediante la aplicación de una métrica denominada Ontometric. Ésta métrica toma como base



muchos criterios (versión del lenguaje, madurez de la metodología, objetivos del proyecto, etc.) y está destinada para: Ingenieros o administradores de proyectos.

Según Brank y otros (Brank, Grobelnik, & Mladení) las aproximaciones de las evaluaciones de ontologías se encuentran dentro de las siguientes categorías:

- “Aquellas basadas en la comparación de la ontología con el *golden standar ontology*”, (comparación a nivel léxico y conceptual).
- “Basadas en el uso de la ontología dentro de una aplicación para la posterior evaluación de resultados”.
- “Basadas en comparaciones con una fuente de datos (una colección de documentos) acerca del dominio que cubre la ontología”.
- “Basadas en criterios de expertos en ontologías”.

Además en (Brank, Grobelnik, & Mladení) se agrupan estas aproximaciones de acuerdo al nivel de evaluación en:

Léxica, vocabulario.- Principalmente se centra en evaluar los conceptos, instancias, etc., que han sido incluidos en la ontología.

Jerarquía o taxonomía.- Las relaciones existentes entre los conceptos de la ontología del tipo, es *un*, y cualquier otra existente en el modelo.

Contexto o nivel de evaluación.- La ontología puede formar parte de un conjunto de ontologías y puede ser usada para varias definiciones o dominios.

Estructura, arquitectura, diseño.- Otros criterios tales como los que se puede apreciar en (Gangemi, Catenacci, Ciaramita, & Lehmann), en donde se hace énfasis en 3 características para la evaluación de una ontología, y que son: a) medida estructural; es decir, la estructura del modelo ontológico (representación mediante grafos), b) funcionalidad, hace referencia al diseño, tareas para la que fue creada, acuerdo de opiniones entre los expertos, y, c) usabilidad, el nivel de anotación que posee una ontología, facilidad para reconocer sus propiedades, etc.

Para medir las ontologías UTPL y DSpace se tomó en consideración los criterios provistos y descritos en (Gangemi, Catenacci, Ciaramita, & Lehmann), esto debido a que permiten validar específicamente una ontología en términos de su propósito de



construcción (estructura, anotaciones, funcionalidad) y no en términos de reutilización.

2.3.1. Validación a nivel estructural

En el nivel estructural se evalúa la representación de la ontología como un grafo, propiedades: lógica y topológicas.

Dentro de este nivel se puede evaluar la profundidad, distribución de nodos hojas, nodos hermanos, densidad, consistencia, cohesión, etc.

Las pruebas de evaluación y validación se las realiza a la ontología para las recomendaciones del contenido almacenado en el Repositorio de Objetos de Aprendizaje (ROA). Como se puede observar a continuación, las métricas de profundidad y amplitud se aplican a las relaciones del tipo "es un", por lo que no han sido consideradas como parámetros a evaluar en la presente aplicación.

Profundidad.- Es una propiedad relacionada a la cardinalidad de los caminos, en donde las relaciones son del tipo "es un". La profundidad media se define como:

$$PM = \frac{1}{n_{PCG}} \sum_j^P N_{j \in P}$$

De donde:

$N_{j \in P}$: es la cardinalidad de cada camino j

n_{PCG} : es la cardinalidad del conjunto de caminos P

Amplitud.- Es la cardinalidad de las generaciones o niveles en un grafo, en donde los arcos considerados son del tipo "es un".

Matemáticamente la amplitud se define como:

$$Am = \frac{1}{n_{LCG}} \sum_j^L N_{j \in L}$$

En donde:

$N_{j \in L}$: es la cardinalidad de cada generación j del conjunto de generaciones L

n_{LCG} : es la cardinalidad del conjunto de generaciones L



2.3.1.1. Métricas de cohesión

Cohesión.- Se refiere al grado de relación entre los miembros de la clase.

- **Número de clases raíces (NCR).**- Es el número de clases C_j raíces definidas de forma explícita en la ontología O_i . Una clase raíz es aquella que no tiene una clase padre o superclase.

Se define matemáticamente como:

$$NCR(O) = \sum C_j \text{ para todo } 1 \leq j \leq n$$

- **Número de clases hoja (NCH).**- Es el número de clases hojas L_j , definidas de forma explícita en la ontología O_i . Una clase hoja es aquella que no tiene subclases definidas en la ontología.

Matemáticamente se la define como:

$$NCH(O) = \sum L_j \text{ para todo } 1 \leq j \leq n$$

- **Promedio de la profundidad de herencia del árbol de nodos hojas (PPH-ANH).**- Es la suma de la profundidad de todos los caminos dividido para el número total de caminos.

Matemáticamente se define como:

$$PPH - ANH = \frac{\sum D_j}{n} \text{ para todos } D_j, 1 \leq j \leq n$$

En donde:

D_j : es el número total de nodos en un camino j

n : número total de caminos en la ontología

En el estudio de (Yao, Orme, & Etkorn, 2005) se propone una escala para evaluar los resultados obtenidos de métricas, la misma que se presenta a continuación:



0 = Bajo

0.25 = Moderado

0.5 = Promedio

0.75 = Alto

1 = Excelente

Nivel estructural

Criterio	Valor
NCR	13
NCH	58
PPH-ANH	0.75

Tabla 2.19: Validación de la ontología a nivel estructural

Considerando la escala presentada y los resultados obtenidos a nivel estructural, se puede decir que la ontología tiene un nivel de cohesión alto.

2.3.2. Validación de la ontología a nivel funcional

El nivel funcional se relaciona con el principal propósito de una ontología; es decir, determinar la precisión que posee el modelo. En A theoretical framework for ontology evaluation and Validation (Gangemi, Catenacci, Ciaramita, & Lehmann) sugieren tres formas en que se puede medir la precisión: a) juicio de los expertos, b) un conjunto de datos valorados como expresiones calificadas de expertos.

En este nivel los métodos para evaluar se dividen en: caja-negra y caja de vidrio, entre ellos tenemos:

Caja negra	Caja de vidrio
<i>Acuerdo de los expertos</i>	<i>Satisfacción del usuario</i> <i>Tareas soportadas a través de la ontología</i> <i>Modularidad</i>

Tabla 2.20: Clasificación de métodos para medición de precisión



2.3.2.1. Lógica Descriptiva

- **Relaciones Lógico Descriptivas Directas (RLDD).**- Es el cociente entre el número total de relaciones en el modelo y la sumatoria de relaciones directas.

Matemáticamente se define como:

$$RLDD = \frac{\sum R_{dj}}{n_r} \text{ para todo } 1 \leq dj \leq n$$

En donde:

R_{dj} :relación directa j

n_r :número total de relaciones en el modelo

- **Relaciones Lógico Descriptivas Inversas (RLDI).**- Es el cociente entre la sumatoria de relaciones inversas y el número total de relaciones en el modelo.

$$RLDI = \frac{\sum R_{ij}}{n_r} \text{ para todo } 1 \leq ij \leq n$$

En donde:

R_{ij} :relación inversas j

n_r :número total de relaciones en el modelo

2.3.2.2. Lógica Funcional

Se refiere al funcionamiento y tareas soportadas por la ontología. Su valoración se determina por la precisión de los resultados obtenidos al momento de realizar búsquedas.

Se define matemáticamente como:

$$\text{Precisión} = \frac{1}{n_b} \sum \frac{i_c}{n_i}$$

En donde:

n_b :número total de búsquedas

i_c :instancias correctas que satisfacen el criterio de búsqueda

n_i :número total de instancias retornadas en la búsqueda

Criterio	Valor
----------	-------



Lógico descriptiva	Número de relaciones directas	62
	Número de relaciones inversas	14
	Número total de relaciones	76
	RLDD	0.815
	RLDI	0.184
Lógico funcional	Número total de búsquedas	5
	i_e 1era búsqueda	5
	m_e 1era búsqueda	7
	i_e 2da búsqueda	3
	m_e 2da búsqueda	4
	i_e 3era búsqueda	2
	m_e 3era búsqueda	5
	i_e 4ta búsqueda	6
	m_e 4ta búsqueda	7
	i_e 5ta búsqueda	5
	m_e 5ra búsqueda	7
	Precisión	0.686

Tabla 2.21: Validación de la ontología a nivel funcional

Como se puede observar a nivel funcional, se concluye que la ontología posee un nivel de precisión promedio.

2.3.3. Validación de la ontología a nivel del perfil de usabilidad

El perfil de usabilidad de la ontología se refiere al conjunto de anotaciones ontológicas que facilitan su manejo. Algunos ejemplos de anotaciones semánticas son: precio, versión, autor, etc.

En este nivel se distinguen tres criterios a evaluar: la eficiencia, reconocimiento, interfaz.

- Eficiencia, incluye anotaciones como: organización desarrolladora de la ontología, software usado, costo, esfuerzo requerido para la creación, etc.
- Reconocimiento, anotaciones sobre el ciclo de vida de la ontología, métodos empleados, versionamiento, etc.



- Interfaz, relacionado al proceso de comunicar una ontología con una interfaz de usuario.

Las anotaciones semánticas consideradas para la evaluación en el perfil de usabilidad de la ontología son:

- Anotaciones de clases;
- Anotaciones de instancias;
- Anotaciones de relaciones.

En el anexo tres se adjunta el formato usado para la validación de anotaciones semánticas.

Perfil de usabilidad

Anotaciones semánticas		
Eficiencia	Organización	✓
	Costo	X
	Herramientas usado	✓
	Autor	✓
	Metodología	✓
Reconocimiento	Versionamiento	✓
	Ciclo de Vida	X
	Definición de relaciones	✓
	Definición de atributos	✓
	Anotaciones de clases	✓
	Anotaciones de instancias	✓
	Anotaciones de relaciones	✓

Tabla 2.22: Validación de la ontología a nivel de perfil de usabilidad.



De acuerdo al formato presentado, podemos decir que la ontología posee una gran cantidad de anotaciones semánticas que permiten identificarla y conocer su propósito, facilitando así el mantenimiento de la misma.



CAPITULO III



3. Búsqueda semántica

3.1. *Introducción*

En el presente capítulo se presenta una breve definición de lo que es un buscador semántico y la arquitectura propuesta para la implementación de un componente adicional que permitirá realizar búsquedas semánticas en el ROA DSpace, constituyéndose así, en un sistema recomendador de OA.

3.2. *Contexto actual*

La gran cantidad de información que se halla actualmente en la Web, y la falta de organización de la misma, constituye un problema al momento en que un usuario realiza una búsqueda de acuerdo a sus necesidades, como resultado de esto lo que se obtiene a cambio, es información irrelevante y poco precisa, de esta forma el mismo usuario tiene que realizar un proceso manual de selección de aquellos documentos o información requerida.

La falta de precisión de la información se debe a que los buscadores basan sus criterios de búsquedas en palabras claves que definen un documento o recurso en la Web, otros como el buscador Google usan el ranking que los sitios poseen. Toda esta combinación de factores (algoritmos de búsqueda, falta de organización de la información, etc.) forma una búsqueda basada en sintaxis; es decir, coincidencias exactas de palabras.

Según Sánchez & Fernández (2005) Hay al menos tres tipos de situaciones que pueden producir errores en las búsquedas:

“Polisemia: 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.

Sinonimia: 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).

Multilingüismo: realizamos una consulta en el buscador introduciendo términos en inglés y no encontramos páginas Web relacionadas con lo que nos interesa pero escritas en español (o cualquier otra lengua).



En todos estos casos, el problema es que la consulta no identifica de forma precisa aquello en lo que estamos interesados sino que solamente identifica un término que en cierto idioma tiene entre sus significados aquello que estamos buscando. Se denomina "búsqueda sintáctica" a un procedimiento de búsqueda en el que el resultado de la búsqueda es el conjunto de documentos que contienen los términos introducidos en la consulta".

Para contrarrestar estos inconvenientes los buscadores tendrán que implementar mecanismos que permitan realizar búsquedas de acuerdo al significado de las palabras, es por este motivo que se denominan buscadores semánticos.

Algunos autores como en (Criado, 2008) clasifican a los buscadores semánticos en dos categorías: buscadores semánticos orientados y no orientados al usuario. Se entiende por buscadores semánticos no orientados al usuario a aquellos que realizan las búsquedas en ficheros RDF, microformatos, u OWL, y que son útiles para expertos de un área en específico; es decir, buscadores ontológicos.

Los buscadores semánticos orientados hacia el usuario, son aquellos usados por una gran cantidad de personas en Internet, poseen diversas fuentes de confianza, además de información actualizada.

Según (Ding, Finin, Joshi, Peng, Pan, & Reddivari) los buscadores semánticos deben cumplir algunas tareas como los buscadores convencionales tales como: descubrimiento de documentos en línea, procesamiento de las consultas de los usuarios, y ordenamiento de los resultados de búsqueda. En el buscador Swoogle la primera tarea se realiza a través del API del buscador Google, esto permite encontrar documentos candidatos iniciales basados en sus nombres de archivos (por medio de una crawler semántico). Para el procesamiento de consultas se identifica tres tipos de resultado: a) el término URI, representa un término único RDF -una clase, propiedad o instancia-, b) un grafo RDF, es decir, el usuario necesita obtener específicamente un sub-grafo o una parte de un grafo RDF, y c) la URL de un documento Web semántico, es decir, una "referencia física" al documento.

En el buscador Swoogle, el ordenamiento de los resultados de búsqueda se logra a través del ranking mediante los algoritmos OntoRank (basado en un modelo racional que emula el comportamiento de navegación de un agente a través de diversos documentos) y TermRank (Ding, Finin, Joshi, Peng, Pan, & Reddivari)



(ordena los resultados de acuerdo a la cantidad de veces que un término es usado).

En cambio en (Movva, Ramachandran, Li, Cherukuri, & Graves) se desarrolló un buscador denominado Noesis para el ámbito científico atmosférico. Al igual que Swoogle usa servicios Web de otros buscadores tales como: Yahoo y Google para obtener recursos iniciales. Noesis utiliza un algoritmo para la búsqueda que consiste en dos pasos: a) análisis de la consulta, b) presentación semántica.

El primer paso, es realizado en el momento en el que el usuario proporciona su criterio de búsqueda, el mismo que es descompuesto con el propósito de identificar conceptos (iniciales) que se encuentran definidos en una ontología. Luego de esto se asocian otros términos provenientes de la ontología a los conceptos iniciales.

Los conceptos asociados se usan para realizar una búsqueda a través de un Servicio de Inferencia Ontológico (OIS, por sus siglas en inglés), con la finalidad de obtener generalizaciones y especializaciones de los conceptos de búsqueda del usuario, a este paso se lo denomina presentación semántica.

3.3. Definición

Como se menciona en (Harth, Jürgen Umbrich, & Hogan, 2008) “los buscadores semánticos permiten consultas más expresivas sobre información integrada proveniente de múltiples fuentes”; es decir, al igual que los buscadores actuales se procesará información provenientes de diversas fuentes, pero como resultado se obtendrá información más precisa, esto debido a la estructuración y organización de la información, así como al uso de métodos para gestionar el conocimiento tales como: ontologías, o, uso de microformatos².

