**ARQUITECTURA DE UN SISTEMA BASADO EN AGENTES PARA LA RECUPERACIÓN DE METADATOS RDF EN BASE A UNA ONTOLOGÍA DE DOCUMENTOS**

Por
Juan Larry Dinos Rojas

Tesis sometida en cumplimiento parcial de los requisitos para el grado de

MAESTRO EN CIENCIAS
en
INGENIERÍA DE COMPUTADORAS

UNIVERSIDAD DE PUERTO RICO
MAYAGÜEZ CAMPUS
2004

Aprobado por:

| | |
|---|---|
| _____ | _____ |
| J. Fernando Vega Riveros, Ph. D.<br>Presidente, Comité Graduado | Fecha |
| _____ | _____ |
| Manuel Rodríguez Martínez, Ph. D.<br>Miembro, Comité Graduado | Fecha |
| _____ | _____ |
| Néstor Rodríguez, Ph. D.<br>Miembro, Comité Graduado | Fecha |
| _____ | _____ |
| Jorge Ortiz Álvarez, Ph. D.<br>Director del Departamento | Fecha |
| _____ | _____ |
| Awilda Valle, M. S.<br>Representante de Estudios Graduados | Fecha |

# ABSTRACT

This thesis describes the architecture of an agent-based system to retrieve information resources for a higher education environment. An ontology representing documents used by students and faculty members is also described, where RDF-S (Resource Description Framework Schema) is used. The RDF-S recommendation offers terms that allow the representation of concepts, their relationships and their attributes, all of which formed the metadata. The architecture uses three types of agent (UserAgent, SearchAgent and OntologyAgent), where each agent is in charge of different tasks such as user interaction, ontology retrieval and metadata search. The prototype called oSEMA was development using this architecture, to recover information in a distributed environment. Tests were conducted to measure the execution time to process a request in two scenarios. As a result, a decentralized organization of agents with their own repositories RDF presents a better response time. Also, using this ontology to describe documents provided greater advantages.

# RESUMEN

Esta tesis describe la arquitectura de un sistema basado en agentes para la recuperación de recursos de información en un entorno de educación. Se construyó una ontología en la cual se representan los documentos utilizados por estudiantes y profesores. Se utilizó el lenguaje *Resource Description Framework Schema* (RDF-S) para representar la ontología de documentos. En la arquitectura del sistema se desarrollaron tres tipos de agentes (*UserAgent, SearchAgent* y *OntologyAgent*) que se encargan de la interacción con el usuario, la recuperación de la ontología y la búsqueda de meta-datos. A base de esta arquitectura se desarrolló un prototipo llamado *oSEMA* para recuperar información en un entorno distribuido. Se realizaron pruebas para medir el tiempo de respuesta en dos escenarios. Se encontró que, una organización descentralizada de agentes con sus propios repositorios RDF presenta un mejor tiempo de respuesta. Además, resultó ventajoso usar esta ontología para describir y compartir los documentos.

©Copyright por Juan Larry Dinos-Rojas, Mayo 2004

Dedicatoria

A mi padre Tomás, a mi madre Julia y a mis hermanos por todo el apoyo incondicional que me han brindado.

# RECONOCIMIENTOS

Al Dr. J. Fernando Vega-Riveros por todo el apoyo incondicional y consejos para la realización del presente trabajo. Y también agradezco a los miembros del Comité Graduado los doctores: Manuel Rodríguez y Néstor Rodríguez, y a la profesora. Awilda Valle por las sugerencias y correcciones del presente manuscrito. Deseo agradecer a todos mis amigos y en especial a mi novia por todo su apoyo. Finalmente, deseo agradecer al programa "PRECISE" por su apoyo económico para la investigación realizada en los veranos de los años 2002 y 2003.

# TABLA DE CONTENIDO

**LISTA DE TABLAS** ..... **X**

**LISTA DE FIGURAS** ..... **XI**

**LISTA DE APÉNDICES** ..... **XIII**

**CAPÍTULO 1 INTRODUCCIÓN** ..... **1**

1.1 MOTIVACIÓN ..... 4
1.2 OBJETIVOS ..... 5
1.3 CONTRIBUCIONES ..... 6
1.4 ORGANIZACIÓN DE LA TESIS ..... 6

**CAPÍTULO 2 ESTADO DEL ARTE** ..... **7**

2.1 INTRODUCCIÓN ..... 7
2.2 LOS LENGUAJES SEMÁNTICOS Y LAS ONTOLOGÍAS ..... 7
*2.2.1 El Lenguaje RDF* ..... *8*
*2.2.2 RDF-S* ..... *10*
*2.2.3 Ontology Inference Layer* ..... *12*
*2.2.4 Darpa Agent Markup Language + OIL* ..... *12*
*2.2.5 Ontology Web Language* ..... *14*
*2.2.6 Ontologías en la representación del conocimiento* ..... *15*
2.3 AGENTES ..... 17
*2.3.1 Plataforma de Agentes* ..... *18*
*2.3.2 Lenguajes de Comunicación entre Agentes* ..... *20*

**CAPÍTULO 3 TRABAJOS RELACIONADOS** ..... **24**

3.1 INTRODUCCIÓN ..... 24
3.2 KBS-HYPERBOOK ..... 24
3.3 UNA HERRAMIENTA PARA LA CAPTURA DE APUNTES ..... 25
3.4 ENTORNO DE TRABAJO PARA ANOTACIONES: ONTOANNOTATE ..... 26
3.5 HYPERCLIP ..... 27
3.6 MANTHA ..... 27

**CAPÍTULO 4 REPRESENTACIÓN DE LOS METADATOS** ..... **29**

4.1 INTRODUCCIÓN ..... 29
4.2 IDENTIFICACIÓN DE LOS METADATOS ..... 30

*4.2.1 Identificación de los documentos* ..... *30*
*4.2.2 Principales Datos Identificados* ..... *31*
4.3 CONSTRUCCIÓN DE LA ONTOLOGÍA: LA ONTOLOGÍA DE DOCUMENTOS ..... 32
*4.3.1 Las clases y la Jerarquía de Clases* ..... *32*
*4.3.2 Las Propiedades de las Clases* ..... *33*
4.4 EDICIÓN DE LA ONTOLOGÍA CON PROTÉGÉ ..... 34
4.5 REPRESENTANDO LA ONTOLOGÍA DE DOCUMENTOS EN EL LENGUAJE RDF-S ..... 36

**CAPÍTULO 5 ARQUITECTURA BASADA EN AGENTES PARA LA RECUPERACIÓN DE METADATOS ..... 41**

5.1 INTRODUCCIÓN ..... 41
5.2 JADE: UNA ALTERNATIVA PARA EL DESARROLLO DE AGENTES ..... 41
5.3 ACERCA DE LA ARQUITECTURA ..... 43
5.4 IMPLEMENTACIÓN DE LOS AGENTES ..... 44
*5.4.1 UserAgent* ..... *44*
*5.4.2 OntologyAgent* ..... *45*
*5.4.3 SearchAgent* ..... *46*
5.5 MENSAJES UTILIZADOS EN LA COMUNICACIÓN ENTRE LOS AGENTES ..... 47
5.6 BÚSQUEDA DE METADATOS ..... 50
5.7 IMPLEMENTACIÓN DE UN PROTOTIPO: OSEMA ..... 52
*5.7.1 Explorador Semántico* ..... *52*
*5.7.2 Pantalla de Anotaciones* ..... *53*

**CAPÍTULO 6 METODOLOGÍA ..... 55**

6.1 INTRODUCCIÓN ..... 55
6.2 PROPÓSITO DE LA PRUEBA ..... 55
6.3 METODOLOGÍA ..... 56
*6.3.1 Equipos y Herramientas* ..... *56*
*6.3.2 Escenarios* ..... *57*
*6.3.3 Configuración de los Repositorios RDF* ..... *60*
*6.3.4 Consultas* ..... *61*

**CAPÍTULO 7 RESULTADOS ..... 63**

7.1 PRIMER ESCENARIO ..... 63
7.2 SEGUNDO ESCENARIO ..... 65
7.3 DISCUSIÓN ..... 68

**CAPÍTULO 8 CONCLUSIONES Y TRABAJOS FUTUROS ..... 69**

8.1 CONCLUSIONES ..... 69
8.2 TRABAJOS FUTUROS ..... 71

**BIBLIOGRAFÍA ..... 73**

**APÉNDICE A ..... 78**

**ENCUESTA REALIZADA A LOS PROFESORES ..... 78**

CUESTIONARIO UTILIZADO PARA LA ENCUESTA ..... 78
RESULTADOS DE LA ENCUESTA ..... 86

**APÉNDICE B ..... 94**

**ACRÓNIMOS ..... 94**

# LISTA DE TABLAS

Tabla 2-1 Partes de una Expresión..... 9

Tabla 4-1 Descripción de las Propiedades de la Clase *Document* ..... 34

Tabla 6-1 Descripción de las Computadoras Utilizadas en las Pruebas ..... 56

Tabla 7-1 Tiempo de Respuesta de las Consultas Realizadas por los Agentes en un Ambiente Centralizado (en milisegundos)..... 64

Tabla 7-2 Tiempo de Respuesta de la Consultas Realizadas por 8 Agentes en un Ambiente Descentralizado ..... 67

# LISTA DE FIGURAS

Figura 2.1 Grafo de una Expresión RDF ..... 10

Figura 2.2 Expresión RDF ..... 10

Figura 2.3 Ejemplo de una RDF Schema..... 11

Figura 2.4 Representación de una Ontología en DAML+OIL ..... 14

Figura 2.5 Representación de una Ontología en OWL ..... 15

Figura 2.6 Representación de una Ontología que Describe a Profesores y Estudiantes... 17

Figura 2.7 Arquitectura de la Plataforma de Agentes Propuesta por FIPA [FIPA02a].... 19

Figura 2.8 Capas del Lenguaje KQML..... 21

Figura 2.9 Estructura del Mensaje ACL ..... 22

Figura 4.1 Representación de la Ontología de Documentos..... 33

Figura 4.2 Representación de la Ontología en *Protege* ..... 35

Figura 4.3 Representación en Grafos RDF de una Expresión ..... 37

Figura 4.4 Descripción de los Datos de un Documento en RDF ..... 38

Figura 4.5 Parte de la Ontología de Documentos en Grafos RDF..... 39

Figura 4.6 Parte de la Representación de la Ontología de Documentos en el Lenguaje RDF-S ..... 40

Figura 5.1 Arquitectura de JADE ..... 42

Figura 5.2 Arquitectura Basada en Agentes ..... 43

Figura 5.3 Estructura del Agente *UserAgent* ..... 45

Figura 5.4 Estructura del Agente *OntologyAgent* ..... 46

Figura 5.5 Estructura del Agente *SearchAgent* ..... 47

Figura 5.6 Estructura del Mensaje ACL para Verificar la Existencia de un Concepto .... 48

Figura 5.7 Estructura del Mensaje ACL para Guardar un Recurso ..... 49

Figura 5.8 Estructura del Mensaje ACL para Realizar una Búsqueda ..... 49

Figura 5.9 Ejemplo de Representación de los Metadatos en el Lenguaje RDF..... 50

Figura 5.10 Escenario de una Búsqueda de Metadatos de un Documento ..... 51

Figura 5.11 Pantalla del Explorador Semántico ..... 53

Figura 5.12 Pantalla de Anotaciones ..... 54

Figura 6.1 Configuración de un Escenario Centralizado de Agentes(8 Instancias del Agente UserAgent) ..... 58

Figura 6.2 Configuración de un Escenario Descentralizado de 8 Agentes (Instancias del Agente UserAgent) ..... 60

Figura 6.3 Consulta RDQL para Obtener los Metadatos de los Documentos ..... 61

Figura 7.1 Tiempo de Respuesta de las Consultas Realizadas por 8 Agentes (Instancias del Agente UserAgent) ..... 65

Figura 7.2 Representación Gráfica del Tiempo de Respuesta de las Consultas Realizadas por los Agentes en un Ambiente Distribuido..... 67

# LISTA DE APÉNDICES

**APÉNDICE A** ..... **78**

**ENCUESTA REALIZADA A LOS PROFESORES** ..... **78**

CUESTIONARIO UTILIZADO PARA LA ENCUESTA ..... 78

RESULTADOS DE LA ENCUESTA ..... 86

**APÉNDICE B** ..... **94**

**ACRÓNIMOS** ..... **94**

# CAPÍTULO 1

# INTRODUCCIÓN

El actual crecimiento de la Internet ha permitido el acceso a grandes volúmenes de información localizados en diferentes y heterogéneos sistemas. El acceso a esta información es realizado a través de exploradores (*browsers*) que usan *Hypertext Markup Language* (HTML) para mostrar la información. La principal desventaja de HTML es que no permite una adecuada estructuración de la información. De esta forma, la organización *World Wide Web Consortium* (W3C) desarrolló otras alternativas para codificar la información en las páginas Web, para darle mayor significado y reducir la ambigüedad en los recursos de información. Una de estas alternativas es el lenguaje *Resource Description Framework (*RDF) y su extensión *Resource Description Framework Schema* (RDF-S), el cual ofrece un conjunto de primitivas para la representación de modelos de conocimiento que son similares a alternativas basadas en marcos [Gomez02].

El lenguaje RDF provee la base para la interoperatividad de metadatos a través de diferentes comunidades para la descripción de recursos como las páginas Web [Iannella98]. Para describir las propiedades de los recursos y sus relaciones con otros recursos se utiliza el lenguaje RDF-S que permite definir una taxonomía de los recursos basada en clases y sus propiedades asociadas.

Estos lenguajes son importantes en la representación de los elementos de una ontología. Una ontología permite representar el conocimiento en un dominio definiendo

términos, relaciones y un conjunto de reglas. Generalmente una ontología tiene una representación jerárquica de objetos y tipos de objetos, lo cual permite representar las relaciones y propiedades de estos objetos.

El conocimiento que proveen las ontologías puede ser representado a través de los lenguajes RDF y RDF-S con los cuales se formarán los metadatos de los recursos de información, que pueden estar almacenados en diferentes computadoras de una red heterogénea. La búsqueda y recuperación de esta información puede ser realizada por los motores de búsqueda o agentes de software. En particular, los agentes presentan importantes características, tales como autonomía, pro-actividad y cooperación, entre otras, lo cual ha llevado a que la visión de los agentes sea muy convincente y muchas personas crean que se volverán necesarios a medida que la complejidad del *World Wide Web* se incremente [Hendler99]. Se espera que estos agentes ayuden a encontrar información con sólo proporcionarles algunas palabras, logrando los resultados deseados de una manera más efectiva.

En esta tesis se propone aplicar las tecnologías anteriormente mencionadas, en la descripción, representación y búsqueda de recursos educativos que son utilizados en el salón de clase. Uno de los principales problemas que se tiene en la actualidad es la carencia de herramientas que permitan acceder a este tipo de recursos. Se observa principalmente este problema por la falta de mecanismos o herramientas que permitan registrar los procesos que ocurren en el salón de clase o sus resultados en forma de informes, ensayos, asignaciones, exámenes y otros. Las herramientas que registran los procesos que ocurren en el salón de clase son mínimos, de tal forma que el acceso a experiencias anteriores y a conocimientos técnicos acumulados es extremadamente difícil o imposible. Así, la carencia de la reutilización del conocimiento puede no animar la construcción o al crecimiento del conocimiento a su capacidad máxima en un salón de clase, de esta forma, limitando las oportunidades para la innovación y animando la repetición. No podemos categóricamente indicar que no está sucediendo la innovación en

el salón de clase, sino, que la "innovación está por todas partes; la dificultad es aprender de ella" [Fischer01].

