

TRATAMIENTO SEMÁNTICO DE LOS LOGS DE UN SERVIDOR USANDO
RDF

CARLOS ARTURO PARRA LLANOS

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR EN CONVENIO CON EL ITESM
MAESTRÍA EN CIENCIAS COMPUTACIONALES
CARTAGENA DE INDIAS D. T. Y C.
JUNIO DE 2007

TRATAMIENTO SEMÁNTICO DE LOS LOGS DE UN SERVIDOR USANDO

RDF



CARLOS ARTURO PARRA LLANOS

TESIS

REQUISITO PARA OBTENCIÓN DEL TÍTULO DE MAGISTER EN CIENCIAS
COMPUTACIONALES CON ÉNFASIS EN REDES DE COMPUTADORES

DIRECTOR

EDUARDO CARRILLO ZAMBRANO

Doctor en Tecnología de la Información, Computación y Comunicaciones

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR EN CONVENIO CON EL ITESM

MAESTRÍA EN CIENCIAS COMPUTACIONALES

CARTAGENA DE INDIAS D. T. Y C.

JUNIO DE 2007

A mi esposa e hijos.

Carlos Arturo Parra Llanos

AGRADECIMIENTOS

El autor expresa sus agradecimientos:

Al Doctor Eduardo Carrillo Zambrano, Director del Laboratorio de Cómputo Especializado de la Universidad Autónoma de Bucaramanga, por la asesoría y apoyo irrestrictos prestados como Director de tesis durante la realización del presente trabajo.

A la Doctora Elsa Ruiz, porque más que la Coordinadora del Programa de Maestrías en el Campus Universidad Tecnológica de Bolívar fue la amiga que apoyó y facilitó la labor.

A los Directivos y al personal del Departamento Informático de la Universidad Autónoma de Bucaramanga, por haber facilitado el empleo de la información de su servidor Web para el desarrollo del presente trabajo, especialmente a la Ing. Karol Dalila Reyes.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	1
1.1. Antecedentes	1
1.2. Motivación del trabajo	3
1.3. Planteamiento del problema y justificación	5
1.4. Objetivo general	7
1.4.1. Objetivos específicos	7
1.5. Descripción de la estructura del documento de tesis	8
Referencias Bibliográficas	9
2. MARCO TEÓRICO (ESTADO DEL ARTE)	10
2.1. El RDF	10
2.1.1. Relación del RDF con otros estándares utilizados en Internet	26
2.1.2. Aplicaciones del RDF	33
2.2. Ontología	39
2.2.1. Representación del conocimiento	41
2.2.2. Lenguajes de especificación ontológica	42
2.3. La Web Semántica	44
2.3.1. La Ciencia de la Web	46
2.4. Logs	48
2.4.1. Logs por ICMP	50
2.4.2. Web logs	51
2.5. Herramientas	57
2.5.1. Protégé	58
2.5.2. Sesame	59
2.5.3. OntoStudio	64
2.5.4. R ₂ O + ODEMapster	65
2.5.5. RDF en el Contexto de los Negocios	65
2.5.6. Jena	66
2.6. Aplicaciones	66
2.6.1. UNSPSC	66
2.6.2. Open Biomedical Ontologies - OBO	67
2.6.3. Ontología de los logs con RDF	68
2.6.4. Swoogle	68
Referencias Bibliográficas	70
3. METODO DE INVESTIGACIÓN	78
Referencias bibliográficas	81
4. RESULTADOS DE LA INVESTIGACIÓN	82
4.1. Propuesta de vocabulario y ontología Logging	82
4.1.1. Definición de vocabulario y ontología	86
4.1.2. Diferencias de la ontología propuesta con la antecesora	92
4.1.3. Código RDF/XML	94
4.1.4. Almacenamiento de la ontología Log en Sesame	96

4.1.5. Procedimiento de inclusión de los logs en el repositorio	100
4.1.6. Evaluación de la ontología	101
4.2. Análisis de los logs	116
4.2.1. Solicitud del cliente.....	116
4.2.2. Códigos de estado del servidor.....	117
4.2.3. Los Logs en el servidor Web Apache.....	117
4.2.4. Logs de los UNAB	119
4.3. Aplicación de análisis de Logs.....	121
Referencias bibliográficas.....	124
5. CONCLUSIONES Y TRABAJO FUTURO	126
BIBLIOGRAFÍA	128
ANEXOS	130
Anexo I - Código RDF/XML generado por Protégé.....	131

LISTA DE TABLAS Y FIGURAS

TABLAS

TABLA 2.1. COMPARACIÓN DE LAS CARACTERÍSTICAS DE LOS LEGUAJES DE ESPECIFICACIÓN ONTOLÓGICA.....	43
TABLA 4.1. EJEMPLO DE UNA INSTANCIA LOG EN FORMATO NETLOGGER.	94
TABLA 4.2. EJEMPLO DE INSTANCIAS LOGS EN FORMATO DAMED-WG.....	95
TABLA 4.3. EJEMPLO DE INSTANCIAS LOGS EN FORMATO LOG4J.	96
TABLA 4.4. RESPUESTA A CONSULTA 1 EN SERQL.	103
TABLA 4.5. RESPUESTA A CONSULTA 2 EN SERQL.	104
TABLA 4.6. RESPUESTA A CONSULTA 3 EN SERQL.	104
TABLA 4.7. RESPUESTA A CONSULTA 4 EN SERQL.	105
TABLA 4.8. RESPUESTA A CONSULTA 5 EN SERQL.	109
TABLA 4.9. RESPUESTA A CONSULTA 6 EN SERQL.	109
TABLA 4.10. RESPUESTA A CONSULTA 7 EN SERQL.	111
TABLA 4.11. RESPUESTA A CONSULTA 8 EN SERQL.	112
TABLA 4.12. RESPUESTA A CONSULTA 9 EN SERQL.	113
TABLA 4.13. RESPUESTA A CONSULTA 10 EN SERQL.	114

FIGURAS

FIGURA 2.1. ARQUITECTURA DEL ENTORNO DEL RDF.	11
FIGURA 2.2. DIAGRAMA SENCILLO DE NODO Y ARCO. TOMADO DE REC. RDF-SYNTAX-19990222.	18
FIGURA 2.3. PROPIEDADES CON VALOR ESTRUCTURADO. TOMADO DE REC. RDF-SYNTAX-19990222.	19
FIGURA 2.4. VALOR ESTRUCTURADO CON IDENTIFICADOR. TOMADO DE REC. RDF-SYNTAX-19990222.	21
FIGURA 2.5. FORMATOS DE DATOS DEL W3C Y SUS RELACIONES. TOMADO DE W3C NOTE-RDF-19971029.	27
FIGURA 2.6. REPRESENTACIÓN GRÁFICA DE LA CODIFICACIÓN UML/RDF. TOMADO DE REPRESENTING UML IN RDF, HTTP://INFOLAB.STANFORD.EDU/~MELNIK/RDF/UML/.39	

FIGURA 2.7. DISTRIBUCIÓN DEL MERCADO DE SERVIDORES WEB, ENTRE AGOSTO DE 1995 Y ABRIL DE 2007. TOMADO DE HTTP://NEWS.NETCRAFT.COM/ARCHIVES/WEB_SERVER_SURVEY.HTML.....	50
FIGURA 2.8. DIAGRAMA DE FLUJO DEL LOG. TOMADA DE JAVA LOGGING OVERVIEW, HTTP HTTP://JAVA.SUN.COM/J2SE/1.4.2/DOCS/API/JAVA/UTIL/LOGGING/LOGRECORD.HTML.	53
FIGURA 2.9. DIAGRAMA DE FLUJO CON MANEJADOR DE MEMORIA. TOMADA DE JAVA LOGGING OVERVIEW, HTTP://JAVA.SUN.COM/J2SE/1.4.2/DOCS/API/JAVA/UTIL/LOGGING/LOGRECORD.HTML.	54
FIGURA 2.10. EL SERVIDOR SESAME, TOMADO DEL MANUAL DE SESAME.	60
FIGURA 2.11. LA ARQUITECTURA DE SESAME. TOMADO DEL MANUAL DE SESAME.	62
FIGURA 2.12. MUESTRA DE LA LISTA DEL DESARROLLO DE LA ONTOLOGÍA OBO.....	67
FIGURA 2.13. VISUALIZACIÓN DE LA METADATA DE UNA ONTOLOGÍA CONSULTADA EN SWOOGLE.	69
FIGURA 4.1. JERARQUÍA DE LA CLASE GULF2 EN RDF SCHEMA.....	89
FIGURA 4.2. MUESTRA DE SOLICITUDES PROMEDIOS POR HORA AL SERVIDOR WEB UNAB.	93
FIGURA 4.3. ADICIÓN DEL VOCABULARIO Y LA ONTOLOGÍA LOGS A SESAME.....	97
FIGURA 4.4. EXPLORACIÓN DE LA ONTOLOGÍA LOGS EN SESAME.	98
FIGURA 4.5. ESTRUCTURA DE LA ONTOLOGÍA EN EL REPOSITORIO EN MYSQL MANEJADO POR SESAME.....	98
FIGURA 4.6. CONTENIDO DE LA TABLA LITERALS DEL REPOSITORIO.	99
FIGURA 4.7. FACILIDADES PARA CONFIGURACIÓN DE PRUEBA A LA ONTOLOGÍA EN PROTÉGÉ.....	102
FIGURA 4.7. CONJUNTO DE NOMBRES DE ARCHIVOS LOGS DEL SERVIDOR UNAB.	120

RESUMEN

La investigación plantea el manejo semántico de los logs de un servidor Web usando RDF. Para lo cual se establece conocer RDF como formato de metadato, elaborar una propuesta de ontología de logs y cómo emplearla en el diseño de un sistema de información para la Web semántica, disponiendo de un servidor de la Universidad Autónoma de Bucaramanga.

El trabajo inicia con el estudio del RDF, Web Semántica, ontología, repositorios y de algunas de sus aplicaciones, al mismo tiempo que se seleccionan logs del servidor unab.edu.co y, partiendo de una propuesta anterior, se emplea un método conocido para establecer el vocabulario y la ontología de los logs, que para la implementación hace con las herramientas de uso libre Protégé y Sesame y como repositorio al manejador de base de datos MySQL.

Se llega al establecimiento del vocabulario y la ontología de los logs; planteamiento que presenta ventajas en su manejo sobre su antecesor y define un procedimiento para manejo de los logs en repositorios que son implementados con herramientas comunes y de uso libre.

1. INTRODUCCIÓN

1.1. ANTECEDENTES

El hipertexto, nacido hace aproximadamente 15 años, es la posibilidad técnica de enlazar textos y gráficos mediante "links". En términos muy simples, el hipertexto le permite abandonar el desarrollo secuencial de la información que implicaba una estructura definida de inicio, desarrollo y fin; el hipertexto refleja la forma asociativa en que funciona nuestro cerebro, donde los "saltos" creativos son los links hacia otras páginas. El World Wide Web Consortium (W3C) desarrolla desde 1994 los protocolos comunes para la evolución de la WWW, desarrollando otras tecnologías que permitan la automatización de la Web en cuanto a intercambio de información entre plataformas y aplicaciones muy variadas. El estándar que ha dominado en los documentos WWW ha sido el HTML, el que resulta adecuado para mostrar datos pero no nos permite describirlos, y esto limita enormemente las posibilidades de manipulación de los mismos dentro y fuera del navegador [1], sin embargo, éste está evolucionando en múltiples direcciones, XML (Extensible Markup Language) es una de ellas. La aparición del XML es una respuesta a la necesidad que existe dentro de Internet de encontrar formas adecuadas de manejar los datos. XML permite la automatización de la Web

en cuanto a intercambio de información entre plataformas y aplicaciones muy variadas, provee una sintaxis que puede ser utilizada para crear Metadatos, esto es, información acerca de la información. De esta forma los autores pueden describir por sí mismos los contenidos que albergan en la red. Dentro de XML se han creado diferentes formatos de Metadatos cuyo principal objetivo es permitir la localización de información en la Web de manera eficiente y eficaz. A continuación se enuncian los formatos de metadatos en XML:

RDF (Resource Description Framework): Es una manera automática de describir qué información está disponible. El vocabulario utilizado depende de la aplicación. RDF deja a los programadores la oportunidad de elegir el vocabulario esencial de cada género en particular. Existe una dirección de Web para cada vocabulario a la cual se puede recurrir. Está basado en nodos (direcciones Web) y atributos.

“El RDF soporta el uso de convenciones para facilitar la interoperabilidad modular entre grupos independientes del elemento metadatos. En el modelo de datos, estas convenciones incluyen mecanismos estándares para la representación semántica agrupada en forma simple, pero poderosa. El RDF adicionalmente provee una manera para publicar vocabularios entendibles por el hombre y procesables por la máquina. Los vocabularios son el grupo de propiedades o elementos de los metadatos, definidos por las

comunidades para la descripción de recurso. La capacidad para estandarizar la declaración de los vocabularios es anticipada para incentivar la reutilización y extensión de semánticas entre comunidades de información diferentes. Por ejemplo, la Iniciativa Dublin Core (The Dublin Core Initiative), una comunidad internacional de descripción de recurso dedicada a la descripción simple de recurso para el descubrimiento, adoptado por RDF. Educom's IMS Instructional Metadata System, diseñada para proporcionar acceso a material educativo, adoptó el Dublin Core y la arquitectura correspondiente y fue extendida con la semántica de dominio específico. El RDF está diseñado para soportar este tipo de modularidad semántica para la creación de una infraestructura que soporte la combinación de registros de atributos distribuidos. De esa manera, no se requiere de un registro central. Esto permite a las comunidades declarar vocabularios que puedan ser reutilizados, ampliados y/o refinados en aplicaciones o requerimientos descriptivos para especificación de dominio" [2].

1.2. MOTIVACIÓN DEL TRABAJO

El proyecto tiene como fin establecer qué es el RDF y cuáles son sus beneficios en el manejo de ontologías, de tal forma que sea una fuente teórico-práctica para futuros desarrollos en el área.

Para el planteamiento de la ontología se dispondrá de los logs de un servidor de la Universidad Autónoma de Bucaramanga, tendiente a facilitar la

accesibilidad a los datos e información generado por un servidor Web, a través de los conceptos de RDF aplicados al diseño ontológico, que permita realizar un análisis al comportamiento de los usuarios de un servidor Web.

Se ha seleccionado el análisis de los logs del servidor de la Universidad Autónoma de Bucaramanga, dado que este proyecto va a emplear facilidades del RDF, debido a las siguientes características:

- Interoperabilidad entre aplicaciones que intercambian metadatos.
- Descripción de recursos. En el manejo y disposición de este tipo de datos es fundamental su descripción, conocer la información sobre los datos dispuestos.
- Servicios colaborativos. Al ser un servicio prestado por un ente académico de formación superior.
- Comercio electrónico. En el futuro cercano el sistema podría comercializar algún producto.

La importancia de este trabajo de investigación radica en su intención de contribuir en la mejora del uso de los recursos que nos ofrece la red de redes Internet y en su proyección hacia la comunidad.

1.3. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

En la actualidad el resultado del empleo de motores de búsqueda en la Web arroja gran cantidad de información que en su mayoría no está relacionada con el interés del usuario debido a que las estructuras actuales de disposición y manejo de la información no permiten un filtro adecuado.

¿Qué beneficios tiene el uso de RDF en el diseño de sistemas de información en la Web? ¿Permitirá su utilización una mejora en el diseño, estableciéndose como una alternativa al momento de desarrollar sistemas de información en Internet? ¿Los resultados de las búsquedas serán inteligentes, es decir, se le presentará al usuario lo que realmente está buscando? ¿Cuál es la relación de RDF, XML, XLS, DOM, etc.? ¿El manejo semántico de los logs con RDF permitirá realizar un análisis al comportamiento de los usuarios de un servidor Web?

El proyecto tiene como fin establecer cuáles son los beneficios de utilizar RDF en el diseño de sistemas de información en la WEB y a la vez aplicarlo desarrollando un vocabulario y ontología para el manejo de los logs de un servidor Web, de tal forma que sea una fuente teórico-práctica para futuros desarrollos en el área.

La selección del RDF entre los estándares para metadatos obedece a que:

- El RDF soporta metadatos en la Web.

- El RDF permite la codificación, intercambio y reutilización de metadatos estructurados bajo cualquiera de los formatos de descripción de recursos disponibles, como el DC, PICS, IAFA, etc., permitiendo la elección del formato de descripción que se adecúe a las necesidades.
- El RDF utiliza XML como sintaxis común de intercambio y procesamiento de metadatos y éste se perfila como el estándar de la Web.
- Los metadatos acogidos en el marco RDF pueden ser utilizados en una gran variedad de áreas de aplicación como búsqueda y recuperación de recursos, catalogación, descripción de contenido y temática de un sitio Web, buscadores y rastreadores, clasificación de contenidos, etc.
- El RDF como firma digital contribuiría con la seguridad en la Web requerida en el comercio electrónico [3].

Uno de los mayores obstáculos encontrados por la comunidad en la descripción de recursos es la multiplicidad de estándares incompatibles para la sintaxis de metadatos y los esquemas de los lenguajes de definición. Esto nos lleva al escaso desarrollo de aplicaciones interdisciplinarias y servicios para la comunidad descriptora de recursos. RDF proporciona la solución a este problema por medio de una sintaxis de especificación [4] y un esquema de especificación [5].

Se ha seleccionado el servidor Web de la Universidad Autónoma de Bucaramanga, dado que este proyecto va a emplear facilidades del RDF.

1.4. OBJETIVO GENERAL

Realizar un estudio del RDF (Resource Description Framework), de su importancia en el diseño de sistemas de información en la Web y su aplicación en una propuesta de ontología para contribuir con el mejoramiento de los procesos de localización de información en la Web de manera inteligente.

1.4.1. Objetivos específicos

Determinar qué es el RDF mediante la recopilación, clasificación y análisis de la información referente a este formato de metadatos, para contribuir con la aprehensión de este conocimiento y utilizarlo en el mejoramiento del diseño de los sistemas de información para la Web.

Establecer cómo se realiza el diseño de los sistemas de información en la Web con el uso de metadatos, como herramienta de soporte investigativo para nuevos desarrollos.

Analizar la relación de RDF con otros nuevos estándares utilizados en Internet como XML, XSL, DOM, etc., con el fin de definir las diferencias y similitudes en la concepción e implementación entre estos estándares.

Elaborar una propuesta de ontología de logs a un servidor de la Universidad Autónoma de Bucaramanga, en RDF, para destacar las bondades de su empleo.

1.5. DESCRIPCIÓN DE LA ESTRUCTURA DEL DOCUMENTO DE TESIS

El presente documento de tesis está conformado por cuatro capítulos que van conduciendo al lector paso a paso desde los aspectos históricos, hasta la propuesta y conclusiones obtenidas.

En el presente capítulo se presentan los antecedentes, la motivación del trabajo, el planteamiento del problema y justificación, objetivos y la descripción de la estructura del documento. El capítulo dos contiene el marco teórico (estado del arte), descripción del RDF, estado de la especificación del RDF en el W3C, relación del RDF con otros estándares utilizados en Internet y aplicaciones del RDF, ontología, la Web Semántica, herramientas y aplicaciones. El capítulo tres describe el método de investigación. El capítulo cuatro presenta los resultados de la investigación. El capítulo quinto contiene las conclusiones y trabajo futuro. Y finalmente, se presenta la bibliografía empleada y anexos.

Se ha aplicado Vancouver como norma de estilo a la bibliografía y las referencias bibliográficas.

REFERENCIAS BIBLIOGRÁFICAS

[1] W3C World Wide Web Consortium. LASSILA O. Introduction to RDF Metadata [online]. [actualizado 1997 nov 13; citado 2006 dic 17]. [aprox. 6 pantallas]. Disponible en: <http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html>.

[2] Miller E. An Introduction to the Resource Description Framework. D-Lib Magazine May 1998 [serial online]. 1998 may [citado 2006 dic 16]; 4(1) [aprox. 25 pantallas]. Disponible en: <http://www.dlib.org/dlib/may98/miller/05miller.html>.

[3] W3C World Wide Web Consortium. Resource Description Framework (RDF) Schema Specification [online]. [actualizado 1999 mar 3; citado 2006 dic 19]. [aprox. 4 pantallas]. Disponible en: <http://www.w3.org/TR/1999/PR-rdf-schema-19990303>.

[4] W3C World Wide Web Consortium. Resource Description Framework (RDF) Model and Syntax. [online]. [actualizado 1999 ene 5; citado 2007 ene 29]. [aprox. 120 pantallas]. Disponible en: <http://www.w3.org/TR/PR-rdf-syntax/>.

[5] W3C World Wide Web Consortium. RDF Vocabulary Description Language 1.0: RDF Schema [online]. [actualizado 2004 feb 10; citado 2006 dic 8]. [aprox. 62 pantallas]. Disponible en: <http://www.w3.org/TR/rdf-schema/>.

2. MARCO TEÓRICO (ESTADO DEL ARTE)

Iniciemos recordando la definición de metadatos o metacontenido, como la información sobre la información (data about data), “Un metacontenido no es más que la información relativa al contenido de un documento, como su título, autor, tamaño del archivo, fecha de creación, historial de cambios, palabras claves y demás información asociada.” [1].

2.1. EL RDF

El Resource Description Framework o Marco de Trabajo de Descripción de Recursos (RDF) [2] es el fruto de un esfuerzo conjunto para crear un modelo capaz de dar soporte a los metadatos en la Web, inspirado tanto en el Warwick Framework, como en el Dublin Core o PICS, así como en el nuevo lenguaje de edición XML (eXtensible Markup Language, diseñado con la inclusión de propuestas tanto de Microsoft como de NetScape). El RDF es un estándar que permite la codificación, intercambio y reutilización de metadatos estructurados bajo cualquiera de los formatos de descripción de recursos analizados anteriormente (DC, PICS, IAFA, etc.). El RDF no estipula la semántica de descripción, sino que proporciona a estos colectivos la

posibilidad de elegir el formato de descripción que se adecúe mejor a sus necesidades.

El RDF utiliza XML como sintaxis común de intercambio y procesamiento de metadatos. La sintaxis XML es un subgrupo del estándar internacional de procesadores de texto SGML (Standar Generalized Markup Language) específicamente pensado para utilizar en la Web. La Figura 2.1 presenta la arquitectura del entorno RDF.

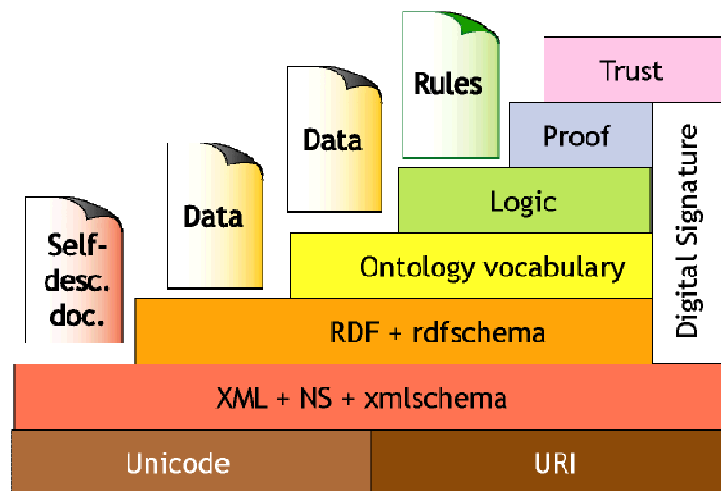


FIGURA 2.1. ARQUITECTURA DEL ENTORNO DEL RDF.

Los objetivos del RDF son amplios y las oportunidades potenciales son enormes. Los metadatos acogidos en el marco RDF pueden ser utilizados en una gran variedad de áreas de aplicación, por ejemplo: en la búsqueda y recuperación de recursos; en catalogación; para describir el contenido y temáticas relacionadas disponibles en un sitio Web particular; en una página Web o en una biblioteca digital; en buscadores y rastreadores, al lado de

agentes inteligentes de software para facilitar la distribución e intercambio de conocimiento; en clasificación de contenidos, para describir colecciones de páginas que representan un sólo documento lógico; para describir los derechos de propiedad intelectual de las páginas Web, etc. El RDF con "firma digital" será la clave para construir la "Web segura", indispensable para el comercio electrónico, la colaboración segura y otras aplicaciones. El RDF es una fundamentación para el procesamiento de metadatos; provee interoperabilidad entre aplicaciones que intercambian información entendible por la máquina en la Web. El RDF enfatiza las facilidades para permitir el procesamiento automatizado de recursos de la Web.

La falta de la semántica de los lenguajes de marcado (markup) fue una de las mayores barreras al desarrollo de un procesamiento de documentos más inteligente en la Web. Hasta hace muy poco el marcado del HTML fue usado solamente para indicar la estructura y diseño de los documentos, pero no su semántica [3].

Modelo y sintaxis del RDF

El modelo y la sintaxis del RDF están formalizados en la Recomendación del W3C, Especificación del Modelo y la Sintaxis del RDF [4], en la que se plantea un modelo para representar metadatos RDF así como una sintaxis para su codificación y transporte de manera en que se maximice la

interoperabilidad, independientemente de lo desarrollado de los servidores Web y los clientes. La sintaxis presentada emplea el Lenguaje de Mercado eXtensible - XML [5]: una de las metas de RDF es hacer posible la especificación semántica para los datos basándose en la manera estandarizada e interoperable del XML. El RDF y XML son complementarios: RDF es un modelo de metadatos, direccionable solamente por las referencias del tipo de codificación que requieren transporte y almacenamiento de archivos (tales como internacionalización, grupo de caracteres, etc.). Por esta razón, RDF depende del soporte de XML. Es también importante entender que esta sintaxis XML es solamente una sintaxis posible para RDF y existen formas alternativas para representar el mismo modelo de datos RDF (se presentan más adelante).