En la web podemos encontrar algunos buscadores semánticos, tales como:

- Twine: <http://www.twine.com/>
- Sindice: <http://sindice.com>
- SWoogle 2006: <http://swoogle.umbc.edu>

² “Es una forma simple de agregar significado semántico a un contenido legible por el humano y que para la máquina es sólo texto plano”. Obtenido de: <http://es.wikipedia.org/wiki/Microformat>



A pesar de estos primeros intentos de buscadores semánticos, aún existen algunas dificultades y desafíos que tendrán que superarse a futuro tal como los mencionados en (Harth, Jürgen Umbrich, & Hogan, 2008), entre ellos:

- “La arquitectura del motor de búsquedas semántico debe ser escalable hacia la Web.
- Desorden actual de los recursos existentes en la Web, la dificultad se aprecia en el desafío de asociar y relacionar los documentos de acuerdo a un dominio de interés”.

Podemos definir un buscador semántico como aquel que permite realizar consultas expresadas en lenguaje natural, y que pueden ser procesables por las máquinas; la búsqueda de la información se encuentra determinada por un sentido estrictamente semántico más no por las etiquetas o palabras clave que definen un recurso contenido en la Web.

3.4. *Modelo lógico del buscador semántico*

Con el propósito de proveer al usuario (de forma independiente al funcionamiento del mecanismo de recomendación de OA, basado en materias que se encuentra matriculado) un mecanismo de búsqueda, se ha creído conveniente construir un buscador semántico para el ROA, que responda a las consultas generadas en lenguaje natural.

En la siguiente figura se puede apreciar la estructura lógica usada para el buscador semántico.

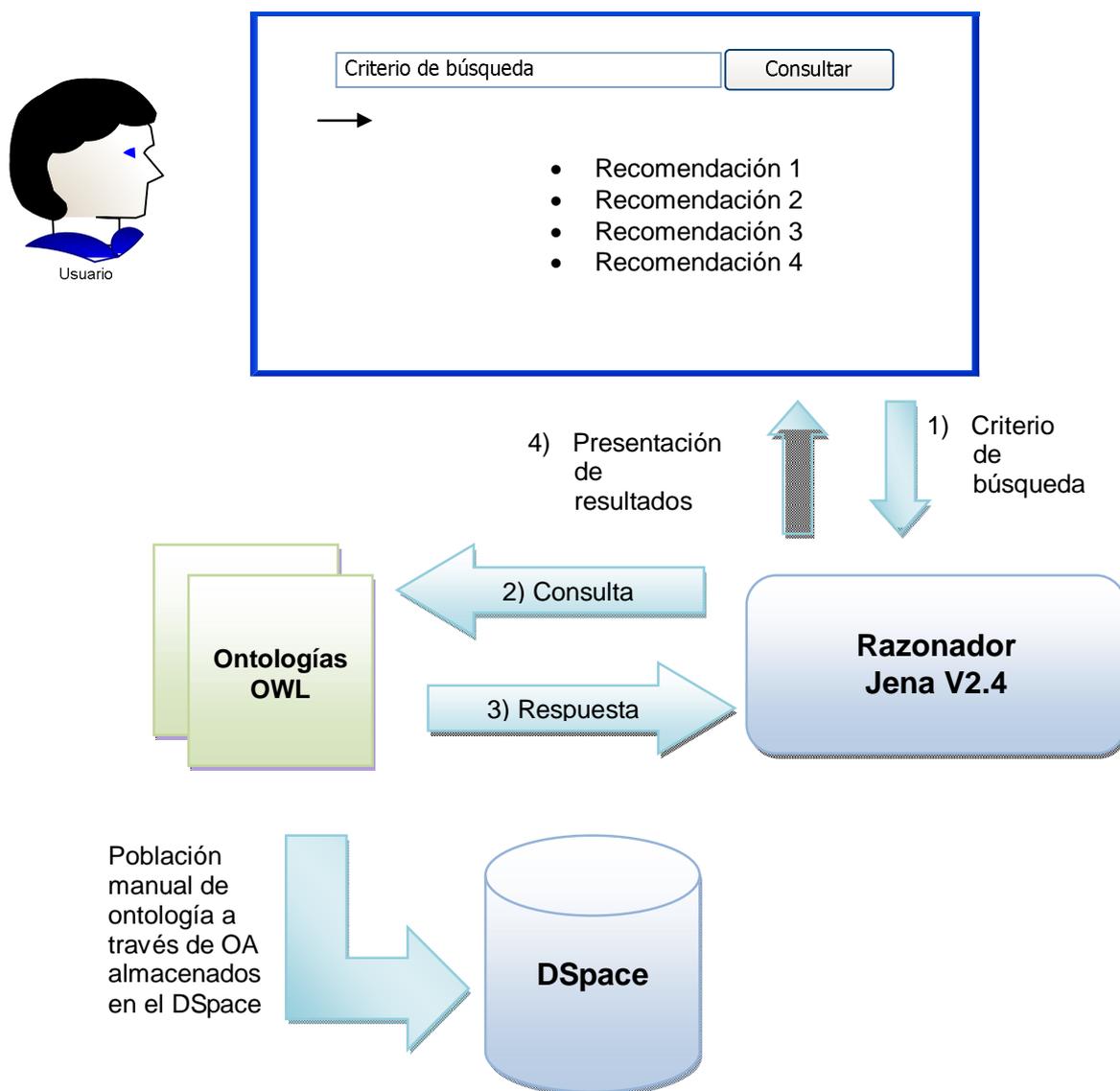


Figura 3.1: Estructura lógica del buscador semántico

Como se observa en la figura anterior, el funcionamiento del buscador semántico, es el siguiente: 1) el criterio de búsqueda del usuario expresada en lenguaje natural es enviada al razonador; 2) el razonador, consulta en los archivos de las ontologías, esto permite determinar los OA relevantes para el usuario (relación entre contenidos, asignaturas y meta información de los OA), 3) se infiere el conocimiento a través de la ontología; y 4) los resultados obtenidos son presentados en la capa de presentación. Cabe señalar que el buscador semántico es un componente más a la arquitectura propuesta a continuación.



3.5. Diseño Arquitectónico

A continuación se presenta la definición del marco conceptual básico de los componentes de la ontología y de la forma de comunicación con otros componentes, con la cual se puede observar el proceso de recomendación de enlaces que incluye: la interfaz Web, razonador, ontología, y en este caso el ROA.

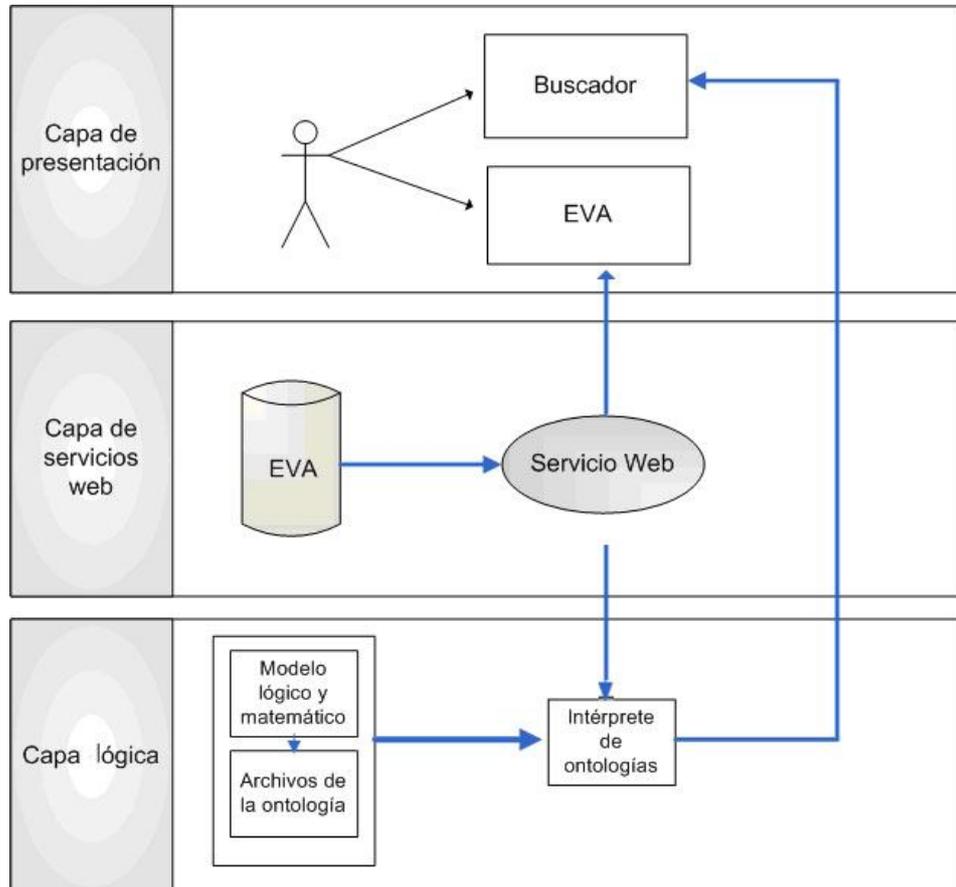


Figura 3.2: Arquitectura de la aplicación

3.6. Implementación del sistema recomendador

La implementación de la aplicación se realizó tomando en consideración algunos aspectos tales como:

- Lenguaje de programación Java v1.6, que se puede obtener desde: <http://www.java.com/es/download/>
- Plataforma de desarrollo Netbeans 6.1, se encuentra disponible en: <http://netbeans.org>
- Servidor Web Tomcat 5.5, disponible en: <http://tomcat.apache.org/download-55.cgi>



- Framework semántico Jena v2.4 para el razonamiento lógico, esta librería se encuentra disponible en: <http://jena.sourceforge.net>

La arquitectura de la aplicación se encuentra representada en la figura 3.4, consta de tres capas: la primera de ellas, permite realizar el razonamiento lógico sobre los datos que se encuentran en la ontología (denominada: capa lógica); la segunda capa la conforma la capa de servicios Web, mientras que la tercera capa la constituye la interfaz de usuario, el detalle del código fuente de cada una de las capas se detalla en el Anexo II.

3.7. Pruebas para componentes que hacen uso de la ontología

Información del documento

TÍTULO:	Plan de pruebas inicial (QA-PPI)
SUBTÍTULO:	
VERSIÓN:	1
ARCHIVO:	
AUTOR:	Manuel A. Cartuche F. (MACF)
ESTADO:	Terminado

Lista de cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0.0	2008-11-28	MACF	Emisión Inicial

Firmas y aprobaciones

ELABORADO	Manuel A. Cartuche F.	
FECHA:	2008-12-03	Firma:

REVISADO POR:	Dunia Inés Jara	
FECHA:	2008-12-09	Firma:

APROBADO	Dunia Inés Jara	
FECHA:	2008-12-09	Firma:



Introducción

Propósito

El plan de pruebas tiene como objetivo verificar que cada componente del sistema funcione correctamente ante escenarios representativos del entorno en el que se ejecutará y validará que el software desarrollado, corresponda con los requisitos definidos inicialmente.

El objetivo del plan de pruebas es:

- Identificar la información del proyecto: documentación generada en la etapa de análisis y los componentes de software de la aplicación que deben ser probados.
- Describir las estrategias de prueba a ser empleadas.
- Identificar los recursos necesarios y suministro estimado de los esfuerzos de pruebas.

Alcance

Las pruebas sobre las que se hacen énfasis son:

- Pruebas de desempeño.
- Pruebas de aceptación (realizadas por el usuario final).

Audiencia

La audiencia involucrada en el proyecto es la siguiente:

- Desarrolladores.
- Usuarios (estudiantes de la modalidad abierta y a distancia).

Escenario de pruebas

Para la validación de los componentes software desarrollados, esto es: a) módulo del EVA, b) SER (aplicación Web independiente), se invitó a los estudiantes de informática (supuestamente cuentan con computador con acceso a Internet) de todos



los centros universitarios (nacionales e internacionales) que hacen uso de la plataforma virtual disponible en la UTPL, y que se encuentran matriculados en la asignatura de Metodología de Estudio de la Modalidad Abierta y a Distancia del período de estudios Octubre 2008 – Febrero 2009, para que sean partícipes en la evaluación del software que hace uso de las ontologías (363).

Para la validación se elaboró una encuesta que permita determinar tanto la aceptación del usuario (usabilidad, facilidad de uso), precisión de los OA (sugerencias de OA relacionados a la asignatura Metodología de Estudio), y los resultados brindados por el buscador en base a las consultas ingresadas. La encuesta (Anexo III) se la hizo llegar vía correo electrónico con las direcciones Web en las que debía acceder para realizar las búsquedas y obtener las recomendaciones ligadas a sus asignaturas.

En el correo se señalaba que una vez que usen la aplicación deberían responder a la encuesta adjunta, luego de tres semanas y de continuas invitaciones únicamente se receptaron ciento cincuenta encuestas llenas, las que nos permitieron probar cada uno de los componentes que hacen uso de la ontología.

Referencias

Se toma como referencia los requerimientos funcionales y no funcionales definidos para el desarrollo del proyecto definidos en el inciso 2.2.1.4.

Factores de motivación

Es necesario efectuar pruebas por varias razones entre ellas:

- Evaluar y mitigar riesgos del proyecto.
- Evaluar y mitigar riesgos técnicos.
- Considerar los elementos del diseño.
- Detectar posibles fallas o defectos al momento de consumir la ontología, y de los componentes que presentan los resultados del buscador.

Estrategia de pruebas

La estrategia de pruebas presenta la aproximación recomendada para las pruebas de acuerdo al objetivo que se persigue. Para cada tipo de prueba, se deberá suministrar una descripción de la misma y la razón por la que está siendo implementada y



ejecutada. Las principales consideraciones para la estrategia de pruebas son las técnicas a ser usadas y el criterio de conocimiento cuando la prueba esté completa.

Pruebas de desempeño

Las pruebas de desempeño tienen la finalidad de evaluar el rendimiento del software en tiempo de ejecución, en este caso la evaluación se realiza una vez que se ha integrado todos los componentes que hacen uso y consumen la ontología, esto con la finalidad de determinar su desempeño real.

Este tipo de prueba se ejecuta para medir el uso de los recursos del servidor en el momento en que el software responde a las peticiones realizadas.

Objetivo de la prueba:	Evaluación del desempeño del servidor durante el consumo de la ontología.
Técnica:	Para el acceso a las ontologías fue necesario crear un componente en el EVA y una interfaz principal para la búsqueda de OAs, esto con el fin de determinar: <ul style="list-style-type: none">· Tiempo de respuesta del servidor.· Monitoreo del consumo de memoria del servidor. Estas tareas se realizan a través de la comparación del desempeño del servidor, antes y durante el consumo de la ontología.
Criterio de Conclusión:	Evaluación de la carga de procesamiento del servidor en el momento de ejecutar los diferentes componentes (módulo EVA, servicio web, buscador) que usan la ontología.