En esta investigación se diseñó una arquitectura basada en agentes que permita acceder a los recursos educativos en entornos distribuidos. Se desarrolló un prototipo llamado *Ontology Semantic Agent* (oSEMA), que permite el acceso a los recursos a través de los agentes. Y, tiene las siguientes funciones, realizar anotaciones de los recursos educativos, buscar los metadatos de los documentos que un usuario necesite y guardar los metadatos de los documentos del usuario en un servidor o localmente (en el mismo computador). *oSEMA* fue desarrollado en Java y utiliza el *middleware* llamado *Java Agent Development Framework* (JADE), para el desarrollo de los agentes y su comunicación. Un elemento importante en la arquitectura es la Ontología de Documentos, que describe a los principales documentos utilizados por profesores y estudiantes. *oSEMA* utiliza la Ontología de Documentos para realizar las anotaciones de los recursos educativos, esto permitirá darle más significado a la información que comparten e intercambian los agentes y también tener un vocabulario común para representar los documentos. En la práctica, cada estudiante podrá acceder, modificar y buscar información de sus documentos desde cualquier salón de clase. También, se tiene una manera conveniente de compartir información semánticamente representada y no simples cadenas de caracteres que en la actualidad se utiliza en muchas aplicaciones para comunicarse con otras aplicaciones.

Las ventajas que pueden traer la combinación de estas tecnologías aplicada a la recuperación y búsqueda de recursos utilizados en el salón de clase son muy significativas. Idealmente en el salón de clase los participantes involucrados en un proceso de aprendizaje construyen conocimiento [Wilson97], en donde utiliza los lenguajes semánticos para representar conocimiento y los agentes para recuperar y buscar recursos educativos que podrían apoyar el proceso de aprendizaje en el salón de clase.

## 1.1 Motivación

La educación es una actividad intensiva en conocimiento, en la cual mucho permanece tácito. Este trabajo es una aproximación a la construcción de sistemas y bases de conocimiento para uso en la educación. Se requiere que tales sistemas y bases de conocimiento sean construidos eficientemente para el almacenamiento y recuperación del conocimiento generado por estudiantes y profesores en el contexto idóneo para mejorar las habilidades de aprendizaje y el desarrollo de competencias necesarias para que los futuros profesionales estén preparados para ser exitosos en los modelos económicos modernos. Las universidades y las instituciones educativas están introduciendo la Gestión de Conocimiento en su organización. Sin embargo, los procesos complejos de colaboración que ocurren en el proceso de enseñanza-aprendizaje todavía plantean desafíos importantes para investigar, principalmente en la Gestión de Conocimiento y la Inteligencia Artificial como campo de soporte, debido a la falta de formalidad que en la mayoría de los casos encierra el aprendizaje [Vega04].

Las actuales iniciativas como *Dublín Core Metadata Initiative* (DCMI) [Beckett02], *Resource Description Framework* (RDF) [Manola04] y *Ontology Inference Layer* (OIL) [Fensel00] se están desarrollando para identificar y describir conocimientos y recursos de información. Es importante que las tecnologías de gestión de conocimiento sean investigadas, desarrolladas y aplicadas en la educación para enriquecer el entorno del aprendizaje.

Estas iniciativas son importantes para describir los diferentes recursos usados en la educación, tales como: asignaciones, ensayos, reportes y otros, que son parte del acervo de materiales de aprendizaje, cuyo contenido de conocimiento no se ha logrado compartir y acumular en forma amplia y abierta por estudiantes y profesores. De ahí la importancia y la motivación para investigar y desarrollar herramientas que permitan almacenar y recuperar información de todos estos recursos como documentos que son el resultado del proceso de enseñanza-aprendizaje, y también la descripción y la representación de los

dominios de conocimiento y su importancia en la recuperación de información en el contexto adecuado.

## 1.2 Objetivos

El principal objetivo de la investigación es el uso de las tecnologías de la Web semántica para la descripción de recursos educativos, como los documentos utilizados por estudiantes y profesores en el proceso de enseñanza-aprendizaje, y el desarrollo de una arquitectura para la búsqueda y la recuperación de tales recursos educativos en un ambiente distribuido.

Los objetivos específicos son:

- Identificar los recursos educativos y su representación a través de la tecnología de la Web semántica;
- Diseñar y desarrollar, una base de conocimiento de documentos bajo un estándar, que permita el almacenamiento, la búsqueda y la recuperación de estos documentos;
- Diseñar y desarrollar una herramienta de bajo costo que permita la recuperación de la información de los recursos educativos en un ambiente distribuido;
- Realizar pruebas para evaluar la factibilidad de desarrollar estas tecnologías en apoyo de la educación.

## 1.3 Contribuciones

Las principales contribuciones de esta tesis se resumen en lo siguiente:

- Desarrollo de una ontología de documentos, con base en los principales documentos utilizados por los estudiantes y profesores.
- Desarrollo de un vocabulario de términos para describir los documentos, el cual permitirá compartir y reutilizar información de los documentos utilizados por los estudiantes y profesores.
- El desarrollo de una arquitectura basada en agentes para la recuperación de los recursos educativos, como los documentos utilizados por estudiantes en el salón de clase, en donde estos documentos son representados a través de metadatos estructurados semánticamente.
- Desarrollo de un prototipo que permite realizar anotaciones sobre documentos utilizando la ontología de documentos y la búsqueda de metadatos de los documentos en un entorno distribuido.

## 1.4 Organización de la Tesis

Esta tesis está dividida en ocho capítulos. El Capítulo 2 da una descripción del estado del arte de las principales tecnologías relacionadas a esta investigación. El Capítulo 3 describe los trabajos previos relacionados. El Capítulo 4 describe la construcción de la ontología de documentos y la representación de los metadatos en un lenguaje semántico. En el Capítulo 5 se describe una arquitectura basada en agentes para la recuperación de los documentos. El Capítulo 6 presenta la metodología para realizar las pruebas en dos escenarios. El Capítulo 7 presenta los resultados y el Capítulo 8 presenta las conclusiones.

# CAPÍTULO 2

# ESTADO DEL ARTE

## 2.1 Introducción

En este capítulo se dará una descripción de las tecnologías que en la actualidad se utilizan en la Web semántica. En la sección 2.2 se presenta un estudio sobre las ontologías y los nuevos lenguajes de representación de conocimiento. En la sección 2.3 se describe la tecnología de agentes, la plataforma propuesta por la organización *Foundation for Intelligent Physical Agents* (FIPA) y los lenguajes de comunicación entre estos agentes.

## 2.2 Los Lenguajes Semánticos y las Ontologías

En estos últimos años, muchas organizaciones, grupos de investigación, y comunidades de Internet están desarrollando tecnologías, herramientas y lenguajes para representar el conocimiento y hacer posible compartir y re-usar este conocimiento. El *World Wide Web Consortium* (W3C) es una de las organizaciones más representativas en el desarrollo de herramientas y lenguajes para representar recursos en el Web. Los lenguajes RDF y RDF-S se utilizan para expresar las descripciones de los recursos en el Web con base en objetos relacionados u ontologías. Otros lenguajes, desarrollados por otras organizaciones, son *Darpa Agent Markup Language + Ontology Inference Layer*

*(*DAML+OIL) y *Ontology Web Language (*OWL), que permiten representaciones más sofisticadas de recursos Web. A continuación se da una descripción de estos lenguajes.

### 2.2.1 El Lenguaje RDF

*Resource Description Framework* (RDF) es uno de los lenguajes desarrollados por el *World Wide Web Consortium* (W3C), que permite la representación de metadatos de los recursos de Internet [Lassila99], el cual facilitará la interoperatibilidad entre aplicaciones que intercambian información en forma entendible entre las máquinas (*machine-understandable*). Este lenguaje puede ser utilizado en diferentes aplicaciones, tales como: búsquedas, recuperación de recursos, catalogación, bibliotecas digitales y agentes inteligentes.

RDF tiene un modelo de datos, el cual nos permite representar expresiones basadas en un modelo formado por tres elementos:

- Recursos: todos los objetos que pueden ser expresados con expresiones RDF son llamados recursos; por ejemplo, una página Web o parte de una página Web; también puede ser un conjunto de páginas Web o un documento.

- Propiedades: es un aspecto específico, atributo o relación que permite describir un recurso.

- Expresión: un recurso específico junto con el nombre de una propiedad, más el valor de ésta formarán una expresión RDF. Estos tres elementos que forman una expresión son llamados, también: sujeto, predicado y objeto

En el siguiente ejemplo, se identifican los tres componentes de una expresión RDF y su representación en el lenguaje RDF.

Juan Rojas es el autor de la tesis http://www.ece.uprm.edu/~juanr/tesis.pdf

Sus partes se muestran en la Tabla 2-1.

Tabla 2-1 Partes de una Expresión

| | |
|---|---|
| Sujeto(Recurso) | http://www.ece.uprm.edu/~juanr/tesis.pdf |
| Predicado(Propiedad) | Autor |
| Objeto(Literal) | “Juan Rojas” |

Esta expresión, también se puede representar gráficamente, en donde los arcos representan las propiedades, los recursos son representados por nodos circulares y los nodos rectangulares representaran los literales (que vienen a ser los valores de la propiedad) como se muestra en la Figura 2.1.

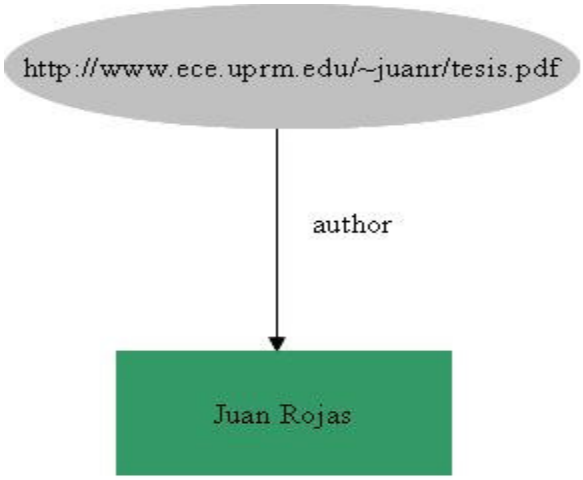


Figura 2.1 Grafo de una Expresión RDF

RDF provee un conjunto de términos para representar una expresión; así la expresión anterior se muestra en la Figura 2.2

```
<?xml version="1.0">
<rdf:RDF>
   <rdf:Description about="http://www.ece.uprm.edu/~juanr/tesis.pdf">
         <s:autor>Juan Rojas</s:autor>
   </rdf:Description>
</rdf>
```

Figura 2.2 Expresión RDF

### 2.2.2 RDF-S

*Resource Description Framework Schema* [Broekstra01] provee mecanismos para definir un vocabulario para datos RDF. A través de RDF-S se podrán definir atributos, los

cuales identificarán las características de los recursos, y las relaciones entre los recursos. Se podrán definir las clases de recursos que serán usados, restringir posibles combinaciones de clases, relaciones y detectar violaciones a las restricciones.

RDF-S tiene un conjunto de términos los cuales nos permitirán construir las expresiones RDF-S válidas. Así, se tienen los términos *Class*, *subClassOf* y *Property* que nos permitirán expresar expresiones acerca de los recursos (clases), representar sus propiedades y a la vez representar una jerarquía de recursos. Los objetos podrán ser instanciados a partir de las clases usando la propiedad *type*. Las restricciones de las propiedades podrán ser especificadas usando *domain* y *range*. A continuación, se muestra un ejemplo en la Figura 2.3

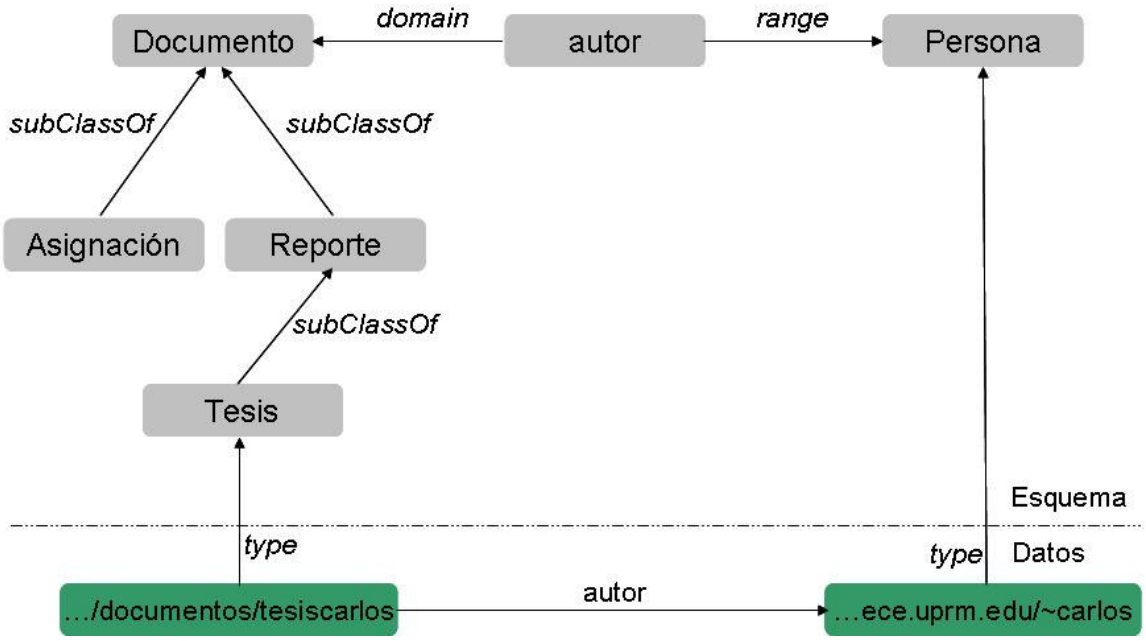


Figura 2.3 Ejemplo de una RDF Schema

En el ejemplo de la figura anterior se muestra un grafo RDF-S en el cual se define un vocabulario: *Documento*, *Asignación*, *Reporte*, *Tesis* y *Persona* que son definidos como clases y *autor* que es definido como una propiedad. También, se identifican ejemplos de instancias como *"…documentos/tesiscarlos*" y *"…ece.uprm.edu/~carlos"*.

### 2.2.3 Ontology Inference Layer

*Ontology Inference Layer* (OIL) es uno de los lenguajes que permite expresar y representar ontologías. OIL provee un modelado de primitivas que son comúnmente utilizadas en alternativas basadas en *frames* para la ingeniería de ontologías (conceptos, taxonomías de conceptos, relaciones y otros), la semántica formal y el soporte a razonamiento [Gomez02]. OIL unifica tres criterios importantes provistos por diferentes comunidades:

- la semántica formal y el soporte eficiente para razonamiento, como las provistas en descripciones lógicas;
- el enriquecimiento epistemológico de las primitivas, como el provisto por los *frames*; y
- una propuesta estándar para notaciones de intercambio sintáctico como la provista por la comunidad del Internet. [Gomez02][Fensel01].