El objetivo principal del RDF es definir un mecanismo para describir recursos que no hagan supuestos sobre un dominio de aplicación particular, ni defina (a priori) la semántica de cualquier dominio de aplicación. La definición del mecanismo podría ser de neutral dominio, no obstante el mecanismo podría ser adecuado para describir información sobre cualquier dominio.

Esta especificación estará seguida por otros documentos que completan el marco de trabajo. Lo más importante es facilitar la definición de metadatos. El RDF tendría un sistema de clases semejante a la programación orientada a objetos y a los sistemas de modelado. Una colección de clases (creadas

típicamente para un propósito o dominio específico) es llamada un *esquema*. Las clases son organizadas jerárquicamente y brindan extensibilidad a través del perfeccionamiento de subclases. En esta forma, con el propósito de crear un esquema ligeramente diferente de uno existente, esto no es necesariamente “reinventar la rueda” pero se puede solamente proporcionar modificaciones incrementales al esquema de base. A través del uso compartido de esquemas RDF se soportará la reusabilidad de definiciones de metadatos. Debido a la extensibilidad incremental del RDF, estarán disponibles agentes de procesamiento de metadatos para rastrear los orígenes de esquemas desconocidos con respaldo de esquemas conocidos y ejecutar acciones significativas en metadatos, que no fueron originalmente diseñados para procesos. El uso compartido y extensibilidad del RDF, adicionalmente permite a los autores de metadatos usar herencia múltiple para “mezclar” definiciones, proporcionar vistas múltiples a sus datos, liberando del trabajo hecho por otros. Adicionalmente, es posible crear instancias de datos RDF basado en múltiples esquema para múltiples orígenes, ejemplo intercalar diferentes tipos de metadatos. Los esquemas pueden ellos mismos ser escritos en RDF; un documento que acompaña a esta especificación, esquema RDF, describe un conjunto de propiedades y clases para describir esquemas RDF.

Como un resultado de muchas comunidades reunidas y de acuerdo en los principios básicos de la representación y transporte de metadatos, RDF tiene

marcada influencia de diferentes orígenes. La influencia principal llega de la misma comunidad de estandarización de la Web en forma de metadatos HTML y PICS, la comunidad de bibliotecas, la comunidad de documentos estructurados en forma de SGML y más importante del XML, y también de la comunidad de Representación del Conocimiento (RC). Existen también otras áreas de tecnología que contribuyen al diseño del RDF; éstas incluyen la programación orientada a objetos y los lenguajes de modelado, así como las bases de datos. Mientras que la comunidad RC bosqueja el RDF, conocidos lectores del tema están cuestionando que el RDF no especifica un mecanismo para razonamiento. El RDF puede ser caracterizado como un sistema marco sencillo. Un mecanismo de razonamiento podría ser construido por encima del sistema marco.

Modelo básico del RDF

La fundamentación del RDF es un modelo para representar las denominaciones y valores de propiedades. El modelo del RDF se obtiene de los principios adecuadamente establecidos de varias comunidades de representación de datos. Las propiedades del RDF pueden ser pensadas como atributos de recursos y en este sentido corresponden a los pares de atributo-valor tradicionales. Las propiedades del RDF además representan relaciones entre recursos y un modelo RDF que puede por lo tanto reensamblarse en un diagrama entidad-relación. (Más precisamente, los esquemas RDF – siendo ellos mismos instancias de modelos de datos RDF -

son diagramas ER). En la terminología del diseño orientado a objetos, los recursos corresponden a objetos y las propiedades corresponden a instancias u ocurrencias de variables.

El modelo de datos RDF es una forma de sintaxis neutral de representación de expresiones RDF. La representación del modelo de datos es usado para evaluar la equivalencia entre contenidos. Dos expresiones RDF son equivalentes sí y solamente sí las representaciones del modelo de datos son las mismas. Esta definición de equivalencia permite algunas variaciones sintácticas en la expresión sin alterar el significado.

El modelo de datos básico consiste de tres tipos de objetos:

Recursos (Resource): Todas las cosas descritas por expresiones RDF son llamadas *recursos*. Un recurso puede ser una página Web completa; tales como por ejemplo el documento HTML <http://www.w3.org/Overview.html>. Un recurso puede ser una parte de una página Web, por ejemplo un elemento específico HTML o XML dentro de un documento origen. Un recurso puede también ser una colección completa de páginas, por ejemplo un sitio Web completo. Un recurso puede también ser un objeto que no es accesible directamente vía la Web, por ejemplo un libro impreso. Los recursos son siempre mencionados por URIs más identificadores opcionales. Cualquier

cosa puede tener un URI; la amplitud de los URI permite la introducción de identificadores para cualquier entidad imaginable.

Propiedades (Properties): Una propiedad es un aspecto específico, característica, atributo o relación usada para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos y sus relaciones con otras propiedades pueden ser descritos. En este aparte no se describe cómo se expresan las características de las propiedades, ya que de esto se encarga el Esquema del RDF en su especificación.

Declaración (Statements): Un recurso específico conjuntamente con una denominada propiedad, más el valor de esa propiedad para ese recurso es una declaración RDF. Estas tres partes individuales de una declaración son llamadas, respectivamente, el *sujeto* (the subject), el *predicado* (the predicate) y el *objeto* (the object). El objeto de una declaración (ejemplo: el valor de la propiedad) puede ser otro recurso o puede ser un literal; por ejemplo un recurso (especificado por un URI) o una simple cadena de caracteres u otro tipo de dato primitivo definido por el XML Schema. En términos del RDF, un *literal* puede contener marcas de XML pero no es adicionalmente evaluado por el procesador RDF. Existen algunas restricciones sintácticas que pueden ser expresadas en lo referente a como marcar en literales.

Ejemplos

Los recursos son identificados por un identificador de recurso. Un identificador de recurso es un URI más un identificador opcional.

Considere como un simple ejemplo la instrucción:

Ora Lassila es creador del recurso <http://www.w3.org/Home/Lassila>.

Esta declaración tiene las siguientes partes:

Sujeto (Recurso)	http://www.w3.org/Home/Lassila
Predicado (Propiedad)	Creador
Objeto (Literal)	"Ora Lassila"

Una declaración RDF se podría representar gráficamente usando un grafo dirigido etiquetado (también llamado "diagramas de nodos y arcos"). En estos diagramas, los nodos (dibujados como óvalos) representan recursos y los arcos representan propiedades. Los nodos que representan literales de cadenas de caracteres se dibujarán como rectángulos. La declaración anterior podría ser diagramada como lo muestra la Figura 2.2.

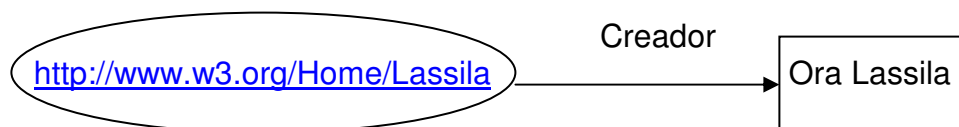


FIGURA 2.2. DIAGRAMA SENCILLO DE NODO Y ARCO. TOMADO DE REC. RDF-SYNTAX-19990222.

Nota: La dirección de la flecha es importante. El arco siempre inicia en el sujeto y apunta al objeto de la declaración. El sencillo diagrama anterior

puede ser leído como” <http://www.w3.org/Home/Lassila> tiene de creador a Ora Lassila”, o en general “<sujeeto> TIENE <predicado><objeto>”.

Ahora, considere el caso en que deseamos decir más sobre las características del creador de este recurso. En prosa, tal instrucción podría ser:

El individuo cuyo nombre es Ora Lassila, email <lassila@w3.org>, es el creador de <http://www.w3.org/Home/Lassila>.

La intención de esta declaración es hacer el valor de la propiedad Creador una entidad estructurada. En RDF tal entidad es representada por otro recurso. La instrucción anterior no da un nombre a este recurso; este es anónimo, así en la Figura 2.3 se representa el empleo del un óvalo vacío:

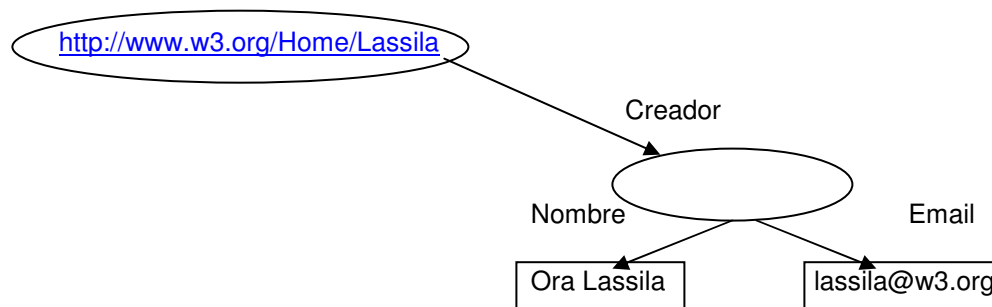


FIGURA 2.3. PROPIEDADES CON VALOR ESTRUCTURADO. TOMADO DE REC. RDF-SYNTAX-19990222.

Nota: correspondiendo a lo leído en la nota anterior, este diagrama podría leerse “<http://www.w3.org/Home/Lassila> tiene un creador y el creador tiene el nombre de Ora Lassila e email lassila@w3.org”.

La entidad estructurada de los ejemplos anteriores puede también ser asignada a un identificador único. La elección del identificador es hecha por el diseñador de la aplicación de base de datos. Para continuar el ejemplo, imagine que un ID de un empleado es usado como el único identificador para un recurso “persona”. Los URIs que sirven como llaves únicas para cada empleado (como lo define la organización) puede entonces ser algo parecido a <http://www.w3.org/staffId/85740>. Ahora podemos escribir dos declaraciones:

El individuo a quien se refiere el empleado con ID 85740 es el de nombre Ora Lassila y tiene dirección e-mail lassila@w3.org. El recurso <http://www.w3.org/Home/Lassila> fue creado por este individuo.

El modelo RDF para esta declaración se muestra en la Figura 2.4.

Note que este diagrama es idéntico al inmediatamente anterior con la adición del URI para el recurso anteriormente anónimo. Desde el punto de vista de una segunda aplicación de consulta de este modelo, no existe distinción entre la instrucción hecha con una sola declaración y la declaración hecha en

las oraciones separadas. Algunas aplicaciones necesitarán que se haga tal distinción, sin embargo esto es soportado por el RDF.

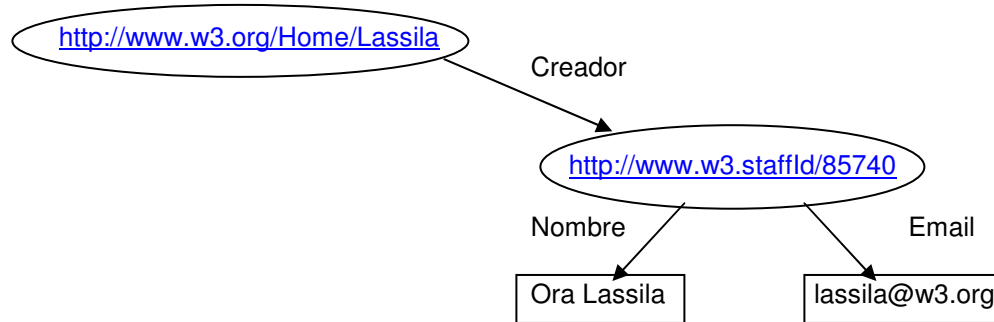


FIGURA 2.4. VALOR ESTRUCTURADO CON IDENTIFICADOR. TOMADO DE REC. RDF-SYNTAX-19990222.

EL RDFS

El RDF(S) (Se usa “RDF(S)” para referirse a la tecnología combinada de RDF y el Esquema RDF) constituye otro estándar para metadatos que está cambiando la WWW a una base de conocimiento entendible por la máquina. Este es una aplicación de XML que permite la denotación de hechos y esquema en un formato compatible en la Web, construido sobre un modelo de objeto elaborado para describir conceptos y relaciones. Así, este puede aparecer como una escogencia natural para un lenguaje de descripción ampliamente-emporado en ontología. No obstante, su falta de capacidad para describir la semántica de los conceptos y relaciones más allá de los proveídos por los mecanismos de herencias, hacen de éste un lenguaje débil, incluso, para los más austeros sistemas basados en conocimiento. Nuestro método sigue el espíritu de la WWW, tampoco se asume como un

lenguaje de especificación de axioma global, de tal manera que también es inaceptable para un propósito y es débil para otros, pero en cambio es una metodología que permite a las comunidades de usuarios especificar que axiomas son de interés en su dominio.

Para expresar el papel del RDFS (Resource Description Framework Schema) en la terminología ingeniería del conocimiento, se define como una ontología común que documentos particulares RDF pueden ser revisados otra vez para determinar consistencia.

La ontología muestra su utilidad en aplicaciones en áreas tales como integración o fraccionamiento inteligente de información. Por lo tanto su uso es muy interesante para aplicaciones Web, que puede también beneficiarse por la amplia experiencia hecha en la comunidad de adquisición de conocimiento. Al mismo tiempo, es una gran oportunidad para la comunidad de adquisición de conocimiento como el RDF(S) puede cambiar la ingeniería del conocimiento, en una poderosa tecnología y metodología.

El RDF(S) solamente ofrece las más básicas primitivas de modelación para la modelación ontológica. Incluso aunque hay buena y mala escogencia para los lenguajes formales en particular, uno debe encarar el principal intercambio entre docilidad y expresividad de un lenguaje. El RDF(S) se ha situado cerca al extremo bajo de expresividad, porque ha sido concebido

para ser aplicado a amplios recursos de la Web. En contraste al lenguaje de representación de conocimiento común, RDF(S) no da la respuesta definitiva a todos los problemas de representación del conocimiento, pero en cambio provee un lenguaje de núcleo extensible. El mecanismo de espacio de nombres y redefinición del RDF(S) permite a la comunidad de usuarios, definir su propio formato estándar RDF(S) – expandiendo las definiciones y semánticas del núcleo. Como el RDF(S) deja la huella del camino a la ingeniería del conocimiento; en este punto, debemos reconsiderar asuntos cruciales concernientes al modelado y aplicaciones de la ontología. Para nombrar un poco, se menciona el problema de integración y direccionamiento entre espacio de nombres, el tema de la escalabilidad, o la definición y uso de axiomas ontológicos.

La atención se concentra sobre lo más reciente, particularmente sobre cómo modelar axiomas en RDF(S), seguido de las cláusulas, (i), que el núcleo de la semántica del RDF(S) es re-usado tal que aplicaciones “puras” del RDF(S) puedan aún procesar las definiciones del núcleo del objeto-modelo, (ii), que la semántica es preservada entre diferentes herramientas de inferencia, y, (iii), que el axioma de modelación es adaptable a la demanda de divergencia reflejada de distintas comunidades. Por ejemplo, el primer requerimiento es violado por el lenguaje de intercambio ontológico XOL, haciendo todas las definiciones de objeto-modelo inasimilable para la mayoría de las aplicaciones RDF(S). La cláusula de intercambiabilidad y adaptabilidad es

extremadamente difícil de mantener por los analizadores basados en la representación por árboles de MetaLog, puesto que esto obliga a la formulación lógica de primer-orden. Se enseña como adaptar a la metodología general de tal manera que se está proponiendo para modelado de axioma aplicado a la ingeniería de la ontología con RDF(S). Esto está basado en la traslación de especificaciones de axioma RDF(S) a varios sistemas que proveen los servicios de inferencia. En el ejemplo, se configuran especificaciones de axiomas en un formato F-Logic que está siendo servido como el núcleo del sistema por SilRi, un servicio de inferencia para el núcleo del RDF [6]. La metodología está centrada alrededor de la categorización de axiomas, porque esto permite una descripción más concisa del significado semántico, más bien que una representación sintáctica particular de los axiomas. Esto es, se logra lo mejor en extensión y adaptabilidad para sistemas de inferencia particulares.

EI RDF en el W3C

Después de la definición de la primera especificación del RDF en 1997, de la aprobación del primer conjunto como recomendación en 1999 y de su última revisión, cuenta con una suite conformada por la Especificación de la Sintaxis RDF/XML, el Lenguaje de Descripción del Vocabulario RDF 1.0: RDF Schema, Introducción al RDF, RDF: Sintaxis de Conceptos y Resumen, Semántica del RDF y los Casos de Prueba. Todas estas especificaciones pasaron a la clasificación de Recomendaciones del W3C desde el 10 de

febrero de 2004 [6-11] por parte del W3C, dando paso al desarrollo de nuevas tecnologías, hasta el punto de estar hablándose de la segunda Web, impulsada con la aparición de la Web semántica. Avances que hoy se ven reflejados con el trabajo más recientemente de los grupos de trabajo XHTML2 y de Divulgación de la Web Semántica en el W3C vienen trabajando en los escenarios para incrustar (Embedding), RDF en HTML. El 30 de marzo de 2007, el W3C lanzó el primer documento público con estatus de Borrador de Trabajo, Casos de uso del RDFa: Escenarios para Embeber el RDF en HTML, su nombre exacto en inglés es “RDFa Use Cases: Scenarios for Embedding RDF in HTML” [12].

La importancia de esta propuesta radica, como lo presentan en el borrador de trabajo, en que las páginas Web comunes, escritas en HTML, contienen datos estructurados de importancia inherente. Cuando los publicadores puedan expresar estos datos más completamente y cuando las herramientas puedan leerlos, un nuevo mundo de funcionalidad para el usuario estará disponible, permitiéndole al usuario transferir datos estructurados entre aplicaciones y sitios Web. Un evento en una página Web puede ser directamente importado al calendario del PC de un usuario. Una licencia de un documento puede ser detectada para que el usuario automáticamente esté informado de sus derechos. Un creador de una foto, información de la configuración de la cámara, resolución y otros tópicos pueden ser publicados

tan fácilmente como la foto misma, permitiendo la búsqueda y el compartir estructurados.

RDFa es una sintaxis para expresar datos RDF estructurados en HTML. RDFa expresa metadatos en XHTML-construcción y extensión compatible, disponiendo un nuevo mundo de funcionalidades para el usuario. Escribir para lectores un tanto familiarizados con HTML, RDF y la notación N3, este escenario considera los publicadores, las herramientas constructoras y los usuarios.

2.1.1. Relación del RDF con otros estándares utilizados en Internet

Existen un conjunto de formatos de datos [13] del W3C que han nacido de la creciente adopción del XML como una sintaxis común para expresar estructuras de datos. El RDF, como un nivel superior del XML, proporciona una base común para expresiones semánticas. Aplicaciones que permiten a los programas combinación de datos lógicamente contruidos usando RDF (y por lo tanto XML) y esto ampliará la modularidad y extensibilidad de la Web. En la Figura 2.5 se presentan los formatos de datos del W3C y su relación.

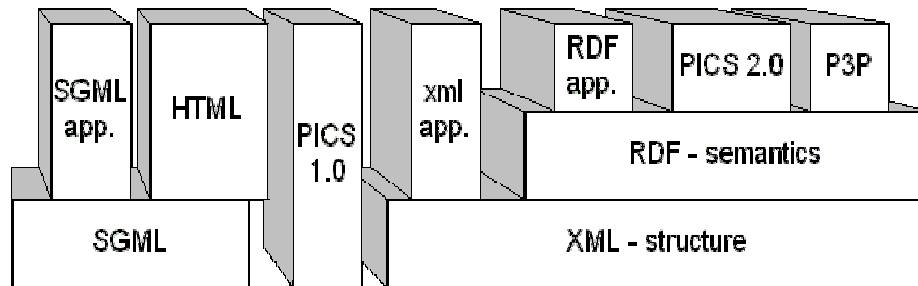


FIGURA 2.5. FORMATOS DE DATOS DEL W3C Y SUS RELACIONES. TOMADO DE W3C NOTE-RDF-19971029

Relación entre el RDF y SGML

El Lenguaje de Marcado Normalizado Estándar o Standard Generalized Markup Language (SGML) es la primera tecnología importante sobre información estructurada normalizada, que fue creada como resultado de un trabajo hecho por IBM para ofrecer una forma de formatear y mantener documentos estructurados como los documentos legales. XML es un subconjunto simplificado de SGM, que está orientado a la Web. Por tanto RDF es un subconjunto aun más restringido del SGML.

Recordemos que el W3C presentó un equipo de expertos en SGML cuyo objetivo era el de crear una nueva tecnología de marcado con las ventajas principales de SGML y con la relativa simplicidad del HTML. En 1986 el equipo lanzó la especificación XML 1.0.

Relación entre el RDF y PICS

La historia de los metadatos en el W3C [14] comenzó en 1995 con Plataforma para Selección de Contenidos en Internet, o PICS [15] o Platform for Internet Content Selection. PICS es un mecanismo para establecer clasificaciones de las páginas Web de un servidor [16]. Estas clasificaciones, o etiquetas de clasificación, contienen información acerca del contenido de las páginas Web. PICS es una infraestructura para asociar descripciones (metadatos) con contenidos de Internet. Originalmente se diseñó para ayudar a padres y profesores a controlar los contenidos a los que accedían los niños a través de Internet, pero esto también facilitó otros usos como códigos de reconocimiento, confidencialidad y administración de derechos de la propiedad intelectual. Las etiquetas PICS pueden describir cualquier aspecto de un documento o sede Web.

En vez de ser un grupo fijo de criterios, PICS introduce un mecanismo general para crear sistemas de clasificación. Quien utilice este mecanismo puede clasificar el contenido basándose en sus objetivos y valores; y los usuarios pueden configurar sus buscadores de modo que filtren las páginas Web que no cumplan los criterios que ellos han establecido. El desarrollo de PICS estuvo motivado por la anticipación, por parte de sus diseñadores, de posibles restricciones futuras en los contenidos de Internet en Estados Unidos y en otros lugares. Durante sucesivas reuniones sobre bibliotecas

digitales se pusieron de manifiesto los límites de las especificaciones PICS [17], por lo que el W3C formó un nuevo grupo de trabajo, PICS-NG (Next Generation) para dirigir las investigaciones en torno a la descripción de recursos. Posteriormente, el desarrollo de aplicaciones adicionales dio lugar a la creación del grupo de trabajo W3C RDF.

La plataforma PICS es una en la que otros servicios de evaluación y filtrado de software están siendo construidos. Padres interesados en software de búsqueda con filtro o ISPs que ofrezcan filtrado querían consultar el sitio <http://www.netparents.org>.

El RDF permitirá expresar todo lo que el PICS puede expresar, pero permite también valores, cadena de caracteres y estructurados, y algunas otras ingeniosas características.

Relación entre el RDF y DC

El Dublin Core (DC) [18] nació hace cinco años, de manos de OCLC (Online Computer Library Centre) y de NCSA (National Centre for Supercomputing Applications). Su objetivo: definir un conjunto básico de atributos que sirvan para describir todos los recursos existentes en la red.

Las principales características [19] del Dublin Core son:

Simplicidad, pensado para que pueda ser utilizado tanto por bibliotecarios como por cualquier autor que desee describir sus documentos y aumentar su visibilidad.

Consenso internacional en el número y definición de los elementos.

Flexibilidad, nada en el DC es obligatorio, todos los elementos son opcionales y repetibles, así el usuario elige la profundidad de una descripción.

El formato Dublin Core ya ha sido declarado RFC 2431 (Request for Comments) por la IETF en septiembre de 1998 y se espera que en breve se publiquen otras RFCs que especifiquen el Dublin Core Extendido.

Los elementos metadatos de DC, versión 1.1, del 02/07/1999, se definen en el RFC2413 (<http://dublincore.org/documents/1999/07/02/dces/#rfc2413>) y también en <http://dublincore.org/documents/1999/07/02/dces/>.

Se han presentado las características de DC y queda entendido que éste es otro formato de metadatos más, con un interés particular. Como se ha presentado RDF es una propuesta generalizada, que está por encima de DC.

Relación entre RDF y XML

La relación entre RDF y XML [20] es semejante a la del hijo y el padre, al emplear el primero la sintaxis del último. También se afirmó, en el ítem 2.1.1., que son complementarios. Es así como RDF ofrece una manera de describir recursos Web por medio de un vocabulario XML.

Relación entre el RDF y XSL

Traigamos la definición de XSL de Morrisón

“XSL (Lenguaje de Estilos eXtensible) es una tecnología de hojas de estilos que está específicamente diseñada para XML, aunque, de alguna forma, XSL lleva a cabo la misma tarea que la CSS (Hoja de Estilos en Cascada). Sin embargo, XSL va mucho más allá que la CSS a la hora de manipular la estructura lógica de los documentos XML. Esta potencia adicional tiene un coste: XSL incluye algunas construcciones programáticas de gran potencia que pueden ser complejas. Aparte de ser más difícil de aprender que las CSS, XSL es una técnica demasiado nueva.” [21].

Al ser RDF un subconjunto, una aplicación, de XML, por ende XSL es empleado en RDF, como una solución, para proporcionar la forma de dotar de estilo a los documentos RDF.

Relación entre el RDF y DOM

El Modelo de Objetos de Documentos o Document Object Model (DOM) del XML, es el medio por el que se puede acceder y manipular un documento XML. Sin el DOM, XML no sería más que un sistema de almacenamiento para datos a los que se accede en virtud de distintos métodos originales. Con el DOM, XML empieza a plantear su promesa como lenguaje de programación universal, de plataforma cruzada e independiente de la aplicación [22]. Al ser RDF un subconjunto, una aplicación, de XML, implícitamente DOM es empleada en RDF.

Mozilla emplea la DOM en RDF. Existen diversas formas para que los usuarios puedan emplearla, una de ellas es descrita por Gregor Bornemann en <http://www.geocrawler.com/archives/3/125/2001/7/0/6204381/>.