Tabla 3.1: Prueba para medir los recursos del servidor

La siguiente figura muestra el funcionamiento del servidor antes de realizar operaciones sobre la ontología.

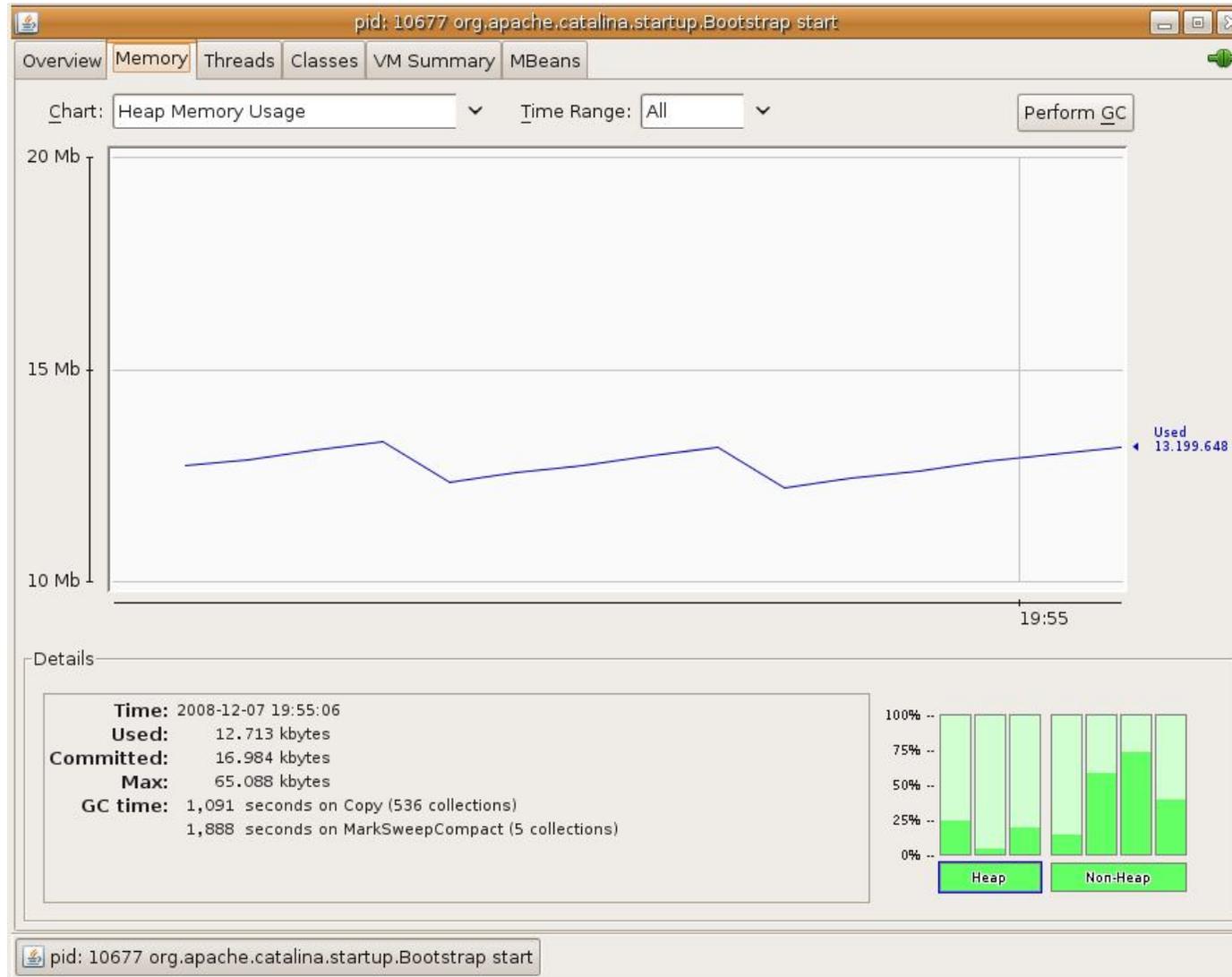


Figura 3.3: Funcionamiento del servidor antes de la ejecución de los componentes que permiten el consumo de la ontología



Recomendaciones a estudiantes a través del componente existente en el EVA.

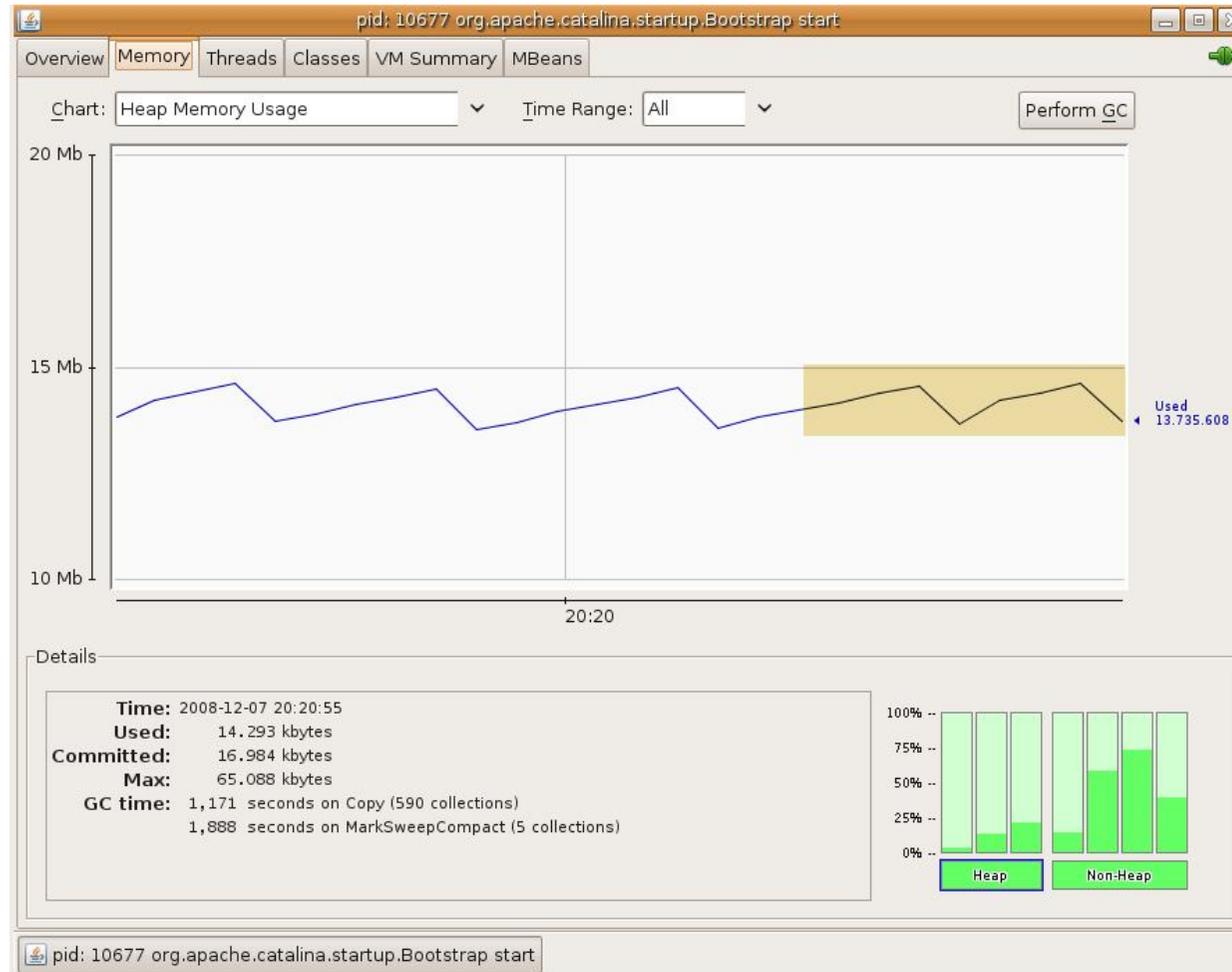


Figura 3.4: Consumo de memoria del servidor mediante el uso del servicio Web



Consultas a través del buscador.

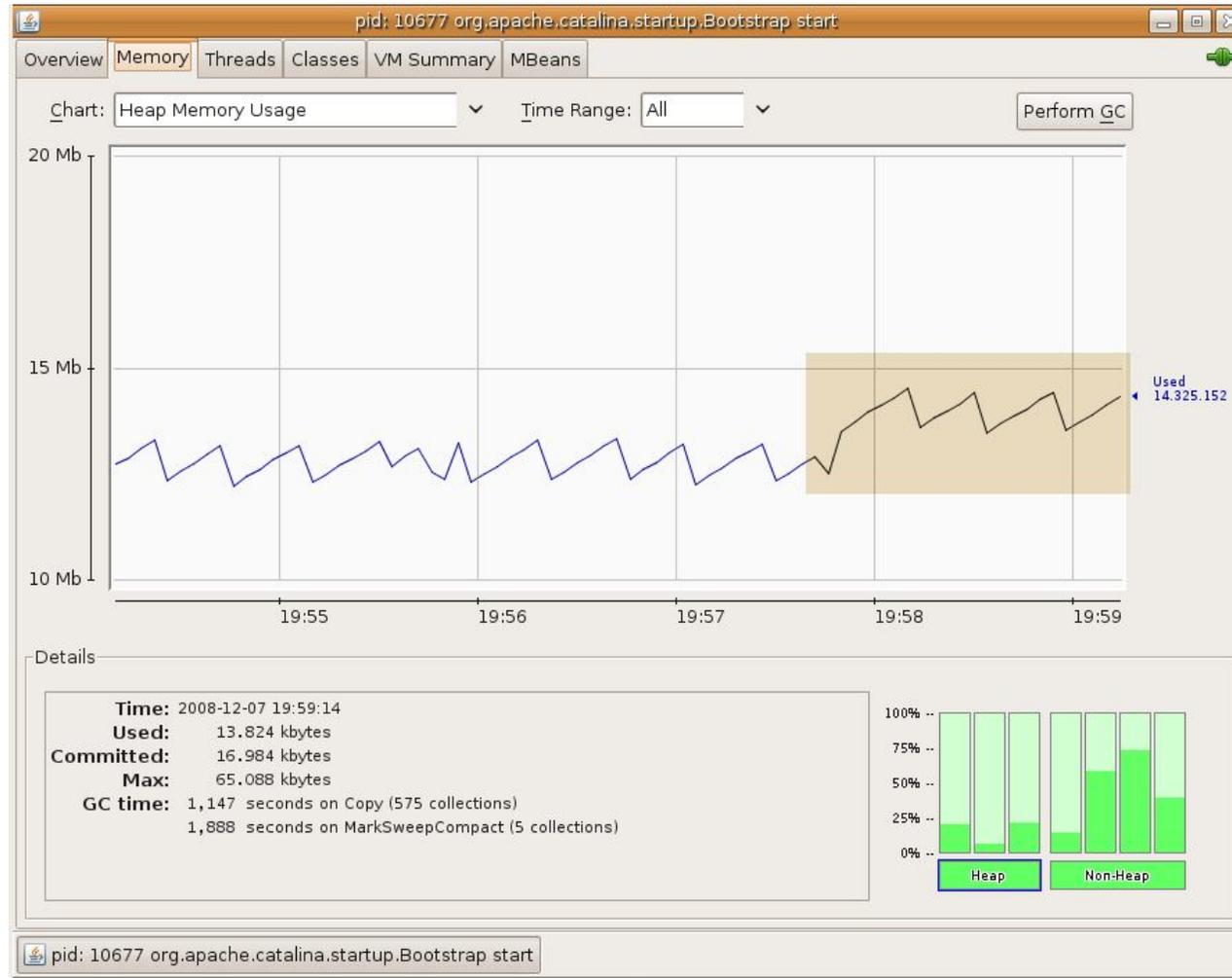


Figura 3.5: Consumo de memoria mediante el uso del buscador



Como se puede observar en las figuras anteriores, cuando se hace uso de los diferentes componentes, el servidor tiende a aumentar la carga de procesamiento y consumo de la memoria RAM, lo cual oscilan entre:

Componente	Incremento de memoria (MB)
Consultas a ontologías a través del buscador	1,073364258
Recomendaciones de Objetos de Aprendizaje a través del EVA	0,511131287

Tabla 3.2: Incremento en el consumo de memoria RAM de acuerdo al uso de los componentes

Estos valores dependen de los recursos hardware que se dispongan y la cantidad de accesos que se realicen.

Tiempos de respuesta

Cantidad de personas que intervinieron	Proceso	Tiempo promedio (seg)
150	Consultas a través del buscador	1.5
150	Recomendaciones por medio del EVA	1.3

Tabla 3.3: Tiempo de respuestas del servidor de acuerdo al proceso.

Pruebas de aceptación del usuario

Al aplicar este tipo de pruebas podemos determinar el nivel de aceptación de los usuarios que utilizarán los componentes creados. Para realizar las pruebas se informó sobre el contexto general del proyecto y ciento cincuenta usuarios hicieron uso de los componentes desarrollados.

Objetivo de la prueba:	Asegurar que los OA presentados tengan relación con los contenidos de la asignatura Metodología de Estudio, y/o que los OA presentados satisfacen los requerimientos ingresados por los estudiantes.
-------------------------------	--



Técnica:	Se ejecutan todas las opciones de la aplicación para verificar: <ul style="list-style-type: none"> • Que las pantallas son fáciles de usar. • Que se realizan todas las funciones especificadas por los estudiantes. • Que se desplieguen los mensajes apropiados de error y precaución cuando se usan datos inválidos.
Criterio de Conclusión:	<ul style="list-style-type: none"> • Que las pruebas planeadas se ejecuten satisfactoriamente. • Todos los defectos identificados han sido corregidos.

Tabla 3.4: Prueba de aceptación de usuario

Resultados obtenidos

	Comprobación de datos	Confiabilidad de datos	Fácil de usar y amigable al usuario
Muy bueno	54.3%	76%	46%
Bueno	33.7%	20%	46%
Regular	12%	4%	8%
Total	100%	100%	100%

Tabla 3.5: Resultados de las pruebas de aceptación de usuarios

De acuerdo a los resultados presentados se puede apreciar que los componentes que permiten hacer uso de la ontología son amigables para los usuarios, por consiguiente de fácil uso; así mismo, existe una confiabilidad de datos de 76%, la que puede ser mejorada mediante el uso de un analizador léxico-sintáctico y el empleo de agentes inteligentes que permitan realizar sugerencias considerando el perfil del usuario.

3.8. Validación

Según Pressman (PRESSMAN Roger, 2005) la validación del software se logra mediante una serie de pruebas que demuestran que se cumplen con los requisitos planteados inicialmente.