OIL es un lenguaje que permite expresar ontologías, el cual consiste de una lista de definiciones de clases *class-def,* definiciones de expresiones *class-expression* que pueden estar conformados por un nombre de clase, una restricción o una combinación lógica de clases. Una restricción *slot-constraint* especificará una lista de una o más restricciones aplicadas al *slot,* y por último se tiene una definición *slot-def* que es asociada a un nombre y a una descripción del *slot*.

### 2.2.4 Darpa Agent Markup Language + OIL

Se requieren lenguajes que permitan expresar con más detalle y claridad los recursos del Web, y que tales descripciones sean fácilmente usadas en el razonamiento automático. Por tal motivo, se desarrolló OIL y luego DAML+OIL, lenguajes

ontológicos que extienden RDF-S con un rico conjunto de primitivas. Dentro de las principales características que presenta este lenguaje se tienen: [Sean01]

- Una bien definida semántica y una clara comprensión de las propiedades del lenguaje, en donde las descripciones lógicas proveen a DAML+OIL con la habilidad y la flexibilidad de componer clases y propiedades para formar nuevas expresiones.

- Una sintaxis entendible por las máquinas codificadas en lenguajes del Web. DAML+OIL es una extensión de RDF-S, haciendo que las ontologías construidas en DAML+OIL sean accesibles a cualquier aplicación basada en RDF-S.

- Una arquitectura por capas, evitando colocar cualquier objeto dentro del núcleo del lenguaje.

DAML+OIL es un importante lenguaje desarrollado por DARPA y *OntoKnowledge*, el cual presenta características muy importantes para la construcción de ontologías, la captura de conocimiento y el razonamiento automático. En la Figura 2.4 se muestra una parte de la representación en DAML+OIL de la ontología mostrada en la Figura 2.6.

```
<daml:Class rdf:ID=”Persona”/>
<daml:Class rdf:ID=”Profesor”>
   <rdfs:subClassOf rdf:resource=”#Persona”/>
</daml:Class>
<daml:Class rdf:ID=”Profesor_Asociado’>
   <rdfs:subClassOf rdf:resource=”#Profesor”/>
</daml:Class>
<daml:Class rdf:ID=”Profesor_Asistente”>
   <rdfs:subClassOf rdf:resource=”#Profesor”/>
</daml:Class>
<daml:Class rdf:ID=”Instructor”>
   <rdfs:subClassOf rdf:resource=”#Profesor”/>
</daml:Class>
<daml:Class rdf:ID=”Estudiante”>
   <rdfs:subClassOf rdf:resource=”#Persona”/>
</daml:Class>
<daml:Class rdf:ID=”Graduado”>
   <rdfs:subClassOf rdf:resource=”#Estudiante”/>
</daml:Class>
<daml:Class rdf:ID=”Subgraduado”>
   <rdfs:subClassOf rdf:resource=”#Estudiante”/>
</daml:Class>
```

Figura 2.4 Representación de una Ontología en DAML+OIL

### 2.2.5 Ontology Web Language

OWL es otro de los lenguajes recomendados por la organización W3C que incorpora lecciones aprendidas en el diseño y aplicación del lenguaje DAML+OIL. AL igual que los anteriores lenguajes es usado cuando la información contenida en los documentos necesita ser procesada por las aplicaciones. Este lenguaje provee un mayor conjunto de primitivas para representar el significado de los elementos y sus relaciones con otros elementos en una ontología.

*Ontology Web Language* está divido en tres sub-lenguajes [McGuinness04]: *OWL Lite*, en donde los usuarios pueden realizar clasificaciones jerárquicas y aplicar simples restricciones. *OWL DL* permite dar una mayor expresividad a los elementos de una ontología y además incluye todas las construcciones del lenguaje OWL. Y por último, el sub-lenguaje *OWL Full* permite a los usuarios dar una mayor expresividad a sus sentencias y tienen la libertad de usar la sintaxis en RDF. En la Figura 2.5 se muestra un ejemplo.

```
<owl:Class rdf:ID=”Persona”/>
<owl:Class rdf:ID=”Profesor”>
   <rdfs:subClassOf rdf:resource=”#Persona”/>
</owl:Class>
<owl:Class rdf:ID=”Profesor_Asociado’>
   <rdfs:subClassOf rdf:resource=”#Profesor”/>
</owl:Class>
<owl:Class rdf:ID=”Profesor_Asistente”>
   <rdfs:subClassOf rdf:resource=”#Profesor”/>
</owl:Class>
<owl:Class rdf:ID=”Instructor”>
   <rdfs:subClassOf rdf:resource=”#Profesor”/>
</owl:Class>
```

Figura 2.5 Representación de una Ontología en OWL

### 2.2.6 Ontologías en la representación del conocimiento

En estos últimos años se incrementó la cantidad de información en la Internet, lo cual motivó el incremento de la complejidad y la diversidad de la información. Esto, también motivó el interés de automatizar muchas actividades que tradicionalmente se realizan en forma manual [Baclawski01]. La principal motivación en el uso de las ontologías es que permiten compartir y reutilizar conocimiento en un dominio [Duineveld00]. De ahí el

interés de muchos investigadores en el área de inteligencia artificial para tratar de aplicar las ontologías en la representación de conocimiento.

Las ontologías son un elemento importante en la Web Semántica, y es un término filosófico que trata sobre la naturaleza de la existencia y los tipos de objetos que existen [Berners01]. También, la ontología se puede definir como una especificación explícita de una conceptualización compartida [Duineveld00]. La *conceptualización* se refiere a un modelo abstracto de un fenómeno en el mundo, en donde se tienen que identificar los conceptos relevantes de este fenómeno. El término *explícito* significa qué tipos de conceptos son usados y las restricciones de estos conceptos. Y, por último, el término *compartido* se refiere a que las ontologías capturan conocimiento que es aceptado por todos y no es individual de un grupo.

Actualmente, se les está dando más uso a las ontologías en el campo de la Inteligencia Artificial. En este campo se define la ontología como una descripción formal y explícita de conceptos de un dominio (*classes,* a las que nos referiremos como conceptos), propiedades de cada concepto, las cuales describen las características y atributos de los conceptos (*slots* llamados también como propiedades o roles) y restricciones de las propiedades (*facets* conocidas como restricciones) [Fridman01]. La Figura 2.6 muestra la ontología acerca de los profesores y estudiantes.

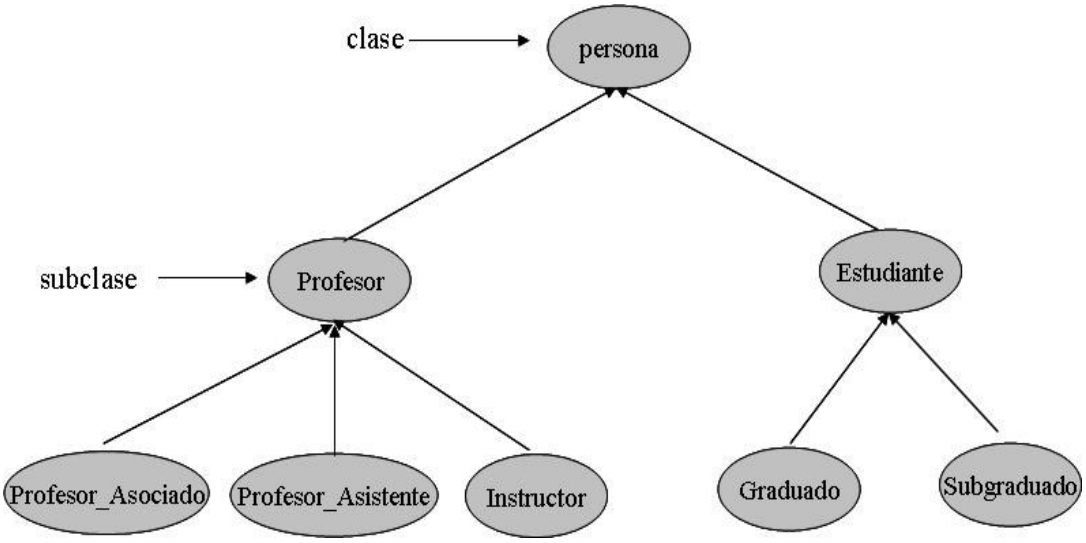


Figura 2.6 Representación de una Ontología que Describe a Profesores y Estudiantes

La información que es representada a través de los lenguajes semánticos, puede ser compartida entre aplicaciones como los agentes. En la siguiente sección se da una descripción del paradigma de la tecnología de agentes.

## 2.3 Agentes

Se tienen muchas definiciones acerca de agentes. La definición de agente dada por Hermans [Hermans00] en su trabajo de agentes inteligentes, se refiere a una pieza de *software* la cual realiza una tarea dada usando información de su entorno y que actúa de manera conveniente hasta completar dicha tarea. El agente debería ser capaz de adaptarse a base de los cambios ocurridos en su entorno.

También, un agente se define [Nwana96] como un componente de *software* el cual es capaz de realizar ciertas tareas en representación de un usuario. Los agentes son autónomos y se comportan según los objetivos que van alcanzar, reaccionando a eventos externos, comunicándose y colaborando con otros agentes para realizar una tarea determinada.

Tratar de definir un agente de manera uniforme es bastante difícil. Otra forma aceptable para definir un agente es en función a las propiedades que lo caracterizan, como:

- Autonomía: el agente es capaz de trabajar sin intervención humana o de otros agentes;
- Sociabilidad: el agente puede interactuar con otro agente a través de un lenguaje que pueda entender;
- Reactividad: el agente puede percibir eventos de su entorno y reaccionar a esos eventos;
- Pro-actividad: el agente puede realizar acciones por iniciativa propia en función de sus objetivos;
- Movilidad: el agente posee la capacidad de moverse de una computadora a otra a través de una red de computadoras;
- Veracidad: se asume que todo agente no comunica información falsa a propósito;
- Benevolencia: se asume que un agente está dispuesto a ayudar a otro agente, si esto, no entra en conflicto con sus objetivos;
- Racionalidad: el agente actúa de forma racional, para cumplir con sus objetivos.

### 2.3.1 Plataforma de Agentes

Para el funcionamiento adecuado de los agentes, se tiene que proveer los servicios y los componentes necesarios para su funcionamiento. La organización *Foundation for Intelligent Physical Agents* (FIPA) [FIPA96] se encarga de estandarizar aspectos relacionados con la tecnología de agentes y de sistemas multi-agentes. Uno de los

estándares desarrollados por esta organización es el *FIPA Agent Management Specification* [FIPA02a] en el que se describe el modelo de referencia FIPA para la plataforma de agentes y la funcionalidad de cada uno de sus componentes (ver Figura 2.7).

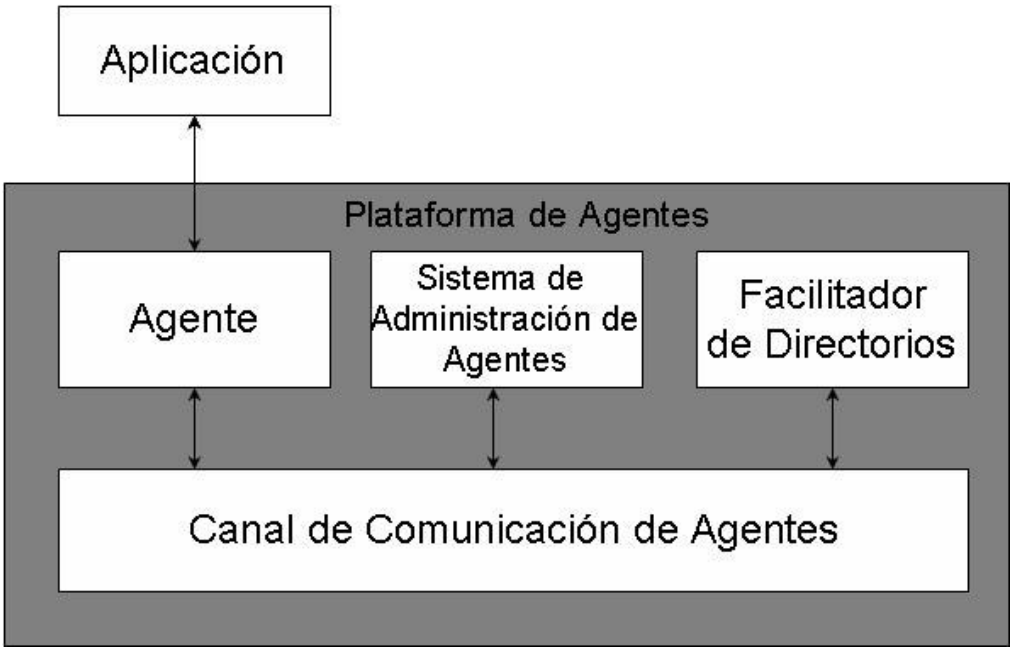


Figura 2.7 Arquitectura de la Plataforma de Agentes Propuesta por FIPA [FIPA02a]

La plataforma de agentes consiste de los siguientes componentes:

**Sistema de Administración de Agentes**: controla el acceso y el uso de la plataforma de agentes, se encarga de gestionar el ciclo de vida de los agentes, de los recursos locales y de los canales de comunicación y proporciona el servicio de páginas blancas que permite localizar a un agente por su nombre.

**Facilitador de Directorios**: provee el servicio de páginas amarillas, que permite localizar a los agentes por sus capacidades y no por su nombre.

**Canal de Comunicación de Agentes**: se encarga del control de todos los mensajes que se envían entre los agentes, incluyendo los mensajes hacia y desde las plataformas remotas.

### 2.3.2 Lenguajes de Comunicación entre Agentes

La comunicación es uno de los aspectos importantes dentro de una comunidad de agentes, la cual permitirá compartir y enviar mensajes entre agentes. Los métodos tradicionales permiten la comunicación entre los agentes, pero éstos no son suficientes para lograr un comportamiento social entre los agentes; para ello es necesario que los mensajes tengan un significado y contenido semántico. Se desarrollaron varios lenguajes los cuales son usados para el envío de mensajes, pero los más importantes se dan a conocer a continuación.

#### 2.3.2.1 KQML

*Knowledge Query and Manipulation Language* (KQML) es uno de los lenguajes que nos provee la estructura para enviar mensajes entre agentes. Dentro de sus principales características tenemos que es independiente de la sintaxis, del contenido y la ontología que se utilizará. Así mismo, es independiente del mecanismo de transporte (TCP/IP, SMTP, IIOP, etc.) e independiente del contenido del lenguaje (KIF, SQL, STEP, Prolog, etc.).

KQML fue desarrollado por el consorcio *DARPA Knowledge Sharing Effort* (KSE) [Ramesh93] cuyo principal objetivo fue desarrollar una base tecnológica para construir en forma progresiva sistemas cada vez con más funcionalidad, complejidad y amplitud.

KQML provee un lenguaje y un protocolo para intercambiar información entre agentes inteligentes, y para lograr este intercambio se tiene que compartir una sintaxis y una semántica común para que los mensajes sean entendidos por los agentes.

Al lenguaje KQML se le dividió en tres capas [Finin94] como se muestra en la Figura 2.8. La capa de contenido alberga el contenido del mensaje, que es especificado en cualquier formato ya sea texto o binario. La capa de comunicación contiene parámetros como el identificador único asociado a la comunicación, identidad del remitente e identidad del destinatario. La capa de mensaje define tanto los diferentes tipos de interacción que se pueden establecer entre los agentes, como los protocolos que se pueden establecer entre los agentes. También, incluye otras características como el lenguaje de representación o la ontología, siendo ambos opcionales.