Relación entre el RDF y P3P

La Plataforma para Proyectos de Preferencias de Privacidad o Platform for Privacy Preferences Project (P3P) [23] nace al reconocer el W3C la necesidad de estandarizar las normas de privacidad de la Web. P3P permite a los usuarios establecer preferencias que rigen el modo en la que la información se va a compartir con los sitios Web. Una parte de este proceso se ha automatizado a través de un vocabulario basado en XML que anuncian normas de privacidad. Su formalización se encuentra en la

Nota del 25 de enero de 2002 del W3C, "An RDF Schema for P3P", ubicada en <http://www.w3.org/TR/2002/NOTE-p3p-rdfschema-20020125>.

Cabe mencionar que TRUSTE [24] es una organización que trabaja para hacer cumplir las normas de privacidad llevando a cabo comprobaciones regulares en las organizaciones que hayan aceptado someterse a tal control.

Dado que la codificación manual de las normas de privacidad es tediosa existen servicios automáticos y en línea como el que presta LinkExchange de Microsoft, solo basta con registrarse en <http://privacy.linkexchange.com/pwSignup.asp>.

El vocabulario XML que utiliza P3P se conoce como vocabulario armonizado y se basa en el RDF. Por lo que la relación existente entre RDF y P3P radica en que este último emplea al RDF en su vocabulario armonizado.

2.1.2. Aplicaciones del RDF

Dublin Core con RDF

Se ha efectuado la especificación y empleo del metadato Dublin Core haciendo uso de la Especificación de Dato XML y el RDF [25]. Al respecto se ha considerado:

- Repetición y preservación del orden del elemento,

- Estructura y agrupación de los elementos, y
- Extensión y herencia del esquema.

Adicionalmente se encuentran afirmaciones de que el desarrollo del RDF provee un medio en que la meta semántica del Dublin Core puede ser preservada y expresada, mientras que permite la definición simultánea de una sintaxis utilizando el XML. Un rango de otras sintaxis pueden seguirse empleando para la implementación del Dublin Core, sin embargo solo la combinación del RDF y el XML actualmente ofrecen en rango de capacidades requeridas para expresar ampliamente la riqueza, tanto del Dublin Core limitado como del ilimitado, sin ambigüedad [26]. Se concluye que originalmente convergen ideas concurrentes de los conceptos detrás del Dublin Core con las facilidades sintácticas y estructurales ofrecidas por el XML y RDF dándole un arma poderosa en el arsenal del uso y manipulación de información. Así también, Dublin Core ofrece una manera para que los diversos recursos puedan ser descritos por descubrimiento en un contexto interdisciplinario y el XML/RDF provee la estructura para las expresiones sin ambigüedad de la información Dublin Core [27], así como también las más detalladas descripciones de las comunidades involucradas.

La estrategia de Netscape de implementación del RDF

El RDF es una tecnología que está siendo usada en Mozilla para integrar y agregar recursos de Internet. En particular, es usado en los servicios de

metadatos Aurora [28] y SmartBrowsing [29]. Aurora es una interface de usuario sencilla para manejar palabras claves, crónicas, resultados de búsquedas, sistemas de archivo, ftp, mapas de sitios, etc. Estos “tipos de datos” referenciados en el modelo de datos del RDF. SmartBrowsing es un mecanismo que permite confiar en los servicios de terceros de suministro de anotaciones metadatos para las páginas vistas por los buscadores en la Web.

En la implementación de RDF en Mozilla, Mozilla tiene su modelo RDF que tiene leves diferencias de la Especificación del Modelo y la Sintaxis RDF, ítem 4.2.1, diferencias que no afectan el uso del modelo RDF. Las interfaces empleadas para interactuar con RDF son:

- nsIRDFService. Servicio RDF.
- nsIRDFNode. Interface para un nodo en grafo RDF.
- nsIRDFDataSource. Provee el acceso a la colección de “instrucciones relacionadas” (o un “subgrafo”).
- nsIRDFCompsiteDataSource. Es derivada de nsIRDFDataSource.
- nsIRDFObserver. Es una interface que implementa un cliente RDF.
- nIRDFContainer. Permite el acceso simplificado a un objeto contenedor (container) RDF.

Los detalles de la implementación del Modelo RDF en Mozilla se describen en el documento Back-End Architecture [30].

Representación del vCard en RDF

Se hace referencia al vCard como la tarjeta de presentación de perfil comercial definida por el RFC 2425. No se intenta crear una definición independiente para el esquema vCard, el objetivo es definir una alternativa de codificación en RDF para el formato definido tradicionalmente [31].

La mayoría de las propiedades-tipo de vCard son cadenas de caracteres como sus valores y simplemente son representadas por su nombre de propiedad-tipo y en valor como lo especifica vCard:

FN, NICKNAME, BDAY, MAILER, GEO, TITLE, ROLE, CATEGORIES, NAME, SOURCE, NOTE, PRODID, REV, SORT-STRING y CLASS.

Por ejemplo,

```
<vCard:FN> Corky Crystal </vCard:FN>  
<vCard:BDAY> 1980-01-01 </vCard:BDAY>  
<vCard:TITLE> Computer Officer Class 3 </vCard:TITLE>  
<vCard:ROLE> Programmer </vCard:ROLE>
```

El espacio de nombres de RDF para vCard es <http://imc.org/vCard/3.0#>.

Representando UML en RDF

La comunidad UML desarrolló un grupo de modelos muy usados para representar componentes estáticos y dinámicos de sistemas de software-concertado. El UML es un estándar de la industria y sirve como una base de modelación para estándares emergentes en otras áreas como OIM, CWN, etc. Es así, como hoy existe una variedad de vocabularios UML para describir modelos de objetos, tipos de datos, esquemas de base de datos, transformaciones, etc.

La meta de este trabajo es hacer “Compatible a RDF” con UML. Esto permite mezclar y extender los modelos UML y los elementos del lenguaje del mismo UML en la Web de una manera abierta. El XMI, el estándar común para codificar UML en XML por OMG, no ofrece esta capacidad. Él está basado en un laborioso-alambrado de la DTD. Por ejemplo, si un tercero fuera a clarificar el concepto “Event” definido dentro de la carta del UML dice “ExternalEven” e “InternalEven”, no sería posible publicar la correspondiente instancia del evento en XMI.

Aquí el mapeo sugerido del UML/RDF [32] no es una codificación especial para UML. La idea principal fue crear un URL para cualquier identificación del ente UML (Ej. <http://www.omg.org/uml/1.3/Foundation.Core.Class> o <http://www.omg.org/uml/1.3/Foundation.Data.Types.Boolean.true>). El propósito de la asociación del UML corresponde con las propiedades del

RDF, el resto es conversión mecánica. Los URLs para UML/RDF se escogieron para incluir las etiquetas de los nombres usados en XMI como sufijos para facilitar la conversión hacia/desde XMI.

La representación del modelo conceptual del UML en RDF es similar a definir una especificación alternativa del Esquema RDF. El cargador mismo del UML usa el paquete `Foundation.Core` en una forma similar a RDFS. Sin embargo, éste contiene muchos detalles que van más allá del ámbito del RDFS y es significativamente más elocuente.

El siguiente código usa solamente los constructores básicos del RDF tales como `rdf:type` y `rdf:Seq`. La codificación es incompleta (ej. sin “elementos auxiliares” del paquete `Foundation.Core`) y puede contener errores “semánticos”. Los modelos UML de Mayor-nivel (igual que cartas de los Estados) pueden ser codificados usando el vocabulario UML/RDF propuesto. Ilustra como la flexibilidad del RDF puede ser desplegada para mezclar el vocabulario de la máquina de estado del UML con `DublinCore`. Se preserva la consistencia del modelo. La instancia de los datos bosquejados a continuación no puede ser representada en XMI sin la violación de la DTM del XMI.

```
<rdf:RDF
  xmlns:s="http://www.omg.org/uml/1.3/Behavioral_Elements.State_Machines."
  xmlns:dc="http://purl.org/dc/elements/1.1/">
```

```
<s:StateMachine rdf:ID="MyMicrowave">
```

```

<s:StateMachine.top>
  <s:State rdf:ID="PowerOn"
    dc:Description="This is the start state for the microwave upon
powering on"/>
</s:StateMachine.top>
</s:StateMachine>

```

La representación gráfica se presenta en la Figura 2.6.

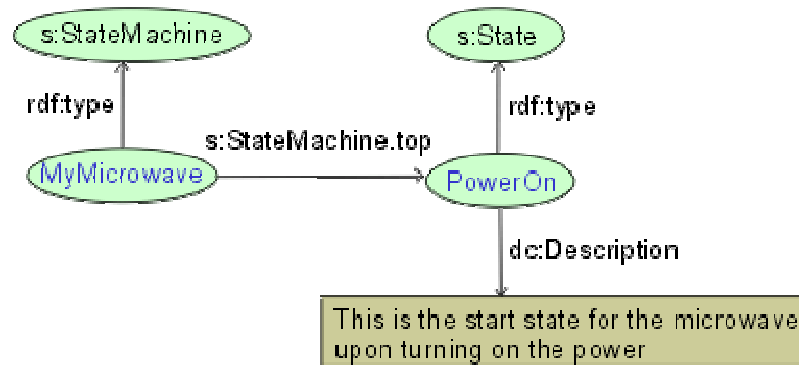


FIGURA 2.6. REPRESENTACIÓN GRÁFICA DE LA CODIFICACIÓN UML/RDF. TOMADO DE REPRESENTING UML IN RDF, [HTTP://INFOLAB.STANFORD.EDU/~MELNIK/RDF/UML/](http://infolab.stanford.edu/~melnik/rdf/uml/).

Para mayor información dirigirse a A Discussion of the Relationship Between RDF-Schema and UML, Nota del W3C, ubicada en <http://www.w3.org/TR/NOTE-rdf-uml/>.

2.2. ONTOLOGÍA

En filosofía, la **ontología** (del griego ὄν, genitivo ὄντος: de ser (parte de εἶναι: ser) y -λογία: ciencia, estudio, teoría) o *Metafísica general* es el estudio de lo que es, en tanto que es y existe. Por ello es llamada *la teoría del ser*, es decir, el estudio de todo lo que es: qué es, cómo es y cómo es posible. La Ontología se ocupa de la definición del ser y de establecer las categorías

fundamentales o modos generales de ser de las cosas a partir del estudio de sus propiedades.

Una ontología es también una entidad computacional, por lo que nos referiremos a la definición formal propuesta por W3C [33], para lo que el autor se ha permitido traducirla a continuación.

Una ontología define los términos usados para describir y representar un área del conocimiento. Las ontologías son usadas por personas, bases de datos y aplicaciones que necesitan compartir información específica de un sujeto (dominio), tales como medicina, herramientas para la manufactura, bienes inmuebles, reparación de automóviles, administración financiera, etc. La ontología incluye definiciones, usables por el computador, de conceptos básicos en el dominio y sus relaciones. Ellas encierran conocimientos en un dominio y también conocimiento de los dominios que abarcan. En esta forma, ellas hacen reusable el conocimiento.

Para mayor claridad del tema bien vale la pena abstraer la conceptualización realizada por Abián M. [34], en La Web Semántica Hoy, quien aborda con claridad y sencillez el tema.

Las ontologías son imprescindibles para conseguir la Web Semántica. Los usuarios organizarán la información para que los agentes de software

puedan interpretar el significado y podrán buscar e integrar datos mejor que en la actualidad. El conocimiento disponible en las otologías será la fuente de extracción automática de datos en la Web para las aplicaciones, procesarlos y obtener conclusiones, permitiéndoles a partir de éstas obtener conclusiones que apoyen la toma de decisiones y de esta forma negociar con otros agentes o personas.

2.2.1. Representación del conocimiento

Para que la Web Semántica funcione, los computadores deben tener acceso a colecciones estructuradas de información y a un grupo de reglas de inferencia que puedan usar para conducirlo al razonamiento autónomo. Los investigadores de la Inteligencia Artificial han estudiado tales sistemas antes del desarrollo de la Web: La representación del conocimiento (Knowledge representation), como esta tecnología es a menudo llamada, está corrientemente en el estado comparable al hipertexto antes del advenimiento de la Web: ésta es claramente una buena idea, y una muy buena demostración existente, pero esto no ha cambiado aún el mundo. Éste contiene las semillas de importantes aplicaciones, pero para realizar este potencial completo debe ser enlazado a un sistema global único.

El sistema de representación de conocimiento tradicional ha sido centralizado, requiriendo que todos compartan exactamente la misma definición de conceptos comunes tales como “padre” o “vehículo”. Pero el

control central es agobiante, y el incremento del tamaño y alcance de tal sistema rápidamente puede ser inmanejable.

Además, este sistema tiene límites usualmente cuidadoso de las preguntas que pueden ser contestadas para que el computador pueda contestar realmente – o contestar de todo. El problema es reminiscencia del teorema de Godel para las matemáticas: un sistema que es bastante complejo para ser muy usado también rodea preguntas incontestables, las versiones más sofisticadas de la paradoja básica “Esta frase es falsa”. Para evitar tales problemas, generalmente los sistemas tradicionales de representación del conocimiento tienen su propia dirección e idiosincrasia del grupo de reglas para hacer inferencia sobre sus datos. Por ejemplo, un sistema de genealogía, actuando en una base de datos o árbol familiar, puede incluir la regla “una esposa de un tío es una tía”

2.2.2. Lenguajes de especificación ontológica

El desarrollo logrado y por ende el subsiguiente no hubiese sido ni será posible sin los lenguajes disponibles y en desarrollo, que son la plataforma de lanzamiento para promover la aplicación de esta tecnología. Entre ellos se mencionan los siguientes, sin demeritar los no nombrados.

Corcho y Gómez [35] presentan un análisis de los lenguajes de especificación ontológica (Ontology Specification Languages); a) los que

vienen siendo empleados ampliamente por la comunidad ontológica, tales como Ontolingua [36], OKBC [37] (<http://www-ksl.stanford.edu/software/OKBC/>), OCML [38] (<http://kmi.open.ac.uk/projects/ocml/>), FLogic [39] y LOOM [40]; b) lenguajes creados en el contexto de Internet, recomendados por el W3C como XML [41], RDF [42] y RDFS [43] y; c) nuevos lenguajes para la especificación ontológica, como XOL [44] (<http://www.ai.sri.com/pkarp/xol/xol.html>), SHOE [45] y OIL [46].

Con base en la información disponible y con el fin de disponer de una concepción fundamentada en las características de los lenguajes se presenta la Tabla 2.1.

TABLA 2.1. COMPARACIÓN DE LAS CARACTERÍSTICAS DE LOS LENGUAJES DE ESPECIFICACIÓN ONTOLÓGICA.

Lenguaje	Elementos	Semántica	Base	Metadata	Reglas raz
Ontolingua	Objetos, funciones y relaciones	Declarativa	Cálculo de predicado de primer orden	Si	No monotónica
OCML	Clases, instancias, Funciones, relaciones, reglas	Declarativa "assertion"	Lenguaje estructurado	Si	Si
FLogic	O-O	Modelo-teórico	Lenguaje estructurado y Cálculo de predicado de primer orden		No
LOOM	Objeto, relaciones	Descriptiva	Lenguaje estructurado	Si	Si
RDF	Recursos, propiedades, objeto	Declarativa	Lenguaje estructurado	Si	No

Continuación Tabla 2.1.

Lenguaje	Elementos	Semántica	Base	Metadata	Reglas raz
RDF Schema	Clases, recursos, propiedades	Declarativo	Lenguaje estructurado	Si	No
XOL	Clases, slot , facets	object-oriented knowledge representation systems, OKBC [47]		Si	No
SHOE	instancias	Agentes		No	monotonic, sin negación
OIL	Concepto, relaciones, funciones y axiomas	RDFs	Lenguaje estructurado.	Si	Si
OWL		RDF	Lenguaje estructurado.	Si	Si
Lenguaje	Elementos	Semántica	Base	Metadata	Reglas raz

2.3. LA WEB SEMÁNTICA

Hasta el momento se ha presentado el RDF y los lenguajes ontológicos, de este último, el de mayor relevancia es el Lenguaje de Ontología para la Web (OWL). Ahora estas dos tecnologías han causado un gran impacto, hasta el punto de ser los responsables de la nueva tendencia en la WWW hacia la Web Semántica como afirmó Berners-Lee [48] y que algunos la llaman la Web 2.0. Por lo que se presenta a continuación sus principales conceptos:

“El primer paso es colocar los datos en la Web de un modo en que las máquinas puedan entenderlos naturalmente o convertirlos a esa forma. Esto

crea lo que yo llamo una Web semántica: una red de datos que pueden ser procesados directamente o indirectamente por máquinas” [49].

La Web semántica proporcionará estructura al contenido importante de las páginas Web, creando un entorno donde los agentes de software que viajan de página en página puedan enseguida llevar a cabo complicadas tareas para los usuarios. Así, un agente que venga de la página Web de la clínica no sólo sabrá que la página tiene palabras como “tratamiento, medicina, médico, terapia” (como puede codificarse hoy), sino también que el Dr. Hartman trabaja en la clínica los lunes, miércoles y viernes, y que el script (programa no compilado realizado en un lenguaje de programación sencillo) toma un rango de datos en el formato año-mes-día y devuelve fechas y horas de consulta [50].

La definición oficial de la W3C, La Web Semántica provee un marco de trabajo común que permite compartir y re-usar datos a través de una aplicación, empresa y comunidad delimitada. Es un esfuerzo colaborativo liderado por W3C con participación de un gran número de investigadores e industriales. Está basada en RDF.

Se dispone entonces de lenguajes especializados, como es el caso del OWL, el Lenguaje de Ontología de la Web (The Web Ontology Language) que proporciona un lenguaje para definición estructurada, ontologías basadas en

la Web que entregan rica integración e interoperabilidad de datos entre comunidades descriptivas. Donde recientemente se han usado lenguajes para desarrollar herramientas y ontologías para comunidades de usuarios específicos (particularmente en las ciencias y en aplicaciones de comercio electrónico en compañías específicas), no fueron definidas para ser compatibles con la arquitectura de la WWW en general y la Web Semántica en particular.

Hay que destacar la fuerza del tema en el sector, es la tendencia, hacia allá vamos, por lo que se dispone de diversos grupos de trabajo, grupos de discusión, eventos especializados y un grupo amplio de herramientas, tanto del sector comercial como del software libre. Tanto es así que se cuenta hasta con cooperativas, como es el caso de Cooperative Ontologies Programme [51] y en éste hasta se dispone de cursos detallados, como el caso de An Introduction to Semantics! [52]. Algunos investigadores vienen anunciando que la Web Semántica es una revolución, como lo expresó la Dra. Asunción de María Gómez Pérez en la entrevista concedida al Heraldo de Aragón, del día martes 7 de marzo de 2006 [53].

2.3.1. La Ciencia de la Web

Debido al desarrollo de la WWW, aparece la iniciativa de Investigación de Ciencia de la Web (Web Science Research Initiative - WSRI), un esfuerzo

entre la Ciencia de la Computación y el Laboratorio de Inteligencia Artificial (CSAIL) en MIT y la Escuela de Electrónica y Ciencias de la computación (ECS) en la Universidad de Southampton. La meta de la WSRI es facilitar y producir los avances científicos fundamentales necesarios para informar del diseño futuro y uso de la WWW [54].

Aquí se emplea el término “ciencia” en dos formas. Ciencias física y biológica analizan el mundo natural y tratan de hallar leyes microscópicas que, extrapoladas a la realidad macroscópica, puedan generar el comportamiento observado. La ciencia de la computación, en contraste, aunque en parte analítica, es principalmente sintética: Es la encargada de la construcción de nuevos lenguajes y algoritmos para producir el comportamiento deseado del computador. La ciencia de la Web es una combinación de estas dos características.

Dado el panorama, actualidad e importancia del tema se presenta al lector una relación de los organismos reconocidos y sitios Web como guía para profundizar el tema, así:

W3C – World Wide Web Consortium, relación de actividades, <http://www.w3.org/2001/sw/> y su Grupo Coordinador en Web Semántica, <http://www.w3.org/2001/sw/CG/>.

Maryland information and network dynamics lab semantic web agents project
– mindswap, <http://www.mindswap.org/>.

[Laboratorio MIND](http://www.mindswap.org/) de la [University of Maryland Institute for Advanced Computer Studies](http://www.mindswap.org/), <http://www.mindswap.org/>.

2.4. LOGS

A menudo se requiere que eventos y otros parámetros operacionales sean guardados en logs, que pueden de ser analizados por intereses de medición. Hay diferentes tipos de eventos que pueden ser gravados, tales como intercambio de mensajes, utilización de recursos, latencia o falla y sucesos comunes.

Como consecuencia se dispone de numerosos sistemas que emplean diversos formatos para guardar o registrar los eventos, y a su vez se propicia la existencia de herramientas para el análisis de logs.

Los servidores Web, como software, difieren en detalles pero todos comparten algunas características básicas [55]:

HTTP. Responde a requerimientos http. Cada programa servidor Web opera por aceptación de requerimientos http de la red y proveyendo una respuesta http al requerimiento. La respuesta http típicamente consiste en un

documento html, pero puede ser un archivo de texto en bruto, una imagen o algún otro tipo de documento; si se haya algo malo en el requerimiento de un cliente o mientras intenta servir el requerimiento, un servidor Web envía una respuesta de error que puede incluir algún mensaje HTML o de texto para mejor explicación del problema al usuario final.

Logging. Usualmente los servidores Web tienen además la capacidad de registrar (logging) información detallada, sobre los requerimientos del cliente y las respuestas, en un archivo log; éste permite al Webmaster reunir estadísticas mediante el empleo de analizadores log sobre los archivos logs.

Los cuatro programas servidores HTTP más comunes son:

- Servidor HTTP Apache de la Apache Software Foundation (<http://www.apache.org/>).
- Servicios de Información Internet (IIS) de Microsoft (www.microsoft.com).
- Sistema de Servidor Web Java de Sun de Sun Microsystems, formalmente Servidor Web Sun ONE, iPlanet y Netscape Enterprise (<http://java.sun.com/j2ee/1.3/docs/tutorial/doc/>).
- Servidor Web Zeus de Zeus Technology (<http://www.zeus.com/>).

La Figura 2.7, presenta la segmentación del mercado entre los servidores Web más comunes, indicando porcentaje del 59 para Apache [56].

Hay cientos de programas servidores Web disponibles, muchos de los cuales son especializados para muchos propósitos específicos.

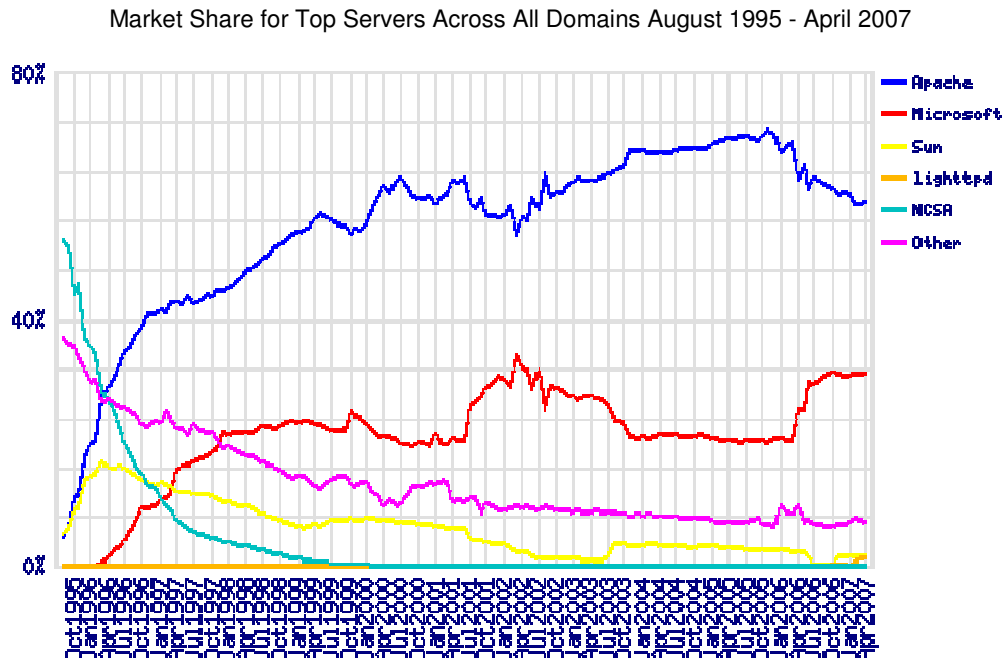


FIGURA 2.7. DISTRIBUCIÓN DEL MERCADO DE SERVIDORES WEB, ENTRE AGOSTO DE 1995 Y ABRIL DE 2007. TOMADO DE [HTTP://NEWS.NETCRAFT.COM/ARCHIVES/WEB_SERVER_SURVEY.HTML](http://news.netcraft.com/archives/web_server_survey.html).

2.4.1. LOGS POR ICMP

Dado que en el protocolo TCP/IP se encuentra una de las fuentes de generación de los logs al disponerse de un mecanismo de mensajes de propósito especial, a continuación se presenta a grandes rasgos su funcionamiento el Protocolo de Mensajes de Control Internet, ICMP [57].

Cuando un datagrama causa un error, el ICPM sólo puede reportar la condición de error a la fuente original del datagrama; la fuente debe relacionar el error con un programa de aplicación individual o debe tomar otra acción para corregir el problema. El programa individual al que se refiere aquí puede ser, entre otros, el servidor Web, que es el que maneja los mensajes de nuestro interés al ser una fuente ésta para el origen de los mensajes Logs. Recordemos que este tipo de mensaje viaja encapsulado en el datagrama IP.

Aunque este tipo de logs no son específicamente de nuestro interés, dada su importancia y complementariedad se trajo a mención; sin embargo, son los logs de los servicios Webs (Web Services) los que ocuparán toda nuestra atención, por lo que se abordan en la siguiente sección.

2.4.2. WEB LOGS

Holliday et all. [58], contempla la ontología de los logs. El autor presenta las propuestas de manejo de los logs más empleados: Jakarta Commons Logging - JCL, Apache Log4J - log4J y NetLogger, y adicionalmente el estándar ontológico propuesto por ellos el GULF, como formato común.