Identificador de requisito	Requisitos	Cumplimiento de requisito
REQ001	Extracción de información, metadatos	✓



	correspondientes a un OA.	
REQ002	Presentación de enlaces.	✓
REQ003	Comunicación intérprete – interfaz de usuario.	✓

Tabla 3.5: Verificación del cumplimiento de requisitos

Al desarrollar las pruebas de funcionamiento de la aplicación, se pudo verificar, la extracción, y procesamiento de los datos obtenidos de las ontologías, así como también la cantidad de errores para el cumplimiento de cada uno de los requisitos en base a pruebas con usuarios que validen la certeza de los resultados o enlaces obtenidos en sus consultas.

Identificador de requisito	No generan error (%)	Generan error (%)
REQ001	60	40
REQ002	80	20

Tabla 3.6: Porcentaje de errores generados en el cumplimiento de requisitos

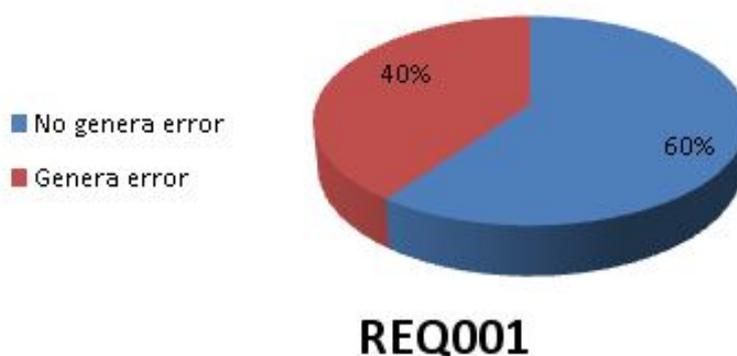
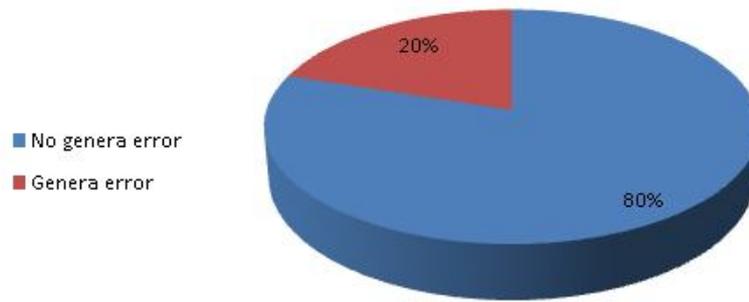


Figura 3.6: Porcentaje de errores en requisito REQ001



REQ002

Figura 3.7: Porcentaje de errores en requisito REQ002

El tercer requisito (comunicación intérprete-interfaz de usuario) se encuentra embebido o inmerso en los dos primeros; es decir, es el medio por el cual existe comunicación entre la ontología y la presentación de información hacia el usuario, razón por la cual no existe una validación que indique el porcentaje de aciertos y errores del mismo.



Discusión

Dentro de la UTPL en la Unidad de Virtualización, se maneja un Repositorio Digital de Objetos de Aprendizaje ROA, el que permite almacenar diversas clases de documentos digitales a los que se agrega un conjunto de metadatos, los que permiten describir a mayor detalle cada uno de los recursos digitales.

El ROA se encuentra disponible para que cualquier usuario de la Web pueda registrarse y hacer uso de esta herramienta; sin embargo, en la actualidad el repositorio no es utilizado en todo su potencial por la comunidad utepelina. Por lo que se hace necesario poner a disposición del estudiante el material educativo clasificado y catalogado en el repositorio, a través de la implementación de una ontología, que permita ofrecer recomendaciones de los recursos almacenados en el ROA por el contenido de cada una de las materias que curse el estudiante.

Para el desarrollo de la ontología fue necesario el estudio y el análisis comparativo de metodologías ontológicas que guían el proceso de creación de ontologías. Además el dominio del problema se dividió en dos subdominios, el primero permite representar parte de la oferta académica dentro de la UTPL, mientras que el otro permite la representación de los recursos almacenados en el repositorio mediante un perfil de aplicación del estándar LOM V1.0.

Una de las fases de mayor importancia dentro de la metodología con la cual se elaboraron las ontologías es la evaluación de las mismas, ya que mediante este proceso se obtiene un mayor refinamiento de las versiones ontológicas y nos permite medir el grado de reusabilidad de las mismas.

Las recomendaciones de Objetos de Aprendizaje son presentadas al estudiante mediante un módulo denominado "*Recomendaciones de Objetos de Aprendizaje provenientes del DSpace*" en el Entorno Virtual de Aprendizaje (EVA), para lo cual fue necesario la construcción de un servicio web que facilite la comunicación entre los lenguajes JAVA y PHP, el mismo que se encuentra disponible en la dirección <http://evadev.utpl.edu.ec:8080/serService/serWSDLService?wsdl>, y que hace uso del protocolo SOAP. Así mismo, existe la posibilidad de realizar consultas de recursos almacenados en el repositorio mediante un buscador que se encuentra implementado bajo la tecnología Java.

El buscador realiza consultas a través del motor de inferencias denominado Jena, para las cuales el criterio de búsqueda deber ser escrito de la misma forma a cómo se encuentran representadas las instancias dentro de los archivos de las ontologías.



Construida la ontología es necesario su instanciamiento, proceso que actualmente se realiza de forma manual en base a los recursos almacenados en el repositorio DSpace y al contenido de las materias que se ofertan en la universidad.

El acceso a las ontologías es pública y se lo puede hacer a través de las URLs http://evadev.utpl.edu.ec:8080/ser/ontology_files/dspace_model.owl y http://evadev.utpl.edu.ec:8080/ser/ontology_files/utpl_model.owl, para cualquier otro proyecto [semántico](#).

Trabajos Futuros

Para potenciar el uso de la ontología desarrollada es necesario realizar el instanciamiento del contenido de todas las asignaturas, al encontrarse el pensum académico sujeto a cambios, es necesario proponer un tema de investigación que permita realizar este proceso de manera semiautomática o completamente automática.

Por otra parte, el estándar LOM consta de nueve categorías y alrededor de setenta y cinco elementos, por consiguiente el ingreso de los metadatos es un proceso tedioso y lento, para suplir esta limitante es necesario proponer un mecanismo que permita el ingreso de los mismos de manera ágil y confiable.

Considerando la metodología de trabajo implícita en la Web 2.0 (trabajo colaborativo) y con el deseo de definir metadatos colaborativos sería importante un trabajo investigativo que permita pasar de una aplicación profile a una aplicación social, facilitando así el enriquecimiento de metadatos LOM. Para lo cual es necesario realizar un análisis del soporte de la herramienta DSpace y del proceso de ingreso de metadatos de forma colaborativa.

Con el objeto de agilizar la inclusión de la UTPL en la Web Semántica, es necesario realizar un estudio de las ontologías existentes y disponibles en la actual Web, a fin de determinar las que pueden ser reusadas, adaptadas e instanciadas.

Así mismo, para potenciar el desarrollo de aplicaciones semánticas, es necesario la investigación y elaboración de componentes adicionales a las ontologías, como los denominados "agentes inteligentes", esto permitirá en base a los diferentes perfiles de usuario brindar recomendaciones mucho mas personalizadas.



Conclusiones

- El desarrollo de ontologías (modelos DSpace y UTPL) que se encuentren accesibles al mundo, facilitan la definición de otros proyectos semánticos en donde se utilicen dominios definidos, verificados, evaluados y aprobados, como: el estándar LOM y oferta académica.
- El poseer una fuente de datos (como el caso del repositorio DSpace) para las recomendaciones asociadas a las materias que se imparten dentro de la UTPL, facilitan el incremento de individuos o instancias para los modelos ontológicos desarrollados.
- Para seleccionar la metodología de desarrollo de una ontología, es preciso considerar el dominio de la aplicación.
- En el desarrollo de ontologías es necesario pensar en instancias.
- El desarrollo de ontologías aplicadas a un proyecto en particular involucra el seguimiento de una serie de pasos, como: selección del lenguaje para la representación de la ontología, selección de la herramienta para la gestión de ontologías, investigación y selección de una metodología de desarrollo.
- Los metadatos constituyen un pilar fundamental dentro de la semántica ya que a través de ellos se puede identificar un recurso dentro de la web, pese a estas descripciones que los metadatos otorgan, existe una carencia de semántica en los mismos, es por esta razón el uso de ontologías que permite adicionar relaciones, axiomas y reglas de inferencia para definir un recurso dentro de la web.
- El desarrollo de un sistema recomendador brindará la posibilidad de acceder a OA que contengan información más precisa y que se encuentren relacionados a su oferta académica.
- La web semántica es una iniciativa que tiene como objetivo principal automatizar algunos procesos humanos, (servicios, búsqueda de información) a través de la dotación semántica a los recursos que se hallan en la web, lo que constituye un gran reto, debido a la cantidad de información que se halla en la web sin organizar y categorizar.



- En el contexto de la Web semántica, se considera metadatos, ontologías y siglas (RDF, XML, OWL...) con buscadores, sin embargo, aún no existen buscadores de carácter global que permitan realizar búsquedas en toda la Web, basándose en RDF o en metadatos (DC, LOM, IMS...)
- El ingreso de metadatos, dentro de la estructura ontológica, que al realizarse de forma manual resulta tedioso, confuso y hasta costoso, por lo que muchas veces es incorrecto.
- La Web semántica, en la actualidad, está constituida por un conjunto de islas semánticas, en donde se utilizan ontologías desarrolladas a la medida y motores de búsqueda que usan esa información; por consiguiente, la Web 3.0 es algo relativamente inicial.
- Cuando se habla de Web semántica se piensa en recuperación semántica de información, pero al no considerar un método raking, o la relevancia de un documento, es más apropiado denominar recuperación semántica de datos, ya que las búsquedas devuelven instancias específicas de un dominio (ontología).
- Se ha demostrado que es posible el desarrollo de un buscador semántico que hace uso de un razonador para inferir conocimiento almacenado en una ontología.
- Uno de los campos de la Web Semántica y que merece investigación es el mantenimiento y evolución de ontologías.
- La URL definida por defecto en un nuevo proyecto mediante la herramienta Protégé, permite el uso y consumo de la ontología, más no su acceso y reutilización. Cuando la ontología sea parte de otro proyecto se sugiere cambiar la URL, bajo el mismo dominio en el que se desarrollará el proyecto, y reemplazando directamente el prefijo de la ontología en el archivo owl.
- Las métricas de evaluación y validación de ontologías, a nivel estructural de: amplitud y profundidad, dependen de la representación lógica del dominio y directamente de las relaciones del tipo "es un".
- Para la aplicación de reglas de inferencia en una ontología es necesario el uso de un software adicional denominado Jess, el cual permite crear y validar el motor de inferencias.



Recomendaciones

- Para agregar contenido semántico a la Web se hace necesario que quienes crean el contenido en la Web, tengan la cultura del uso de metadatos para que los buscadores, que indexan en función de metadatos, nos den el resultado esperado.
- Utilizar mecanismos para la etiquetación automática para el instanciamiento de las ontologías ya que el proceso manual es una actividad, confusa, tediosa y hasta costosa.
- Para aprovechar el verdadero potencial de una ontología es conveniente el desarrollo de software que permita su explotación de una manera eficaz.
- Para futuros proyectos semánticos, que involucren el desarrollo de ontologías, se debe especificar la URL en donde se encontrará alojada la ontología, esto con la finalidad de identificarla como recurso único dentro de la Web, haciendo posible su acceso y reutilización.
- Adquirir la licencia del software Jess, lo cual permitirá una correcta definición y manipulación de inferencias



Bibliografía

- BARCHINI Graciela, Á. M. (2006). JISTEM - Journal of Information Systems and Technology Management. Revista de Gestão da Tecnologia e Sistemas de Informação,3-18.
Disponble en: <http://www.tecsi.fea.usp.br/revistatecsi/edicoesanteriores/v03n01-2006/v3n1a1.pdf>
- BRANK, J., Grobelnik, M., & Mladenié, D. (s.f.). A SURVEY OF ONTOLOGY EVALUATION TECHNIQUES. Ljubljana, Slovenia. Disponible en: <http://kt.ijs.si/dunja/sikdd2005/Papers/BrankEvaluationSiKDD2005.pdf>
- CORCHO, M. F.-L.-P. (2001). Methodologies tools and languages for building ontologies. Where is their meeting point? Universidad Politecnica de Madrid, Campus de Montegancedo s/n, Madrid 28660, España. Disponible en: http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf
- CRIADO, L. (06 de 06 de 2008). Web Semántica. Recuperado el 25 de 02 de 2009, Disponible en: <http://lcriadof.blogspot.com/2008/06/situacin-de-los-buscadores-semnticos.html>
- DING, L., Finin, T., Joshi, A., Peng, Y., Pan, R., & Reddivari, P. (s.f.). Search on the Semantic Web. University of Maryland, Baltimore County, USA. Disponible en: <http://www2.computer.org/plugins/dl/pdf/mags/co/2005/10/rx062.pdf?template=1&loginState=1&userData=anonymous-IP1235611200520>
- Dublin Core Metadata Initiative-DCMI Metadata Terms. (14 de Enero de 2008). Recuperado el 20 de Febrero de 2008, Disponible en: <http://dublincore.org/documents/dcmi-terms/>
- Dublin Core Metadata Initiative-Using Dublin Core - Dublin Core Qualifiers. (07 de Noviembre de 2005). Recuperado el 20 de Febrero de 2008, de <http://dublincore.org/documents/usageguide/qualifiers.shtml>
- FENSEL D, V. H. (1999). Project Presentation On-To-Knowledge: Content-driven Knowledge-Management Tools through Evolving Ontologies. Vrije Universiteit Amsterdam. Disponible en: <http://www.ontoknowledge.org/download/slides/otk2-1.pdf>
- FENSEL D. Van Hermelen F., K. M. (s.f.). On-To-Knowledge: Ontology-based Tools



for Knowledge Management. Free University Amsterdam VUA, Division of Mathematics and Informatics De Boelelaan 1081a, NL-1081 HV Amsterdam, The Netherlands. Disponible en: <http://www.cs.vu.nl/~frankh/postscript/eBeW00.pdf>