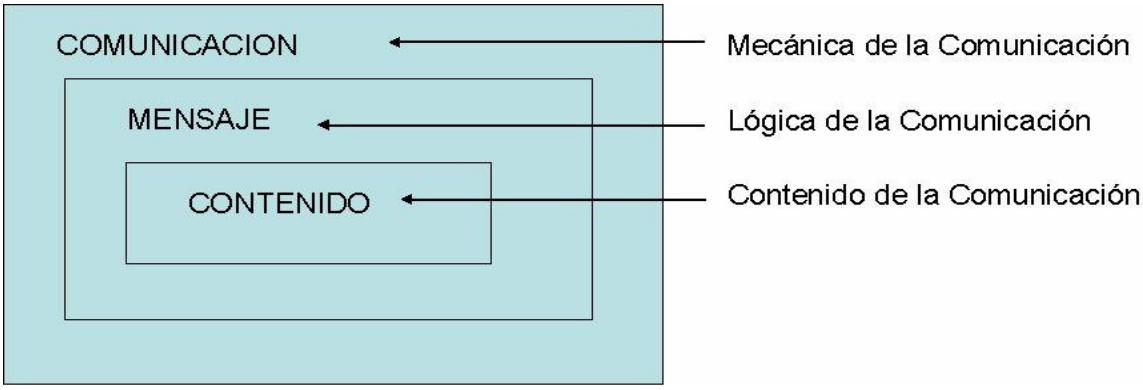


Figura 2.8 Capas del Lenguaje KQML

#### 2.3.2.2 ACL-FIPA

La organización internacional FIPA desarrolló un lenguaje estandarizado para la comunicación de agentes denominado *Agent Comunicación Language FIPA* (ACL FIPA) [FIPA02c], cuyo principal objetivo fue darle un sentido semántico a los mensajes que intercambian los agentes. Para lograrlo se basaron en los *actos comunicativos*, los cuales se basan principalmente en la solicitud de una acción a un agente.

Para la realización de una acción, el agente A envía un mensaje *request* al agente B; el receptor del mensaje podría rechazar o aceptar la acción a través de un mensaje *accept*

o *refuse* respectivamente. Si el agente B aceptara la petición tiene que notificarlo e indicarle al agente A cuando finalice la realización de la acción a través del mensaje *agree*. Si en el proceso de realización de una acción se produce un fallo, se notificará al agente A mediante el mensaje *failure*.

En el párrafo anterior se menciona un típico intercambio de mensajes entre dos agentes para la realización de una acción, en donde se hace uso de algunos actos comunicativos como *request* y *refuse*. Éstos son parte importante en la estructura de un mensaje ACL FIPA.

La estructura de un mensaje ACL FIPA está formada por varios parámetros necesarios para la comunicación de los agentes. Un mensaje se compone de un identificador, indicando el tipo del acto comunicativo y también, incluye un conjunto de parámetros de la forma clave-valor que es la información necesaria para la realización de una acción. La estructura del mensaje ACL FIPA [FIPA02c] se muestra en la Figura 2.9.

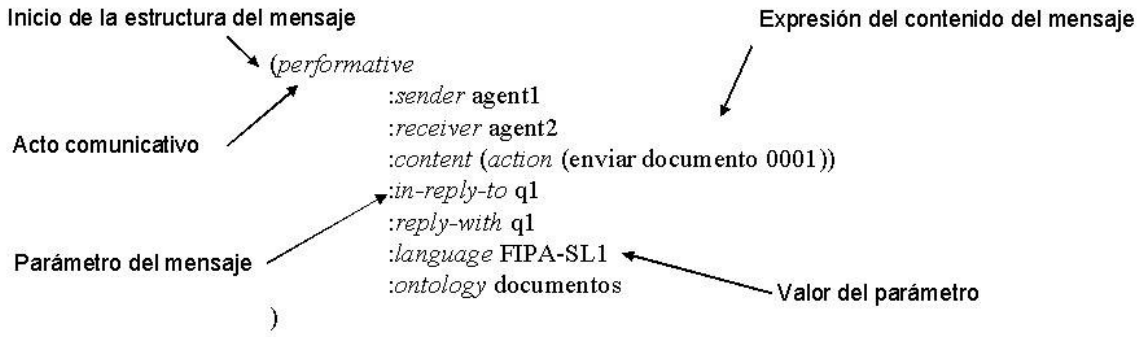


Figura 2.9 Estructura del Mensaje ACL

En la palabra *performative* se especifica el tipo de acto comunicativo que identificará al mensaje que se esté enviando. FIPA identificó varios actos comunicativos [FIPA02b] como: *Accept Proposal, Agree, Cancel, Call for Proposal, Confirm, Failure, Propose, Proxy, Query If, Query Ref, Refuse* y *Request.*

El parámetro *sender* identificará al agente que origina el mensaje, en tanto el parámetro *receiver* especifica el agente que recibirá el mensaje. En *content* se especifica el contenido del mensaje y en los parámetros *reply-with* y *in-reply*-to se especifican las etiquetas que irán junto al mensaje. El parámetro *language* contiene el lenguaje con el que se expresará el contenido del parámetro *content*. El parámetro *ontology* especifica el vocabulario que se usará en los mensajes.

En este capítulo se abarcó dos temas importantes que son utilizados en el desarrollo de esta tesis, como son los lenguajes semánticos en la representación de ontologías y el paradigma de agentes. En los siguientes capítulos se describirá la aplicación de estas tecnologías en la tesis.

# CAPÍTULO 3

# TRABAJOS RELACIONADOS

## 3.1 Introducción

En este capítulo se presenta una síntesis de algunos trabajos realizados sobre agentes en la recuperación de recursos en entornos distribuidos, así como las herramientas para la representación de conocimiento a través de los lenguajes de ontología.

## 3.2 KBS-Hyperbook

*KBS-Hyperbook* [Dhraief01] propone un repositorio de aprendizaje abierto, en donde una de las características principales es el uso del estándar RDF/RDFS como un lenguaje de modelado para expresar información acerca de los diferentes objetos de aprendizaje que están contenidos en el repositorio, y también de cómo estos objetos de aprendizaje están relacionados. En el repositorio se almacenan metadatos que representan información acerca de la estructura y la ruta de acceso en un curso particular. Los *Uniform Resource Locator* (URL) son utilizados como identificadores para elementos simples (módulos, unidades de curso y sub-unidades) y otros metadatos que identifiquen el contenido del curso

Adicionalmente, se desarrollo una variante a RDF llamada *O-Telos-RDF* [Nejdl01], el cual provee funcionalidades mejoradas para el modelado de metadatos y el

refinamiento de sentencias. *O-Telos-RDF*, permite representar conceptos a través de una red semántica la cual consiste de nodos y arcos. Los nodos representan todos los conceptos (como clases, meta-clases, etc.) e individuales (como páginas específicas y literales) y los arcos representan relaciones específicas (propiedades) donde los arcos y los nodos son etiquetados con valores atómicos o URI como nombres.

## 3.3 Una Herramienta para la Captura de Apuntes

Venkatsubramani S. y Raman RKVS en su trabajo [Venkatsubramani02] proponen una herramienta para la captura de apuntes en las páginas de Internet, en donde resaltan la importancia del uso de las tecnologías de la Web semántica para capturar comentarios, apuntes, explicaciones, preguntas, referencias, ejemplos, correcciones y otros. Un elemento importante son los metadatos que proveen información adicional de una pieza de dato existente y para su representación utilizan el estándar RDF. Esta herramienta presenta los siguientes tres componentes:

- Una herramienta para la captura de los apuntes.
- Una herramienta para el desarrollo de ontologías
- Un repositorio para los metadatos y un motor de inferencias para consultas.

Además, permite almacenar los apuntes bajo una jerarquía de temas que es flexible para los usuarios, permitiendo adicionar un tema en el caso de no encontrar uno. Todos estos apuntes pueden ser almacenados localmente o en un servidor central y luego pueden ser recuperados por los usuarios.

## *3.4 Entorno de Trabajo para Anotaciones:* OntoAnnotate

La investigación presentada en [Staab01], resalta el uso de anotaciones como técnica para poner datos en la Web y que sean comprendidas por las máquinas. Los autores de esta investigación profundizaron en dar apoyo a la representación de metadatos enlazados semánticamente; por tal motivo desarrollaron una herramienta que tiene la capacidad de manipular ontologías y capaz de adaptarse a una determinada ontología.

Como principal lenguaje para las anotaciones semánticas y las ontologías usaron RDF y *RDF Schema*, donde aprovecharon las primitivas de estos estándares para definir los metadatos, sus relaciones y sus propiedades. Los autores de esta investigación identificaron 4 categorías semánticas:

- **Identificación de un Objeto**: Los nuevos objetos son creados con la confirmación de que existe un objeto con un único identificador.

- **Relación Objeto-Clase**: Cada objeto es asignado a una clase de objetos por una persona.

- **Relación Atributo-Objeto**: Cada objeto puede ser relacionado a un valor a través de un atributo.

- **Relación Objeto-Objeto**: Cada objeto puede estar relacionado a todos los objetos existentes (incluso a sí mismo) vía una relación.

Esta investigación presenta una alternativa para la creación de metadatos. *OntoAnnotate* es un ambiente de trabajo que provee las herramientas para crear, buscar y extraer los metadatos.

## 3.5 Hyperclip

En el trabajo presentado en [Sato02] se propone una herramienta que permite compartir y juntar metadatos que representan información acerca del contenido electrónico para ayudar al trabajo intelectual. A los metadatos que ellos utilizan lo denominaron *context*, lo cual a su vez está formado por referencias a otros contenidos. Ellos proponen que los metadatos representen el conocimiento referente a diferentes actividades que realizan los usuarios, tales como: creación, referencia y revisión de documentos. Para representar los metadatos utilizan los lenguajes RDF o XLink.

Adicionalmente, presentan una arquitectura para compartir los contenidos entre los usuarios para lo cual utilizaron el modelo de comunicación *peer-to-peer*, y les permitió una mejor comprensión de cómo pueden ser reutilizados los contenidos.

## 3.6 MANTHA

En este trabajo [Roberto99] se propone una arquitectura para la redacción, publicación y almacenamiento de documentos en la Internet, donde se enfocan principalmente en la administración y el procesamiento de documentos que están estructurados con colecciones de multimedia. Para describir cada uno de los elementos del documento utilizan el lenguaje semántico RDF, el cual permitirá una manipulación uniforme. También, desarrollaron una ontología, donde representaron la estructura del documento con el que se formará un vocabulario común para compartir.

Para la manipulación de estos documentos diseñaron una arquitectura basada en agentes, en el cual se utilizó Java en su desarrollo y KQML para el intercambio de mensajes entre los agentes. En esta arquitectura se identifican dos espacios: el espacio de la federación conformado por todos los agentes que se encargan de proveer los servicios

y el espacio del usuario donde los agentes proveen a los usuarios las facilidades para acceder a la federación.

Este capítulo describió algunos trabajos donde resaltan el uso de lenguajes semánticos para representar metadatos y trabajos que utilizan los agentes en el desarrollo de sus aplicaciones. En el siguiente capítulo se presentarán los aspectos y criterios utilizados para la representación del conocimiento en este trabajo.

# CAPÍTULO 4

# REPRESENTACIÓN DE LOS METADATOS

## *4.1 Introducción*

A los metadatos se les define como información acerca de la información; de ahí su importancia para describir objetos, documentos, páginas Web y bibliotecas digitales. Los metadatos pueden ser útiles para [Steinacker01]:

- resumir el significado de los datos;
- permitir a los usuarios la búsqueda de datos;
- permitir a los usuarios determinar si los datos son los que necesita;
- dar información que afecte el uso de los datos; e
- indicar relaciones con otros recursos.

Los metadatos son datos altamente estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos. Así, por ejemplo existen metadatos en los documentos como: reportes, informes y proyectos, utilizados por los estudiantes y los profesores. En este capítulo se da una descripción de estos metadatos, su representación a través de uno de los lenguajes de representación semántica y la importancia de utilizarlos en la construcción de ontologías.

## *4.2 Identificación de los Metadatos*

Los metadatos que se identificaron están basados en los documentos utilizados por los estudiantes y los profesores. De ahí se tomó como caso de estudio los documentos manejados por profesores y estudiantes del Departamento de Ingeniería Eléctrica y de Computadoras de la Universidad de Puerto Rico, Recinto Universitario de Mayaguez. La identificación de estos documentos se realizó a través de encuestas realizadas a profesores.

### 4.2.1 Identificación de los documentos

Se realizó una encuesta para determinar los documentos utilizados por los profesores en sus sesiones de clases (ver Apéndice A). Es importante destacar que a través de las encuestas se capturó información muy importante y, a partir de ésta los metadatos a utilizar.

Con las encuestas se buscó identificar lo siguiente:

- Documentos que utilizan los profesores en sus sesiones de clases
- Frecuencia de uso de estos documentos
- Técnicas y recursos que utilizan en sus sesiones de clases
- Fuentes de información

Estos cuatro elementos fueron los principales puntos que se tomaron en cuenta para la elaboración de las encuestas. Con la información obtenida se empezó a identificar los datos que pudieran representar el dominio de documentos utilizados por profesores y estudiantes.

### 4.2.2 Principales Datos Identificados

Una vez realizada la encuesta se procedió a identificar los datos más resaltantes, y a partir de esta información tener un mejor conocimiento sobre los documentos que se utilizan. Así se identificaron los siguientes documentos:

- asignación de lectura;
- asignación de trabajo;
- informe;
- artículo;
- informe de laboratorio;
- proyecto;
- reporte final de proyecto;
- propuesta de tesis;
- informe final de tesis;
- apuntes del profesor;
- apuntes del estudiante; y
- exámenes.

En cada uno de estos documentos se identificaron los siguientes datos que están presentes en cada uno de ellos:

- título;

- autor (res); y
- tema.

## 4.3 Construcción de la Ontología: La Ontología de Documentos

La construcción de una ontología permitió definir un vocabulario común, en el cual se basa la organización de los documentos. Además, incluye definiciones básicas de conceptos, relaciones y atributos. Las ventajas que presenta el desarrollar una ontología son las siguientes:

- Compartir una estructura común de información entre las personas y los agentes de *software*
- Permitir la reutilización del conocimiento
- Representar el conocimiento en forma clara y explícita
- Analizar el conocimiento del dominio

### 4.3.1 Las clases y la Jerarquía de Clases

Con los datos identificados se empezó a determinar las clases (*class*) y sus respectivas propiedades (*slots*). Una vez identificadas las clases (las clases fueron descritas en el idioma inglés), se empezó a construir la jerarquía de clases. Se usó la metodología *top-down* para construir la jerarquía de clases, en donde se empezó con el concepto más general en el dominio y se continuó luego con una subsiguiente especialización de clases.