Tomcat y Axis han adoptado el Apache Jakarta Commons Logging (JCL) para realizar el registro de logs. Otra de las herramientas de registro de logs (logging), a menudo usada es Apache Log4J.

La combinación de JCL y Log4J es usada para logging de aplicaciones Webs, tales como lo hacen Tomcat y Axis.

El registro de logs o eventos NetLogger está dividido en campos, en que cada campo tiene tres partes: tipo de código, la llave y el valor. En el que el tipo de código indica como interpretar el valor. Los registros NetLogger con “s” denotan una cadena de caracteres (string) y “t” un registro de tiempo. La llave es una cadena de caracteres que describe el nombre del valor.

t DATE: 2004-04-15T21:30:01.425059

s LVL: Information

s TGT: appended

El registro de logs DAMED-WG tiene cuatro partes: medida del tiempo, tipo de evento, destino y valor del tipo de evento.

El flujo de control

Las aplicaciones hacen llamados log sobre objetos Logger. Entendamos por Logger los objetos manejadores de los logs. Los Loggers están organizados en espacios de nombres jerárquico y los Loggers hijos pueden heredar algunas propiedades log de sus padres en el espacio de nombres [59].

Las aplicaciones hacen llamados logs a un objeto Loggers. El objeto Logger destina objetos LogRecord que pueden ser pasados a objetos Manejadores (Handler) para publicación. Los Loggers y los Manejadores pueden emplear Niveles de logs y (opcionalmente) Filtros para decidir si ellos están interesados en un logRecord particular. Cuando es necesario publicar externamente un LogRecord, un Manejador puede (opcionalmente) usar un Formateador para localizar y formatear el mensaje antes de ser publicado en una cadena de I/O. Figura 2.8.

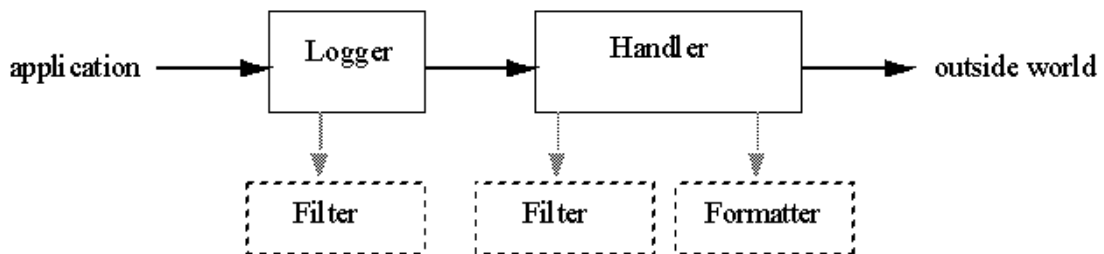


FIGURA 2.8. DIAGRAMA DE FLUJO DEL LOG. TOMADA DE JAVA LOGGING OVERVIEW,
 HTTP
[HTTP://JAVA.SUN.COM/J2SE/1.4.2/DOCS/API/JAVA/UTIL/LOGGING/LOGRECORD.
 HTML.](http://java.sun.com/j2se/1.4.2/docs/api/java/util/logging/LogRecord.html)

Cada Logger mantiene el seguimiento de un grupo de Manejadores de salida. Preestablecida (default) todos los Loggers también envían su salida a su Logger padre. Pero los Loggers pueden también ser configurados para ignorar Manejadores de mayor jerarquía.

Algunos Manejadores tienen salida directa a otros Manejadores. Por ejemplo, el Manejador de Memoria mantiene un buffer en anillo de RegistrosLog, y un disparador de eventos publica sus RegistrosLog a través de un Manejador objetivo. En tales casos, un formateo es hecho por el último Manejador en la cadena. Figura 2.9.

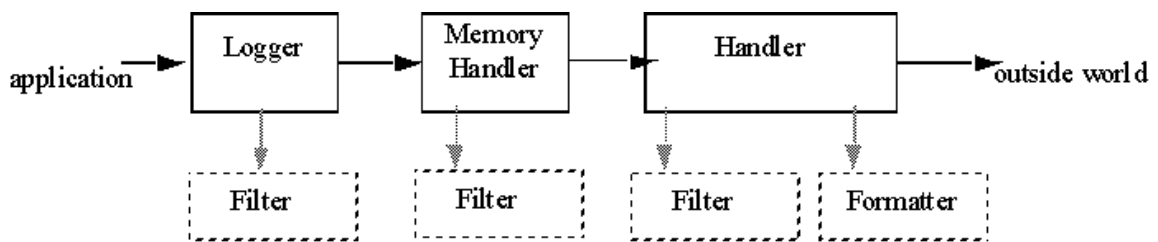


FIGURA 2.9. DIAGRAMA DE FLUJO CON MANEJADOR DE MEMORIA. TOMADA DE JAVA LOGGING OVERVIEW, [HTTP://JAVA.SUN.COM/J2SE/1.4.2/DOCS/API/JAVA/UTIL/LOGGING/LOGRECORD.HTML](http://java.sun.com/j2se/1.4.2/docs/api/java/util/logging/LogRecord.html).

Las APIs son estructuradas para que hagan un llamado a las APIs Logger que puede ser barato cuando el logging es deshabilitado. Si el logging está deshabilitado para un nivel dado de log, entonces el Logger puede hacer una prueba de comparación barata y regresar. Si el logging está habilitado para un nivel log dado, el Logger es aún cuidadoso para minimizar costos ante el revisor gramatical del RegistroLog a los Manejadores. En particular, la localización y formateo (que son relativamente caros) son diferidos hasta los requerimientos al Manejador. Por ejemplo, un ManejadorDeMemoria puede mantener un buffer circular de RegistrosLog sin tener que pagar el costo de formateo.

El formato del archivo log

Se encuentra que W3C definió un **formato común del archivo log** [60], así:

```
remotehost rfc931 authuser [date] "request" status bytes
```

donde,

remotehost. Nombre del anfitrión remoto (o número IP si el servidor DNS no está disponible, o si el DNSLookup está apagado).

rfc931. El nombre de registro remoto del usuario. Permite determinar la identidad de un usuario de una conexión TCP particular.

authuser. Nombre del usuario como él mismo se autenticó.

[date] . Fecha y hora del requerimiento.

"request" . La línea exacta del requerimiento como llega del cliente

status. El código de estado HTTP retornado al cliente.

bytes. La longitud del contenido del documento transferido.

Posteriormente, el W3C trabajó en una propuesta de un **formato extendido para el archivo Log** [61].

Como resultado de las ambigüedades halladas en el análisis del archivo log en el formato común pueden ocurrir entre los campos caracteres

separadores de campo. El formato de archivo log extendido está diseñado para mantener las siguientes necesidades:

- Permite el control sobre el registro de dato.
- Soporta las necesidades de los Proxys, clientes y servidores en formato común.
- Provee manejo robusto de edición de caracteres escaping.
- Permite el intercambio demográfico de dato.
- Permite el resumen de datos.

El formato de archivo log descrito permite personalizar archivos log para ser almacenado en un formato legible por las herramientas de análisis genéricas. Un encabezado especificando el tipo de dato almacenado se escribe antes del inicio de cada log.

Recientemente, el W3C presenta el **Formato de archivo Log para Tiendas**, como una mejora al formato extendido presentado anteriormente, W3C [62].

Las características del formato se presentan a continuación:

- El formato puede ser analizado por todas las herramientas convencionales de archivo log en virtud del principio de compatibilidad especificada por el W3C.

- El formato es fácil de integrar con sistemas estándares o aplicaciones de tiendas (ej. Los basados en Java, Allaire ColdFusion, Microsoft Active Server Page (ASP)).
- El formato es soportado por los sistemas de los vendedores de tiendas tales como AEON, Hybris, Internolix, Intershop, Openshop, and SAP.

Analizadores sistemáticos de logs

Slogger (**S**emantic **L**og-**A**nalys**e**r) es un programa que realiza el análisis y evaluación de aplicaciones e infraestructuras distribuidas. Este usa la tecnología Web semántica para asistir en la unificación de datos heterogéneos.

Existen en el mercado numerosos productos para el análisis de logs y la W3C presenta una evidencia W3C [63]. Adicionalmente, podríamos visitar el sitio de la UPPSALA Universitet [64], que presenta analizadores para diferentes tipos de plataforma.

2.5. HERRAMIENTAS

Estos planteamientos vienen impulsando el desarrollo de herramientas computacionales que facilitan la aplicación de estas tecnologías, como son el caso de Protégé, Sesame, OntoStudio, R2O+ODEMapster y Jena. En la presentación a continuación, se profundiza en las dos primeras herramientas por ser éstas las que el autor empleó para su aplicación.

2.5.1. Protégé

Protégé [65] es una herramienta de software integrada, usada para desarrollar sistemas y dominios expertos para desarrollar sistemas de bases de conocimiento. Las aplicaciones desarrolladas en Protégé son usadas en la solución de problemas y toma de decisiones en un dominio particular. Es la plataforma independiente del medio para crear y editar ontologías y bases de conocimiento que permite a los usuarios comenzar a diseñar rápida e intuitivamente una ontología. La referencia de sus creadores se presenta en su manual “Getting Started with Protégé-Frames”.

Es necesario introducir el marco conceptual expuesto, en la estructura y terminología de Protégé, aspecto que presenta Noy y McGinness [66]. Donde para Protégé una ontología es una descripción explícita y formal de conceptos en un dominio de discurso (**clases** (a veces llamadas conceptos)), propiedades de cada concepto describiendo varias características y atributos del concepto (**slots** (a veces llamados roles o propiedades)), y restricciones sobre los slots (**facetos** (algunas veces llamadas restricciones de rol)).

Desarrollo de ontologías en Protégé

No existe un único procedimiento establecido para desarrollar una ontología; sin embargo, Noy y McGinness [66], hacen una propuesta que presentan y analizan paso a paso y con ejemplo aplica el procedimiento sugerido, e inclusive plantean diferentes formas de abordaje de algunas de las etapas. El

planteamiento, que es aplicado más adelante en el presente trabajo en el ítem 4.1.1. “Definición de vocabulario y ontología” del capítulo 4, y tiene aun mayor valor al estar propuesto para una de las herramientas más comunes en el medio, como lo es Protégé.

2.5.2. Sesame

Sesame [67] es un marco de trabajo de dominio público (open source) en Java para almacenamiento, consulta y razonamiento con RDF y RDF Schema. Llega a ocupar el papel de almacenamiento y recuperación para ontologías y metadatos expresado en RDF y RDF Schema. Puede ser usado como una base de datos para RDF y RDF Schema, o como una librería Java para aplicaciones que necesitan trabajar internamente con RDF. Por ejemplo, suponga que necesita leer un archivo grande de RDF, hallar información relevante para su aplicación y usar esta información. Sesame le proporciona las herramientas necesarias para análisis sintáctico, intérprete, consulta y almacenaje, toda esta información, incorporada en su propia aplicación, si usted quiere, o, si usted prefiere, una base de datos separada o igualmente en un servidor remoto. Generalmente Sesame provee al desarrollador de aplicaciones una caja de herramientas que contiene lo más usado, martillos, destornilladores, etc. para hacer “Hágalo usted mismo” con RDF.

La librería Sesame

La librería Sesame consiste de un solo archivo, `sesame.jar`, que contiene clases Java listas para usar en su aplicación. En el Manual del Usuario de Sesame [68] se haya la explicación y ejemplos de cómo emplear Sesame en sus códigos en: Cómo hacer queries (how to do queries) y cómo adicionar, remover datos (how to add and remove data), etc.

El Servidor Sesame

Sesame puede ser usado como un Servidor en que las aplicaciones clientes (o usuarios humanos) pueden comunicarse sobre http (Ver Figura 2.10). Sesame puede ser desarrollado como una Aplicación Java Servlet en Apache Tomcat, un servidor Web que soporta Servlets de Java y la tecnología JSP.

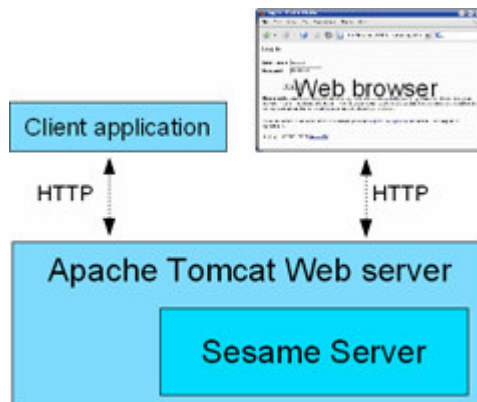


FIGURA 2.10. EL SERVIDOR SESAME, TOMADO DEL MANUAL DE SESAME.

Repositorios e inferencias

Un concepto central en el marco de trabajo de Sesame es el repositorio (repository). Un repositorio es un contenedor de almacenamiento para RDF. Este puede simplemente ser un objeto Java (o grupo de objetos Java) en memoria, o puede ser una base de datos relacional. Se escoge la forma de almacenar, sin embargo, es importante realizar siempre esta operación, en Sesame sucede con respecto a un repositorio: cuando usted adiciona datos RDF, usted adiciona estos a un repositorio.

Sesame, como se ha mencionado, soporta inferencias en RDF Schema. Esto permite que dado un grupo de RDF y/o RDF Schema, Sesame puede hallar la información implícita en el dato. Sesame soporta esto por la simple adición de toda la información implícita al repositorio también cuando el dato es inicialmente almacenado.

Es importante destacar que inferir en Sesame está asociado con el tipo de repositorio que usted usa. Sesame soporta diversos tipos de repositorio (En el manual del Usuario de Sesame se encuentra una sección dedicada a la configuración avanzada de éstos). Algunos de éstos soportan inferencia, otros no. Cuando usted quiere hacer inferencias en Sesame depende en su gran mayoría de su aplicación.

Generalidades de la arquitectura de Sesame

En la Figura 2.11, se presenta de manera general la arquitectura completa de Sesame.

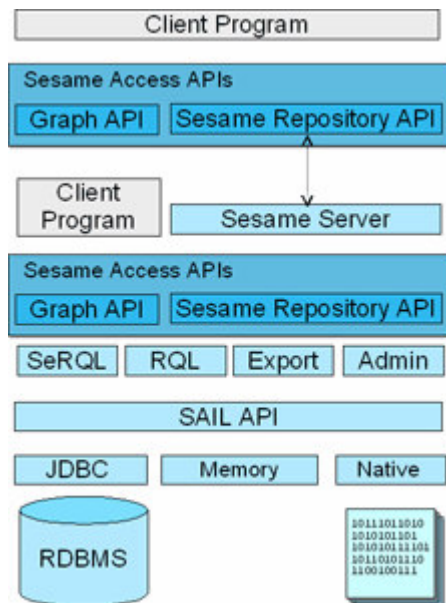


FIGURA 2.11. LA ARQUITECTURA DE SESAME. TOMADO DEL MANUAL DE SESAME.

Comenzando por abajo, el Nivel de Almacenamiento e Inferencia (the Storage And Inference Layer), o SAIL API, es la API Sesame interna que abstrae del formato usado (ejemplo donde el dato es almacenado en un RDBMS, en memoria o en un archivo), y provee soporte de razonamiento. La implementación SAIL puede también ser apilada una sobre otra, para proporcionar funcionalidad tal como manejo de captura o acceso concurrente. Cada repositorio de Sesame tiene su propio objeto SAIL que lo representa.

En la parte superior del SAIL, nosotros hayamos los módulos de funcionalidad de Sesame, tales como los motores de consultas SeRQL, RQL y RDQL, el módulo de administración y de exportar. El acceso a estos módulos funcionales están disponibles a través de las APIs de Acceso de Sesame, consistentes en dos partes separadas: la API del Repositorio y la API de Gráfica. La API del Repositorio proporciona el acceso de alto-nivel a los repositorios de Sesame, tales como consulta, almacenamiento de archivos rdf, extracción de RDF, etc. La API Gráfica provee más soporte de grano-fino para la manipulación de RDF, tal como adición y remoción individual de frases, y creación directa del código de pequeños modelos RDF. Las dos APIs se complementan en funcionalidad y a menudo son usadas juntas en la práctica.

Las APIs de Acceso (Access APIs) proveen acceso directo a los módulos de funcionalidad de Sesame, a cualquiera de los programas de cliente (por ejemplo, una aplicación de escritorio que use a Sesame como una librería), o de los componentes próximos de la arquitectura de Sesame, el servidor Sesame. Este es un componente que proporciona acceso basado en HTTP a las APIs de Sesame. Entonces, en el lado de un cliente remoto http, nosotros le proporcionamos el acceso a las APIs, que son usadas para la comunicación con Sesame, éste no es como una librería, pero si como un servidor corriendo en una ubicación remota.

Mientras cada parte del código de Sesame está públicamente disponible y extensible, la mayoría de los desarrolladores serían los principales interesados en las API de Acceso, para comunicarse con un modelo RDF de Sesame o un repositorio de Sesame desde sus aplicaciones. En el Manual del Usuario de Sesame se presentan detalles sobre las APIs y códigos de ejemplos.

2.5.3. OntoStudio

OntoStudio [69] lo califican sus desarrolladores como el medio ambiente para ontologías y para el desarrollo de aplicaciones semánticas. Es el sucesor de OntoEdit que fue distribuido alrededor del mundo unas 5.000 veces. OntoStudio está basado en el desarrollo de medioambiente Eclipse de IBM, que es recordado como estándar, hablando de herramientas de edición.

Los esquemas de base de datos pueden ser importados para ser integrados semánticamente a Oracle, MS SQL y DB2.

OntoStudio soporta no solamente OWL y RDF, sino también F-Logic y OXML. OWL fue diseñado por W3C para la descripción de ontologías; F-Logic y OXML son optimizaciones para los procesos lógicos basados en ontologías.

2.5.4. R₂O + ODEMapster

R₂O+ODEMapster [70] es la herramienta para la actualización de datos legalmente relacionados a la Web Semántica. Está basado en la descripción declarativa de mapeo entre relaciones y elementos ontológicos. Dispone de la exploración de mapeo para un procesador de dominio independiente.

2.5.5. RDF en el Contexto de los Negocios

Es amplio el uso del RDF en un contexto de negocios [71] en el que se proponen un marco de trabajo compuesto por: 1) Un modelo para describir tres áreas diferentes de recursos: negocios, información y comunicación, e integración de las áreas de descripción de recursos por medio de clases genéricas, obligaciones y relaciones. 2) Una extensión al modelo de descripción e integración de nodos y relaciones en una red organizacional. 3) Una extensión para el modelo de descripción y organización de la comunicación de información en los negocios. Este marco de trabajo sugiere un uso extenso del RDF en el contexto de los negocios.

En éste se concluye que esta visión puede ser realizada, pero requiere una aproximación consistente en cómo agrupar el contexto. Adicionalmente, la implementación de un marco de trabajo como una aproximación, debe tomar en cuenta los cambios constantes del contexto: re-estructuración de

información, re-organización de comunicaciones y re-posicionamiento de nodos en la red de los negocios.

2.5.6. Jena

Jena [72] es una herramienta de uso libre, desarrollada por los laboratorios de Web Semántica de HP (<http://www.hpl.hp.com/semweb/>). Su fin es manipular metadatos desde aplicaciones escritas en Java. En su versión 2, Jena incluye un analizador sintáctico de RDF, una API para RDF, una API de ontologías que permite trabajar con OWL, SPARQL, DALM y RDFS, un subsistema de razonamiento y un sistema de persistencia. Por si fuera poco, permite trabajar con el lenguaje de consultas de RDF, el RDQL.

2.6. APLICACIONES

Ha sido tal la respuesta del público a la creciente necesidad de integración de datos, que en la actualidad son diversas las aplicaciones sobre ontologías y Web Semántica por lo que a continuación se presentan algunas representativas.

2.6.1. UNSPSC

The United Nations Standard Products and Services Code (UNSPSC) [73] proporciona una ontología abierta, estándar global, multi-sector en busca de la eficiencia, clasificación exacta de productos y servicios. La UNSPSC ofrece un sistema único de clasificación global que puede ser usado para:

- Ampliación de la visibilidad de la compañía a partir del análisis de gastos.
- Optimización de la obtención de costos-efectivos.
- Exploración completa de capacidades electrónicas de comercio.

2.6.2. Open Biomedical Ontologies - OBO

Aunque en biología se han presentado numerosas iniciativas, podemos destacar el desarrollo de la ontología Open Biomedical Ontologies [74] en diferentes dominios de la biología y la medicina, ver Figura 2.12.

Domain	Prefix	Files	Format	Foundry	OBO CVS
Animal natural history and life history	ADW	protege source	Protege	no	no
Arabidopsis development	TAIR	arabidopsis development.obo	OBO	no	yes
Arabidopsis gross anatomy	TAIR	po anatomy.obo	OBO	no	yes
Biological imaging methods	FBbi	image.obo	OBO	no	yes
Biological process	GO	gene ontology.obo	OBO	yes	yes
BRENDA tissue / enzyme source	BTO	BrendaTissue.obo	OBO	no	yes
C. elegans development	WBls	worm development.obo	OBO	no	yes
C. elegans gross anatomy	WBbt	[none]	OBO	no	no
Cell type	CL	cell.obo	OBO	yes	yes
Cellular component	GO	gene ontology.obo	OBO	no	yes
Cereal plant development	GRO	cereals development.obo	OBO	no	yes
Cereal plant gross anatomy	GRO	po anatomy.obo	OBO	no	yes

FIGURA 2.12. MUESTRA DE LA LISTA DEL DESARROLLO DE LA ONTOLOGÍA OBO.

Así mismo, muchas otras disciplinas en las ciencias del ambiente están mirando la integración de datos en hidrología, climatología, ecología y oceanografía, como por ejemplo <http://marinemetadata.org/examples/mmihostedwork/ontologieswork>.

2.6.3. Ontología de los logs con RDF

Se ha reservado el último lugar al tema de la ontología de los logs con RDF, y no por eso menos importante. Al respecto, es muy escasa la literatura disponible; sin embargo, hay que destacar el trabajo del Distributed System Group – DSG presentado por Holliday et all. [58], en el que se contempla la ontología de los logs. El autor relaciona las propuestas de manejo de los logs más empleados y presenta su aporte, el estándar ontológico GULF, como formato común.

Trabajo éste que es el antecesor del presente documento, por lo que se ha reservado un espacio especial en el capítulo cuatro, ítem 4.1. “Propuesta de vocabulario y ontología Logging” en el que se presenta en detalle la propuesta, acompañada de un análisis, hasta llegar a los nuevos aportes.

2.6.4. Swoogle

Swoogle [75] es un motor de búsqueda para documentos de la Web Semántica, incluye ontologías OWL, construidas por la University of Maryland Baltimore County patrocinado por la Fundación de Ciencia Nacional (National Science Foundation). El sitio anuncia la disposición de 10.000 ontologías, por lo que se hace bastante atractivo. Al consultar y visualizar la metadata nos presenta su contenido con la metadata correspondiente a la ontología, en el caso de la Figura 2.13, es la de la ontología

<http://www.w3.org/2000/10/swap/log.n3>. Si la ontología consultada está disponible, se tiene acceso a ella.

Swoogle's Metadata

- ◆ [hasDateDiscovered](#): 2005-01-17
- ◆ [hasDatePing](#): 2006-12-14
- ◆ [hasPingState](#): PingModified
- ◆ [type](#): SemanticWebDocument
- ◆ [isEmbedded](#): false
- ◆ [hasGrammar](#): N3
- ◆ [hasParseState](#): ParseSuccess
- ◆ [hasDateLastmodified](#): 2006-11-20
- ◆ [hasDateCache](#): 2006-12-14
- ◆ [hasEncoding](#): ISO-8859-1
- ◆ [hasLength](#): 11K
- ◆ [hasCntTriple](#): 85.00
- ◆ [hasOntoRatio](#): 1.00
- ◆ [hasCntSwi](#): 36.00
- ◆ [hasCntSwiDef](#): 29.00
- ◆ [hasCntInstance](#): 0.00

FIGURA 2.13. VISUALIZACIÓN DE LA METADATA DE UNA ONTOLOGÍA CONSULTADA EN SWOOGLE.

REFERENCIAS BIBLIOGRÁFICAS

[1] Maruyama H, Tamura K, Uramoto N. Sitios Web con XML y Java. Madrid: Prentice Hall; 2000. p. 19.

[2] W3C World Wide Web Consortium. LASSILA O. Introduction to RDF Metadata [online]. [actualizado 1997 nov 13; citado 2006 nov 17]. [aprox. 6 pantallas]. Disponible en: <http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html>.

[3] Harmelen F y Fensel D. Practical Knowledge Representation for the Web. En: IJCAI'99 Workshop on Intelligent Information Integration; 1999 jul 31 – ago 2; Stockholm, Sweden; 1999.

[4] W3C World Wide Web Consortium. Resource Description Framework (RDF) Model and Syntax Specification. [online]. [actualizado 1999 feb; citado 2007 ene 29]. [aprox. 100 pantallas]. Disponible en: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>.

[5] W3 World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Fourth Edition) [online]. [actualizado 2006 ago 16; citado 2007 ene 29]. [aprox. 120 pantallas]. Disponible en: <http://www.w3.org/TR/REC-xml>.

[6] W3C World Wide Web Consortium. Beckett D. RDF/XML Syntax Specification (Revised) [online]. [actualizado 2004 feb 10; citado 2006 dic 18]. [aprox. 36 pantallas]. Disponible en: <http://www.w3.org/TR/rdf-syntax-grammar/>.

[7] W3C World Wide Web Consortium. Brickley D, Guha R. RDF Vocabulary Description Language 1.0: RDF Schema [online]. [actualizado 2004 feb 10; citado 2006 dic 18]. [aprox. 36 pantallas]. Disponible en: <http://www.w3.org/TR/rdf-schema/>.