- GANGEMI, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (s.f.). A theoretical framework for ontology evaluation and validation. Roma, Italia. Disponible en: http://www.loa-cnr.it/Papers/swap_final_v2.pdf
- GARCÍA, R. (2005). Un Entorno para la Extracción Incremental de Conocimiento desde Texto en Lenguaje Natural. TESIS DOCTORAL. Universidad de Murcia. Departamento de Ingeniería de la Información y las Comunicaciones. Disponible en: http://www.tdx.cbuc.es/TDX/TDR_UM/TESIS/AVAILABLE/TDR-1123105-140512//rvalencia.pdf
- GRIGORIS Antoniou, V. H. (2004.). A Semantic Web Primer. The MIT Press Cambridge, Massachusetts, London, England. Disponible en: <http://www.ulb.tu-darmstadt.de/tocs/121667529.pdf>
- GRUBER Thomas R. (1993). A Translation Approach to Portable Ontology Specifications. KNOWLEDGE SYSTEMS LABORATORY Computer Science Department Stanford University Stanford, California. Disponible en: <http://tomgruber.org/writing/ontolingua-kaj-1993.htm>
- HAPPEL Hans-Jörg, SEEDORF. Stefan. (s.f.). Application of ontologies in Software Engineering. Universität Mannheim, Lehrstuhl für Wirtschaftsinformatik III, Schloss, 68131 Mannheim, Germany. Disponible en: http://km.aifb.uni-karlsruhe.de/ws/swese2006/final/happel_full.pdf
- HARTH, A., Jürgen Umbrich, S., & Hogan, A. (2008). Building a Semantic Web Search Engine: Challenges and Solutions. National University of Ireland, Galway. XTech Conference. "The Web on the move". 6 – 9 May 2008. Dublín Ireland. Disponible en: <http://2008.xtech.org/public/schedule/detail/477>
- HELFIN J, H. J. (s.f.). SHOE. Recuperado el 05 de Enero de 2008, Disponible en: <http://www.cs.umd.edu/projects/plus/SHOE/>
- HURTADO, D. (2006). Desarrollo de un módulo ontológico para las consultas avanzadas de información de la biblioteca digital de la escuela de ingeniería en sistemas y computación de la Universidad del Valle. Universidad del Valle. Facultad de Ingeniería.
- JIM, H. (2004). Preguntas frecuentes sobre el Lenguaje de Ontologías Web (OWL)



del W3C. Recuperado el 17, de Febrero de 2008, Disponible en:
<http://www.w3c.es/Traducciones/es/SW/2005/owlfaq>

- LOZANO Tello, A. (Febrero de 2002). MÉTRICA DE IDONEIDAD DE ONTOLOGÍAS. TESIS DOCTORAL. Disponible en:
<http://quercusseg.unex.es/adolfo/tesis.htm>
- Metadato. (s.f.). Wikipedia. Recuperado el 18 de Febrero de 2008, Disponible en:
<http://es.wikipedia.org/wiki/Metadatos>
- MILLS, D. (Febrero de 2008). Project10X's Semantic Wave 2008 Report: Industry Roadmap to Web 3.0 & Multibillion Dollar Market Opportunities, Executive Summary. Disponible en:
http://www.researchandmarkets.com/reports/682250/semantic_wave_report_industry_roadmap_to_web_3_0
- Movva, S., Ramachandran, R., Li, X., Cherukuri, P., & Graves, S. (s.f.). Noesis: A Semantic Search Engine and Resource Aggregator for Atmospheric Information Technology and Systems Center, University of Alabama in Huntsville, Huntsville, AL 35899. Disponible en:
http://esto.nasa.gov/conferences/nstc2007/papers/Ramachandran_Rahul_A3P4_NSTC-07-0084.pdf
- OWL. (s.f.). Wikipedia. Recuperado el 17 de Febrero de 2008, Disponible en:
<http://es.wikipedia.org/wiki/OWL>.
- Pérez, C. (s.f.). Estudios de Lingüística Española (ELiEs). Recuperado el 12 de Enero de 2008. Disponible en: <http://elies.rediris.es/elies18/531.html>
- PRESSMAN, R. (2005) Ingeniería de Software un enfoque práctico. Sexta edición. México. Mc Graw Interamericana de Editores.
- Protégé. (s.f.). Recuperado el 18 de Mayo de 2008, Disponible en:
<http://protege.stanford.edu>
- ROBSON, Roby. (s.f.). A MATHEMATICIAN'S GUIDE TO IMS METADATA. Recuperado el 20 de Febrero de 2008, Disponible en:
<http://mathmetadata.org/ammf/docs/mathguide>
- SÁNCHEZ FERNÁNDEZ Luis, FERNÁNDEZ GARCÍA Norberto. La Web Semántica: fundamentos y breve "estado del arte" (2005). Revista Novatica No. 178, Noviembre – Diciembre 2005, año XXXI. Disponible en:
<http://www.ati.es/novatica/2005/178/178-6.pdf>



- WIKIPEDIA. (s.f.). Resource Description Framework. Recuperado el 17 de Febrero de 2008, Disponible en: http://es.wikipedia.org/wiki/Resource_Description_Framework
- WIKIPEDIA. (s.f.). Wikipedia-Cyc. Recuperado el 6 de Marzo de 2008, Disponible en: <http://es.wikipedia.org/wiki/Cyc>
- WIKIPEDIA. (s.f.). Wikipedia-Learning object metadata. Recuperado el 20 de Febrero de 2008, Disponible en: http://en.wikipedia.org/wiki/Learning_object_metadata
- YAO, H., ORME, A. M., & ETZKORN, L. (2005). Cohesion for Metrics Ontology Design and Application. Hunsteville, Alabama 35889, United States of America. Disponible en: <http://www.scipub.org/fulltext/jcs/jcs11107-113.pdf>



ANEXOS



ANEXO I
MANUAL DEL ADMINISTRADOR



1. Implementación de la Ontología

1.1. *Protégé*

Protégé es un editor de ontologías de código abierto, así como también framework para bases de conocimiento. Fue desarrollado por la universidad de Stanford bajo el lenguaje Java (Protégé), lo que permite a su vez que este editor sea multiplataforma.

Las ventajas que trae consigo el uso de esta herramienta son:

- Permite un modelado muy sencillo, posee varios modos de visualización que facilitan determinar la jerarquía y relaciones entre las clases (comúnmente en Protégé se denominan clases a los conceptos).
- El modelo de la ontología pueden ser exportado dentro de una variedad de formatos, como: RDF(S), OWL, and XML Schema.
- Provee un API con la que se facilita la creación de programas de exploración y edición de ontologías.
- Cuenta además con una gran comunidad de usuarios de desarrolladores, academias, y gobierno quienes usan Protégé para soluciones de diversas áreas como biomedicina, educación, etc.
- Posee gran variedad de plugins que pueden ser añadidos para conectarse a bases de datos tales como: Mysql, Oracle, etc.

1.2. *Creación de la ontología*

Para la representación del modelo de la ontología se ha escogido el editor Protégé, por las características anteriormente mencionadas, además se utilizó la versión Protégé 3.3.1, la cual es una versión estable, y que puede ser descargada de la dirección: <http://protege.stanford.edu/>

Para la creación de un nuevo proyecto de ontología se procede de la siguiente manera:

- Hacemos doble clic en el ícono de Protégé
- Ir a la barra de menú y hacemos clic en el ítem **File>>New Project**. Luego de esto aparecerá un asistente para la creación del proyecto.



Figura 1: Menú de Protege

- Aparecerá un asistente en el cual se encuentra marcada la opción "OWL/RDF Files" que nos permite expresar la ontología en cualquiera de estos dos lenguajes - OWL ó RDF-, damos clic en **Next**.

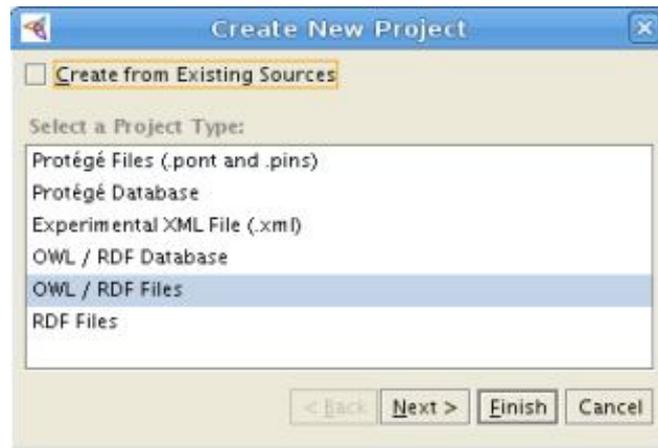


Figura 2: Tipo de proyecto

- Luego el asistente nos solicita escribir la URL. La URL nos permite identificar donde se encuentra ubicada la ontología, para utilizarla en otros proyectos de ontologías, es decir importar el modelo ontológico a través de su URL. Hacemos clic en **Next**.



Figura 3: URL de la ontología

- En la siguiente pantalla se muestra los tipos de lenguaje OWL y RDF, para lo cual seleccionamos el lenguaje OWL DL -OWL Description



Logic -, que nos va a permitir la máxima expresividad -definir restricciones, cardinalidad, etc-, sin exigir tanto costo computacional para el razonamiento (Wikipedia-OWL). Hacemos clic en siguiente.



Figura 4: Lenguaje de la ontología

- Por último se pide que se escoja la forma en la que se visualizará las clases, dentro del editor Protégé.



Figura 5: Selección de la vista de clases dentro de Protégé

1.3. Representación de los conceptos en Protégé

La representación de los conceptos mediante Protégé se los puede realizar al ubicarnos en la pestaña **OWLClasses**, e ir al panel izquierdo **Subclass Explorer** para crear las clases de dos formas



- La primera de ellas es mediante el ícono Create subclass , que permitirá crear una a una cada clase.
- La segunda forma es mediante la ayuda de un wizard que permitirá la creación de todas las clases y subclases a la vez.

Sea cual sea el método usado para la representación de los conceptos, todas las clases creadas serán subclases de owl:Thing, esto es debido a que en Protégé owl:Thing es una clase que representa el conjunto que contiene a todas las otras clases de nuestros modelos.

En la creación de los conceptos de la ontología base se usó el segundo método, para lo cual debemos seguir los siguientes pasos:

- Hacemos clic en la pestaña **OWLClasses**.
- Hacemos clic derecho en la clase owl:Thing y escogemos del menú la opción: **Create subclasses**.
- Aparece un cuadro de diálogo en donde se explica el proceso de creación de múltiples clases. Hacemos clic en Next.
- Luego aparece un cuadro en el cual se selecciona la clase padre o superclase de la que crearemos la(s) subclase(s). Hacemos clic en Next.
- En la siguiente ventana nos aparece un área de texto en donde podemos crear múltiples clases simplemente tipeando el nombre de la clase en una nueva línea del cuadro de texto. Hacemos clic en Next.



Figura 6: Creación de múltiples clases

- En la ventana siguiente, la opción de hacer que todas las clases sean disjuntas se encuentra activada, esto permite que no exista ningún tipo de



dependencia entre las clases. Hacemos clic en Next.



Figura 7: Opción para hacer que las clases sean disjuntas entre ellas.

- Finalmente aparece una ventana donde se muestra un resumen de los pasos realizados para la creación de múltiples clases. El resultado será:



Figura 8: Resultado de creación de clases.

1.4. Propiedades: Object, datatype y annotation

Las propiedades son relaciones binarias que se definen para las clases y que afectan a cada una de sus instancias, por ejemplo, la relación binaria “*agrupa*” entre las clases: área y carrera, se representa a nivel de instancias de la siguiente forma:

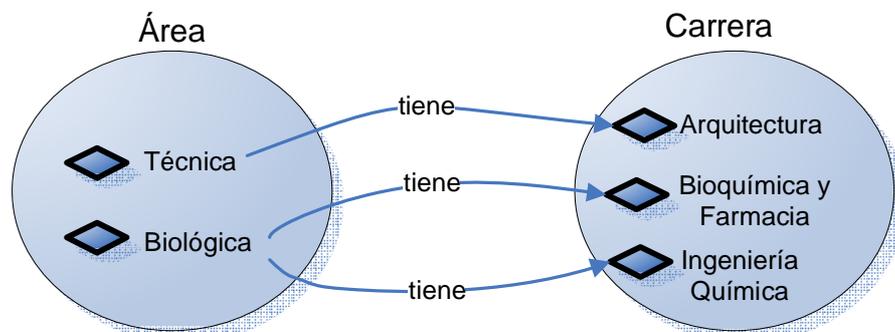


Figura 9: Relación binaria agrupa



Dentro de Protégé se pueden definir tres tipos de propiedades que son: object, datatype y annotation. Las propiedades de tipo object nos permiten enlazar instancia a instancia, en otras palabras relaciones entre instancias, mientras que los datatype permiten definir valores que tendrán las instancias, tales como: nombre string, precio float, etc. Las propiedades de tipo annotation permiten especificar información a las clases (meta datos) tales como creador, fecha, etc.

1.4.1. Representación de las relaciones mediante propiedades Object

Para definir las relaciones binarias en la herramienta Protégé, realizamos los siguientes pasos.

- Hacemos clic en la pestaña **Properties**.
- La interfaz que aparece se divide en dos paneles, el izquierdo se llama property browser, el mismo que se divide en tres sub pestañas donde se encuentran las propiedades object, datatype y annotation- y el derecho property editor, donde se especifica el dominio y/o rango denominado para la propiedad.
- En el panel property browser hacemos clic en la pestaña: Object, creamos una nueva propiedad mediante el ícono  y la denominamos "agrupan".
- Ahora hay que especificar el dominio y rango que tendrá la propiedad (dominio y el rango son las clases que se van a relacionar), para lo cual nos ubicamos en el panel property editor, en la parte inferior en el de dominio, especificamos la clase **área**. De forma similar especificamos de dominio, especificamos la clase **carrera** como rango para la propiedad que acabamos de crear.
- El resultado es el siguiente:

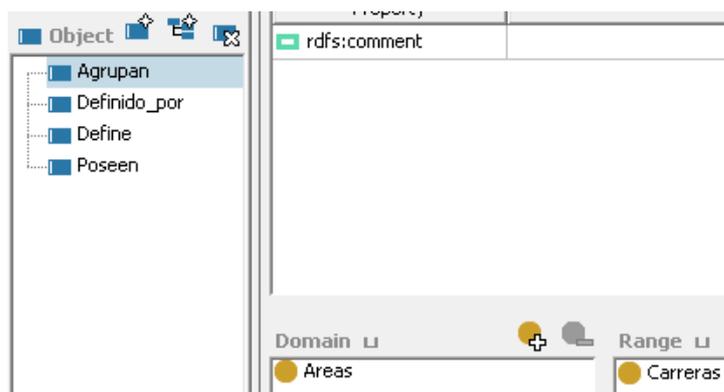


Figura 10: Creación de propiedad object

1.4.2. Representación de propiedades datatype

Las propiedades datatype pueden ser descritas como atributos o características que tienen las clases, y que pertenecen a un tipo de datos en específico, el mismo que se define en XML Schema Datatype (describe tipos de datos básicos, tales como: int, float, string, etc.). Para la creación de propiedades datatype mediante Protégé realizamos lo siguiente:

- Hacemos clic en la pestaña **Properties>>Datatype**.
- Creamos la propiedad llamada oferta, haciendo clic sobre el ícono .
- Especificamos la clase a la cual se le aplicará la nueva propiedad creada, para ello hacemos clic en el icono de dominio para escoger la clase asignaturas.
- Especificamos el rango; es decir, el tipo de datos que tendrá la propiedad. Seleccionamos el tipo de dato: string, que permitirá definir unos valores por defecto de tipo cadena para esta propiedad.
- En la sección: Allowed values, definimos tres valores que son:
 - Pregrado.
 - Postgrado.
 - Pregrado y postgrado.
- La creación de los valores permitidos para la propiedad oferta se muestran en la siguiente figura:



Figura 11: Valores permitidos para la propiedad oferta

1.5. Creación de restricciones

Como su nombre lo sugiere, las restricciones limitan a las instancias que pertenecen a una clase; es decir, al momento de crear una instancia de una determinada clase además de establecer el nombre de la instancia es necesario definir las relaciones y atributos existentes entre instancias, además las restricciones hacen uso de las propiedades de tipo object y datatype. Entre los tipos de restricciones encontramos.