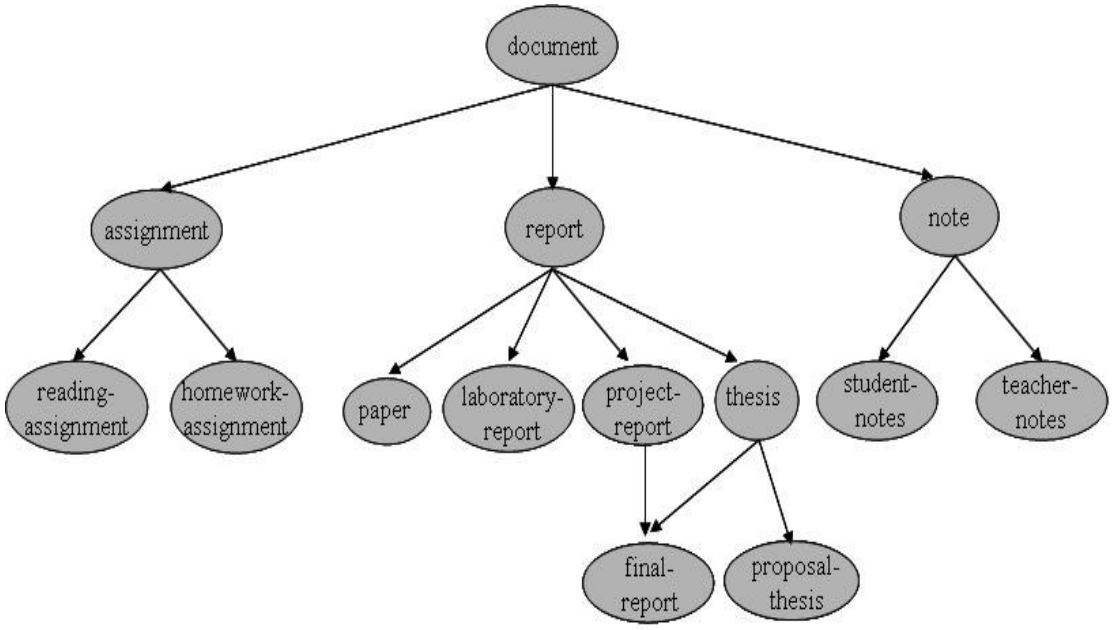


Figura 4.1 Representación de la Ontología de Documentos

De esta forma se tomó a la clase *document* como raíz de la jerarquía, ya que este término puede identificar a cualquiera de los documentos en forma general y luego se formaron las subclases: *assignment*, *report* y *note*. Se pudo también categorizar otras clases como *paper*, *laboratory-report*, *project-report*, *thesis* como subclases de la clase *report*. En la Figura 4.1 se muestra la ontología que representa parte del dominio de documentos. Con base en esta ontología se construirá el vocabulario de documentos que se utilizara en nuestra investigación.

### 4.3.2 Las Propiedades de las Clases

Con sólo la identificación de las clases no es suficiente para proveer suficiente información acerca del dominio y el alcance de la ontología. Por ello, una vez definidas las clases, se debe describir la estructura interna de las clases.

Se describen dos propiedades (*slots*) principales que permitirán describir a los documentos: *title* y *author*. Adicionalmente, se identificaron las restricciones

(*constraints*) que permitan identificar el tipo de valor (*type*), los valores permitidos (*range*), la clase a la que pertenece la propiedad (*domain*), el número de valores (*cardinality*) de cada propiedad.

Tabla 4-1 Descripción de las Propiedades de la Clase *Document*

| | *title* | *author* |
|---|---|---|
| Cardinalidad (*cardinality*) | 1 | *multiple*(0..*) |
| Tipo (*type*) | *String* | *person* |
| Rango (*range*) | *String* | *person* |
| Dominio (*domain*) | *Document* | *document* |

En la Tabla 4-1 se muestran las propiedades de la clase *document*. En la propiedad *title* se ingresará una cadena de caracteres que describirá el título del documento. En la propiedad *author* se describe el autor(es) del documento, esta propiedad es de tipo *person*, en el que se podrán identificar algunas propiedades del autor como el nombre (*name)* y el correo electrónico (*mbox).*

## *4.4 Edición de la Ontología con* Protégé

Protégé [Noy01] es un entorno para la edición de ontologías y la adquisición de conocimiento. Este es un programa *open-source* desarrollado totalmente en el lenguaje Java. Este programa presenta una pantalla gráfica que permite a los desarrolladores concentrarse en el modelo conceptual, permitiendo crear los diferentes elementos que forman parte de una ontología, los cuales son: *class*, *slot*, *facet* y otros. Con esta

herramienta se editó la ontología de documentos, la cual permitió definir la estructura de los metadatos. En la Figura 4.2 se muestra la ontología diseñada con esta herramienta.

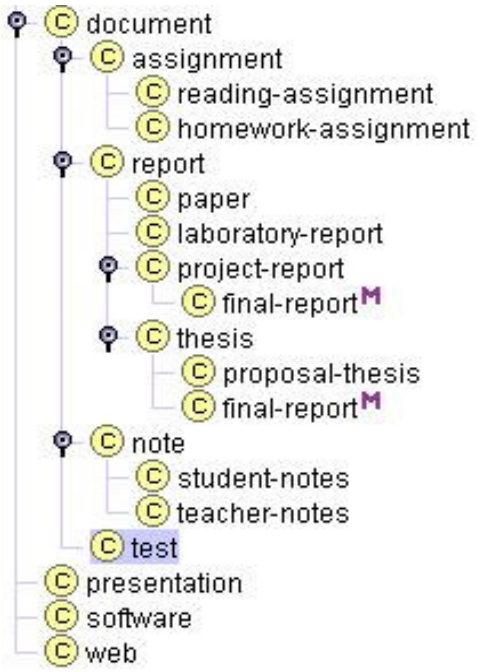


Figura 4.2 Representación de la Ontología en *Protege*

En la ontología se muestra la clase *document* como raíz de la jerarquía y también se ingresaron las subclases *assignment*, *report* y *note*. Es necesario que la clase *final-report* sea una subclase de las clases *thesis* y *project-report* porque en ambos casos se presenta un reporte final, tanto de una tesis como de otro tipo de proyecto. Protégé permite agregar una clase existente a otra (ver Figura 4.2).

Las propiedades *title* y *author* de la clase *document* se representaron utilizando Protégé, estas propiedades son heredadas por las subclases de *document*.

## 4.5 Representando la Ontología de Documentos en el Lenguaje RDF-S

El lenguaje semántico *Resource Description Framework Schema* (RDF-S) provee un conjunto de primitivas para describir los elementos de una ontología como clases, propiedades y relaciones. Con RDF-S se construyó un vocabulario base, el cual permitió describir diferentes documentos a través de expresiones RDF. En estas expresiones se representan las propiedades (*title*, *description*, *url* y otras) de los documentos, y las relaciones (*hasConcept*) usadas para describir otros recursos como un concepto (*Knowledge*, *KnowledgeBase*, etc.).

Las expresiones RDF están formadas por tres elementos *sujeto-predicado-objeto*. De esta forma se tiene la siguiente expresión:

"A. Rojas es autor del artículo titulado *Knowledge Management for Beginners*"

En esta expresión se identifica como objeto a *A. Rojas*, como predicados a *autor* y a *título*, y como sujeto se utiliza "*#articulo01*" que es un identificador único para este recurso.

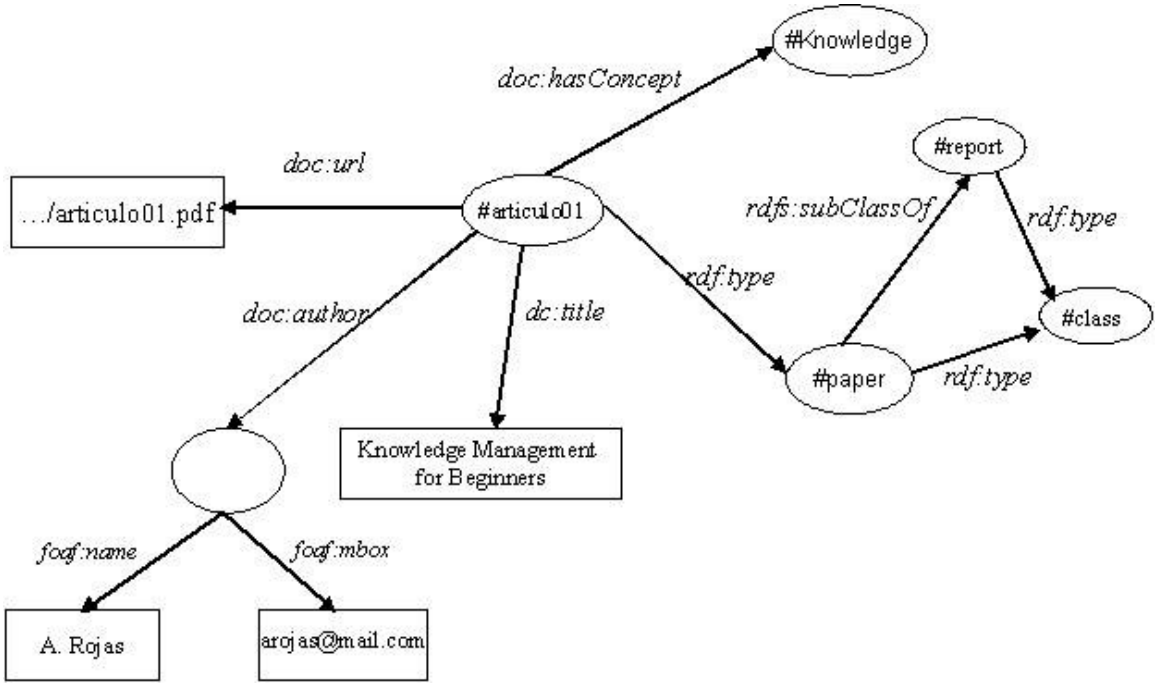


Figura 4.3 Representación en Grafos RDF de una Expresión

En la Figura 4.3 se muestra un grafo RDF en donde se representa información sobre un documento; en esta expresión se utiliza el *namespace doc* en el que se definió el vocabulario común que se utiliza para describir estos documentos. En este ejemplo, se tiene un documento que tiene un identificador (*#articulo01*), un título y un enlace al documento original. También, se tiene información acerca del autor como su nombre y su correo electrónico; el autor es representado a través de un nodo anónimo, por que no se tiene un identificador. Adicionalmente se tiene definido el tipo documento representado por el nodo *#paper* y el concepto *Knowledge* para describir al documento. La descripción en el lenguaje RDF para este ejemplo se muestra en la Figura 4.4.

```
<rdf:Description rdf:about="#article01">
<rdf:type rdf:resource="#paper"/>
    <dc:title>Knowledge Management for
Beginners</dc:title>
<doc:url>www…../article01.pdf</doc:url>
<doc:hasConcept rdf:resource="#Knowledge"/>
<author>
    <foaf:person>
        <foaf:name>A. Rojas</foaf:name>
        <foaf:mbox rdf:resource="mailto:arojas@mail.com" />
    </foaf:person>
</author>
```

Figura 4.4 Descripción de los Datos de un Documento en RDF

En la descripción del documento se utiliza *rdf:type* para definir el tipo de documento, para representar un concepto que identifique al documento se utiliza *doc:hasConcept* y se utiliza *Friend Of A Friend* (FOAF) [Brickley03] para describir los autores del documento. FOAF es un vocabulario para describir personas, simples relaciones entre personas y algunas propiedades. Se utilizó *foaf:name* para representar el nombre del autor y *foaf:mbox* para representar el correo electrónico del autor.

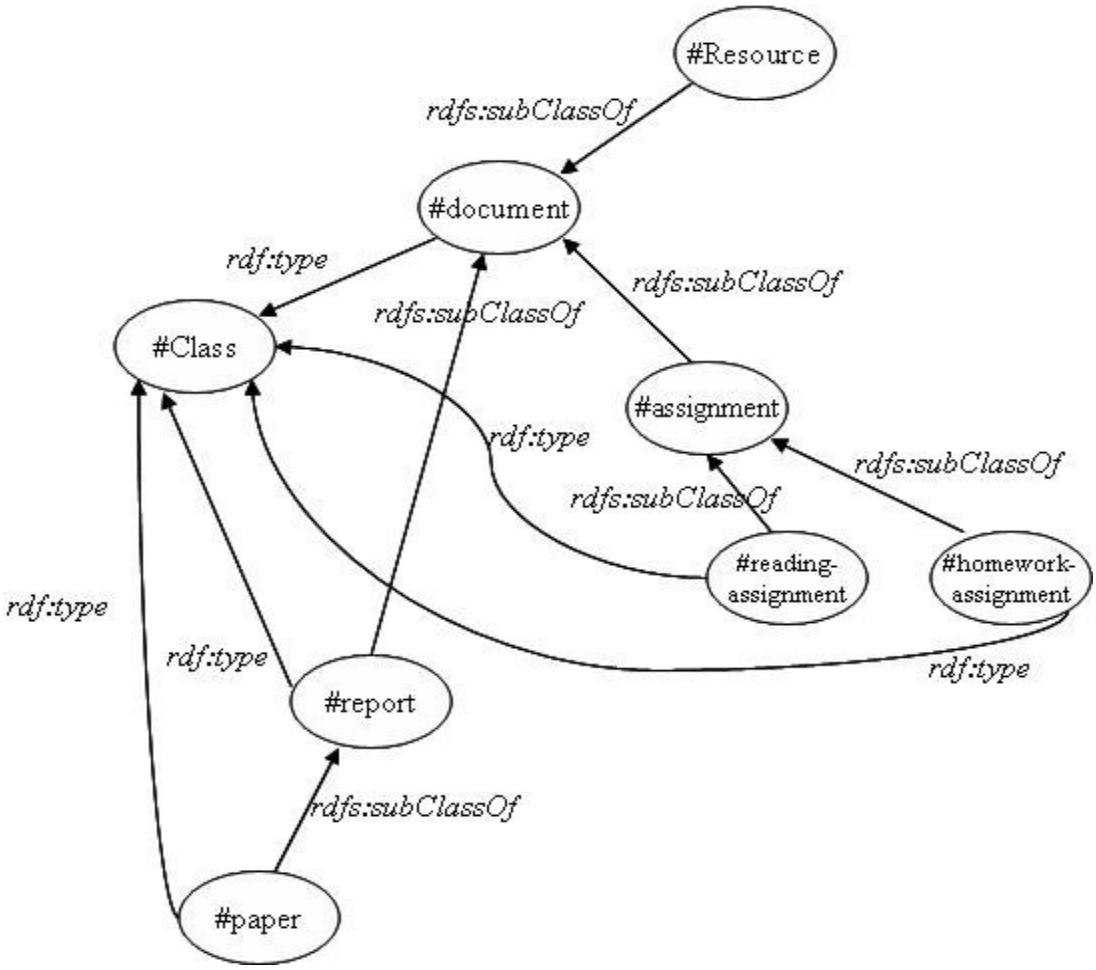


Figura 4.5 Parte de la Ontología de Documentos en Grafos RDF

```
<rdf:Description rdf:ID=”document” rdfs:label=”document”>
   <rdf:type rdf:resource=”#Class”>
   <rdfs:subClassOf rdf:resource=”#resource”>
</rdf:Description>
<rdf:Description rdf:ID=”assignment” rdfs:label=” assignment”>
   <rdf:type rdf:resource=”#Class”>
   <rdfs:subClassOf rdf:resource=”#document”>
</rdf:Description>
<rdf:Description rdf:ID=”reading-assignment” rdfs:label=”reading-assignment”>
   <rdf:type rdf:resource=”#Class”>
   <rdfs:subClassOf rdf:resource=”#assignment”>
</rdf:Description>
```

Figura 4.6 Parte de la Representación de la Ontología de Documentos en el Lenguaje RDF-S

En la Figura 4.5 se muestra parte de la ontología en grafos RDF, en donde se representan algunos documentos (*#document*, #report, #paper, *#assignment,#reading-assignment* y *#homework-assignment* ) a través de nodos y los arcos representan propiedades como *rdf:type* y *rdfs:subClassOf*. Cada uno de estos nodos son representados a través del lenguaje RDF-S (ver Figura 4.6); así tenemos la expresión <*rdf:type rdf:resource=”#Class”*> que indica que el nodo *assignment* es una clase.