[8] W3C World Wide Web Consortium. Manola F, Miller E. RDF Primer [online]. [actualizado 2004 feb 10; citado 2006 dic 18]. [aprox. 36 pantallas]. Disponible en: <http://www.w3.org/TR/rdf-primer/>.

[9] W3C World Wide Web Consortium. Klyne G, Carroll J. Resource Description Framework (RDF): Concepts and Abstract Syntax [online].

[actualizado 2004 feb 10; citado 2006 dic 18]. [aprox. 36 pantallas]. Disponible en: <http://www.w3.org/TR/rdf-concepts/>.

[10] W3C World Wide Web Consortium. Hayes P. RDF Semantics [online]. [actualizado 2004 feb 10; citado 2006 dic 18]. [aprox. 36 pantallas]. Disponible en: <http://www.w3.org/TR/rdf-mt/>.

[11] W3C World Wide Web Consortium. Grant J, Beckett D. RDF Test Cases [online]. [actualizado 2004 feb 10; citado 2006 dic 18]. [aprox. 36 pantallas]. Disponible en: <http://www.w3.org/TR/rdf-testcases/>.

[12] W3C World Wide We Consortium. Aida B, Hausenblas M. RDFa Use Cases: Scenarios for Embedding RDF in HTML [online]. [actualizado 2007 mar 30; citado 2007 abr 2]. [aprox. 6 pantallas]. Disponible en: <http://www.w3.org/TR/2007/WD-xhtml-rdfa-scenarios-20070330/>.

[13] W3C World Wide Web Consortium. Berners T. W3C Data Formats [online]. [actualizado 1997 oct 29; citado 2006 dic 18]. [aprox. 15 pantallas]. Disponible en: <http://www.w3.org/TR/NOTE-rdfarch>.

[14] Miller E. An Introduction to the Resource Description Framework. D-Lib Magazine May 1998 [serial online]. 1998 may [citado 2006 dic 16]; 4(1) [aprox. 25 pantallas]. Disponible en: <http://www.dlib.org/dlib/may98/miller/05miller.html>.

[15] W3C World Wide Web Consortium. Platform for Internet Content Selection (PICS) [online]. [actualizado 2005 jul 8; citado 2006 dic 18]. [aprox. 40 pantallas]. Disponible en: <http://www.w3.org/PICS/>.

[16] W3C World Wide Web Consortium. Lassila O. PICS-NG Metadata Model and Label Syntax. [online]. [actualizado 1997 may 14; citado 2007 ene 29]. [aprox. 80 pantallas]. Disponible en: <http://www.w3.org/TR/NOTE-pics-ng-metadata>.

[17] W3C World Wide Web Consortium. Reagle J y Weitzner D. Statement on the intent and use of PICS : using PICS wel [online]. [actualizado 1998 jun 1; citado 2006 dic 19]. [aprox. 26 pantallas]. Disponible en: <http://www.w3.org/TR/NOTE-PICS-Statement>.

[18] Dublin Core Metadata Initiative [Online] 1995 [citada 2006 dic 19]; [aprox. 4 pantallas]. Disponible en: <http://dublincore.org/>.

[19] Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1: Reference Description [online]. [actualizado 1999 jul 2; citado

2006 dic 19]. [aprox. 28 pantallas]. Disponible en: <http://purl.oclc.org/dc/documents/rec-dces-19990702.htm>.

[20] Bridging the Gap between RDF and XML [online]. [actualizado 1999 dic 16; citado 2006 dic 10]. [aprox. 6 pantallas]. Disponible en: <http://www-db.stanford.edu/~melnik/rdf/fusion.html>.

[21] Enrech M, Maniega D, Serrano J, Comellas N. La gestión del documento electrónico: L' experiència de la Biblioteca de la UOC. &ES. En: Jornades Catalanes de Documentación. Barcelona: Socadi, Cobdc; 1997. p. 115-129 y 158.

[22] Morrison M, et al. XML Al descubierto. Madrid: Prentice-Hall; 2000. p. 268.

[23] W3C World Wide Web Consortium. Platform for Privacy Preferences (P3P) Project [online]. [actualizado 2006 dic 10; citado 2007 ene 10]. [aprox. 16 pantallas]. Disponible en: www.w3.org/P3P/.

[24] Truste [Online]. 2006 ene 5 [citado 2006 dic 29]; [aprox. 3 pantallas]. Disponible en: <http://www.truste.org>.

[25] Layman A. Application of RDF for Extensible Dublin Core Metadata. Resource Discovery Unit 1997 jul 30.

[26] UKOLN. Miller E, Miller P, Bickley D. Guidance on expressing the Dublin Core within the Resource Description Framework (RDF) [online]. [actualizado 1999 jul 1; citado 2006 nov 12]. [aprox. 96 pantallas]. Disponible en: <http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/>.

[27] W3C World Wide Web Consortium. Decker S, Brickley D, Saarela J, Angele J. A query and inference service for RDF. En: QL'98 – The Query Languages Workshop. W3C; 1998 nov 18 [online]. [citado 2006 dic 19]. [aprox. 32 pantallas]. Disponible en: <http://www.w3.org/TandS/QL/QL98/pp/queryservice.html>.

[28] Mozilla.org. RDF: Aurora [online]. [actualizado 1999 mar 18; citado 2006 dic 10]. [aprox. 4 pantallas]. Disponible en: <http://www.mozilla.org/rdf/doc/aurora.html>.

[29] Mozilla.org. RDF. Web services and Mozilla [online]. [actualizado 2002 mar 4; citado 2006 dic 10]. [aprox. 5 pantallas]. Disponible en: <http://www.mozilla.org/rdf/doc/SmartBrowsing.html>.

[30] Mozilla.org. rdf: back-end architecture [online]. [actualizado 2002 dic 16; citado 2006 dic 10]. [aprox. 16 pantallas]. Disponible en: <http://www.mozilla.org/rdf/back-end-architecture.html>.

[31] RFC.net. Dawson F, Howes T. vCard MIME Directory Profile. En: Internet RFC 2426 [online]. [actualizado 1998 sep; citado 2006 dic 10]. [aprox. 128 pantallas]. Disponible en: <http://rfc.net/rfc2426.html>.

[32] Representing UML in RDF [online]. [actualizado 2000 nov; citado 2006 dic 10]. [aprox. 6 pantallas]. Disponible en: <http://www-db.stanford.edu/~melnik/rdf/uml/>.

[33] W3C World Wide Web Consortium. OWL Implementations [online]. [actualizado 2004 dic 6; citado 2006 dic 10]. [aprox. 6 pantallas]. Disponible en: <http://www.w3.org/2001/sw/WebOnt/impls>.

[34] W3C World Wide Web Consortium. Abián M. Web Semántica Hoy [online]. [actualizado 2004 jun 6; citado 2006 dic 10]. [aprox. 14 pantallas]. Disponible en: <http://www.wshoy.sidar.org/index.php>.

[35] Corcho O, Gómez-Pérez A. A Roadmap to Ontology Specification languages. Lecture Notes In Computer Science; Vol. 1937. Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management; 2000 oct 2 - 6; Madrid. Spain. London, UK: Springer-Verlag. 2000. p. 80-96.

[36] Farquhar A, Fikes R, Rice J. The Ontolingua Server: A Tool for Collaborative Ontology Construction. Proceedings of KAW96; Academic Press 1996 sep; 46(6):707-727.

[37] Chaudhri V, Farquhar A, Fikes R, Karp P, Rice J. The Generic Frame Protocol 2.0. Computer Based Learning Unit, University of Leeds [serial online]; 1997 jul 21 [citado 2006 feb 19] [aprox. 35 pantallas]. Disponible en: <http://www.ai.sri.com/~gfp/gfp2/gfp2.html>.

[38] Motta E. Reusable Components for Knowledge Modelling. En: Frontiers in Artificial Intelligence. Amsterdam: IOS Press; 1999. vol 53 pp. 255.

[39] Kifer M, Lausen G, Wu J. Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the Association for Computing Machinery. New York: Stony Brook 1995 jul; 42(4):741-843.

[40] MacGregor R. Inside the LOOM clasifier. SIGART bulletin 1991 jun; 2(3):70-76.

[41] W3C World Wide Web Consortium. Bray T, Paoli J, Sperberg C. Extensible Markup Language (XML) 1.0. (Fourth Edition) [online]. [actualizado 2006 sep 29; citado 2006 nov 24]. [aprox. 320 pantallas]. Disponible en: <http://www.w3.org/TR/REC-xml>.

[42] W3C World Wide Web Consortium. Lassila O, Swick R. Resource Description Framework (RDF) Model and Syntax Specification [online]. [actualizado 1999 ene 5; citado 2006 dic 10]. [aprox. 120 pantallas]. Disponible en: <http://www.w3.org/TR/PR-rdf-syntax>.

[43] W3C World Wide Web Consortium. Brickley, D., Guha, R.V. Resource Description Framework (RDF) Schema Specification [online]. [actualizado 1999 ene 5; citado 2006 feb 10]. [aprox. 4 pantallas]. Disponible en: <http://www.w3.org/1999/.status/PR-rdf-syntax-19990105/status>.

[44] Karp P, Chaudhri V, Thomere J. XOL: An XML-Based Ontology Exchange Language. 1999. Disponible en: <http://www.ai.sri.com/pkarp/xol/xol.html>.

[45] Luke S, Heflin J. SHOE 1.01. Proposed Specification. SHOE Project [online]. [actualizado 2000 feb; citado 2006 feb 10]. [aprox. 15 pantallas]. Disponible en: <http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm>.

[46] Horrocks I, Fensel D, Harmelen F, Decker S, Erdmann M, Klein M. OIL in a Nutshell. Proceedings of the ECAI'00 Workshop on Application of Ontologies and PSMs. Berlin. Germany. August, 2000.

[47] Open Knowledge Base Connectivity, Specification [online]. [actualizado 1999 ene 5; citado 2006 feb 10]. [aprox. 4 pantallas]. Disponible en: <http://www.ai.sri.com/%7Eokbc/>.

[48] Berners-Lee T. Hearing on the "Digital Future of the United States: Part I - The Future of the World Wide Web". 2007 mar 01 [citado 2007 mar 30] [aprox. 9 páginas] Disponible en: <http://dig.csail.mit.edu/2007/03/01-ushouse-future-of-the-web>.

[49] Berners-Lee T, Fischetti M, Dertouzos M. Weaving the Web. Harper San Francisco; 1999.

[50] Berners-Lee T, Hendler J, Lassila O. The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. En Scientific American Magazine [serial online] 2001 may [citada 2007 mar 18]; 1(5):[aprox. 35 pantallas]. Disponible en: <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>.

[51] Coopertative Ontologies Programme. [Online]. 2006 ago 23 [citado 2007 feb 10]; Disponible en: <http://www.co-ode.org/>.

[52] EIC Enterprise Interoperability Centre - An Introduction to Semantics [Online]. 2005 jun [citado 2006 feb 10]; Disponible en: <http://elearning.eic-community.org/ONT1/>.

[53] Heraldo de Aragón. Navarro R. [online]. [actualizado 2006 mar 7; citado 2007 ene 12]. [aprox. 6 pantallas]. Disponible en: http://parla.dia.fi.upm.es/oeg/online%20documentation/press%20releases/HA_P007MIL_20060307CO.pdf.

[54] Berners-Lee T, Hall W, Hendler J, Shadbolt N, Weitzner D. Creating a Science of the Web. Science; 313(8); 2006 ago 11. Disponible en: <http://www.webscience.org/>.

[55] The Internet Engineering Task Force - IETF. Fielding R, et all. Hypertext Transfer Protocol -- HTTP/1.1 [online]. [actualizado junio de 1999; citado 2007 mar 15]. [aprox. 240 pantallas]. Disponible en: <http://www.ietf.org/rfc/rfc2616.txt>.

[56] Netcraft Secure Server Survey [Online] 2007 abr 4 [citado 2007 abr 10]; [aprox. pantallas]. Disponible en: http://news.netcraft.com/archives/web_server_survey.html.

[57] Comer D. Redes globales de información con Internet y TCP/IP. Prentice-Hall Hispanoamericana S. A., México; 1996. p. 126.

[58] Holliday M, Baker M y Boakes R. Grids, Logs, and the Resource Description Framework [online] [actualizado 2004 nov; citado 2007 feb 15]. [aprox. 8 páginas]. Disponible en: <http://dsg.port.ac.uk/projects/slogger/docs/archive/Grids,%20Logs%20and%20the%20Resource%20Description%20Framework%20-%20Holliday,%20Baker,%20Boakes.pdf>.

[59] W3C World Wide Web Consortium. Logging Control In W3C httpd [online] [actualizado 1995 jul; citado 2007 mar 15]. [aprox. 6 pantallas]. Disponible en: <http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format>.

[60] W3C World Wide Web Consortium. A. Luotonen. The Common Logfile Format [online]. [actualizado 1995; citado 2007 mar 15]. [aprox. 7 pantallas]. Disponible en: <http://www.w3.org/pub/WWW/Daemon/User/Config/Logging.html>.

- [61] W3C World Wide Web Consortium. Extended Log File Format [online] [actualizado 1996 mar 23; citado 2007 mar 15]. [aprox. 15 pantallas]. Disponible en: <http://www.w3.org/TR/WD-logfile.html>.
- [62] W3C World Wide Web Consortium. Shop Log File Format [online]. [actualizado 2000 Oct 26; citado 2007 mar 15]. [aprox. 18 pantallas]. Disponible en: <http://www.w3.org/TR/shoplogfileformat/>.
- [63] W3C World Wide Web Consortium. Log analysis tools [online]. [actualizado 1999; citado 2007 mar 15]. [aprox. 4 pantallas]. Disponible en: <http://www.w3.org/WCA/loganalysis-tools.html>.
- [64] UPPSALA Universitet [Online] 2007 feb 5 [citado 2007 mar 20]: [aprox. 16 pantallas] Disponible en: <http://www.uu.se/Software/Analyzers/Access-analyzers.html>.
- [65] Protégé [Online]. 2007 ene 5 [citado 2007 ene 20]; Disponible en: <http://protege.stanford.edu>.
- [66] Noy N, McGuinness D. Ontology Development 101: A Guide to Creating Your First Ontology [online]. [actualizado 2006 may 9; citado 2007 feb 10]. [aprox. 200 pantallas]. Disponible en: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.
- [67] OpenRdf.org. [Online]. 2007 ene 9 [citado 2007 feb 15]; Disponible en: www.openrdf.org.
- [68] Aduna B. V. User Guide for Sesame [online]. [actualizado 2006 abr 22; citado 2007 feb 10]. [aprox. 200 pantallas]. Disponible en: <http://www.openrdf.org/doc/sesame/users/index.html>.
- [69] Ontoprise. Ontostudio [Online]. 2007 ene 8 [citado 2006 dic 10]. [aprox. 3 pantallas]. Disponible en: <http://free.ontostudio.de/>.
- [70] Ontology Engineering Group [online]. [actualizado 2007 ene 16; citado 2007 feb 10]. [aprox. 5 pantallas]. Disponible en: <http://www.oeg-upm.net>.
- [71] Forsberg K, Dannstedt L. Extensible use of RDF in a Business Context [online]. [actualizado 2000 may 19; citado 2007 feb 10]. [aprox. 60 pantallas]. Disponible en: <http://www9.org/w9cdrom/323/323.html>.
- [72] Jena [Online]. 2007 ene 8 [citada 2007 ene 12]; Disponible en: <http://jena.sourceforge.net/>.

[73] UNSPSC [Online]. 2007 [citada en 2007 mar 30]; disponible en: www.unspsc.org.

[74] Open Biomedical Ontologies [Online]. 1999 ago 16 [citada 2007 mar 12]; Disponible en: <http://obo.sourceforge.net/>.

[75] Swoogle. [Online] 2007 mar 26 [citado 2007 abr 3]; [aprox. 12 pantallas]. Disponible en: <http://swoogle.umbc.edu/>.

3. METODO DE INVESTIGACIÓN

La metodología a emplear en la elaboración del presente proyecto de investigación es aplicada y de tipo descriptiva, teniendo en cuenta los objetivos planteados y los resultados que se pretenden alcanzar, como son, modelar una realidad confrontando la teoría con ésta, efectuando un modelo para el tratamiento de los logs.

Se recolectará información y datos tanto de fuentes primarias como secundarias, tal como se describe a continuación:

Se utilizará como fuente primaria la recolección de información por medio de la selección de períodos de registros logs del servidor Web de la Universidad Autónoma de Bucaramanga - UNAB que tiene por nombre unab.edu.co e IP 200.69.124.43. El servidor Web es Apache y la Universidad lo destina al manejo de sus sitios Web principales y se encarga de atender las peticiones a: página principal de la Universidad (www.unab.edu.co), página de egresados UNAB, revistas académicas y radio UNAB.

Los archivos logs empleados son tanto de acceso como de error y se ubican en el servidor en los directorios \$HOME_Apache/Apache/logs/. De éstos la

administradora de los Sistemas de Información de UNAB, Ing. Karol Dalila Reyes Díaz, seleccionó una muestra aleatoria de 110 archivos correspondientes al período entre el 6 de febrero y el 5 de marzo de 2007. Los archivos llevan por nombre `access_log.117xxxxxxx` y `error_log.xxxxxxx` (en que las x corresponden al rango 0763200 a 3096000 con correspondencia en ambos grupos de archivos) y cada archivo contiene registros logs de 12 horas que inician las primeras a las 07:00 hasta las 18:59:59 y las segundas 12 horas desde las 19:00 hasta las 06:59:59 del día siguiente. Los archivos seleccionados han dispuesto en el CD que acompaña la tesis en el directorio `/log/access_log` y `/log/error_log` respectivamente.

La fuente secundaria estará conformada por la bibliografía que se seleccionará sobre cada uno de los tópicos cubiertos por el proyecto, la revisión bibliográfica, haciendo uso de los planteamientos realizados por los autores en RDF, Web Semántica, ontología, repositorios, diseños de sistemas de información, entre otros.

Para la realización de la propuesta producto de la investigación se parte de la investigación previa de Holliday et all. [1] en que se hace un análisis de los logs más comúnmente usados y el planteamiento de una propuesta de un vocabulario y ontología unificada. Para la definición del vocabulario y la ontología se sigue el planteamiento de Noy y McGuinness [2], en el que se

presenta un método para el desarrollo de ontologías compuesto por siete pasos. Cabe destacar, que los siete pasos no incluyen una prueba a la ontología, porque los autores se han referido solamente al desarrollo de ontología y destacan que los investigadores enfatizan también en el análisis de ontologías, involucrando herramientas de diagnóstico como lo presenta Chimaera [3].

En la implementación y prueba del vocabulario y la ontología finalmente se emplearán dos herramientas libres, una primera para la definición, Protégé [4], y una segunda para el almacenamiento en repositorio, manejo y prueba para lo cual se emplea Sesame [5]. Específicamente, la prueba se efectuará haciendo uso del lenguaje de consulta SeRQL de Sesame, en ambos casos se acogen los métodos planteados en los manuales de usuario respectivos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Holliday M, Baker M, Boakes R. Grids, Logs, and the Resource Description Framework [online] [actualizado 2004 nov; citado 2007 feb 15]. [aprox. 8 páginas]. Disponible en: <http://dsg.port.ac.uk/projects/slogger/docs/archive/Grids,%20Logs%20and%20the%20Resource%20Description%20Framework%20-%20Holliday,%20Baker,%20Boakes.pdf>.
- [2] Noy N y McGuinness D. Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología. Stanfor University, Stanford, CA, 94305. 2005 sep 19; pp. 29.
- [3] Chimaera. Chimaera Ontology Environment [online] [actualizado 2003 sep 17; citado 2007 mayo 30]. [aprox. 4 páginas] 2000. Disponible en: <http://www.ksl.stanford.edu/software/chimaera/>.
- [4] Protégé [online]. [actualizado 2007 mar; citado 2007 mar 15]. [aprox. 3 pantallas]. Disponible en: <http://protege.stanford.edu/>.
- [5] Broekstra J, Kampman A y van Harmelen F. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Lecture Notes In Computer Science 2002; 2342:54-68. Disponible en: <http://www.cs.vu.nl/~frankh/postscript/ISWC02.pdf>.

4. RESULTADOS DE LA INVESTIGACIÓN

El capítulo presenta los resultados obtenidos durante la investigación, la propuesta de vocabulario y ontología Logging en la cual se emplea una ontología previa, la GULF; el análisis y aplicación de los logs, como a continuación se describe.

4.1. PROPUESTA DE VOCABULARIO Y ONTOLOGÍA LOGGING

Aplicando el criterio de reusabilidad de ontologías existentes, se parte en el presente trabajo de la propuesta previa de Holliday et. all. [1] en que se definió un vocabulario usando RDF-S para describir los elementos del Formato Unificado de Log Genérico (GULF, del inglés Generic Unified Log Format). El vocabulario está definido en el archivo del esquema cuya dirección es <http://dsg.port.ac.uk/schemas/log> - llamado el QName gulf.

Se consideran tres tipos de orígenes de logs: registros Log4J (gulf:l4j-), NetLogger (gulf:nl-) y de descripción de eventos (Las descripciones “Top 'N' Event”) especificadas por el GGF DAMED-WG (gulf:damed-).

El vocabulario Gulf está compuesto de una definición de clase y varias definiciones de propiedades. La clase describe un único registro log en

formato RDF. Cada propiedad representa un posible campo de un registro log.

Una vez que se define la clase, gulf:Record, usando la siguiente tripleta:

```
<gulf:Record> <rdf:type> <rdfs:Class>.
```

Una clase en RDF Schema es un recurso, en este caso llamada gulf:Record, que tiene una propiedad rdf:type cuyo valor es rdfs:Class. De esta manera, en esta tripleta se está definiendo la clase gulf:Record.

Se especifican las propiedades de la clase gulf:Record. Puesto que esta clase representa un solo registro en un archivo log; un registro llegaría a tener cualquiera de los tipos originales de logs, usamos todos los campos de los logs originales como propiedades de la clase gulf:Record. Se considera, en turno, propiedades halladas de cada una de las tres fuentes log: Log4J, NetLogger y DAMED-WG.

En este punto es necesario hacer un paréntesis, aunque el interés no es profundizar en los formatos de los Logs, en el caso del Log4J [2], amerita hacer una aclaración adicional dado su carácter de universalidad o generalidad en su aplicación y que no fue presentada con claridad por Holliday A. et. all., por lo que nos remitimos al manual [3]. En este el formato se ajusta a lo que el usuario desea, el PatternLayout [4] de, que es parte del

estándar de distribución Log4j, le permite al usuario especificar el formato de la salida de acuerdo a los patrones de conversión similares a los de la función printf del lenguaje C.

Por ejemplo, el PatternLayout con el patrón de conversión “%r [%t] %-5p %c - %m%n” presentaría una salida de algo semejante a:

```
176 [main] INFO org.foo.Bar - Located nearest gas station.
```

Donde, el primer campo es el número del lapso desde el inicio del programa en milisegundos. El segundo campo es el hilo hacedor (thread making) del requerimiento del log. El tercer campo es el nivel del log. El cuarto campo es el nombre del logger asociado con el requerimiento log. El texto después del – es la frase del mensaje.

Log4j crearía el contenido del mensaje log de acuerdo al criterio especificado por el usuario. Por ejemplo, si usted frecuentemente necesita al log Naranjas, un tipo de objeto usado en su proyecto, entonces usted puede registrar un CreadorNaranja que sería invocado cada vez que se necesite registrar en el log (logged) una naranja.

Creadores de objetos son implementados por la interface CreadorObjeto (ObjectRenderer) [5].

Terminado el paréntesis de aclaración del formato Log4j, se retoma la presentación de la propuesta del formato gulf. Para soportar los logs generados por Log4J la clase gulf:Record debe representar cada carácter de conversión en la clase PatternLayout Log4J. Por ejemplo, la conversión del carácter “p” causa el despliegue de la prioridad del evento logging. La propiedad correspondiente es definida por la tripleta:

```
<gulf:l4j-eventPriority> <rdf:type> <rdf:Property> .
```

Esta tripleta ilustra como se define una propiedad en el RDF Schema. Una propiedad en RDF-S es cualquiera que tenga una propiedad rdf:type cuyo valor es rdf:Property. Esto es, en esta tripleta se define una propiedad llamada gulf:l4j-eventPriority. Nótese que en ésta, el propietario de la tripleta sólo define que hay una propiedad con un nombre; nada es indicado acerca del valor posible que ésta puede tener, o la clase a la cual está relacionada.

No se halló una lista de valores estándares para la parte llamada “key” en un campo NetLogger. Esto puede dejarlo de lado la persona para decidir crear un log particular. Es claro que HOST no es parte de las propiedades Log4J. Esto es porque NetLogger fue diseñado para aplicaciones distribuidas donde los logs integrados pueden contener entradas de diferentes anfitriones (hosts). De esta manera se adiciona la propiedad gulf:nl-host al vocabulario gulf.

La especificación DAMED-WG de descripción de evento, los “Top ‘N’ Events”, tienen cuatro propiedades por evento (ej. del registro log): marca de tiempo, nombre del evento, resultado y valor. Nosotros adicionamos una tripleta por cada una de estas propiedades como se muestra a continuación:

```
<gulf:damed-timestamp> <rdf:type> <rdf:Property> .
<gulf:damed-eventName> <rdf:type> <rdf:Property> .
<gulf:damed-target> <rdf:type> <rdf:Property> .
<gulf:damed-value> <rdf:type> <rdf:Property> .
```

El próximo paso es enseñar como un modelo RDF contiene frases log que pueden ser creadas. El programa es un sólo registro log que usa nuestra ontología. El programa tiene una serie de frases que especifican las tripletas RDF. Para cada registro log hay una tripleta RDF para especificar el registro mismo del log y entonces una tripleta RDF para especificar el valor de cada propiedad que es un campo del registro log en el log original.

4.1.1. Definición de vocabulario y ontología

Terminada la presentación de la propuesta Gulf, pasemos a presentar la nueva propuesta, convirtiéndose ésta en otro de los aportes del presente trabajo.