- Restricciones con cuantificadores.
- Restricciones de cardinalidad.

1.5.1. Restricciones con cuantificadores

Su estructura está compuesta de: una propiedad, un cuantificador, y una clase destino, en esta clase de restricciones los cuantificadores son de dos tipos:

Cuantificador	Símbolo	Lectura	Explicación
			Clases: A y B, Propiedad: tieneElemento
Existencial	(\exists)	“al menos un” o “algún”	Todas las instancias de A tienen al menos un elemento de B.
Universal	(\forall)	“sólo”	Las instancias de A sólo tienenElemento(s) correspondientes a instancias que pertenecen a la clase B.

Cuadro 1: Cuantificadores



Existencial

- **Elementos**
 - Instancias de la clase Área
 - Instancias de la clase Carrera
 - Propiedad: Agrupan
- **Restricción:** Cada área *tiene* al menos a una carrera



Figura 12: Cuantificador existencial

- **Creación de instancia: Socio Humanística**

La aplicación de la restricción implica que: al crear una nueva instancia de la clase área, se debe definir al menos una instancia perteneciente a la clase carrera.



Figura 13: Creación de instancia al aplicar la restricción existencial

Universal

- **Elementos**
 - Instancias de la clase Asignatura
 - Instancias de la clase Contenido
 - Propiedad: *estudia*
- **Restricción:** La asignatura estudia algún contenido



Figura 14: Cuantificador universal

- **Creación de instancia.**

Al crear una nueva asignatura, la restricción implica que todas las relaciones del tipo *estudia* deben ser un miembro de la clase contenido.



Figura 15: Creación de instancia al aplicar la restricción universal

1.5.2. Restricciones de cardinalidad

Poseen la siguiente estructura: propiedad, tipo cardinalidad, y número, son usadas para definir la cantidad de relación o valores que tendrán las instancias; es decir, son de tipo cuantitativas. El siguiente cuadro muestra las clases de restricciones de cardinalidad existentes.

Cuantificador	Símbolo	Explicación
		Clases: A y B, Propiedad: tieneElemento
Igual	=	Las instancias de la clase A se relacionan con un cantidad específica de instancias de la clase B, no mayor, ni menor a la cantidad especificada.
Máximo	>=	Las instancias de la clase A se relacionan con al menos una cantidad específica de instancias de la clase B
Mínimo	<=	Las instancias de la clase A se relacionan con las instancias de la clase B, con una cantidad no mayor a la especificada

Cuadro 2: Tipos de cardinalidad

Cardinalidad mínima

Permite describir las instancias que tienen “como mínimo” o “exactamente “, un número específico de relaciones con otras instancias o valores datatype.

- **Elementos**

- Instancias de la clase Asignatura
- Instancias de la clase Contenido



▪ **Propiedad:** *estudia*

- **Restricción:** Las asignaturas se encuentran definidas como mínimo por dos contenidos



Figura 16: Cardinalidad mínima

- **Creación de instancia**

La restricción implica que es necesario definir dos contenidos como mínimo para cada nueva asignatura



Figura 17: Creación de instancia al aplicar la restricción de cardinalidad mínima

Cardinalidad máxima

- **Elementos**

- Instancias de la clase área
- Instancias de la clase carrera
- Propiedad: *tienen*

- **Restricción:** Cada área se encuentra definida por un máximo de cuatro carreras.



Figura 18: Cardinalidad máxima

- **Creación de instancia**

La restricción implica que es necesario definir como máximo 4 carreras para cada nueva instancia de área.





Figura 19: Creación de instancia al aplicar la restricción de cardinalidad máxima

Cardinalidad igual

- **Elementos**
 - Instancias de la clase carrera
 - Instancias de la clase asignatura
 - Propiedad: *imparte*
- **Restricción:** Cada carrera posee exactamente 4 instancias de asignatura



Figura 20: Cardinalidad igual

- **Creación de instancia**
La restricción implica que es necesario definir 4 asignaturas para cada carrera, no una cantidad mayor ni menor a esta.



Figura 21: Creación de instancia al aplicar la restricción de cardinalidad igual.

1.5.3. Clases Enumeradas

Mediante el uso de Protégé podemos definir una lista de individuos que serán miembros de una clase.

	Símbolo	Representación
Clases enumeradas	{Elemento1;Elemento2}	Las instancias de A, serán uno de los valores especificados en la lista de instancias

Cuadro 3: Clases enumeradas

- **Elementos**
 - Clase RecursosDigitales



- **Lista de elementos:** papers, revistas digitales, presentaciones, videos, enlaces web.



Figura 22: Lista de instancias para la clase recursos digitales

1.6. *Instanciamiento de ontologías*

El instanciamiento de ontologías es un proceso que se debería realizar al final del proceso de creación del modelo ontológico; es decir, luego de: la creación de conceptos, definición de relaciones (propiedades object y datatype), definición de restricciones, etc.; estos pasos permiten definir todos los requisitos necesarios para crear una instancia de un concepto en particular.

A continuación se presenta el proceso para crear una instancia perteneciente a cualquier clase es el siguiente:

- Hacer clic en la pestaña **Individuals**.
- Hacer clic sobre el concepto a crear la(s) instancia(s).
- Hacer clic en el icono , ubicado en el panel **Instante Browse**, para crear una nueva instancia
- Cambiar el nombre de la instancia en el cuadro de texto **For Individual** (posee la url que hace referencia al nombre de la instancia), en el panel **Individual Editor**.
- En el caso de haber detallado atributos u objetos de tipo datatype en el mismo panel **Individual Editor** se detallan los mismos, para crear un nuevo valor del atributo hacer clic en el botón ubicado en la parte superior del atributo.
- Si la clase, en la que se está creando una instancia posee relaciones, y/o restricciones, es necesario (mas no obligatorio), definir las instancias pertenecientes al otro concepto o clase relacionada. Esto se realiza mediante el botón , para crear una nueva instancia, caso contrario se procede a seleccionar instancias previamente creadas a través del botón .



1.6.1. Ontología UTPL

Instanciamiento de la clase Asignatura

- **Relación:**
 - Directa: estudia, Asignatura-Contenido.
 - Inversa: se estudia en: Asignatura-Carrera.
- **Atributos:**
 - Nombre asignatura String
 - Ciclo Lista (String)
 - Modalidad Lista (String)
 - Oferta académica Lista (String)
 - Recomendaciones String
 - Tipo Lista (String)

Al crear una nueva instancia correspondiente al concepto Asignatura, se disponen o visualizan slots o casillas correspondientes tanto a los atributos como a las relaciones definidas anteriormente.

Algunos atributos poseen una lista de valores predefinidos que fueron añadidos al momento de crearlo, esto sirve cuando se desea crear un nuevo valor al atributo; para el efecto se debe seleccionar uno de los valores disponibles de la lista.

En cuanto a las relaciones: directa e inversa; definidas anteriormente, se realiza el proceso general de instanciación.

Para la definición de instancias de la relación estudia se selecciona o crea instancias correspondientes a la clase Contenido; es decir, se define el contenido estudiado en una determinada asignatura.

Para la relación se estudia en, se selecciona o crea instancias que pertenecen al concepto Carrera, con esto se define en la ontología las carreras en las que se imparte una determinada asignatura.

En las siguientes figuras se puede apreciar la creación de una instancia correspondiente a la clase: Asignatura.

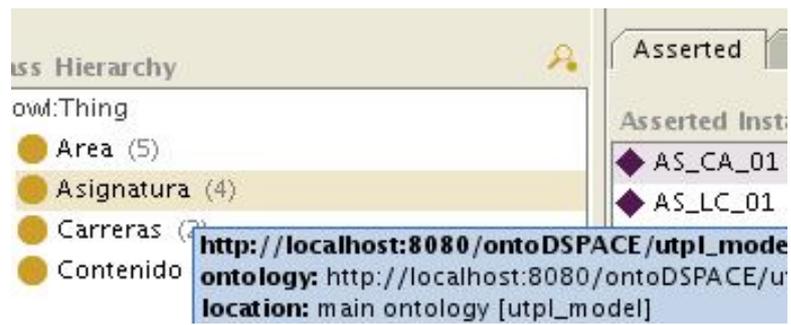


Figura 23: Selección del concepto

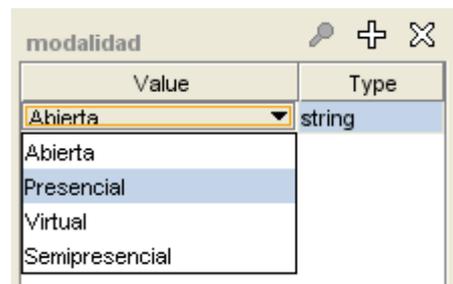


Figura 24: Creación del valor para el atributo *modalidad*

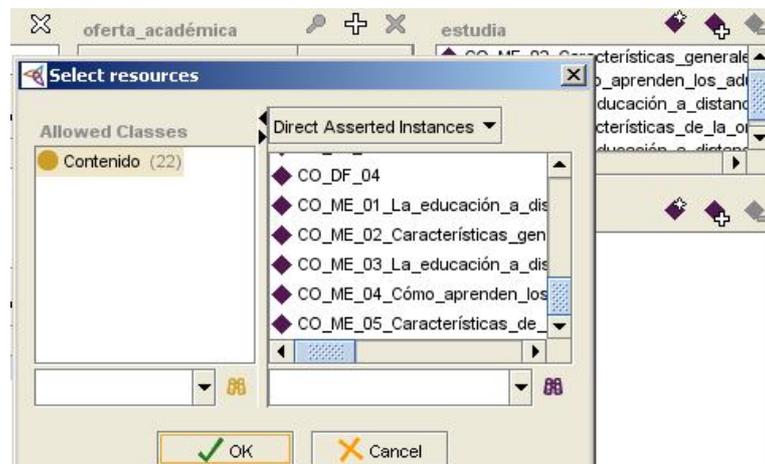


Figura 25: Definición de instancias para la relación *estudia*



UTPL: Área

The screenshot shows the Protege software interface with three main panes:

- CLASS BROWSER:** Shows a class hierarchy for the project 'utpl_model'. The class 'utpl_model:Area (5)' is highlighted with a red box.
- INSTANCE BROWSER:** Shows 'Asserted Instances' for the class 'utpl_model:Area'. The instances listed are: utpl_model:AR_01_Administrativa, utpl_model:AR_02_Biológicas, utpl_model:AR_03_Socio-Humanística, utpl_model:AR_04_Técnica (highlighted), and utpl_model:AR_05_Todas_las_areas.
- INDIVIDUAL EDITOR for utpl_model:AR_04_Técnica:** Shows the URI 'http://localhost:8080/ser/ontology_files/utpl_model.owl#AR_04_Técnica'. It contains a table for 'rdfs:comment' with the following text: 'Está conformada por las escuelas con las que la Universidad abrió sus puertas como: Ingeniería Civil, Arquitectura y cuenta actualmente con tecnología e infraestructura de última generación, que permite potenciar el proceso de enseñanza - aprendizaje.' Below the table, there are two properties: 'utpl_model:nombre_area' with the value 'Técnica' and 'utpl_model:tiene' with two instances: 'utpl_model:CA_TE_01_Sistemas_Informáti' and 'utpl_model:CA_TE_02_Informática'.

Figura 26: Instanciamiento de la clase utpl_model:Area



1.6.2. Ontología DSpace-LOM

Al igual que el caso de la ontología UTPL el proceso de creación de instancias es similar, depende de los atributos, relaciones y restricciones definidas para cada concepto perteneciente al modelo DSpace.

Instanciamiento de la clase General.Identificador.Entrada

- **Relación:**
 - Directa: tieneGeneralTitulo, Catalogo-Titulo
 - Directa: tieneGeneralPalabrClave, Catalogo-PalabraClaves
 - Directa: tieneTécnicaFormato, Catalogo-Formato, entre otras relaciones.

En todas las instancias pertenecientes al concepto General.Identificador.Entrada se deben definir otras instancias pertenecientes a otras clases; esto es, debido a que en esta clase se han definido todas las relaciones que un OA puede tener, como por ejemplo: titulo, autor, fecha, etc.

A continuación se muestran algunas figuras pertenecientes al proceso de creación de instancias del modelo DSpace.



Figura 27: Creación de instancia en la clase Entrada

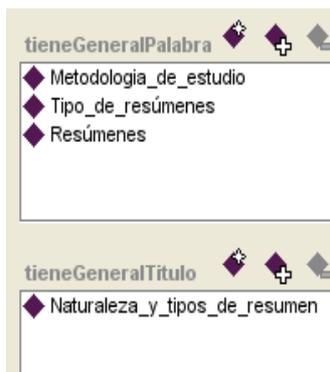


Figura 28: Definición de instancias para las relaciones pertenecientes a la clase General.Identificador.Entrada



DSPACE: General>>Estructura

The screenshot displays the Protege ontology editor interface. It is divided into three main panes:

- CLASS BROWSER:** Shows the class hierarchy for the project 'dspace_model'. The 'General' class is expanded, and 'Estructura' (with 52 instances) is highlighted with a red box.
- INSTANCE BROWSER:** Shows instances for the selected class 'Estructura'. The 'Asserted' tab is active, displaying a list of instances such as 'Atómica_apartado', 'Atómica_cuestionario', 'Atómica_curso', etc.
- INDIVIDUAL EDITOR for Jerárquica_apartado (instance of Estructura):** Shows the URI for the instance and a table for editing its properties. The table has columns for 'Property' and 'Value'. A row for 'rdfs:comment' is visible. Below the table, there are sections for 'Apartado', 'Evaluacion', 'Cuestionario', and 'Modulo', each with a table for editing values and types.

Figura 29: Instanciamiento de la clase General>>Estructura



ANEXO II CÓDIGO FUENTE



1. Arquitectura de la aplicación

El sistema denominado ser se encuentra formado por tres capas que son:

- Capa lógica
- Capa de servicios Web
- Capa de presentación

1.1. Capa lógica

La capa lógica es la encargada del procesamiento de la información y consultas generadas hacia las ontologías; es decir, la capa lógica constituye la base dentro del sistema.

Para la comunicación con la ontología fue necesario trabajar con un framework semántico denominado Jena, el cual permite la construcción de aplicaciones semánticas, y que se lo puede descargar desde: <http://jena.sourceforge.net/>.