En este capítulo se ha descrito el diseño de la Ontología de Documentos. Basado en esta ontología se obtuvo un vocabulario básico para describir los documentos y se utilizó el lenguaje RDF-S para representar este vocabulario.

# CAPÍTULO 5

# ARQUITECTURA BASADA EN AGENTES PARA LA RECUPERACIÓN DE METADATOS

## 5.1 Introducción

Los agentes poseen características que pueden ser utilizadas en aplicaciones para la recuperación de recursos que se encuentran distribuidos en diferentes dispositivos (computadoras, portátiles, etc.) y en diferentes formatos (pdf, doc, etc.). La sociabilidad de los agentes para comunicarse con otros, permite intercambiar mensajes que contengan información acerca de los recursos que se desean recuperar.

En este capítulo se presenta una arquitectura basada en agentes para la recuperación de metadatos y documentos. Se presentará una descripción de los agentes y las herramientas utilizadas en su desarrollo.

## 5.2 JADE: Una Alternativa para el Desarrollo de Agentes

Para la implementación de los agentes se utilizó JADE [Bellifemine99], un *middleware* desarrollado en Java que permite la ejecución y el desarrollo de aplicaciones

que está basado en el paradigma de agentes. JADE es extremadamente versátil al permitir ser integrado dentro de complejas arquitecturas como .NET o J2EE.

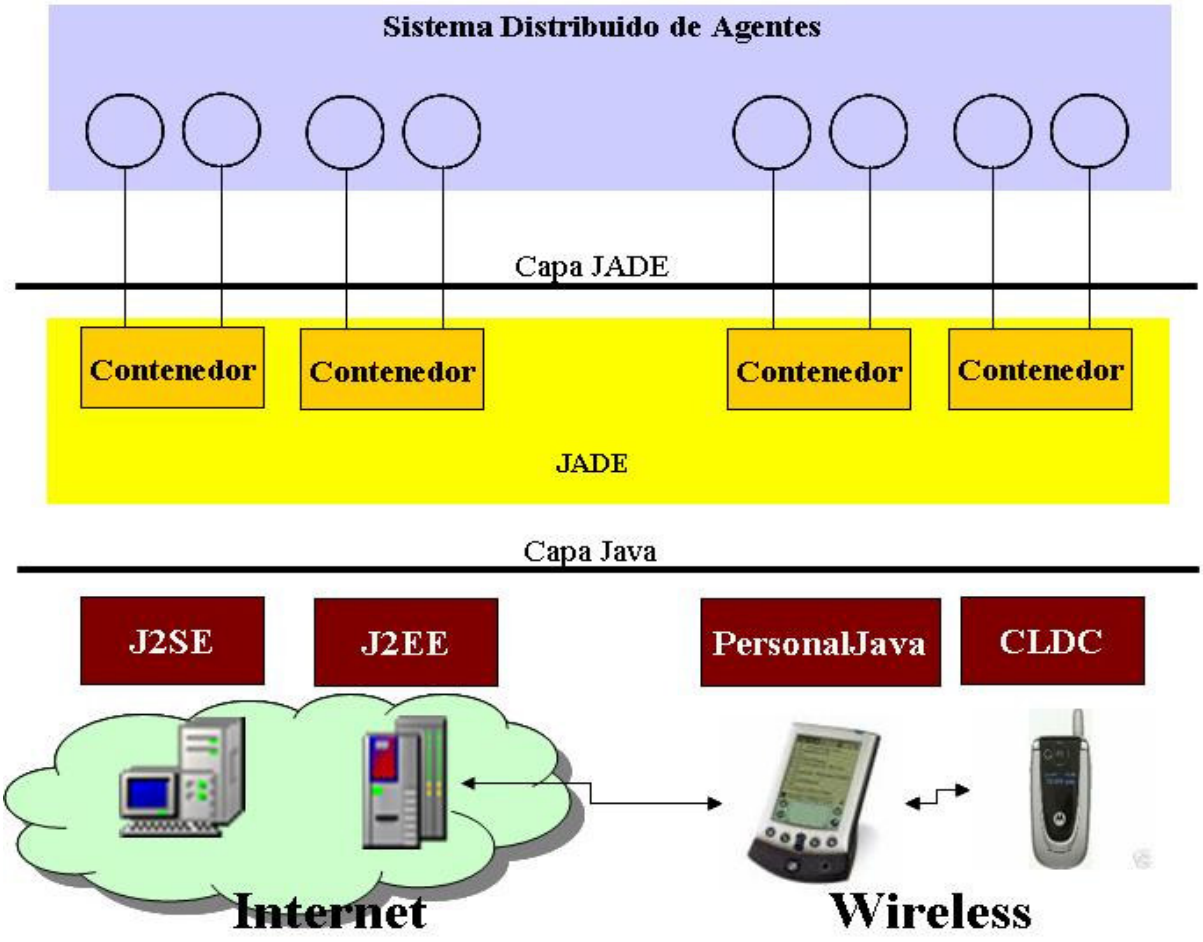


Figura 5.1 Arquitectura de JADE

Se utilizó JADE en esta investigación por su facilidad de uso, donde la complejidad del *middleware* está detrás de un simple e intuitivo conjunto de rutinas, protocolos y herramientas. Tiene las siguientes ventajas: *interoperatibilidad*, puesto que cumple con las especificaciones propuestas por FIPA [FIPA00023], *uniformidad* y *portabilidad*, debido a que JADE provee un conjunto homogéneo de APIs que son independientes de la versión de Java y de las características de la red. En la Figura 5.1 se muestra la arquitectura de JADE, donde cada instancia de un programa JADE es llamado

*contenedor* que contendrá a los agentes. El conjunto de todos los contenedores es llamado *plataforma,* el cual formará la capa JADE.

## 5.3 Acerca de la Arquitectura

En la arquitectura desarrollada en esta tesis se resalta el uso de agentes porque éstas permiten realizar trabajos específicos como la recuperación de recursos o la búsqueda de información. En la Figura 5.2 se muestra la estructura de la arquitectura desarrollada.

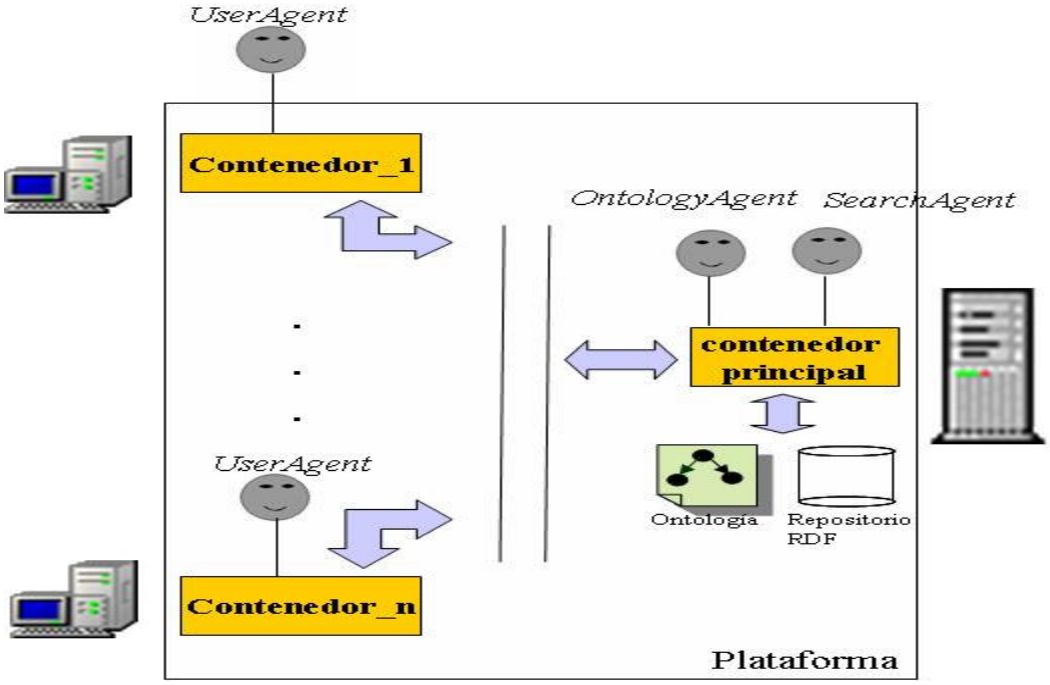


Figura 5.2 Arquitectura Basada en Agentes

En esta arquitectura existen agentes activos en diferentes contenedores, en donde el conjunto de contenedores formarán la plataforma. Se tiene un contenedor principal que registrará a todos los demás contenedores al momento que empiecen a ejecutarse.

Tenemos dos agentes (*SearchAgent* y *OntologyAgent*) definidos en el contenedor principal; uno de estos agentes se encargará de trabajar directamente sobre los metadatos, mientras el otro agente se encargará de realizar las búsquedas. Se tiene un agente (*UserAgent*) por cada computadora, el cual interactuará directamente con el usuario para realizar las tareas de visualización de los resultados de la búsqueda y capturar los metadatos. También se tiene la ontología de documentos (ver Capítulo 4) y el repositorio RDF o base de conocimiento en donde se guardará todas las expresiones RDF.

## 5.4 Implementación de los Agentes

Se utilizó Java para el desarrollo de los agentes y las librerías de JADE que proveen las clases necesarias para su construcción. En estos agentes se utilizan dos clases principales que son parte de la librería de JADE: la clase *agent* de la que heredan los agentes y la clase *behaviour* la cual permite definir las tareas que deben realizar los agentes.

### 5.4.1 UserAgent

El agente *UserAgent* se encarga de enviar los mensajes a los otros agentes de acuerdo a las acciones que realice el usuario. Un usuario puede realizar una acción de búsqueda de información de un documento o recuperar la estructura de la ontología; esta acción será traducida en un mensaje para ser enviada al agente *SearchAgent* o al *OntologyAgent*, y estos agentes procesan el mensaje y envían el resultado al agente *UserAgent* para ser mostrada al usuario.

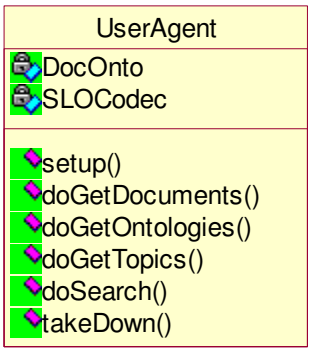


Figura 5.3 Estructura del Agente *UserAgent*

En la Figura 5.3 se muestra la estructura del agente *UserAgent* donde se definieron los métodos para responder a las acciones del usuario. Así tenemos el método *doGetOntologies* que se encarga de construir el mensaje y enviarlo al agente *OntologyAgent*; una vez procesado el mensaje por este último agente, el resultado es enviado al agente *UserAgent* para ser mostrado al usuario.

Dentro de las principales tareas que realiza el agente *UserAgent* son: obtener las ontologías disponibles (*ReceiveOntologiesBehaviour*), obtener la jerarquía de conceptos (*ReceiveTopicsBehaviour*) y obtener los metadatos de un documento de acuerdo al concepto que el usuario seleccione (*ReceiveSearchBehaviour*).

### 5.4.2 OntologyAgent

Este agente se encarga de acceder, modificar y actualizar la ontología de documentos. Los mensajes que llegan a este agente son mayormente para obtener información de la ontología para que luego sea mostrada al usuario. A este agente se le definieron algunas tareas, como por ejemplo: *ObtainOntologiesBehavior* que se encarga de obtener la estructura de la ontología y una vez procesada la tarea se envía el mensaje de respuesta.

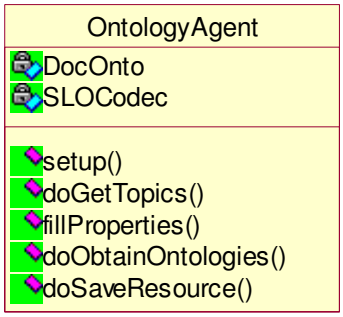


Figura 5.4 Estructura del Agente *OntologyAgent*

Los métodos y atributos de este agente son mostrados en la Figura 5.4, en donde sus principales tareas son: procesar todas las peticiones para obtener la estructura de la ontología (*ObtainOntologiesBehaviour*) y obtener los conceptos (*ObtainTopicOntologyBehaviour*).

### 5.4.3 SearchAgent

El agente *SearchAgent* es otro de los componentes de la arquitectura basada en agentes, el cual se encarga de buscar los metadatos de uno o más recursos dependiendo de las coincidencias que encuentre. Este agente tiene tarea *SearchDocumentsBehaviour* que se encarga de procesar los mensajes y extraer los parámetros de búsqueda que llegan junto con el mensaje. Con estos parámetros se realiza la búsqueda de las expresiones RDF (estas expresiones tienen la forma de sujeto, predicado y objeto) para luego enviar el resultado al agente que originó la petición.

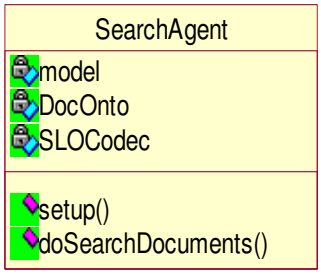


Figura 5.5 Estructura del Agente *SearchAgent*

La estructura de este agente es mostrada en la Figura 5.5, donde la propiedad *model* es utilizada por el motor de búsqueda para almacenar temporalmente todos los *TRIPLES* (llamados también expresiones que tienen la forma de *sujeto*, *predicado* y *objeto*) que están descritos en el lenguaje RDF. Para la búsqueda de las expresiones RDF se utilizó JENA [McBride00] que es una herramienta para búsqueda semántica.

## 5.5 Mensajes Utilizados en la Comunicación entre los Agentes

Se definieron mensajes para intercambiar información entre los agentes, cuya estructura está basada en el estándar FIPA-ACL [FIPA02c]. Cada uno de estos mensajes está orientado a realizar una acción o a transmitir información de un agente a otro; de esta forma se definen los siguientes mensajes:

- Mensajes de petición, en donde se envían mensajes a los agentes para que obtengan información (metadatos) de un determinado documento o de la ontología. De esta forma se definen mensajes como: *GetDocuments*, *GetOntologies* y *GetTopics* cuya estructura en el estándar FIPA-ACL se muestra en la Figura 5.6.

```
(QUERY-IF
:sender: (agent-identifier :name user@ece.uprm.edu.edu:1099/JADE)
:receiver (set (agent-identifier :name ontologyECE@ece.uprm.edu:1099/JADE))
:content "((action (agent-identifier :name user@ece.uprm.edu:1099/JADE)
                (ontologyconcept :title knowledge :
                :pathOnt …/knowledge_ont.rdf
                :baseOnt .../knowledge_ont.rdf)))"
:reply-with topics
:language fipa-s01
:ontology Documents-Ontology
)
```

Figura 5.6 Estructura del Mensaje ACL para Verificar la Existencia de un Concepto

En este mensaje se utiliza el acto comunicativo *QUERY-IF,* el cual permite hacer una consulta a otro agente sobre la existencia de un concepto (en este caso el concepto que se desea buscar es *knowledge*); si existe el concepto, el agente receptor podrá responder con *INFORM* o con *REFUSE,* si la petición ha sido rechazada.