En la definición del vocabulario y ontología se siguió la metodología planteada por Noy y McGuinness [6]:

Paso 1: Determinar el dominio y alcance de la ontología. Para lo que se siguió con las preguntas propuestas ¿Cuál es el dominio que la ontología cubrirá? Representación del comportamiento de los servidores Web, vistas desde sus logs.

¿Para qué usaremos la ontología? Realizar un análisis al comportamiento de los usuarios de un servidor Web.

¿Para qué tipos de preguntas la información en la ontología deberá proveer respuestas?

- a. Estadísticas de atención acertada a los clientes.
- b. Se siente satisfecho un cliente de la atención del servidor Web?
- c. Disponer las estadísticas de solicitudes no satisfechas.
- d. Cuál es el origen de las solicitudes no satisfechas (bases de datos, enlaces, transmisión de archivos, ...).
- e. Cuál es la solicitud típica (evento habitual).
- f. Cuál es el tiempo promedio entre solicitudes.
- g. Cuál es el tiempo requerido para la atención a un usuario.
- h. Cuál es el tiempo ocioso del servidor.

- i. Permitir personalizar las acciones de los usuarios y conocer su comportamiento habitual.

¿Quién usará y mantendrá la ontología? La ontología será usada por el nivel ejecutivo tecnológico de la empresa. La ontología será mantenida por los desarrolladores de software servidor Web y los Web Master.

Paso 2: Considerar la reutilización de ontologías existentes. Se ha tenido en cuenta la propuesta de Holliday et. all. [1].

Paso 3: Enumerar términos importantes para la ontología logs. Propuestas de logs en el mercado, satisfacción de clientes, solicitud típica de los clientes, tipos de solicitudes, aplicaciones que emplearon el servidor, tiempo de empleo de los servicios, fecha de solicitud, dirección IP del solicitante (host), tipos de logs, campos o componentes en los tipos de logs (siguiendo las propuestas de los logs más comunes).

Paso 4: Definir las clases y las jerarquías de clase.

Este nuevo planteamiento del vocabulario y ontología, tuvo como punto de partida la propuesta de A. Holliday et. all. [1], por lo que lo llamaremos Gulf2, ver Figura 4.1.

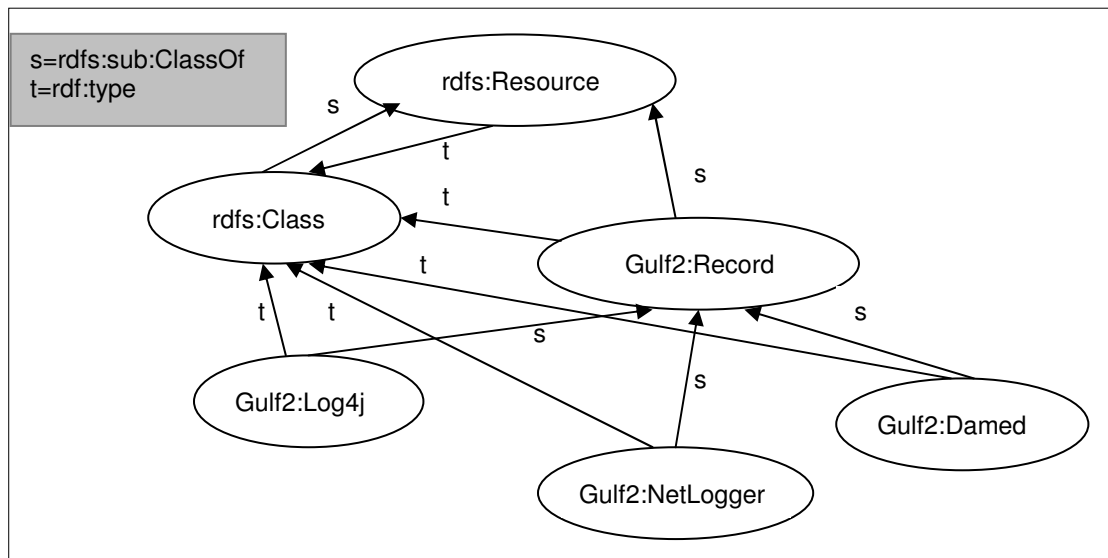


FIGURA 4.1. JERARQUÍA DE LA CLASE GULF2 EN RDF SCHEMA.

Gulf2 está compuesta por una única clase y el resultado es la definición de la clase, Gulf2:Record, que se hace usando la siguiente tripleta:

```
<Gulf2:Record>      <rdf:type>      <rdfs:Class>.
```

Las subclases de la clase Gulf2, que vienen a representar los tres formatos logs más comunes se definen por las tripletas siguientes:

```
<Gulf2:Log4j>      <rdf:type>      <rdfs:Class>.
```

```
<Gulf2:NetLogger> <rdf:type>      <rdfs:Class>.
```

```
<Gulf2:Damed>     <rdf:type>      <rdfs:Class>.
```

Ahora es necesario definir las clases como subclases, así:

```
<Gulf2:Log4j>      <rdfs:subClassOf> <Gulf2:Record>.
```

```
<Gulf2:NetLogger> <rdfs:subClassOf> <Gulf2:Record>.
```

```
<Gulf2:Damed>     <rdfs:subClassOf> <Gulf2:Record>.
```

Paso 5: Definir las propiedades de las clases: slots.

La propiedad correspondiente a la clase Log4j, subclase de Gulf2; es definida por las siguientes tripletas:

```
<Gulf2:Log4j-nHost> <rdf:type> <rdf:Property>.
<Gulf2:Log4j-nRegistro> <rdf:type> <rdf:Property>.
<Gulf2:Log4j-uRequerimiento> <rdf:type> <rdf:Property>.
<Gulf2:Llog4j-hora> <rdf:type> <rdf:Property>.
<Gulf2:Log4j-solicitud> <rdf:type> <rdf:Property>.
<Gulf2:Log4j-cRespuesta> <rdf:type> <rdf:Property>.
<Gulf2:Log4j-bTransferidos> <rdf:type> <rdf:Property>.
<Gulf2:Log4j-navegador> <rdf:type> <rdf:Property>.
```

Las propiedades de la clase Damed-WG, subclase de Gulf2, son definidas por las tripletas siguientes:

```
<Gulf2:Demed-timestamp> <rdf:type> <rdf:Property>.
<Gulf2:Demed-eventName> <rdf:type> <rdf:Property>.
<Gulf2:Demed-target> <rdf:type> <rdf:Property>.
<Gulf2:Demed-value> <rdf:type> <rdf:Property>.
```

Las propiedades de la clase NetLogger, subclase de Gulf2, son definidas por las tripletas siguientes:

```
<Gulf2:NetLogger-date> <rdf:type> <rdf:Property>.
<Gulf2:NetLogger-level> <rdf:type> <rdf:Property>.
```

```

<Gulf2:NetLogger-host>    <rdf:type> <rdf:Property>.
<Gulf2:NetLogger-target> <rdf:type> <rdf:Property>.
<Gulf2:NetLogger-event>  <rdf:type> <rdf:Property>.

```

Paso 6: Definir las facetas de los slots

Para terminar la aplicación de la metodología seleccionada, recordemos que es la de Noy y McGuinness [6]. Dada las características conocidas para cada uno de los tipos de logs, de nuestra jerarquía de subclases de la clase Gulf2, se definen sus campos, los que a su vez vienen a ser propiedades y restricciones (Properties and Restrictions, elemento de Protégé). A cada una de las propiedades, definidas como rdf:Property, se les define el tipo de dato, que para el caso se efectúa empleando el Rango (Range, elemento de Protégé) que puede ser cadena de caracteres (string), fecha y tiempo (date Time), entero (Int), etc. De igual manera se requiere la definición de la cardinalidad, la que dada la naturaleza de nuestros datos, un campo en cada registro, es igual a uno. Posteriormente se define la ontología en Protégé [7] la clase Gulf2 y como sub clases, se definen Log4j, NetLogger y Damed-WG respectivamente, como los tipos de logs más comúnmente empleados.

Cada uno de los elementos definidos ontología, clases, subclases, propiedades e instancias se van autodocumentando empleando rdfs:comment o rdfs:label. Documentación que es complementada con ésta,

que se viene haciendo, paso a paso con el fin de disponer de una ontología completamente documentada.

Para efectos de observar los detalles se remite al lector al Anexo I. Código RDF/XML generado con Protégé.

4.1.2. Diferencias de la ontología propuesta con la antecesora

Esta nueva propuesta tiene diferencias con respecto a su antecesora que hacen necesario destacarlas, como son:

- Se definió un vocabulario compuesto por una súper clase, la clase a la que se le ha asignado por nombre Gulf2:Record y tres subclases, conformadas estas últimas por los tipos más comunes de logs [8].
- La jerarquía o taxonomía planteada, Figura 4.1, permite un manejo más flexible y acorde con la definición del vocabulario y la ontología. Anteriormente, Holliday et. all. [1] había definido los tipos de logs como propiedades de una única clase, GULF, lo cual al crear una instancia de la clase para almacenar la información de un log particular, disponía de todas las propiedades independientes del formato log que se estuviese tratando. Se puede visualizar el alto costo en que se incurre. Costos representados en aumento en la complejidad de manejo, empleo

ineficiente de los espacios de almacenamiento, redundando al final en dificultad para el manejo de los logs.

- Si nos detenemos a mirar la sencillez de la definición de la clase GULF con sus propiedades de manera parcializada y centrada solo en la estructura, podríamos estar cometiendo grandes desaciertos, dado el alto número de logs generados, instancias, en la atención de un día típico, que en el caso del servidor de la UNAB aproximadamente son 203.638 solicitudes/día, es decir 8.485 solicitudes/hora. Mostrándonos el impacto en la relación eficiencia de la estructura y espacio ocupado por las instancias de la clase para el almacenamiento de logs en un día, una semana, un mes, un año o más. Figura 4.2.

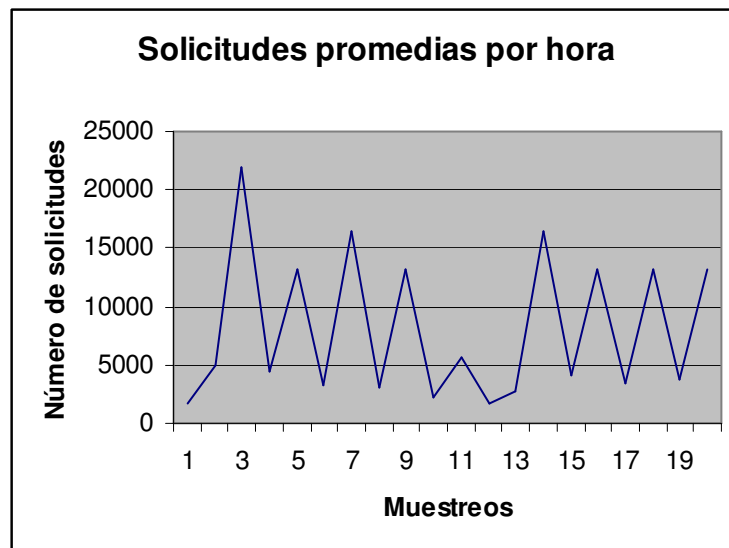


FIGURA 4.2. MUESTRA DE SOLICITUDES PROMEDIOS POR HORA AL SERVIDOR WEB UNAB.

4.1.3. Código RDF/XML

El código RDF/XML generado por Protégé 3.1 al vocabulario y ontología, para las instancias definidas y presentadas en la sección 4.1.1. “Definición de vocabulario y ontología” inmediatamente anterior se presenta en el Anexo I – Código RDF/XML generado por Protégé.

El código RDF/XML correspondiente al vocabulario y la ontología Gulf2 es autodocumentado con comentarios empleando rdfs:comment y rdfs:label, se emplea el espacio de nombres (Namespace) <http://www.unab.edu.co/2007/Gulf2#>.

En la obtención de una muestra del código RDF/XML presentado, se escogió la inclusión de instancias de los tipos de logs, NetLogger, Log4j y Damed-WG.

Para NetLogger, se definió la siguiente instancia o registro log, así:

TABLA 4.1. EJEMPLO DE UNA INSTANCIA LOG EN FORMATO NETLOGGER.

Date	Level	Host	Target	Event
marzo 19, 2007 12:12	Information	172.25.1.16	appended	program.end

Para DAMED-WG, se definieron las siguientes dos instancias o registros log, así:

TABLA 4.2. EJEMPLO DE INSTANCIAS LOGS EN FORMATO DAMED-WG.

Timestamp	EventName	Target	Value
Marzo 19, 2007 22:41:00	delay.rondtrip	23.5	129.42.17.99
Marzo 19, 2007 22:47:00	processor.load	12	129.42.17.99

Para Log4j, se definieron más instancias que en los casos anteriores, 17, por ser éste el formato empleado en el servidor Apache de la UNAB, www.unab.edu.co y con el fin de facilitar las pruebas. Para efectos de ilustración se presentan en la tabla 4.3, sólo cinco instancias de las declaradas.

En la tabla 4.3 se han contraído los nombres de las propiedades, las columnas, de la clase Log4j con el fin de ajustarla al ancho de la página, como a continuación se describen:

nHost. Nombre de host.

nReg, es la propiedad nRegistro, que indica el nombre de registro remoto.

uRem, es la propiedad uRemoto , que indica el Usuario remoto.

hora. Es la fecha y la hora en que se sirvió la solicitud.

solicitud: Es la primera línea de la solicitud que se hizo al servidor.

cRespuesta. Indica si la solicitud ha tenido éxito y, si no, el tipo de error que se ha producido.

bTransferidos. Es el número total de bytes que se transfieren al cliente.

navegador. Navegador empleado.

TABLA 4.3. EJEMPLO DE INSTANCIAS LOGS EN FORMATO LOG4J.

nHost	nReg	uRem	hora	solicitud	cResp	bTransf	navegador
200.69.124.66	-	-	Feb 6, 2007 190004	GET HTTP/1.1	304	0	Mozilla/4.0
200.69.124.66	-	-	Feb 6, 2007 190005	POST /pls/portal/ HTTP/1.1	200	47655	Mozilla/4.0
200.69.124.66	-	-	Feb 6, 2007 190006	POST /pls/portal/ HTTP/1.1	200	11853	Mozilla/4.0
200.69.124.66	-	-	Feb 6, 2007 190006	GET /portal HTTP/1.1	200	59066	Mozilla/4.0
200.69.124.66	-	-	Feb 6, 2007 190006	GET /pls/portal/ HTTP/1.1	200	39210	Mozilla/4.0

4.1.4. Almacenamiento de la ontología Log en Sesame

Partiendo de la presentación realizada en el capítulo dos sobre Sesame y con base en los planteamientos [9], pasaremos a su empleo en la transferencia a Sesame, según indica el manual [10], del vocabulario y

ontología de los Logs generado por Protégé. En este procedimiento se requirió obtener el código RDF/XML de los datos, instancias de la clase, almacenarlos en un archivo ASCII y posteriormente adicionar el archivo en Sesame empleando la opción Add File. Ante lo que se obtuvo un resultado exitoso como se muestra en la Figura 4.3 y al explorar la ontología en la Figura 4.4.

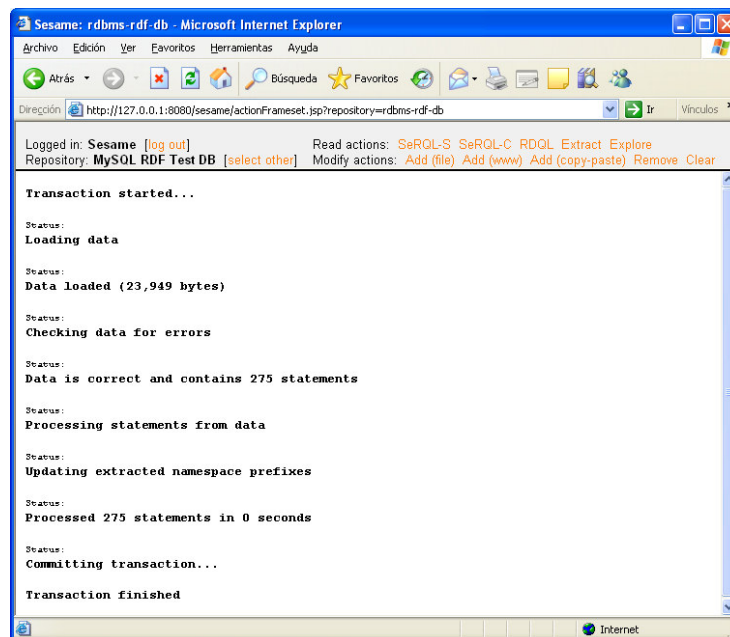


FIGURA 4.3. ADICIÓN DEL VOCABULARIO Y LA ONTOLOGÍA LOGS A SESAME.

Dado que la configuración de Sesame emplea el repositorio en MySQL para el almacenamiento de la información, se hace necesario describir la existencia de la base de datos testdb, declarada para su empleo como repositorio. Base de datos que fue declarada y empleada con la información de la ontología Logs, como se aprecia en las Figuras 4.5 y 4.6 respectivamente.

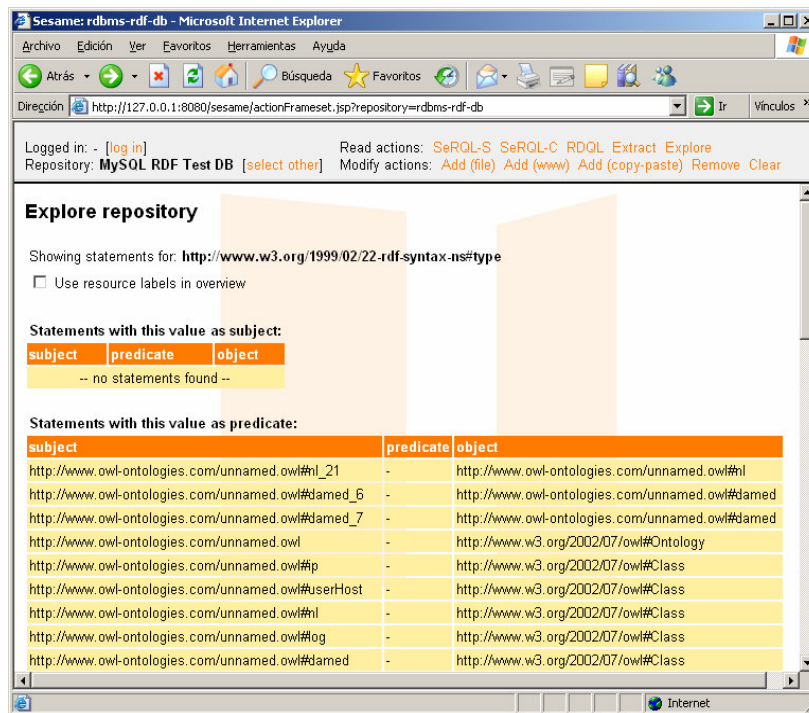


FIGURA 4.4. EXPLORACIÓN DE LA ONTOLOGÍA LOGS EN SESAME.

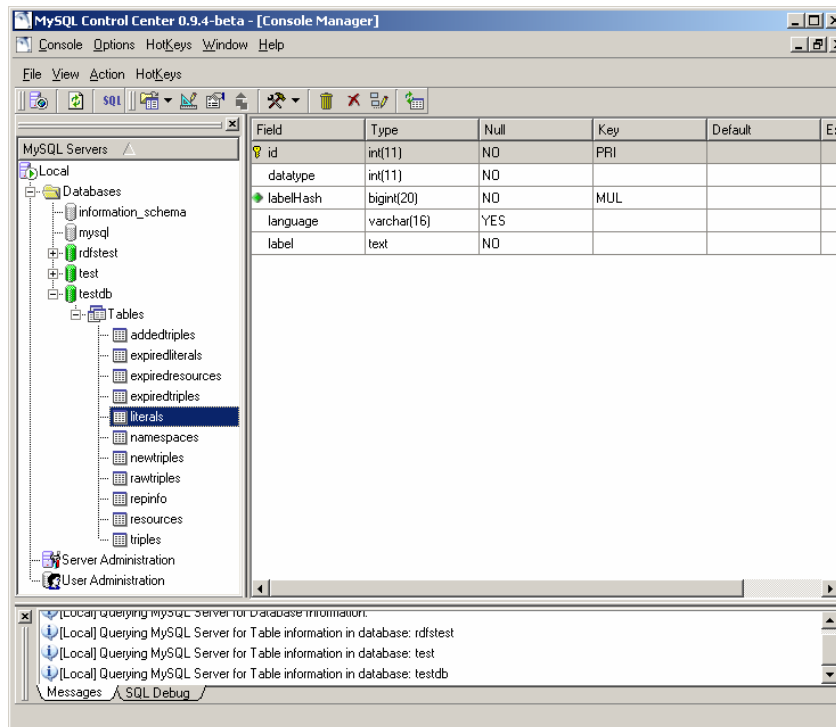


FIGURA 4.5. ESTRUCTURA DE LA ONTOLOGÍA EN EL REPOSITORIO EN MYSQL MANEJADO POR SESAME.

En la Figura 4.6, se destaca en el campo Label de la tabla Literals los elementos correspondientes a los registros logs utilizados como prueba de la definición de instancias presentadas en las tablas 4.1 y 4.2 y posteriormente relacionados en el ítem Código RDF/XML. Por supuesto que en este último, el código, es necesario realizar una búsqueda cuidadosa con el fin de ubicarlos.

id	datatype	labelHash	language	label
1	-36	127	[NULL]	12
2	-35	126	[NULL]	2007-03-19T22:47:0
3	-34	124	[NULL]	42
4	-33	127	[NULL]	processor.load
5	-32	126	[NULL]	2007-03-19T22:41:0
6	-31	127	[NULL]	23.5
7	-30	124	[NULL]	17
8	-29	124	[NULL]	129
9	-28	127	[NULL]	delay.roundtrip
10	-27	124	[NULL]	99
11	-26	127	[NULL]	arteco01
12	-25	124	[NULL]	16
13	-24	124	[NULL]	25
14	-23	127	[NULL]	Information
15	-22	127	[NULL]	program.end
16	-21	124	[NULL]	172
17	-20	127	[NULL]	appended
18	-19	126	[NULL]	2007-03-19T12:12:0
19	-18	124	[NULL]	1

id	select_type	table	type	possible_keys	key	key_len
1	SIMPLE	literals	ALL	[NULL]	[NULL]	[NULL]

FIGURA 4.6. CONTENIDO DE LA TABLA LITERALS DEL REPOSITORIO.

El siguiente paso es generar mediante un programa Java una interface que transforme los registros logs de los servidores mencionados y adicionarlos posteriormente a Sesame para realizar con éstos el análisis propuesto.

4.1.5. Procedimiento de inclusión de los logs en el repositorio

Recopilando los pasos realizados y que anteriormente fueron descritos en los ítem 4.1.1 a 4.1.3, en un procedimiento conformado por etapas, se obtiene:

Compilación de los logs. Esta etapa corresponde a la generación de las instancias de clases, realizadas al final del ítem 4.1.1. “Definición de vocabulario y ontología” y que, para efectos de definición del procedimiento, será la recopilación de los archivos logs a tratar.

Conversión de los logs a códigos RDF/XML. Su nombre resume bien la idea. Sin embargo, para efectos de total claridad, resumamos los pasos realizados anteriormente así: Generación de instancias en Protégé y posterior conversión del vocabulario y ontología e instancias, con esta misma herramienta, a código RDF/XML. Al respecto se podría pensar en importar los archivos logs a Protégé o en disponer de un programa que convierta los logs a código RDF/XML.

Almacenamiento de la ontología logs en Sesame

Obtenido el archivo en código RDF/XML, producto de la conversión, se procede a adicionarlo en Sesame empleando la opción Add File. De esta forma se ponen los logs en el repositorio y quedan disponibles para ser empleados en el momento de ser requeridos. Los logs ahora en el repositorio

gozan de las ventajas ofrecidas por el manejo de éste, respaldado por toda la funcionalidad de un manejador de base de datos relacional y que para nuestro caso concreto es MySQL.

El siguiente paso es la explotación de los logs para los fines definidos anteriormente en el primer y segundo paso realizados en la definición del vocabulario y ontología. Tema que será tratado posteriormente y en el que se describe la aplicación desarrollada para tal fin.

4.1.6. Evaluación de la ontología

Definido el vocabulario y ontología, obtenido el código RDF/XML y efectuado el almacenamiento de la ontología en Sesame, como respectivamente se presentó en los ítems 4.1.1, 4.1.3 y 4.1.4, es necesario pasar a la etapa de evaluación o análisis de la ontología propuesta, involucrando herramientas de diagnóstico, como fue presentado en el capítulo 3. “Método de Investigación”.

Para la evaluación de la ontología se empleó uno de los lenguajes de consulta de Sesame, SeRQL. Este lenguaje que fue presentado en el capítulo 2, sección 2.5.2 “Sesame” y a donde puede remitirse el lector para mayor información. La estructuración de las consultas mostrará la potencialidad de la ontología creada y dará una estimación de las posibilidades de las aplicaciones futuras.

Previa a la prueba empleando SeRQL, se había realizado una prueba a la ontología empleando la opción Run ontology test... en el menú OWL de Protégé obteniéndose un resultado positivo. Esta función puede ser configurada por el usuario, opción Test setting del mismo menú, donde se presentan todas las facilidades para realizar una prueba exhaustiva, como se muestra en la figura 4.7.