En el siguiente fragmento de código se muestra la forma de comunicarse con una ontología a través de Jena

Clase: QuerybyIdSubject.java

```
/*
 * QuerybyIdSubject.java
 *
 * Created on 8 de julio de 2008, 10:47
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

public static void main(String[] args)
{
    try {
        ConsumerOntology.ConsumerUTPLOntology UTPLModel =
        new
            ConsumerOntology.ConsumerUTPLOntology();
        .....
    }catch(Exception e){}
}
```

Clase: ConsumerUTPLOntology

```
/**
 *
 * @author manuel
 */
public class ConsumerUTPLOntology
```



```
{
    //ontologia utpl
    private String URI =
        "/home/manuel/NetBeansProjects/ser/web/ontology_files/
        utpl_model.owl";
    private OntModel model; //El modelo de la ontologia usado

    public ConsumerUTPLOntology()
    {
        this.model = new OntologyCreation.InstanceOntology(URI).getModel();
    }

    public OntModel getModel()
    {
        return model;
    }

    public void setModel(OntModel model)
    {
        this.model = model;
    }
}
```

En esta capa se han creado las clases necesarias para el consumo de ambas ontologías: UTPL y DSpace, las clases correspondientes son:

UTPL	DSpace
Contenido	1. LOM <ul style="list-style-type: none">○ Anotation○ General○ LifeCycle○ Metadata○ EducationalUse○ Relation○ Classification○ Right○ Technical
Clase contenedora de toda la información: DataContainer	

Constructor de la clase: DataContainer

```
public DataContainer(String Area, String Carrera, String Asignatura,
    String OfertaAcademica, String Ciclo,
    String Tipo, String Modalidad, ArrayList<Contenido>
    content, ArrayList<LOM> lom)
{
    this.Area = Area;
    this.Carrera = Carrera;
    this.Asignatura = Asignatura;
    this.OfertaAcademica = OfertaAcademica;
    this.Ciclo = Ciclo;
}
```



```
this.Tipo = Tipo;  
this.Modalidad = Modalidad;  
this.content = content;  
this.lom = lom;  
}
```

El proyecto fue realizado en Netbeans 6.1 para la creación de las capas (lógica, interfaz de usuario, servicios web), para esto fue necesario añadir a los proyectos hechos en lenguaje Java, la librería Jena, esto con la finalidad de la compilación y ejecución de los proyectos.

Para realizar esto es necesario hacer lo siguiente:

1. Descargar el paquete Jena de: <http://jena.sourceforge.net/>, luego descomprimirlo (puede ser ubicado en un directorio donde se encuentren el resto de librerías Java).
2. Abrir el IDE Netbeans 6.1.
3. Ir al menú Tools>>Libraries. Se abre una nueva ventana llamada “**Library Manager**”.

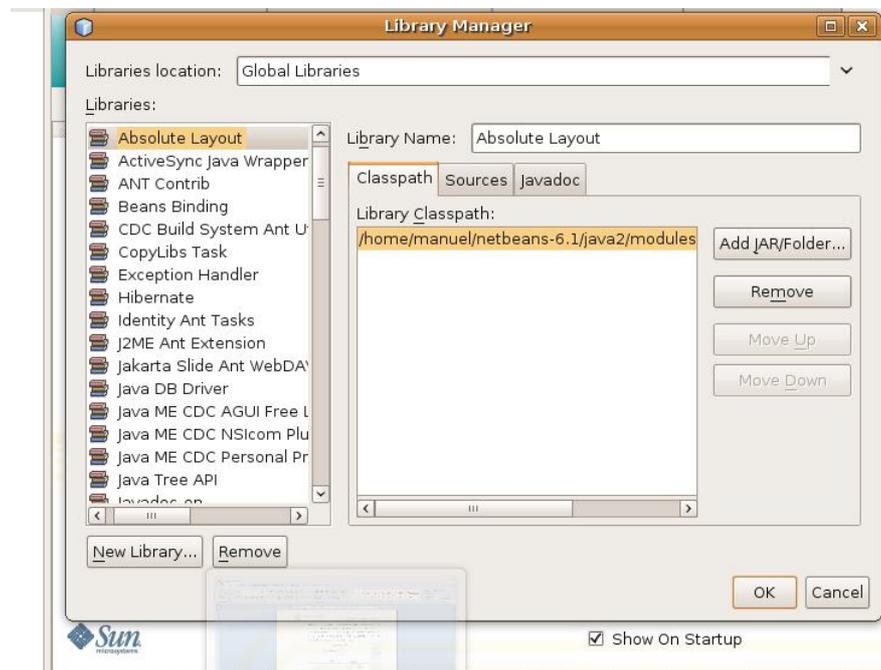


Figura 1: Cuadro de diálogo Library Manager

4. En la parte inferior de la ventana, existe un el botón “New Library”, en la que se abre un cuadro de diálogo para especificar el nombre de la nueva librería. Colocamos el nombre de Jena.

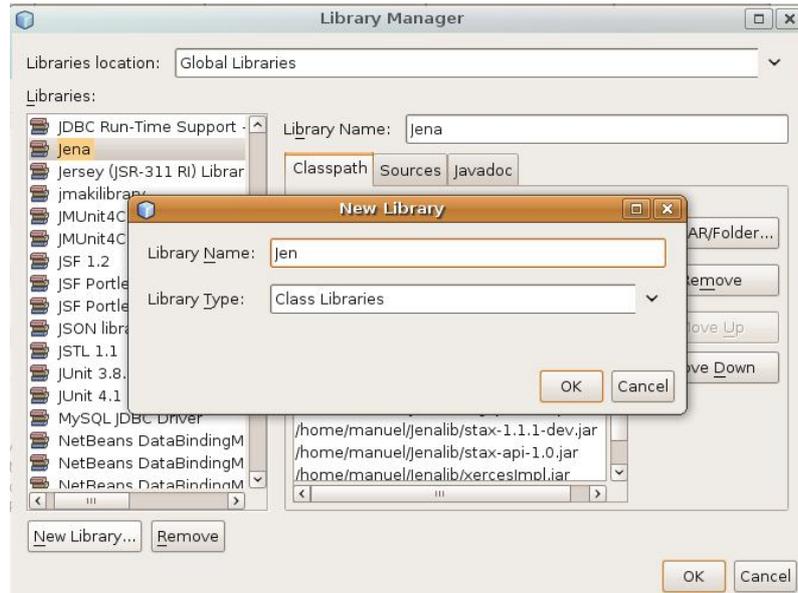


Figura 2: Adición de una nueva librería dentro de Netbeans

5. En la parte derecha de la ventana “Library Manager”, se encuentra un área de texto en donde se muestran todas las clases correspondientes a la nueva librería. Hacemos clic en el botón “Add JAR/Folder”, para ubicar las clases correspondientes al paquete Jena.

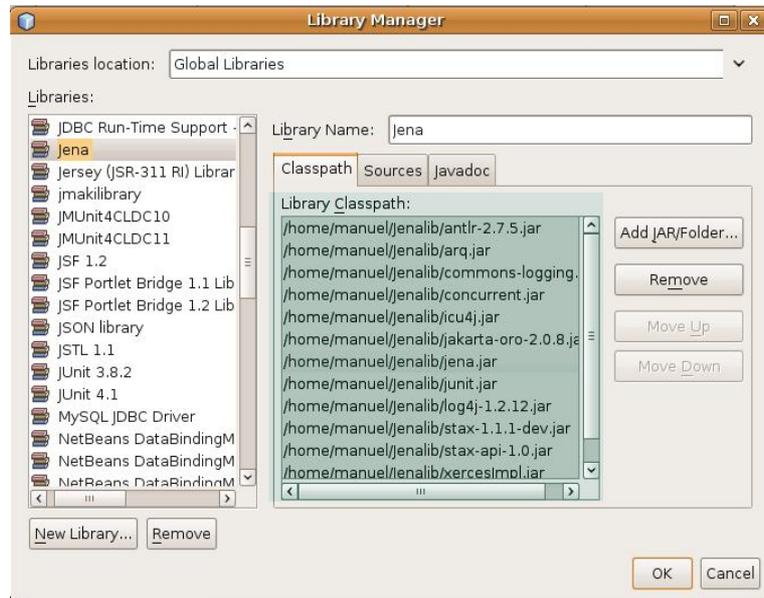


Figura 3: Selección de clases que pertenecen a una librería

6. Luego hacer clic en el botón “OK” de la ventana “Library Manager”.



1.1.2. Añadir la librería Jena a un proyecto Java

Para realizar el consumo de ontologías dentro de Netbeans, es necesario vincular la librería Jena dentro del proyecto Java, para lo cual realizamos lo siguiente:

1. Abrir Netbeans 6.1 IDE.
2. En la pestaña **"Projects"**, hacer clic sobre el proyecto, esto con la finalidad de mostrar la estructura de árbol dentro de Netbeans, con la que se encuentra conformado el proyecto.



Figura 4: Estructura de árbol de un proyecto dentro de Netbeans

3. Hacer clic derecho sobre la carpeta "Libraries", aparecerá un menú con la opciones de: añadir un proyecto, un archivo Jar o una librería.
4. Hacer clic en la opción "Add Library", aparecerá todas las librerías disponibles en Netbeans y aquellas añadidas por el usuario. Seleccionar la librería Jena anteriormente creada.

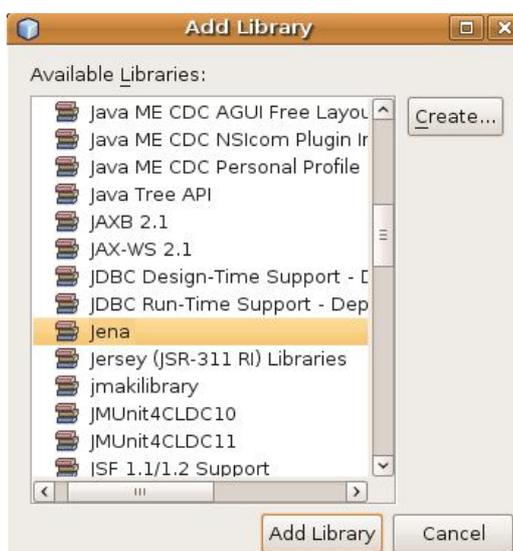


Figura 5: Selección de la librería Jena



5. Hacer clic en el botón “Add Library”.

1.2. Capa de servicios Web

La capa de servicios Web, permite consultas y consumo de las ontologías desde otros sistemas, capa que fue creada con la finalidad de realizar recomendaciones dentro del Entorno Virtual de Aprendizaje EVA, y dar soporte a cualquier otro proyecto semántico futuro.

1.2.1. Creación del servicio Web

Dentro de Netbeans se realizó la construcción del Web service, usando SOAP como protocolo de comunicación entre el servidor y cliente(s). La UDDI (Universal Directory and Description) mediante la cual es accesible y posible el consumo del servicio Web es:

<http://localhost:8080/serService/serWSDL?wsdl>.

El servicio Web se encuentra compuesto de tres métodos que son:

1. **browserSer.-** Permite realizar búsquedas dentro de las ontologías de acuerdo a un criterio especificado por el cliente.
2. **getDatosMateria.-** Permite obtener recomendaciones específicas para una determinada asignatura dentro de la UTPL.
3. **getTitle_and_URI.-** Permite obtener el título y la URI de un recurso en específico dentro de las ontologías.

A continuación se presenta el código fuente del servicio web.

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package services;
import Queries.browser;
import java.io.Serializable;
import java.util.ArrayList;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import process.Catalogo;
import process.DataContainer;
import process.LOM.LOM;

/**
 *
 * @author manuel
 */
@WebService()
```



```
public class serWSDL implements Serializable
{
    /**
     * browserSer
     * Permite obtener recomendaciones de objetos de
     * aprendizaje cuando un usuario ingresa algun criterio
     * @param String query=>La consulta proveniente del cliente
     * @return ArrayList<LOM>=> Un arrayList con todas las
recomendaciones
     * que satisfacen el criterio del cliente
     */

    @WebMethod(operationName = "browserSer")
    public ArrayList<LOM> browserSer(@WebParam(name = "query")
String query)
    {
        //TODO write your implementation code here:
        ArrayList<LOM> lom =null;
        if(!query.isEmpty()){
            browser br = new browser(query);
            br.QueryInOntologies();
            lom = new ArrayList<LOM>();
            lom = br.getLom();
        }
        return lom;
    }

    /**
     * getDatosMateria
     * Permite obtener recomendaciones especificas
     * para la asignatura especificada como parametro
     * @param: String name=> el nombre de la materia
     * @return Datacontainer => Contenedor de datos q especifica todas
     * las recomendaciones ligadas a la materia
     */

    @WebMethod(operationName = "getDatosMateria")

    public DataContainer getDatosMateria(@WebParam(name = "materia")
String materia)
    {
        return new process.ConsumidorOntologias().getDatosMateria(materia);
    }

    /**
     * getTitle_and_URI
     * Permite obtener el titulo y la localizacion de un determinado
     * objeto de aprendizaje
     * @param: String Catalogo=> el nombre del catalogo
     * @return Catalogo => Clase Catalogo (titulo y URL)
     */

    @WebMethod(operationName = "getTitle_and_URI")

    public Catalogo getTitle_and_URI(@WebParam(name = "Catalogo")
String Catalogo)
    {
        return new
```

new



```
process.ConsumidorOntologias().getTitle_and_URI(Catalogo);  
    }  
}
```

1.3. Capa interfaz de presentación

La capa de presentación o interfaz de usuario es la encargada de brindar la información proveniente de las ontologías, esto es, cuando existe alguna interacción o evento por parte del usuario. El framework de trabajo para la creación de páginas Web dentro de Netbeans 6.1 es: JSP

1.3.1. Buscador

La página principal del buscador (denominada **SER**), posee la siguiente estructura:

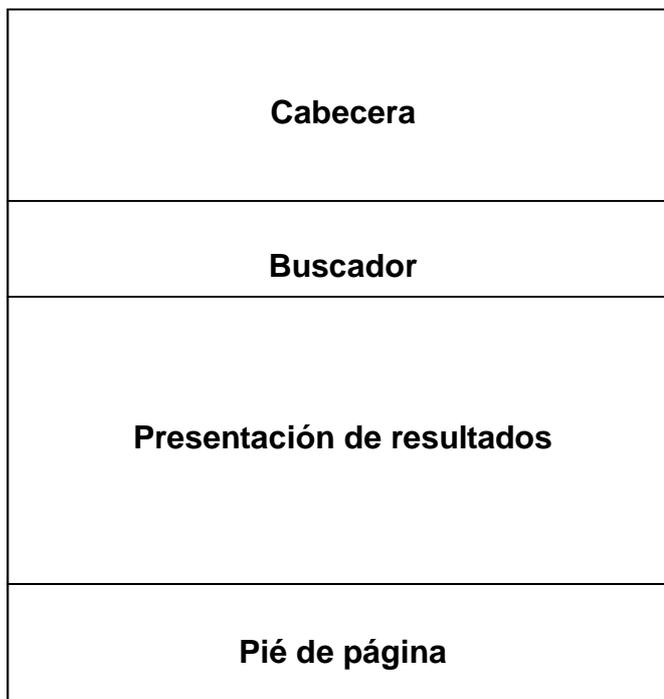


Figura 7: Composición de la página principal de la interfaz de usuario

El buscador presenta OA relacionados con los criterios de búsqueda del usuario; es decir, los enlaces tienen vinculación con las palabras ingresadas y que forman parte de la definición de conceptos de la ontología.



1.3.2. Cliente EVA

Dentro del entorno virtual de aprendizaje se desarrolló un módulo, que es un cliente del sistema “**SER**”. La comunicación entre ambos sistemas se realiza mediante el consumo del servicio Web, especificado en la sección 1.2 del presente Anexo. Para esto fue necesario adicionalmente descargar el paquete **nusoap** (paquete para aplicaciones en PHP que permite crear y consumir servicios Web), el mismo que se puede descargar desde: <http://sourceforge.net/projects/nusoap/>

A continuación se presenta el código fuente en PHP para el consumo del servicio Web.