- Mensajes de acción, en los cuales se indica a un agente que realice una acción, como por ejemplo el mensaje *SaveResource*, el cual le indica a un agente que guarde los metadatos de un recurso; la estructura de este mensaje es mostrada en la Figura 5.7

```
(INFORM
:sender: (agent-identifier :name user@ece.uprm.edu.edu:1099/JADE)
:receiver (set (agent-identifier :name searchECE@ece.uprm.edu:1099/JADE))
:content "((action (agent-identifier :name user@ece.uprm.edu:1099/JADE)
                (ontologyconcept :title knowledge :
                :pathOnt …/knowledge_ont.rdf
                :baseOnt .../knowledge_ont.rdf)))"
:reply-with "new resource"
:language fipa-s01
:ontology Documents-Ontology
)
```

Figura 5.7 Estructura del Mensaje ACL para Guardar un Recurso

- Mensajes de búsqueda, con los cuales podremos iniciar la búsqueda de un recurso en la base de conocimiento; el agente encargado de la búsqueda recibirá este mensaje con los datos necesarios. Así tenemos la estructura del mensaje *Search* que es mostrado en la Figura 5.8.

```
(QUERY-REF
:receiver (set (agent-identifier :name searchECE@ece.uprm.edu:1099/JADE))
:content "((action (agent-identifier :name user@ece.uprm.edu:1099/JADE
  :addresses (sequence IOR:00000…)) (search_resources :topic KnowledgeBase
  :ontology (ontologyconcept :title Knowledge :pathOnt …/knowledge_ont.rdf
  :baseOnt …/knowledge_ont.rdf :pathKB …/knowledge_kb.rdf))))"
:reply-with search
:language fipa-s10
:ontology Documents-Ontology
)
```

Figura 5.8 Estructura del Mensaje ACL para Realizar una Búsqueda

En este mensaje se indica al agente que realice la búsqueda de los documentos que estén relacionados al concepto *KnowledgeBase*; si el documento es encontrado

devolverá los resultados; en caso contrario enviará un mensaje con el acto comunicativo *REFUSE* indicando que no se encontraron los metadatos.

## 5.6 Búsqueda de Metadatos

Como se ha dicho anteriormente, el lenguaje semántico RDF provee primitivas para representar expresiones que en su estructura tiene tres elementos importantes *sujeto, predicado* y *objeto*. Los metadatos de los documentos serán representados en expresiones RDF, de esta forma se representarán el título, autor(es), tipo y otras características de un documento. En la Figura 5.9 se muestra un ejemplo sobre la descripción de un documento en el lenguaje RDF.

```
<rdf:Description rdf:about="#article02">
    <rdf:type rdf:resource="#article" />
    <dc:title>Collaborative Mapping with RDF</dc:title>
    <doc:url.>http://…/article02</doc:url>
    <doc:hasConcept rdf:resource="#KnowledgeBase" />
    <doc:author>
        <foaf:person>
            <foaf:name>Jo Walsh</foaf:name>
            <foaf:mbox rdf:resource="mailto:jw@frot.org" />
        </foaf:person>
    </doc:author>
</rdf:Description>
```

Figura 5.9 Ejemplo de Representación de los Metadatos en el Lenguaje RDF

Para extraer los metadatos de la base de conocimiento utilizamos JENA [McBride00] que provee un conjunto de librerías, protocolos y herramientas que permiten manipular el modelo RDF como un conjunto de *RDF TRIPLES*. La búsqueda de los metadatos está a cargo del agente *SearchAgent* y un escenario de búsqueda se muestra en la Figura 5.10.

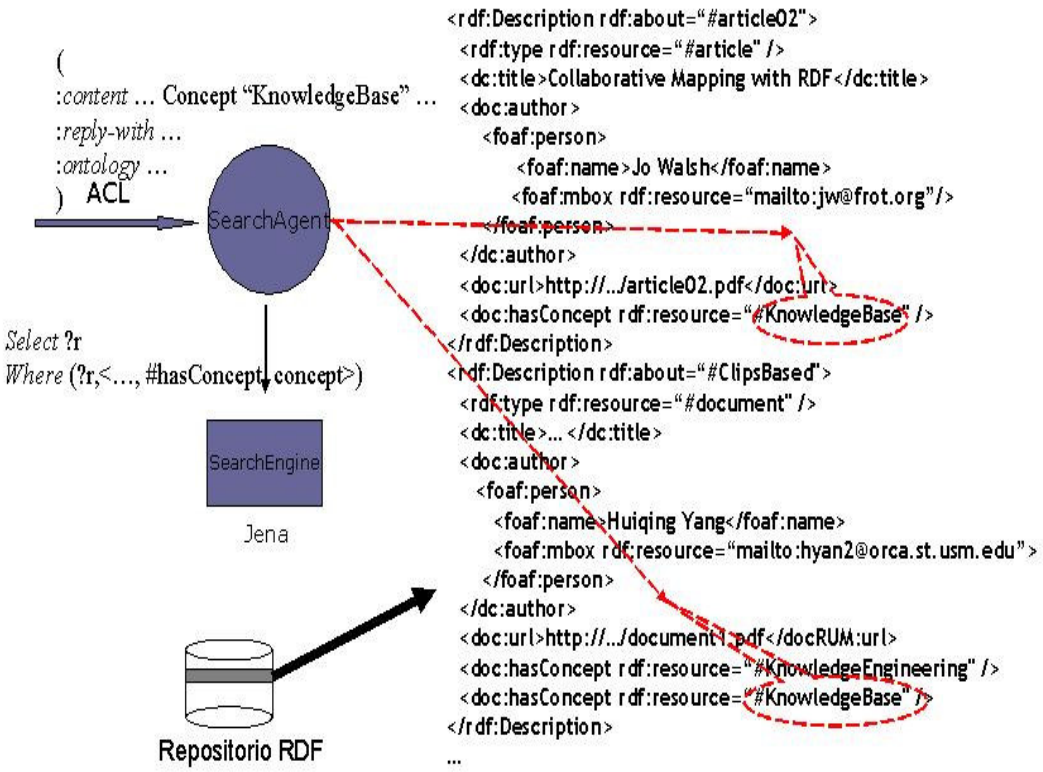


Figura 5.10 Escenario de una Búsqueda de Metadatos de un Documento

El agente *SearchAgent* se encarga de obtener los parámetros que lleva el mensaje para la búsqueda. Se utilizó el lenguaje de consulta *RDF Data Query Language* (RDQL) [Seaborne04] para construir las consultas, de esta forma se estructuró la consulta *select* con base en los parámetros obtenidos por el agente *SearchAgent*, en donde se tiene como condición una sentencia de la forma *(?r ,<#hasConcept>, Concept)*, en el cual C*oncept* representa la información que deseamos encontrar(en este caso un concepto). Luego, la consulta es enviada al motor de búsqueda JENA que extraerá todas las sentencias que cumplan con la condición. Por último, se envía un mensaje con el resultado al agente *UserAgent* para luego ser visualizado.

## *5.7 Implementación de un Prototipo:* oSEMA

*Ontology Semantic Agent (oSEMA)* es un prototipo desarrollado en Java que permite realizar anotaciones basado en la ontología de documentos. Adicionalmente este prototipo provee una pantalla para realizar búsquedas de metadatos de los documentos. Estos metadatos pueden ser almacenados localmente o en un repositorio RDF del servidor.

Este prototipo tiene como componente principal al agente *UserAgent,* y es a través de la pantalla de *oSEMA* que los usuarios podrán interactuar con el agente. *oSEMA* tiene dos interfaces importantes que se describen en los próximos párrafos.

### 5.7.1 Explorador Semántico

El explorador semántico permite buscar los metadatos de los documentos y los resultados de la búsqueda son mostrados en el lado derecho de la pantalla. En el lado izquierdo de la pantalla se tiene una jerarquía conceptos, que es utilizado para buscar los documentos. En la Figura 5.11 se muestra la pantalla.

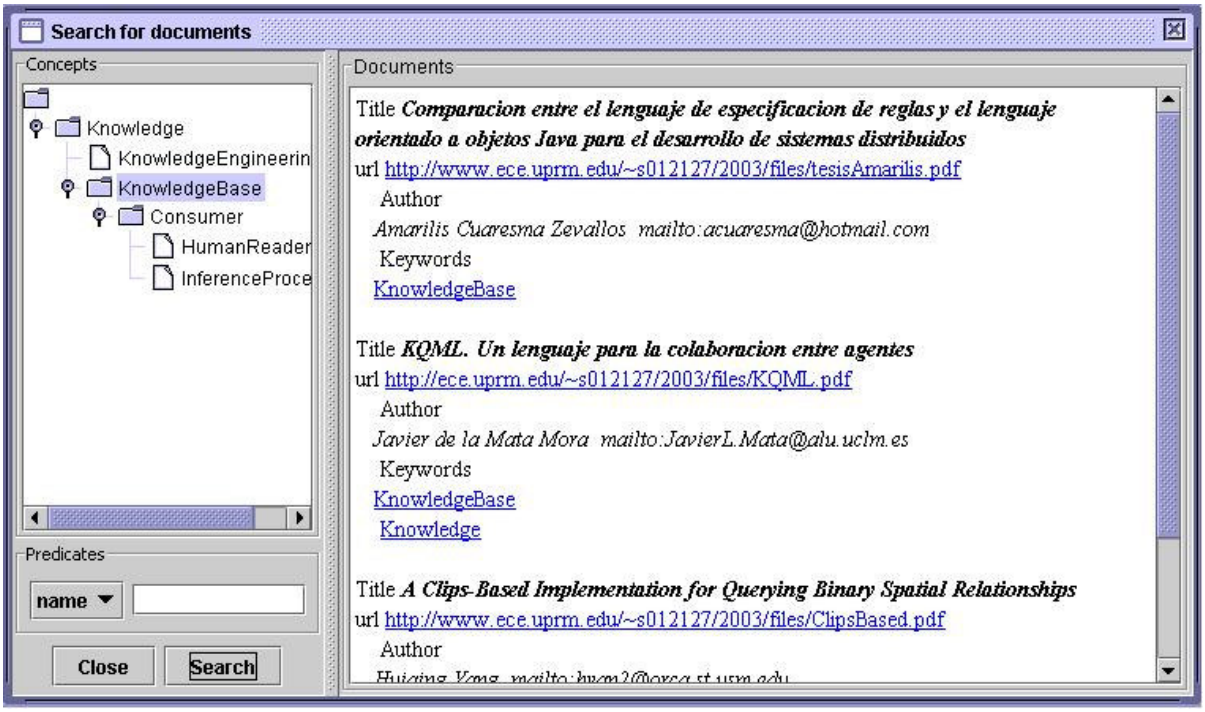


Figura 5.11 Pantalla del Explorador Semántico

### 5.7.2 Pantalla de Anotaciones

A través de esta pantalla se pueden realizar las anotaciones de los documentos y luego ser almacenados localmente o en el servidor. Estas anotaciones serán realizadas utilizando el vocabulario definido en la ontología de documentos y serán expresadas en el lenguaje RDF, para luego ser almacenadas en el repositorio RDF. En la Figura 5.12 se muestra la pantalla.

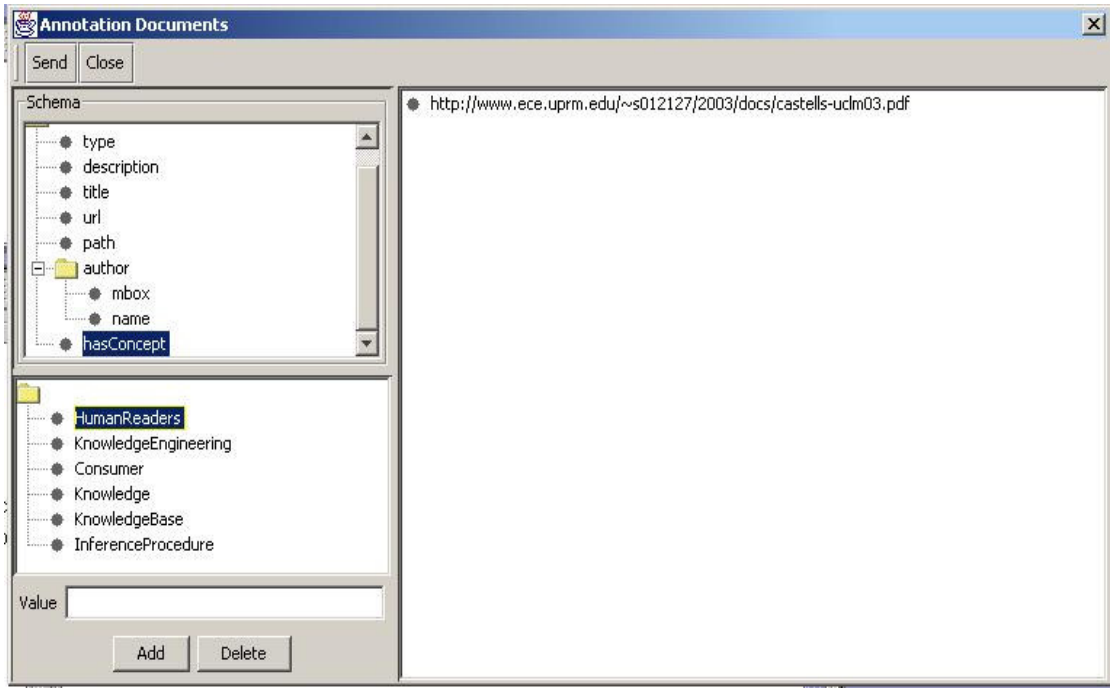


Figura 5.12 Pantalla de Anotaciones

En este capítulo se describió la arquitectura basada en agentes para la recuperación de metadatos de los documentos. También, se describió cada uno de los agentes y el prototipo construido. En el siguiente capítulo se describirá la metodología para realizar pruebas de tiempo de respuesta de consultas.

# CAPÍTULO 6

# METODOLOGÍA

## 6.1 Introducción

Con los metadatos identificados es posible describir las características de los documentos que están representados en una ontología. Para este propósito se utilizó el lenguaje semántico RDF. Estos metadatos, son almacenados en repositorios RDF o base de conocimiento, que contendrá sentencias de la forma *sujeto-predicado-objeto*. Para buscar los metadatos se utilizó el lenguaje de consulta RDQL (ver sección 5.6), el cual nos permitió buscar los metadatos sobre los documentos que deseamos recuperar.

El tiempo de respuesta de las consultas pueden afectar el comportamiento del sistema, de ahí que es necesario realizar pruebas para medir el tiempo de respuesta cuando se tienen varios agentes que realizan consultas simultáneamente. En este capítulo, se presentan la metodología y los resultados de estas pruebas.

## 6.2 Propósito de la Prueba

El propósito de estas pruebas es para determinar una organización ideal de los agentes y del repositorio de conocimiento. Se tomó el tiempo de respuesta de las consultas porque pueden afectar al rendimiento del sistema, considerando que se tendrán

varios agentes distribuidos en diferentes computadoras que estén realizando consultas simultáneamente.

## 6.3 Metodología

En esta sección, se describe la metodología seguida para realizar las pruebas de tiempo de respuesta de consultas que realizan los agentes (instancias del agente *UserAgent*) configurados en diferentes máquinas.