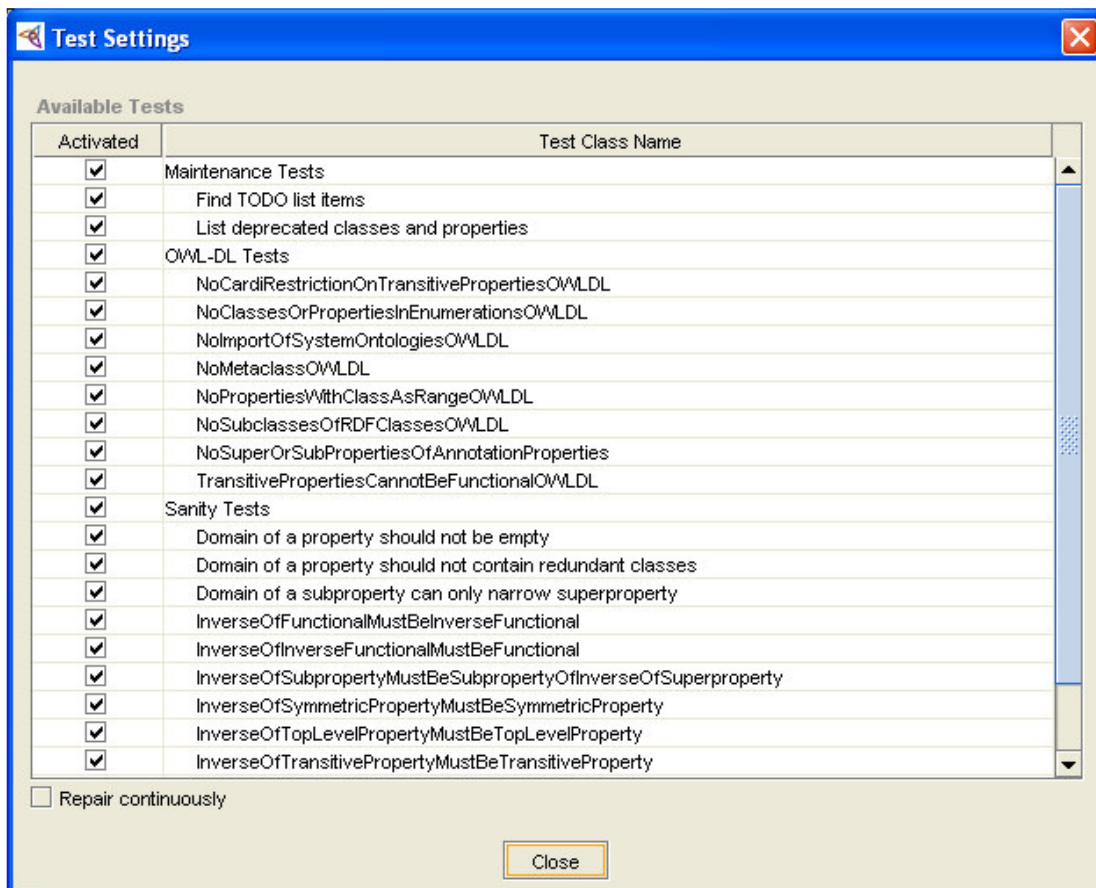


FIGURA 4.7. FACILIDADES PARA CONFIGURACIÓN DE PRUEBA A LA ONTOLOGÍA EN PROTÉGÉ.

Pasemos entonces a la siguiente etapa, la evaluación realizada a la ontología Gulf2 empleando consultas en SeRQL. Iniciemos esta sección

aclarando que la muestra de los logs almacenados en el repositorio no es representativa y lo aceptaremos así dado que para el propósito del presente trabajo es más importante mostrar la aplicación de SeRQL en esta situación. Con esta salvedad se presentan a continuación las consultas en SeRQL y el resultado obtenido en Sesame para la evaluación de la ontología Gulf2.

Consulta uno, que devuelve el sujeto y valor de todas las tripletas con la propiedad o:hostName.

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y
      from {X} o:hostName{Y}
```

TABLA 4.4. RESPUESTA A CONSULTA 1 EN SeRQL.

Query results:	
X	Y
http://www.unab.edu.co/2007/Gulf2#NetLogger_instancia_1	"artec01"^^ http://www.w3.org/2001/XMLSchema#string
1 results found in 151 ms.	

Responde “artec01” en la columna Y, que corresponde a la instancia1 del tipo de logs NetLogger en el espacio de nombres o indicado (<http://www.unab.edu.co/2007/Gulf2#>), Tabla 4.4.

Consulta dos, que devuelve todos los sujetos de la clase o:Damed.

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X
      from {X} rdf:type {o:Damed}
```

TABLA 4.5. RESPUESTA A CONSULTA 2 EN SERQL.

Query results:	
X	
	http://www.unab.edu.co/2007/Gulf2#damed_instancia_2
	http://www.unab.edu.co/2007/Gulf2#damed_instancia_1
2 results found in 30 ms.	

La Tabla 4.5 muestra el resultado a la consulta desplegando los logs de la clase Damed.

Consulta tres, se extraen todos los registros de la clase o:Damed con las propiedades Dirección IP del host (dhost) y Nombre del evento (eventName) esta clase.

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y, Z
from
{X} rdf:type {o:Damed};
o:dhost {Y};
o:eventName {Z}
```

TABLA 4.6. RESPUESTA A CONSULTA 3 EN SERQL.

Query results:		
X	Y	Z
http://www.unab.edu.co/2007/Gulf2#damed_instancia_2	"129.42.17.99"^^http://www.w3.org/2001/XMLSchema#string	"processor.load"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#damed_instancia_1	"129.42.17.99"^^http://www.w3.org/2001/XMLSchema#string	"delay.roundtrip"^^http://www.w3.org/2001/XMLSchema#string
2 results found in 121 ms.		

La Tabla 4.6 muestra el resultado a la consulta desplegando los logs de la clase Damed con sus propiedades Dirección IP del host y Nombre del evento.

Consulta cuatro, se extraen todos los registros de la clase o:Log4j con las propiedades Nombre de host (nHost), Hora (hora), Solicitud (solicitud) y Código de respuesta (cRespuesta) esta clase.

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select U, X, Y, Z, W
from
{U} rdf:type {o:Log4j};
o:nHost {X};
o:hora {Y};
o:solicitud {Z};
o:cRespuesta {W}
```

TABLA 4.7. RESPUESTA A CONSULTA 4 EN SERQL.

Query results:				
U	X	Y	Z	W
http://www.unab.edu.co/2007/Gulf2#Log4j_2	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-06T19:00:05"^^ http://www.w3.org/2001/XMLSchema#dateTime	"POST /pls/portal/HTTP/1.1"@en	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_1	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-08T19:00:00"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /pls/portal/docs/1/398935.PNG HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_4	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-08T19:00:00"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /portal/HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string

Continuación Tabla 4.7.

Query results:				
U	X	Y	Z	W
http://www.unab.edu.co/2007/Gulf2#Log4j10	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-07T07:00:04"^^ http://www.w3.org/2001/XMLSchema#dateTime	""GET /pls/portal/docs/1/398929.GIF HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j12	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-08T19:00:00"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /pls/portal/docs/1/398938.PNG HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j16	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-24T19:00:02"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /pls/portal/url/ITEM/1ED6F74F9260167AE0440003BA3D5405 HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"302"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j18	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-07T07:00:04"^^ http://www.w3.org/2001/XMLSchema#dateTime	"POST /pls/portal/!PORTAL HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j17	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-24T19:00:03"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /pls/portal/url/ITEM/1ED6F74F9260167AE0440003BA3D5405 HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"302"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j11	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-06T09:00:04"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET HTTP/1.1"@en	"304"^^ http://www.w3.org/2001/XMLSchema#string

Continuación Tabla 4.7.

Query results:				
U	X	Y	Z	W
http://www.unab.edu.co/2007/Gulf2#Log4j_15	"200.69.124.66"^ ^http://www.w3.org/2001/XMLSchema#string	"2007-02-08T19:00:00"^^ http://www.w3.org/2001/XMLSchema#dateTime	"POST /pls/portal/ HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_9	"200.69.124.66"^ ^http://www.w3.org/2001/XMLSchema#string	"2007-02-07T07:00:04"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /pls/portal/PORTAL HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_3	"200.69.124.66"^ ^http://www.w3.org/2001/XMLSchema#string	"2007-02-06T19:00:06"^^ http://www.w3.org/2001/XMLSchema#dateTime	"POST /pls/portal/ HTTP/1.1"@en	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_6	"200.69.124.66"^ ^http://www.w3.org/2001/XMLSchema#string	"2007-02-07T07:00:04"^^ http://www.w3.org/2001/XMLSchema#dateTime	"POST /pls/portal/IPORTAL HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_7	"200.69.124.66"^ ^http://www.w3.org/2001/XMLSchema#string	"2007-02-07T07:00:04"^^ http://www.w3.org/2001/XMLSchema#dateTime	"ET /portal/ HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_13	"200.69.124.66"^ ^http://www.w3.org/2001/XMLSchema#string	"2007-02-08T19:00:00"^^ http://www.w3.org/2001/XMLSchema#dateTime	"POST /pls/portal/IPORTAL HTTP/1.1"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_5	"200.69.124.66"^ ^http://www.w3.org/2001/XMLSchema#string	"2007-02-06T19:00:06"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /pls/portal/ HTTP/1.1"@en	"200"^^ http://www.w3.org/2001/XMLSchema#string

Continuación Tabla 4.7.

Query results:				
U	X	Y	Z	W
http://www.unab.edu.co/2007/Gulf2#Log4j_4	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"2007-02-06T19:00:06"^^ http://www.w3.org/2001/XMLSchema#dateTime	"GET /portal HTTP/1.1"@en	"200"^^ http://www.w3.org/2001/XMLSchema#string

17 results found in 321 ms.

En la respuesta podemos apreciar los host (se aprecia el más frecuente, para el caso 200.69.124.66, este debido al Proxy, por lo que no tenemos el nombre del host que en realidad realiza la consulta) que consultan el sitio Web de la UNAB, www.unab.edu.co; adicionalmente, obtenemos información de las solicitudes realizadas (de donde podemos obtener la más frecuente, para el caso GET) y el código de respuesta (más frecuente, para el caso 200), Tabla 4.7.

Como podemos apreciar, esta consulta nos presenta diversos resultados por ser muy general, por lo que es necesario realizar preguntas más puntuales y dirigidas a resolver asuntos específicos.

Consulta cinco, se extraen todas las propiedades Dirección IP del host (dhost) y Nombre del evento (eventName) del registro de la clase o:Damed cuyo código es igual a "processor.load".

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y, Z
```



```

from {X} rdf:type {o:Damed};
o:dhost {Y};
o:eventName {Z}
where Z like "processor.load"

```

TABLA 4.8. RESPUESTA A CONSULTA 5 EN SERQL.

Query results:		
X	Y	Z
http://www.unab.edu.co/2007/Gulf2#damed_instancia_2	"129.42.17.99"^^ http://www.w3.org/2001/XMLSchema#string	"processor.load"^^ http://www.w3.org/2001/XMLSchema#string
1 results found in 20 ms.		

La Tabla 4.8 presenta un logs del tipo Demed cuyo Nombre de Evento es igual a "processor.load".

Consulta seis, se extraen todas las propiedades Nombre del host (nHost) y Código de Respuesta (cRespuesta) del registro de la clase o:Log4j cuyo código es igual a "200".

>consulta

```

using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y, Z
from
{X} rdf:type {o:Log4j};
o:nHost {Y};
o:cRespuesta {Z}
where Z like "200"

```

TABLA 4.9. RESPUESTA A CONSULTA 6 EN SERQL.

Query results:		
X	Y	Z
http://www.unab.edu.co/2007/Gulf2#Log4j_2	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_11	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"200"^^ http://www.w3.org/2001/XMLSchema#string

Continuación Tabla 4.9.

Query results:		
X	Y	Z
http://www.unab.edu.co/2007/Gulf2#Log4j_14	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_10	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_12	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_8	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_15	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_9	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_3	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_6	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_7	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_13	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_5	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_4	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"200"^^http://www.w3.org/2001/XMLSchema#string
14 results found in 150 ms.		

Como se presentó en el ítem 4.2.2. “Códigos de estado del servidor” los mensajes de la serie 200 indican que se ha completado satisfactoriamente la

solicitud de un cliente. En este caso, los clientes fueron atendidos satisfactoriamente por el servidor. Es necesario comparar este resultado con los otros resultados porcentuales de las consultas complementarias, que a continuación se realizarán y dependiendo del tipo de muestra tomada y si el porcentaje de clientes satisfechos es mayor que los otros casos en las consultas complementarias (como en este caso se presenta), se podría inferir que los clientes que visitan el servidor de la UNAB, www.unab.edu.co, son clientes satisfechos. Tabla 4.9.

Consulta siete, se extraen todas las propiedades Nombre del host (nHost) y Código de Respuesta (cRespuesta) del registro de la clase o:Log4j cuyo código es igual a "302".

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y, Z
from
{X} rdf:type {o:Log4j};
o:nHost {Y};
o:cRespuesta {Z}
where Z like "302"
```

TABLA 4.10. RESPUESTA A CONSULTA 7 EN SERQL.

Query results:		
X	Y	Z
http://www.unab.edu.co/2007/Gulf2#Log4j_16	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"302"^^http://www.w3.org/2001/XMLSchema#string
http://www.unab.edu.co/2007/Gulf2#Log4j_17	"200.69.124.66"^^http://www.w3.org/2001/XMLSchema#string	"302"^^http://www.w3.org/2001/XMLSchema#string

2 results found in 40 ms.

Se obtuvo las solicitudes redirigidas (Código de respuesta 302) de los registros logs del tipo Log4j, indicándose el nombre del host, Tabla 4.10.

Consulta ocho, se extraen todas las propiedades Nombre del host (nHost) y Código de Respuesta (cRespuesta) del registro de la clase o:Log4j cuyo código es igual a "400".

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y, Z
from
{X} rdf:type {o:Log4j};
o:nHost {Y};
o:cRespuesta {Z}
where Z like "400"
```

TABLA 4.11. RESPUESTA A CONSULTA 8 EN SERQL.

Query results:		
X	Y	Z
0 results found in 331 ms.		

Como se presentó en el ítem 4.2.2. "Códigos de estado del servidor"400 los mensajes de la serie 400 indican que ha habido un error en el lado del cliente. En este caso, no existió en los logs almacenados este tipo de situación. Dependiendo del tipo de muestra tomada, se podría inferir que los clientes no tienen limitaciones para consultar al servidor, Tabla 4.11.

Consulta nueve, se extraen todas las propiedades Nombre del host (nHost) y Código de Respuesta (cRespuesta) del registro de la clase o:Log4j cuyo código es igual a "500".

>consulta

```

using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y, Z
from
{X} rdf:type {o:Log4j};
o:nHost {Y};
o:cRespuesta {Z}
where Z like "500"

```

TABLA 4.12. RESPUESTA A CONSULTA 9 EN SERQL.

Query results:		
X	Y	Z
0 results found in 120 ms.		

Como se presentó en el ítem 4.2.2. “Códigos de estado del servidor”400 los mensajes de la serie 500 indican que ha habido un error en el lado del servidor. En este caso, no existió en los logs almacenados este tipo de situación. Ahora, dependiendo del tipo de muestra tomada, se podría inferir que hay un buen desempeño de parte del servidor, es confiable su servicio y por tanto el cliente recibe un servicio estable, Tabla 4.12.

Consulta diez, se extraen todas las propiedades Nombre del host (nHost) y Bytes Transferidos (bTransferidos) del registro de la clase o:Log4j cuyo valor sea mayor a 20.000 bytes.

>consulta

```

using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select X, Y, Z
from
{X} rdf:type {o:Log4j};
o:nHost {Y};
o:bTransferidos {Z}
where Z > 20000

```

TABLA 4.13. RESPUESTA A CONSULTA 10 EN SERQL.

Query results:		
X	Y	Z
http://www.unab.edu.co/2007/Gulf2#Log4j_2	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"47655"^^ http://www.w3.org/2001/XMLSchema#int
http://www.unab.edu.co/2007/Gulf2#Log4j_14	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"59167"^^ http://www.w3.org/2001/XMLSchema#int
http://www.unab.edu.co/2007/Gulf2#Log4j_9	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"39210"^^ http://www.w3.org/2001/XMLSchema#int
http://www.unab.edu.co/2007/Gulf2#Log4j_6	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"47728"^^ http://www.w3.org/2001/XMLSchema#int
http://www.unab.edu.co/2007/Gulf2#Log4j_7	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"590660"^^ http://www.w3.org/2001/XMLSchema#int
http://www.unab.edu.co/2007/Gulf2#Log4j_13	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"47641"^^ http://www.w3.org/2001/XMLSchema#int
http://www.unab.edu.co/2007/Gulf2#Log4j_5	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"39210"^^ http://www.w3.org/2001/XMLSchema#int
http://www.unab.edu.co/2007/Gulf2#Log4j_4	"200.69.124.66"^^ http://www.w3.org/2001/XMLSchema#string	"59066"^^ http://www.w3.org/2001/XMLSchema#int

8 results found in 50 ms.

En este caso, el 50% de las solicitudes al servidor fueron transferidos al cliente más de 20.000 bytes. De esta forma se podrían establecer parámetros estimativos de los volúmenes de bytes transferidos a los usuarios y hacer una previsión de la capacidad de volumen de despacho del servidor con el fin de prestar un servicio adecuado al cliente al menor costo, Tabla 4.13.

La ontología Gulf2 podría ser optimizada tratando de evitar la redundancia de información, específicamente en lo que respecta a las propiedades Dirección Ip del Host (host) y en Nombre del Host (hostName) en el tipo de log

NetLogger, representado por la clase NetLogger. Para lo que se dispondría de la siguiente jerarquía:

La clase NetLogger (date, event, host, lvl, tgt) optimizada y una nueva subclase HostName (hostName, codigoHostName).

Para lo cual podríamos tener un agente capaz de entender que el significado de la columna “host” de la subclase NetLogger es equivalente al de la propiedad codigoHostName de la subclase HostName. El agente es capaz entonces de combinar ambas subclases o tablas realizando un “mapping” semántico entre ambas propiedades, columnas, mediante una consulta.

Consulta once, Se realizará dicha consulta y se extraerá todas las direcciones IP (host) de la clase o:NetLogger seguidos del nombre de los nombres de los host (o:hostName) que acceden al servidor.

>consulta

```
using namespace o = <http://www.unab.edu.co/2007/Gulf2#>
select DI, NHO
from
{X} rdf:type {o:NetLogger};
o:host {DI};
o:hostName {HN},
{Y} rdf:type {o:HostName};
o:codigoHostName {HHN};
o:hostName {HNA};
where HN = HHN
```

Se podría seguir con la formulación de preguntas, pero ahora se puede dar un parte de aceptación de la ontología planteada, después de haber pasado la evaluación.

4.2. ANÁLISIS DE LOS LOGS

Dado que se busca efectuar un análisis de los logs generados por un servidor Web se hace necesario realizar un breve recuento de los aspectos relacionados con las solicitudes de los clientes y los códigos de estado del servidor.

4.2.1. Solicitud del cliente

Cada solicitud de un cliente, es realizada por un navegador; las solicitudes del cliente Web constan de tres partes: método, el recurso que está solicitando y el número de versión HTTP. Los métodos empleados son GET, POST o HEAD.

- GET. Solicita un documento al servidor.
- HEAD. Como GET, excepto que sólo se devuelven los encabezados.
- POST. Envía datos a algún manipulador que indica el URI.

Existen otros métodos que son menos usados y que se pueden consultar en RFC 2616 [11].

4.2.2. Códigos de estado del servidor

Cuando se recibe la solicitud de parte del cliente, el servidor devolverá un código de estado (Status-Code) y encabezado de respuesta antes de enviar la respuesta.

Los mensajes se clasifican en cinco grupos, y cada uno de ellos en tipos distinto de condición de estado RFC 2616 [11]:

- Los mensajes de la serie 100 son de tipo informativo.
- Los mensajes de la serie 200 indican que se ha completado satisfactoriamente la solicitud de un cliente.
- Los mensajes de la serie 300 indican que la solicitud ha sido redirigida.
- Los mensajes de la serie 400 indican que ha habido un error en el lado del cliente.
- Los mensajes de la serie 500 indican que ha habido un error en el lado del servidor.

4.2.3. Los Logs en el servidor Web Apache

La información que se intercambia cuando un cliente http se conecta a un servidor Web Apache se almacena por defecto en los archivos de registro error_log (todo lo que no funciona) y acces_log (las transacciones http), pero

se pueden crear cualquier número de archivos de registros con ubicación y contenido personalizados con las directivas LogFormat y CustomLog [12]. La ubicación por defecto de los archivos de registro logs son logs/access_log y logs/error_log respectivamente. La generación de los archivos sólo se produce a menos que los archivos de configuración así lo indiquen, porque por defecto esto no se efectúa.

El contenido de access_log, como sabemos sigue el Formato de Registro Común, contiene los siguientes campos:

Nombre del host. Que es la dirección IP del cliente que solicitó el documento del servidor.

Nombre del registro remoto. Por lo general este campo tiene por valor -, esto dado los problemas actuales de ética y la aparición de los spam.

Usuario remoto. Es el nombre del usuario remoto escrito en respuesta a una consulta nombre de usuario/contraseña.

Hora. Es el grupo fecha y hora en que se sirvió la solicitud, incluyendo la zona horaria.

Solicitud. Es la primera línea de solicitud que hizo al servidor. Normalmente, HEAD, GET o POST, seguido del URL solicitado y la versión http en que se esperaba la respuesta.

Código de respuesta. Indica si la solicitud ha tenido éxito y, si no, el tipo de error que se ha producido, acorde a los códigos de estado del servidor, anteriormente presentados, en el ítem 4.2.2. Códigos de estado del servidor.

Bytes transferidos. Es el número de bytes transferidos al cliente.

4.2.4. Logs de los UNAB

Para propósitos de aplicación en el análisis de los logs, se emplean logs del servidor Web de la UNAB [13]. Éste es un servidor Web Apache y para acceso de uso específico del presente trabajo los archivos logs seleccionados se dispusieron en http://www.unab.edu.co/access_log.tar.gz. Para conocer sobre las características y detalles correspondientes a los logs empleados hay que remitirse al capítulo 3. Metodología. Una muestra de los nombres del conjunto de archivos Logs de acceso empleados se presentan en la Figura 4.7.

Se trae como ejemplo el contenido de dos mensajes logs del servidor mencionado, así:

```
200.69.124.66 - - [05/Mar/2007:07:00:21 -0500] "POST
/reports/rwserverlet/getserverinfo HTTP/1.1" 200 48 "-" "-"
```