```
<?PHP

//la clase original extiende de block_list, se ha modificado para que extienda de
block_base
/**
 * Bloque para recomendaciones de acuerdo a las asignaturas
 *
 */
class block_serEva extends block_base {

    public $content;
    public $config;
    public $record;

    function init() {
        $this->title = '<b>Recomendaciones del repositorio DSpace</b>';
        $this->version = 2008111100;
    }

    function has_config() {
        return false;
    }

    function get_content()
    {
        global $THEME, $CFG, $USER;

        if($this->content !== NULL) {
            return $this->content;
        }

        $this->content = new stdClass;
        $this->content->items = array();
        $this->content->icons = array();
        $this->content->footer = "";
        if ($this->instance->pageid != SITEID ||
            $CFG->showsiteparticipantslist > 1 ||
            ($CFG->showsiteparticipantslist == 1 && isteacherinanycourse()) ||
            isteacher(SITEID))
        {
```



```
//eva actual
// 1 = cursos introductorios CI
// 2 = Abierta y a distancia MA
// 3 = Clasica MC
// 4 = Semipresencial MS
// 5 = Virtual MV
//obtener el id del curso
$Id_curso=$this->instance->pageid;
//obtener la categoría del curso
$parents = get_record("course", "id", $Id_curso);
$materia_name = $parents->fullname;
$materia_name1 = str_replace(" ", "_", $materia_name);

$category = get_record("course_categories", "id",
    $parents->category);
//determinar la modalidad en la que se imparte la materia
if ($category->parent == 8)//modalidad abierta..
{
    $modalidad = "_MA_";
} else if($category->parent == 7){
    $modalidad = "_MP_";
}

//la materia
//completa es: AS_MODALIDAD_ID_NOMBRE_MATERIA
$Id_materia_codificada =
"AS".$modalidad.$Id_curso."_".$materia_name1;
echo $Id_materia_codificada;
$this->content->text = $this->getPestana
    ($Id_materia_codificada,$materia_name);
}
unset($filteropt); // memory footprint
} //fin get_content

function getPestana($Id_materia,$materia)
{

    $client = $this->Serclient();

    //Obtener el id de la materia
    $params =array('materia'=>$Id_materia);
    //realizar la llamada del metodo Java
    $result = $client->getDatosMateria($params);

    //en el caso de que exista al menos el nombre
    //del area al que pertenece la materia se
    //presenta el resultado

    $pestanda = "";
    if(!empty($result->return->area))
    {
        //Buscamos errores
        //Dar formato de presentacion
        //print_r($result);
```



```
echo '<style type="text/css">';
echo '.clear{clear:both;};
    echo '.descripcion{';
        echo 'font-family: Verdana,Arial,Helvetica,sans-serif;';
        echo 'font-size: 11px;';
    echo '};

    echo '.autor{';
        echo 'color: #a95c33;';
        echo 'font-size: 11px;';
        echo 'font-family: Verdana,Arial,Helvetica,sans-serif;';
        echo 'font-weight: bold;';
    echo '};
echo '</style>';
echo '<link rel="stylesheet"
    href="http://localhost/serClient/web/javascript/css/tab-view.css"
    type="text/css" media="screen">';
echo '<script type="text/javascript"
    src="http://localhost/serClient/web/javascript/js/ajax.js"></script
>';
echo '<script type="text/javascript"
    src="http://localhost/serClient/web/javascript/js/tab-
view.js"></script>';

$pestanda.= '<br /><div><b>Algunas recomendaciones para la
    materia: '.$materia.'<b></div><br />';
$pestanda.= '<div class="clear"></div>';
$pestanda.= '<div id="dhtmlgoodies_tabView2">';

foreach ($result as $r)
{

    //recomendaciones por materia
    $pestanda.= "<div class='dhtmlgoodies_aTab'>";
    foreach ($r->lom as $lom)
    {
        $pestanda.= $this->show_signature($lom);
        $pestanda.= '<hr>';
    }
    $pestanda.= "</div>";

    //recomendaciones por contenido
    $pestanda.= "<div class='dhtmlgoodies_aTab'>";
    foreach ($r->content as $content)
    {
        $pestanda.= $this->show_chapter($content);
    }
    $pestanda.= "</div>";
}
$pestanda.= '</div>';
$pestanda.= '<div><br /><center><b>Búsquedas</b></center>';
$pestanda.= '<hr>';
$pestanda.= '<form name="buscadorSer" method="post"
    action="http://localhost:8080/ser/index.jsp">';
$pestanda.= '<input type="text" name="query" value=""
    size="18" disabled/><input
    type="submit" value="Buscar"
    disabled/>';
```




```
    }
    $tmp.= "</ul>";

    return $tmp;
}

/*
 * Obtener la(s) recomendacion(es) de cada capitulo
 * del web service
 */
function getResult($Catalogo)
{
    //invocar el web service
    $client = $this->Serclient();
    //Obtener el id de la materia
    $params =array('Catalogo'=>$Catalogo);
    //realizar la llamada del metodo Java
    return $client->getTitle_and_URI($params);
}

/*
 * Formar las recomendaciones por asignatura
 * @param $lom: Array q contiene todas las recomendaciones
 * especificadas mediante el estandar lomv1.0
 */
function show_signature($lom)
{
    $tmp = "";
    //recorrer la categoría general
    $tmp.= "<b><a href='javascript:Abrir_ventana(\"".$lom->technical-
    >localizacion.\"')' style='color: #2c641a; font-weight: bold;font-
    family: Verdana,Arial,Helvetica,sans-serif; font-size:
    13px;'>".$lom->general->titulo."</a></b><br>";
    $tmp.= "<div style='font-family: Verdana,Arial,Helvetica,sans-serif;font-
size:
    12px;'>".$lom->general->descripcion."</div>";
    $tmp.= "<div style='color: #a95c33; font-size: 11px; font-family:
    Verdana,Arial,Helvetica,sans-serif;font-weight:bold;'>Autor:
    ".$lom->lifecycle->contribucion->entidad."</div>";
    return $tmp;
}

function show_lom($lom)
{
    $tmp = "";
    //recorrer la categoría general
    if( !empty($lom->general) )
    {
        //print_r($lom->general);
        $tmp.= "<b>Categoria General</b><br>";
        $tmp.= $lom->general->catalogo."<br>";
        $tmp.= $lom->general->descripcion."<br>";
        $tmp.= $lom->general->idioma."<br>";

        foreach($lom->general->palabrasClave as $keywords){
            $tmp.= $keywords."<br>";
        }
        $tmp.= $lom->general->titulo."<br>";
    }
}
```



```
//recorrer la categoria ciclo de vida
if( !empty($lom->lifecycle) )
{
    print_r($lom->lifecycle);
    $tmp.= "<b>Categoria Ciclo de vida</b><br>";
    $tmp.= $lom->lifecycle->estado."<br>";
    $tmp.= $lom->lifecycle->version."-<br>";
    $tmp.= $lom->lifecycle->contribucion->entidad."<br>";
    $tmp.= $lom->lifecycle->contribucion->fecha."<br>";
    $tmp.= $lom->lifecycle->contribucion->tipoRol."<br>";
}

//recorrer la categoria metadata
if( !empty($lom->metadata) )
{
    //print_r($lom->metadata);
    $tmp.= "<b>Categoria Metadata</b><br>";
    $tmp.= $lom->metadata->contribucionEntidad;
    $tmp.= $lom->metadata->contribucionFecha;
    $tmp.= $lom->metadata->contribucionTipo;
    $tmp.= $lom->metadata->esquema;
    $tmp.= $lom->metadata->idioma;
}

//recorrer la categoria technical
if( !empty($lom->technical) )
{
    //print_r($lom->technical);
    $tmp.= "<b>Categoria technical</b><br>";
    $tmp.= $lom->technical->localización;
    $tmp.= $lom->technical->formato;
    $tmp.= $lom->technical->duracion;
    $tmp.= $lom->technical->otrosRequisitos;
    $tmp.= $lom->technical->requisitosNombre;
    $tmp.= $lom->technical->requisitosTipo;
    $tmp.= $lom->technical->requisitosVersionminima;
    $tmp.= $lom->technical->requisitosVersionmaxima;
    $tmp.= $lom->technical->tamaño;
}

//recorrer la categoria educationaluse
if( !empty($lom->educationaluse) )
{
    $tmp.= "<b>Categoria educationaluse</b><br>";
    //print_r($lom->educationaluse);
    $tmp.= $lom->educationaluse->idioma;
    $tmp.= $lom->educationaluse->interactividad;
    $tmp.= $lom->educationaluse->procesoCognitivo;
    $tmp.= $lom->educationaluse->rangoTipicoedad;
    $tmp.= $lom->educationaluse->tiempoTipicoAprendizaje;
    $tmp.= $lom->educationaluse->context->asistencia;
    $tmp.= $lom->educationaluse->context->lugar;
    $tmp.= $lom->educationaluse->context->modalidad;
    $tmp.= $lom->educationaluse->descripcion->conocimientoPrevio;
    $tmp.= $lom->educationaluse->descripcion->objetivosDidacticos;
    $tmp.= $lom->educationaluse->descripcion->tiposdeConocimiento;
    $tmp.= $lom->educationaluse->descripcion->valoresHumanos;
```



```
$tmp.= $lom->educationaluse->evaluacion->descripcion;  
$tmp.= $lom->educationaluse->evaluacion->ovservacion;  
$tmp.= $lom->educationaluse->evaluacion->tipoRol;  
  
//Destinatario  
if( !empty($lom->educationaluse->destinatario) ){  
    foreach( $lom->educationaluse->destinatario as $destinatario )  
    {  
        $tmp.= $destinatario;  
    }  
}  
  
//Tipo de recurso educativo  
foreach( $lom->educationaluse->tipoRecursoEducativo as $tipo )  
{  
    $tmp.= $tipo;  
}  
}  
  
//recorrer la categoria anotacion  
if( !empty($lom->anotation) )  
{  
    //print_r($lom->anotation);  
    $tmp.= "<b>Categoria anotacion</b><br>";  
    $tmp.= $lom->anotation->descripcion;  
    $tmp.= $lom->anotation->entidad;  
    $tmp.= $lom->anotation->fecha;  
}  
//recorrer la categoria classification  
if( !empty($lom->classification) )  
{  
    //print_r($lom->classification);  
    $tmp.= "<b>Categoria classification</b><br>";  
    $tmp.= $lom->classification->descripcion;  
    $tmp.= $lom->classification->proposito;  
    $tmp.= $lom->classification->rtfuente;  
    $tmp.= $lom->classification->rtentrada;  
    $tmp.= $lom->classification->rtidentificador;  
}  
  
//recorrer la categoria relation  
if( !empty($lom->relation) )  
{  
    //print_r($lom->relation);  
    $tmp.= "<b>Categoria relation</b><br>";  
    //descripcion  
    if( !empty($lom->relation->recursoDescripcion) )  
    {  
        foreach($lom->relation->recursoDescripcion as $Rdescripcion)  
        {  
            $tmp.= $Rdescripcion;  
        }  
    }  
}  
  
//identificador Catalogo  
if( !empty($lom->relation->identificadorCatalogo) )  
{  
    foreach($lom->relation->identificadorCatalogo as $idCatalogo)  
    {
```



```
        $tmp.= $idCatalogo;
    }
}

//identificador Entrada
if( !empty($lom->relation->identificadorEntrada) )
{
    foreach($lom->relation->identificadorEntrada as $idEntrada)
    {
        $tmp.= $idEntrada;
    }
}

//Tipo relacion
if( !empty($lom->relation->tipoRelacion) )
{
    foreach($lom->relation->tipoRelacion as $Trelacion)
    {
        $tmp.= $Trelacion;
    }
}

}

//recorrer la categoria rights
if( !empty($lom->rights) )
{
    //print_r($lom->rights);
    $tmp.= "<b>Categoria rights</b><br>";
    $tmp.= $lom->rights->accesoDescripcion;
    $tmp.= $lom->rights->accesoTipo;
    $tmp.= $lom->rights->costo;
    $tmp.= $lom->rights->autoryOtrasrestricciones;
    $tmp.= $lom->rights->descripcion;
}
return $tmp;
}

/*
 * Consumo de servicio Web
 */

function Serclient()
{
    require_once("nusoap/lib/nusoap.php");
    $client = new
        soapclient('http://localhost:8080/serService/serWSDL?wsdl');
    return $client;
}

}
?>
<script>
    function Abrir_ventana (pagina) {
```



```
var opciones="toolbar=no, location=no, directories=no, status=no,  
menubar=no, scrollbars=yes, resizable=yes, width=720,  
height=580, top=125, left=140";  
window.open(pagina,"",opciones);  
}  
</script>
```

La siguiente figura muestra el cliente EVA para los servicios provenientes del sistema SER.



Figura 7: Módulo de recomendaciones de objetos de aprendizaje en el EVA



ANEXO III FORMATOS USADOS



1. Formato usado para las pruebas de aceptación de usuario

Nivel de aceptación de usuario				
	Porcentajes			
	1 – 25%	26% - 50%	51 -75%	76-100%
Los resultados obtenidos satisfacen los criterios de búsquedas.				
El programa es fácil de usar y de entender.				
Nivel de calidad de las recomendaciones presentadas con respecto al contenido de la asignatura.				
Evaluación de la aplicación a nivel general en cuanto a confiabilidad de resultados.				

2. Formato de anotaciones semánticas

Anotaciones semánticas		
Eficiencia	Organización	
	Costo	
	Software usado	
	Autor	
	Metodología	
Reconocimiento	Versionamiento	
	Ciclo de Vida	
	Definición de relaciones	



	Definición de atributos	
	Anotaciones de clases	
	Anotaciones de instancias	
	Anotaciones de relaciones	