```
200.69.124.66 - - [05/Mar/2007:07:00:26 -0500] "POST
/pls/portal/!PORTAL.wwpob_page.show?_pageid=233,1 HTTP/1.1" 200
45780 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 1.1.4322) RPT-HTTPClient/0.3-3"
```

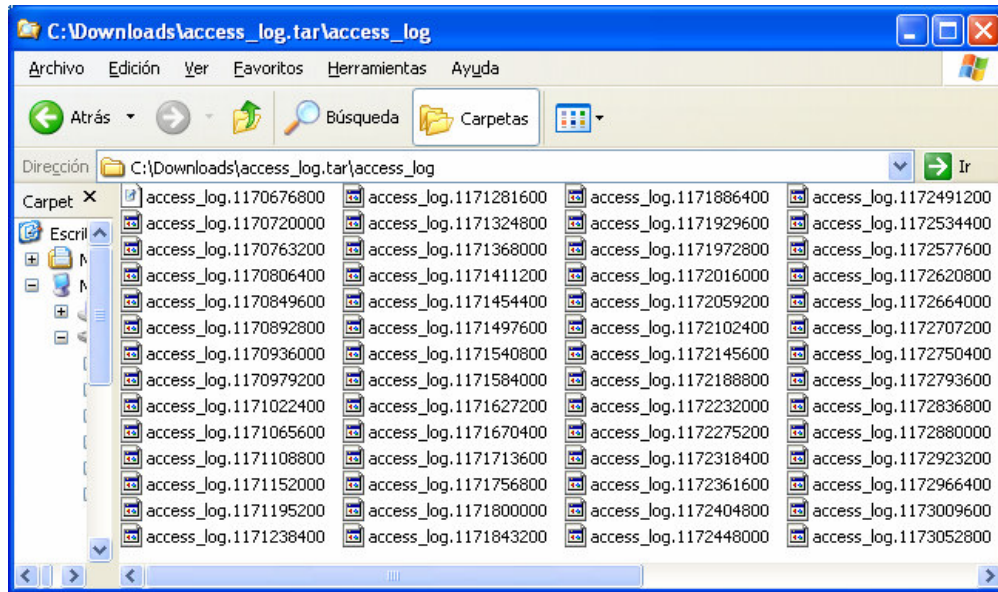


FIGURA 4.7. CONJUNTO DE NOMBRES DE ARCHIVOS LOGS DEL SERVIDOR UNAB.

De éstos se puede evidenciar que los datos contenidos tienen la siguiente forma:

1. 200.69.124.66
2. -
3. -
4. [05/Mar/2007:07:00:26 -0500]
5. "POST /pls/portal/!PORTAL.wwpob_page.show?_pageid=233,1
HTTP/1.1"
6. 200
7. 45780
8. "-"
9. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET
CLR 1.1.4322) RPT-HTTPClient/0.3-3"

1. 200.69.124.66

2. -
3. -
4. [05/Mar/2007:07:00:21 -0500]
5. "POST /reports/rwservlet/getserverinfo HTTP/1.1"
6. 200
7. 48
8. "-"
9. "-"

Se puede evidenciar el formato empleado:

1. IP del usuario
2. Guión (-), en representación del nombre del registro remoto.
3. Guión (-), en ausencia del usuario remoto.
4. [Fecha:hora - referencia GMT]
5. "Solicitud /ubicación y fecha Versión http"
6. Código de respuesta
7. 29268
8. Guión entre comillas "-"
9. "Navegador/Versión (compatible, aplicación; dirección " o un guión en su defecto.

Formato que corresponde con lo planteado para este tipo de servidor.

4.3. APLICACIÓN DE ANÁLISIS DE LOGS

Las aplicaciones de los logs son diversas y el usuario define el uso dependiendo de la aplicación en que se emplean. Se trae a colación algunos datos de interés de los logs de la UNAB seleccionados solo como una muestra de la potencialidad de éstos.

La atención a las solicitudes de los clientes en un día típico en el servidor de la UNAB es aproximadamente de 46.000 solicitudes/día, como se muestra en la Figura 4.2.

Las solicitudes satisfechas exitosamente ascienden al 83% de las solicitudes realizadas al servidor Web de la UNAB. Éstas corresponden a las solicitudes de los clientes calificadas con código de estado del servidor serie 200. Cuando se hace referencia al código de estado del servidor, se efectúa como se presentó en el ítem 4.2.2. Códigos de estado del servidor.

El método típico de solicitudes de los clientes de la UNAB es el GET, representado en un 90%. Los métodos de las solicitudes restantes, correspondientes al 10%, están agrupadas en los métodos HEAD, OPT y POST.

El producto mayormente solicitado por los clientes del servidor Web de la UNAB está representado por el acceso a los productos dispuestos en el directorio /pls/portal y representa un 70% de todas las solicitudes que se efectúan. Por lo que es necesario mejorar y mantener estos productos con el fin: primero, de mantener los clientes que actualmente solicitan estos productos y segundo, partiendo del lema, cliente satisfecho trae más clientes, se espera aumentar el número de clientes que hacen uso de los productos ofrecidos por el servidor Web de la UNAB.

Los datos o estadísticas anteriores, que fueron estimados manualmente a partir de una muestra con el fin de visualizar la potencialidad de la

información de los logs, deben ser obtenidos a través de herramientas de análisis como las anteriormente presentadas. Particularmente la UNAB emplea para el análisis de los logs de su servidor Web la herramienta Advanced Web Statistics 6.5 - Awstats (build 1.857) [14].

REFERENCIAS BIBLIOGRÁFICAS

- [1] Holliday M, Baker M, Boakes R. Grids, Logs, and the Resource Description Framework [online] [actualizado 2004 nov; citado 2007 feb 15]. [aprox. 8 páginas]. Disponible en: <http://dsg.port.ac.uk/projects/slogger/docs/archive/Grids,%20Logs%20and%20the%20Resource%20Description%20Framework%20-%20Holliday,%20Baker,%20Boakes.pdf>.
- [2] O' REILLY On Java.com. Build Flexible Logs With log4j [online]. [actualizado 2002 ago 7; citado 2007 mar 15]. [aprox. 19 pantallas]. Disponible en: <http://www.onjava.com/pub/a/onjava/2002/08/07/log4j.html>.
- [3] Ceki Gülcü, Short introduction to log4j, The Apache Software Foundation, March 2002.
- [4] Logging Services. Class PatternLayout [online]. [actualizado 2005; citado 2007 mar 15]. [aprox. 18 pantallas]. Disponible en: <http://logging.apache.org/log4j/docs/api/org/apache/log4j/PatternLayout.html>.
- [5] Logging Services. Interface ObjectRenderer [online]. [actualizado 2005; citado 2007 mar 15]. [aprox. 2 pantallas]. Disponible en: <http://logging.apache.org/log4j/docs/api/org/apache/log4j/or/ObjectRenderer.html>.
- [6] Noy N y McGuinness D. Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología. Stanfor University, Stanford, CA, 94305. 2005 sep 19; pp. 29.
- [7] Protégé [online]. [actualizado 2007 mar; citado 2007 mar 15]. [aprox. 3 pantallas]. Disponible en: <http://protege.stanford.edu/>.
- [8] W3C World Wide Web Consortium. A. Luotonen. The Common Logfile Format [online]. [actualizado 1995; citado 2007 mar 15]. [aprox. 7 pantallas]. Disponible en: <http://www.w3.org/pub/WWW/Daemon/User/Config/Logging.html>.

[9] OpenRDF.org. Home of Sesame [online]. [actualizado marzo de 2007; citado 2007 mar 15]. [aprox. 8 pantallas]. Disponible en: <http://www.openrdf.org>.

[10] Broekstra J, Kampman A y van Harmelen F. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Lecture Notes In Computer Science 2002; 2342:54-68. Disponible en: <http://www.cs.vu.nl/~frankh/postscript/ISWC02.pdf>.

[11] Fielding R., Gettys J., Mogul J. , Frystyk F., Masinter L., Leach P., et all., "Hypertext Transfer Protocol - HTTP/1.1", Request for Comments: 2616, 1999 jun. Sitio Web: <http://rfc.net/rfc2616.html>.

[12] Bowen R, Coar K. Servidor Apache Al descubierto. Madrid: Pearson Educación. S. A.; 2000. p.334-353.

[13] Universidad Autónoma de Bucaramanga [online]. [actualizado 2007 mar; citado 2007 mar 15]. [aprox. 20 pantallas]. Disponible en: <http://www.unab.edu.co>.

[14] AWStats. Log Analyzer [Online]. 2006 dic 23 [citado 2007 mar 30]; [aprox. 11 pantallas]. Disponible en: <http://awstats.sourceforge.net/>.

5. CONCLUSIONES Y TRABAJO FUTURO

La revisión bibliográfica cumplió el objetivo y presenta el estudio de qué es el RDF, la ontología y la Web semántica mediante la recopilación, clasificación y análisis de la información referente a estas tecnologías como contribución a la aprehensión de este conocimiento y puede ser utilizado en el mejoramiento del diseño de los sistemas de información para la Web, quedando en evidencia la madurez y amplitud del bagaje del conocimiento que las soporta y sus diversas aplicaciones, amparada en una bibliografía extensa que lo acompaña.

La presente investigación plantea una propuesta para el tratamiento semántico de los logs usando RDF, del que hacen parte la definición del vocabulario y ontología Gulf2 como una mejora a la propuesta GULF. El vocabulario y ontología con base en la semántica del RDF permiten el tratamiento y aprovechamiento congruente a los procesos cognitivos de información, permiten unificar los contenidos semánticos y facilitarán el trabajo colaborativo.

Se plantea el procedimiento para el manejo de la ontología empleando herramientas comunes y de uso libre en el medio como son Protégé y Sesame, contemplando el empleo de repositorio soportado, para el caso por MySQL en conexión con Sesame.

Se presenta una propuesta de procedimiento de inclusión de los logs en repositorio, para la semántica del vocabulario y ontología Gulf2 que facilitará su empleo. Debido a que el procedimiento permitirá a los interesados reproducir la experiencia o tener como guía y sobre todo disponer del respaldo y funcionalidad de MySQL como manejador de base de datos relacional, en el manejo de los logs.

Entre los problemas más relevantes encontrados en el uso de RDF están: lo dispendioso en la codificación de la ontología y su posterior empleo, problemas que fueron superados con el empleo de herramientas como Protégé y Sesame, respectivamente. La adición de las instancias, registros logs, en Protégé es bastante dispendioso, esta situación limitó el número de registros logs adicionados.

El trabajo futuro estaría en desarrollar una aplicación para el manejo y explotación semántico de los logs, empleando el vocabulario y la ontología aquí planteada en pro de mejorar los servicios y productos ofrecidos a los clientes prestados por un servidor Web.

BIBLIOGRAFÍA

Bowen R, Coar K. Servidor Apache Al descubierto. Madrid: Pearson Educación. S. A.; 2000.

Comer D. Redes globales de información con Internet y TCP/IP – Principios básicos, protocolos y arquitectura. 3ª ed. México: Prentice-Hall, Inc.; 1998.

Deitel H, Deitel P. Cómo Programar en JAVA. México: Pearson Educación; 2004.

Fowler M, Scout K. UML gota a gota. México: Addison Wesley Longman de México; 1999.

Goslin J, Joy B, Steele G. The Java Language Specification. California: Addison Wesley; 1996.

Horridge M, Knublauch H, Rector A, Stevens R, Wroe C. A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools, Edition 1.0, The University Of Manchester, August 27, 2004.

Larman C. UML y Patrones – Introducción al análisis y diseño orientado a objetos. México: Prentice Hall; 1999.

Maruyama H, Clark A, Murata M, Uramoto M, Tamamura K, Nakamura R, et al. XML and Java: Developing Web Applications. 2ª Ed. New York: Addison Wesley Professional; 2002.

Maruyama H, Tamura K, Uramoto N. Sitios Web con XML y Java. Madrid: Prentice-Hall; 2000.

Meyer B. Construcción de Software Orientado a Objetos. Madrid: Prentice Hall; 1999.

Morrison M, et al. XML Al descubierto. Madrid: Prentice-Hall; 2000.

Nijssen G, Halpin T. Conceptual Schema and Relational Database Design. Sydney: Prentice Hall; 1989.

ANEXOS

ANEXO I - CÓDIGO RDF/XML GENERADO POR PROTÉGÉ

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.unab.edu.co/2007/Gulf2#"
  xml:base="http://www.unab.edu.co/2007/Gulf2">
  <owl:Ontology rdf:about=""/>
  <rdfs:Class rdf:ID="NetLogger">
    <rdfs:subClassOf>
      <rdfs:Class rdf:ID="Gulf2"/>
    </rdfs:subClassOf>
    <rdfs:comment xml:lang="es">NetLogger. Subclase de Gulf2 y uno de los
    formatos logs mas comunes.</rdfs:comment>
  </rdfs:Class>
  <rdfs:Class rdf:about="#Gulf2">
    <rdfs:comment xml:lang="es">Gulf2 viene de la ontologia previa GULF de
    Holliday et. all.</rdfs:comment>
    <rdfs:comment xml:lang="es">Gulf2 esta conformada por tres clases:
    Demed-WG, Log4j y NetLogger</rdfs:comment>
  </rdfs:Class>

```

```

    <rdfs:comment xml:lang="es">Gulf2. Clase propuesta, resultado de la
    investigacion</rdfs:comment>

```

```

</rdfs:Class>

```

```

<rdfs:Class rdf:ID="Damed">

```

```

    <rdfs:subClassOf rdf:resource="#Gulf2"/>

```

```

    <rdfs:comment xml:lang="es">Log4j. Subclase de Gulf2 y uno de los
    formatos logs mas comunes.</rdfs:comment>

```

```

</rdfs:Class>

```

```

<rdfs:Class rdf:ID="Log4j">

```

```

    <rdfs:subClassOf rdf:resource="#Gulf2"/>

```

```

    <rdfs:comment xml:lang="es">Log4j. Subclase de Gulf2 y uno de los
    formatos logs mas comunes.</rdfs:comment>

```

```

</rdfs:Class>

```

```

<rdf:Property rdf:ID="value">

```

```

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

```

```

    <rdfs:comment xml:lang="es">value. Valor del evento. Es una propiedad
    de Damed.</rdfs:comment>

```

```

    <rdfs:domain rdf:resource="#Damed"/>

```

```

    <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>

```

```

</rdf:Property>

```

```

<rdf:Property rdf:ID="hora">

```

```

    <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>

```

```

    <rdfs:range
    rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>

```



```
<rdfs:comment xml:lang="es">hora. Es la fecha y hora en que se sirvio la
solicitud</rdfs:comment>
```

```
<rdfs:comment xml:lang="es">hora. Propiedad de Log4j.</rdfs:comment>
```

```
<rdfs:domain rdf:resource="#Log4j"/>
```

```
</rdf:Property>
```

```
<rdf:Property rdf:ID="uRemoto">
```

```
<rdfs:comment xml:lang="es">uRemoto. Indica el nombre de registro
remoto. Normalmente con valor -.</rdfs:comment>
```

```
<rdfs:comment xml:lang="es">uRemoto. Propiedad de
Log4j.</rdfs:comment>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
<rdfs:domain rdf:resource="#Log4j"/>
```

```
</rdf:Property>
```

```
<rdf:Property rdf:ID="nHost">
```

```
<rdfs:comment xml:lang="es">nHost. Propiedad de Log4j.</rdfs:comment>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
<rdfs:domain rdf:resource="#Log4j"/>
```

```
<rdfs:comment xml:lang="es">nHost. Nombre de host</rdfs:comment>
```

```
<rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

```
</rdf:Property>
```

```
<rdf:Property rdf:ID="nRegistro">
```

```
<rdfs:domain rdf:resource="#Log4j"/>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
<rdfs:comment xml:lang="es">nRegistro. Propiedad de
Log4j.</rdfs:comment>
```

```
<rdfs:comment xml:lang="es">nRegistro. Indica el nombre del registro
remoto. Normalmente -. </rdfs:comment>
```

```
<rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

```
</rdf:Property>
```

```
<rdf:Property rdf:ID="solicitud">
```

```
<rdfs:domain rdf:resource="#Log4j"/>
```

```
<rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

```
<rdfs:comment xml:lang="es">solicitud. Propiedad de
Log4j.</rdfs:comment>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
<rdfs:comment xml:lang="es">solicitud. Es la primera linea de la solicitud
que se hizo al servidor. Normalmente GET, POST o HEAD</rdfs:comment>
```

```
</rdf:Property>
```

```
<rdf:Property rdf:ID="valueN">
```

```
<rdfs:domain rdf:resource="#NetLogger"/>
```

```
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
<rdfs:comment xml:lang="es">valueN. Valor.</rdfs:comment>
```

```
<rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

```
<rdfs:comment xml:lang="es">valueN. Propiedad de la clase
NetLogger.</rdfs:comment>
```

```
</rdf:Property>
```

```

<rdf:Property rdf:ID="target">
  <rdfs:comment xml:lang="es">target. Objetivo. Es una propiedad de
Damed.</rdfs:comment>

  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

  <rdfs:comment xml:lang="es">target. Propiedad de la clase
Damed.</rdfs:comment>

  <rdfs:domain rdf:resource="#Damed"/>

  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</rdf:Property>

<rdf:Property rdf:ID="typecode">

  <rdfs:domain rdf:resource="#NetLogger"/>

  <rdfs:comment xml:lang="es">typecode. Indica como interpretar el
valor.</rdfs:comment>

  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>

  <rdfs:comment xml:lang="es">typecode. Propiedad de la clase
NetLogger.</rdfs:comment>

  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

</rdf:Property>

<owl:FunctionalProperty rdf:ID="timestamp">

  <rdfs:domain rdf:resource="#Damed"/>

  <rdfs:comment xml:lang="es">timestamp. Hora y fecha de la solicitud. Es
una propiedad de Damed.</rdfs:comment>

  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>

```

```

    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>

</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="bTransferidos">

    <rdfs:domain rdf:resource="#Log4j"/>

    <rdfs:comment xml:lang="es">bTransferidos. Propiedad de
Log4j.</rdfs:comment>

    <rdfs:comment xml:lang="es">bTransferidos. Numero total de byte que se
trasfieren al cliente</rdfs:comment>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>

    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>

</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="cRespuesta">

    <rdfs:domain rdf:resource="#Log4j"/>

    <rdfs:comment xml:lang="es">cRespuesta. Propiedad de
Log4j.</rdfs:comment>

    <rdfs:comment xml:lang="es">cRespuesta. Indica si la solicitud ha tenido
exito y, si no, el tipo de error que se ha producido</rdfs:comment>

    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>

    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="navegador">

    <rdfs:comment xml:lang="es">navegador. Especifica el navegador
empleado</rdfs:comment>

```

```

<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Log4j"/>

<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>

<rdfs:comment xml:lang="es">navegador. Propiedad de
Log4j.</rdfs:comment>

</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="key">

<rdfs:comment xml:lang="es">key. Propiedad de la clase
NetLogger.</rdfs:comment>

<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

<rdfs:domain rdf:resource="#NetLogger"/>

<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>

<rdfs:comment xml:lang="es">key. Es un string que describe el valor
principal.</rdfs:comment>

</owl:FunctionalProperty>

<owl:FunctionalProperty rdf:ID="eventName">

<rdfs:comment xml:lang="es">eventName. Propiedad de la clase
Damed.</rdfs:comment>

<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

<rdfs:comment xml:lang="es">eventName. Nombre del evento. Es una
propiedad de Damed.</rdfs:comment>

<rdfs:domain rdf:resource="#Damed"/>

<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>

```

```

</owl:FunctionalProperty>
<Log4j rdf:ID="Log4j_2">
  <nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</nRegistro>
  <cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200</cRespuesta>
  <nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200.69.124.66</nHost>
  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-02-06T19:00:05</hora>
  <solicitud xml:lang="en">POST /pls/portal/ HTTP/1.1</solicitud>
  <navegador xml:lang="en">Mozilla/4.0</navegador>
  <bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >47655</bTransferidos>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Log4j_2. Segundo registro</rdfs:comment>
  <uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</uRemoto>
</Log4j>
<Log4j rdf:ID="Log4j_11">
  <rdfs:comment xml:lang="es">Log4j_11. Instancia once. Segunda
  miniserie de registros</rdfs:comment>
  <navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```
>Mozilla/4.0</navegador>
<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>GET /pls/portal/docs/1/398935.PNG HTTP/1.1</solicitud>
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200</cRespuesta>
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>
<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-08T19:00:00</hora>
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1485</bTransferidos>
<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>
</Log4j>
<Log4j rdf:ID="Log4j_14">
  <uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</uRemoto>
  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-02-08T19:00:00</hora>
  <cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
>200</cRespuesta>
<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>GET /portal/ HTTP/1.1</solicitud>
<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/4.0</navegador>
<rdfs:comment xml:lang="es">Log4j_14. Instancia catorce</rdfs:comment>
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>59167</bTransferidos>
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>
</Log4j>
<NetLogger rdf:ID="NetLogger_1">
  <valueN rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >2007-04-15T21:30:01</valueN>
  <rdfs:comment xml:lang="es">NetLogger_1. Primera
instancia</rdfs:comment>
  <typecode rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >t</typecode>
  <key xml:lang="en">DATE</key>
</NetLogger>
<Log4j rdf:ID="Log4j_10">
```



```
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200</cRespuesta>

<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>

<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-07T07:00:04</hora>

<rdfs:comment xml:lang="es">Log4j_10. Instancia decima</rdfs:comment>

<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/5.0</navegador>

<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>"GET /pls/portal/docs/1/398929.GIF HTTP/1.1</solicitud>

<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>

<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>

<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>67</bTransferidos>

</Log4j>

<Log4j rdf:ID="Log4j_12">

  <navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mozilla/4.0</navegador>

  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
```

```
>2007-02-08T19:00:00</hora>
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200</cRespuesta>
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>
<rdfs:comment xml:lang="es">Log4j_12. Instancia
doceava</rdfs:comment>
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>
<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>GET /pls/portal/docs/1/398938.PNG HTTP/1.1</solicitud>
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1733</bTransferidos>
<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>
</Log4j>
<Log4j rdf:ID="Log4j_16">
<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>GET /pls/portal/url/ITEM/1ED6F74F9260167AE0440003BA3D5405
HTTP/1.1</solicitud>
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>302</cRespuesta>
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
>200.69.124.66</nHost>
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>5</bTransferidos>
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>
<rdfs:comment xml:lang="es">Log4j_16. Instancia dieciseis. Tercera
miniserie de registros</rdfs:comment>
<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/4.0</navegador>
<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>
<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-24T19:00:02</hora>
</Log4j>
<Log4j rdf:ID="Log4j_8">
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>11853</bTransferidos>
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200</cRespuesta>
<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/5.0</navegador>
```

```
<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-07T07:00:04</hora>

<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>

<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>POST /pls/portal/!PORTAL HTTP/1.1</solicitud>

<rdfs:comment xml:lang="es">Log4j_8. Instancia octava</rdfs:comment>

<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>

</Log4j>

<Damed rdf:ID="Damed_2">

  <value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >129.42.17.99</value>

  <rdfs:comment xml:lang="es">Damed_2. Segunda
instancia</rdfs:comment>

  <target rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >12</target>

  <eventName xml:lang="en">processor.load</eventName>

  <timestamp rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-03-19T22:47:00</timestamp>

</Damed>

<Log4j rdf:ID="Log4j_17">

  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
```

```
>2007-02-24T19:00:03</hora>
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66 </nHost>
<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>GET /pls/portal/url/ITEM/1ED6F74F9260167AE0440003BA3D5405
HTTP/1.1</solicitud>
<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>
<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/4.0</navegador>
<rdfs:comment xml:lang="es">Log4j_17. Instancia
diecisiete</rdfs:comment>
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>0</bTransferidos>
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>302</cRespuesta>
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>
</Log4j>
<Log4j rdf:ID="Log4j_1">
<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>
<navegador xml:lang="en">Mozilla/4.0</navegador>
```

```
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>0</bTransferidos>

<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</nRegistro>

<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-06T09:00:04</hora>

<solicitud xml:lang="en">GET HTTP/1.1</solicitud>

<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Primer registro. Primera miniserie de registros.</rdfs:comment>

<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>304</cRespuesta>

<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>

</Log4j>

<Damed rdf:ID="Damed_3">

  <target rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >129.42.17.99</target>

  <rdfs:comment xml:lang="es">Damed_3. Tercera
instancia</rdfs:comment>

  <eventName xml:lang="en">system.OS.name</eventName>

  <value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Linux 2.4.7-10</value>

</Damed>
```

```
<Log4j rdf:ID="Log4j_15">
  <nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200.69.124.66</nHost>
  <solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >POST /pls/portal/ HTTP/1.1</solicitud>
  <nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</nRegistro>
  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-02-08T19:00:00</hora>
  <rdfs:comment xml:lang="es">Log4j_15. Instancia quince</rdfs:comment>
  <bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >11738</bTransferidos>
  <navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mozilla/4.0</navegador>
  <uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</uRemoto>
  <cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200</cRespuesta>
</Log4j>
<Log4j rdf:ID="Log4j_9">
  <nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</nRegistro>
```

```
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200</cRespuesta>

<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/5.0</navegador>

<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-07T07:00:04</hora>

<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>

<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>39210</bTransferidos>

<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>GET /pls/portal/PORTAL HTTP/1.1</solicitud>

<rdfs:comment xml:lang="es">Log4j_9. Instancia novena</rdfs:comment>

<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>

</Log4j>

<Log4j rdf:ID="Log4j_6">

<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200</cRespuesta>

<solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>POST /pls/portal/!PORTAL HTTP/1.1</solicitud>

<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```



```
>--</nRegistro>
<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/5.0</navegador>
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>47728</bTransferidos>
<rdfs:comment xml:lang="es">Log4j_6. Instancia sexta. Segunda miniserie
de registros.</rdfs:comment>
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>
<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>--</uRemoto>
<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-07T07:00:04</hora>
</Log4j>
<Log4j rdf:ID="Log4j_3">
<bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>11853</bTransferidos>
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>--</nRegistro>
<hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-02-06T19:00:06</hora>
<cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200</cRespuesta>
```

```
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>

<solicitud xml:lang="en">POST /pls/portal/ HTTP/1.1</solicitud>

<navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Mozilla/4.0</navegador>

<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Log4j_3. Tercera instancia</rdfs:comment>

<uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>-</uRemoto>

</Log4j>

<Damed rdf:ID="Damed_1">

  <eventName xml:lang="en">delay.roundtrip</eventName>

  <rdfs:comment xml:lang="es">Damed_1. Primera
instancia</rdfs:comment>

  <value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>129.42.17.99, 140.22.9.95</value>

  <target rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>23.5</target>

  <timestamp rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2007-03-19T22:41:00</timestamp>

</Damed>

<NetLogger rdf:ID="NetLogger_3">
```

```
<rdfs:comment xml:lang="es">NetLogger_3. Tercera
instancia</rdfs:comment>

<valueN rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>131.243.2.142</valueN>

<key xml:lang="en">HOST</key>

<typecode rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>s</typecode>

</NetLogger>

<Log4j rdf:ID="Log4j_7">

  <cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200</cRespuesta>

  <nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</nRegistro>

  <bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >590660</bTransferidos>

  <rdfs:comment xml:lang="es">Log4j_7. Instancia septima</rdfs:comment>

  <navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mozilla/5.0</navegador>

  <solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >ET /portal/ HTTP/1.1</solicitud>

  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-02-07T07:00:04</hora>

  <uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
>-</uRemoto>
<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66</nHost>
</Log4j>
<Log4j rdf:ID="Log4j_13">
  <uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</uRemoto>
  <rdfs:comment xml:lang="es">Log4j_13. Instancia
treceava</rdfs:comment>
  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-02-08T19:00:00</hora>
  <cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200</cRespuesta>
  <navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mozilla/4.0</navegador>
  <nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200.69.124.66</nHost>
  <bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >47641</bTransferidos>
  <solicitud rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >POST /pls/portal/!PORTAL HTTP/1.1</solicitud>
  <nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</nRegistro>
```

```
</Log4j>
<NetLogger rdf:ID="NetLogger_2">
  <typecode rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >s</typecode>
  <valueN xml:lang="en">Information</valueN>
  <rdfs:comment xml:lang="es">NetLogger_2. Segunda
instancia</rdfs:comment>
  <key xml:lang="en">LVL</key>
</NetLogger>
<Log4j rdf:ID="Log4j_5">
  <navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mozilla/4.0</navegador>
  <bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >39210</bTransferidos>
  <solicitud xml:lang="en">GET /pls/portal/ HTTP/1.1</solicitud>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Log4j_5. Quinta instancia</rdfs:comment>
  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-02-06T19:00:06</hora>
  <uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >-</uRemoto>
  <cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200</cRespuesta>
```

```
<nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>--</nRegistro>

<nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>200.69.124.66 </nHost>

</Log4j>

<Log4j rdf:ID="Log4j_4">

  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Log4j_4. Cuarta instancia</rdfs:comment>

  <navegador rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Mozilla/4.0</navegador>

  <hora rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
  >2007-02-06T19:00:06</hora>

  <uRemoto rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >--</uRemoto>

  <cRespuesta rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200</cRespuesta>

  <bTransferidos rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >59066</bTransferidos>

  <nRegistro rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >--</nRegistro>

  <solicitud xml:lang="en">GET /portal HTTP/1.1</solicitud>

  <nHost rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
>200.69.124.66</nHost>  
</Log4j>  
</rdf:RDF>
```