### 6.3.1 Equipos y Herramientas

Para medir el tiempo de respuesta se escogió un entorno adecuado donde se pudieran llevar a cabo las pruebas. Se utilizaron nueve computadoras, de las cuales una fue el servidor central (*ece*) y las demás fueron estaciones de trabajo del laboratorio de Estudiantes Graduados (*grads10, grads11, grads12, grads13, grads14, grads15, grads16, grads1 y, grads18*). Todas estas computadoras están conectadas en una red al dominio de *ece.uprm.edu*. La Tabla 6-1 muestra la descripción de estas computadoras utilizadas en la prueba.

Tabla 6-1 Descripción de las Computadoras Utilizadas en las Pruebas

| Nombre | Descripción | Sistema Operativo |
|---|---|---|
| ece | Sun Sparc Ultra Enterprise 450, 4 x 300 MHz, 2GB RAM, 72GB HD | Solaris 8 |
| grads (8 computadoras) | Compaq Precision 340, PIV 1.80GHz, 256 MB RAM, 40GB HD, Monitor 17” | Windows XP |

Para realizar estas pruebas se utilizó el siguiente *software*:

- Forte
- Java versión 1.4.1

En cada computadora se tenía instalado Java, en donde se ejecutaría cada agente sobre la plataforma Windows.

### 6.3.2 Escenarios

Para realizar las pruebas se diseñaron dos escenarios en donde se tiene una configuración básica para los escenarios, así como su organización particular (centralizado o descentralizado). Cada escenario está conformado por agentes en cada terminal o estación de trabajo y dos agentes en el servidor principal.

En la configuración básica se tiene el servidor principal que está conformado por el agente *SearchECE* (instancia del agente *SearchAgent*) que se encargará de la búsqueda de los recursos y el agente *OntologyECE* (instancia del agente *OntologyAgent*) que provee acceso a la ontología de documentos. En cada una de las estaciones de trabajo se crearon los agentes *grads10, grads11, grads12, grads13, grads14, grads15, grads16 y grads17* (instancias del agente *UserAgent*), agentes que realizarán consultas para recuperar los documentos.

A continuación se da una descripción breve de los dos escenarios.

#### 6.3.2.1 Centralizado

En el escenario centralizado se tienen los agentes – instancias del agente *UserAgent* – configurados en cada computadora y los dos agentes, *SearchAgent* y *OntologyAgent* en el servidor (*ece*). En la Figura 6.1 se muestra la configuración del escenario centralizado.

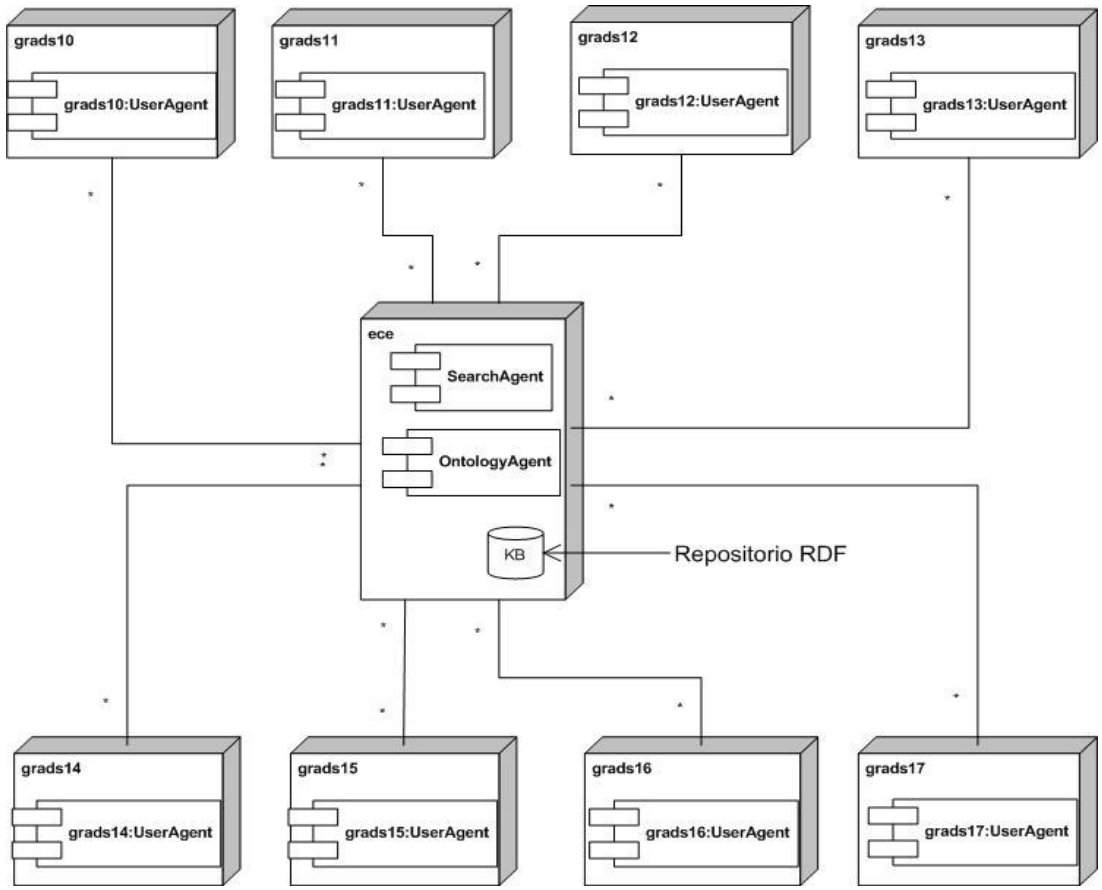


Figura 6.1 Configuración de un Escenario Centralizado de Agentes(8 Instancias del Agente UserAgent)

En el servidor se crea un repositorio RDF y el contenedor principal en donde registrará a todas las instancias de los agentes *UserAgent*, *SearchAgent* y *OntologyAgent*. Para realizar las pruebas, cada uno de los agentes envía consultas para recuperar los metadatos de los documentos al agente *SearchAgent* y, este agente se encarga de buscar

los metadatos en la base de conocimiento, ésta está conformada por los metadatos y los datos de los documentos.

#### 6.3.2.2 Descentralizado

En el escenario descentralizado se tiene a los agentes – instancias del agente *UserAgent* – configurados en cada computadora con su propio repositorio RDF y en el servidor se crea el contenedor principal donde se registrarán las instancias del agente *UserAgent* y la instancia del agente *OntologyAgent*. Cada una de las instancias del agente *UserAgent* envía una consulta para buscar los metadatos y la búsqueda es realizada por cada uno de las otras instancias del agente *UserAgent*. En la Figura 6.2 se muestra la configuración del escenario descentralizado.

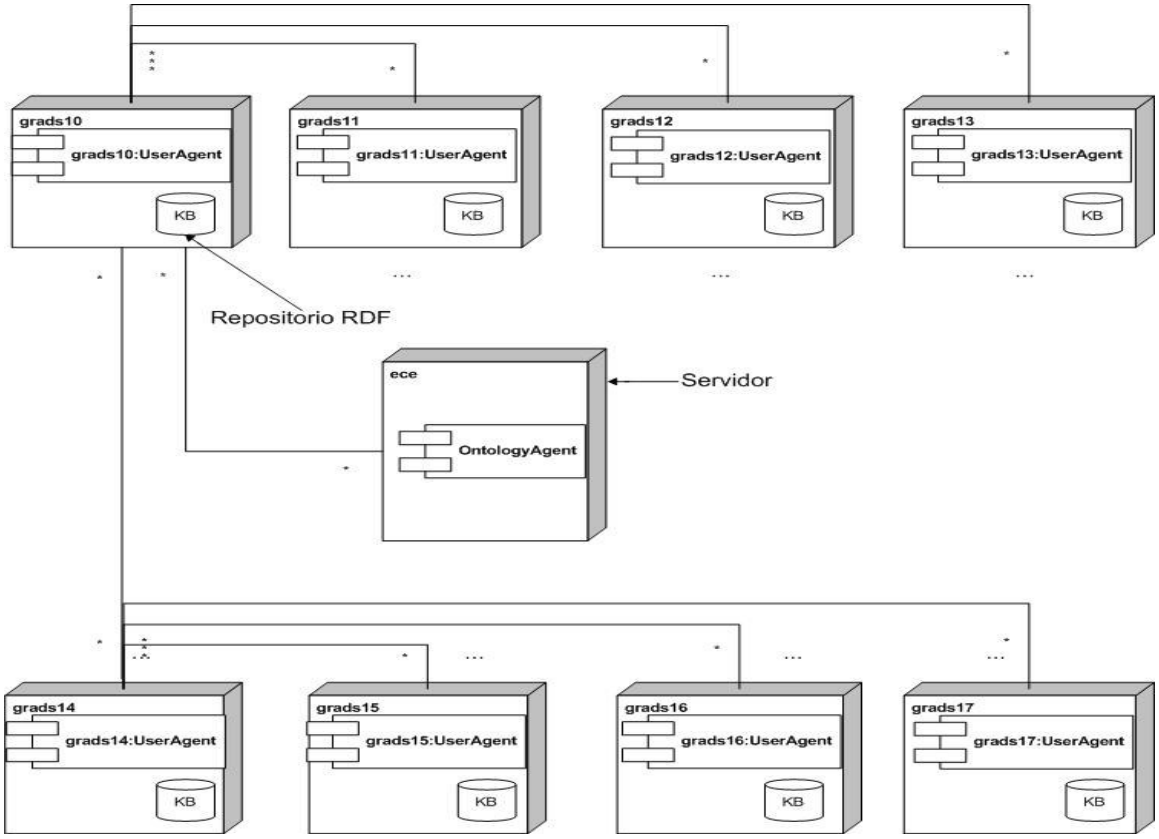


Figura 6.2 Configuración de un Escenario Descentralizado de 8 Agentes (Instancias del Agente UserAgent)

### 6.3.3 Configuración de los Repositorios RDF

En ambos escenarios se utilizó la Ontología de Documentos que fue representado en el lenguaje RDF-S, el cual forma el vocabulario para la representación de los metadatos de los documentos. Este vocabulario es almacenado en un archivo de texto en el servidor *ece*. Para acceder a cada término de este vocabulario es necesario que se cree un *namespace*. En las pruebas se utilizó el siguiente *namespace:*

xmnls:doc=*http://www.ece.uprm.edu/~s012127/2003/docs/documents_ont.rdf*

Se construyó un repositorio RDF de prueba que contiene 400 descripciones de recursos, estos recursos representan documentos. En el caso del escenario centralizado se tiene un único repositorio RDF o base de conocimiento (*Knowledge Base* - KB), que es creado en el servidor *ece* en un archivo de texto. Para el caso del escenario descentralizado, cada instancia del agente *UserAgent* tendrá un repositorio RDF; en este escenario se distribuyeron los 400 recursos en los repositorios RDF de forma uniforme.

### 6.3.4 Consultas

En cada uno de estos escenarios las consultas consisten en buscar los metadatos de los documentos que estén relacionados con un concepto de un dominio de conocimiento. En este caso particular se definió un subconjunto del dominio de sistemas de conocimiento. Para las pruebas de tiempo de respuesta de las consultas se utilizaron los siguientes conceptos: *KnowledgeBase*, *Knowledge*, *KnowledgeEngineering* y *Consumer*. Con base en estos conceptos se construyeron las consultas (ver Figura 6.3) en cada agente.

```
SELECT ?r, ?a
WHERE (?r, <doc:hasConcept>, <concept>), (?r, <doc:author>, ?a)
USING doc FOR
<http://www.ece.uprm.edu/~s012127/2003/docs/documents_ont.rdf#>
```

Figura 6.3 Consulta RDQL para Obtener los Metadatos de los Documentos

En los dos escenarios se utiliza la misma consulta; en el parámetro *concept* se indicará los conceptos a utilizar (*KnowledgeBase*, *Knowledge*, *KnowledgeEngineering* y *Consumer*). Así tenemos, que en el escenario centralizado cada agente envía una consulta con el concepto *KnowledgeBase* al servidor *ece* (la consulta la procesará el agente *SearchECE*), luego otra consulta con el concepto *Knowledge*, y luego las otras consultas con los otros conceptos. En el escenario descentralizado, en cada agente *UserAgent* se

construye una consulta a base de los conceptos mencionados anteriormente, para luego ser enviada a cada uno de los otros agentes *UserAgent*; la búsqueda es realizada en cada uno de estos agentes.

En este capítulo se describe la metodología usada para realizar pruebas de tiempo de respuesta de las consultas realizadas por los agentes en dos escenarios: centralizado y descentralizado. En el siguiente capítulo se presenta los resultados obtenidos.

# CAPÍTULO 7

# RESULTADOS

Este capítulo se presenta los resultados de las pruebas realizadas en los dos escenarios descritos en el capítulo anterior.

## 7.1 Primer Escenario

En el escenario centralizado se crearon los agentes *SearchECE* (instancia del agente *SearchAgent*) y *OntologyECE* (instancia del agente *OntologyAgent*) en el servidor *ece* y se configuraron 8 agentes (instancias del agente *UserAgent*) en cada una de las computadoras de la oficina de estudiantes graduados. Además, se tiene un único repositorio RDF, localizado en el servidor principal. Este repositorio (o base de conocimiento) está formado por todas las sentencias RDF (o hechos) y las descripciones de los metadatos las cuales están dadas por la representación de la Ontología de Documentos en el lenguaje RDFS.

Para realizar las pruebas de tiempo de respuesta de las consultas; los 8 agentes enviaron consultas al agente *SearchECE*, las cuales consistían en buscar documentos que tenían un concepto en particular. Una vez realizada la búsqueda, el resultado es enviado al agente que realizó la petición para ser procesada y mostrada al usuario.

Cada uno de los agentes envió varias consultas consecutivas, mediante las cuales se midió el tiempo de respuesta de cada consulta, desde que se envía la consulta hasta el

momento que es recibida la respuesta. Este tiempo está dado en milisegundos. Los resultados del promedio del tiempo de respuesta se presenta en la Tabla 7-1. La comparación del promedio de tiempo de respuesta de cada agente se muestra en la Figura 7.1. Los agentes se muestran en el eje horizontal y el tiempo en milisegundos es mostrado en el eje vertical. Se realizaron 4 mediciones con los cuatro conceptos: *KnowledgeBase*, *Knowledge*, *KnowledgeEngineering* y *Consumer*.

Tabla 7-1 Tiempo de Respuesta de las Consultas Realizadas por los Agentes en un Ambiente Centralizado (en milisegundos)

| Num. Prueba | grads10 | grads11 | grads12 | grads13 | grads14 | grads15 | grads16 | grads17 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| 1 | 15077 | 9314 | 25860 | 20126 | 43380 | 31016 | 36312 | 46657 |
| 2 | 5328 | 10311 | 42031 | 15968 | 21057 | 26406 | 31796 | 37060 |
| 3 | 5187 | 10300 | 31375 | 41234 | 15352 | 20859 | 25952 | 36387 |
| 4 | 5265 | 10741 | 42969 | 16110 | 21524 | 27109 | 37952 | 32613 |
| | | | | | | | | |
| Promedio | 7714.25 | 10166.5 | 35558.75 | 23359.5 | 25328.25 | 26347.5 | 33003 | 38179.25 |

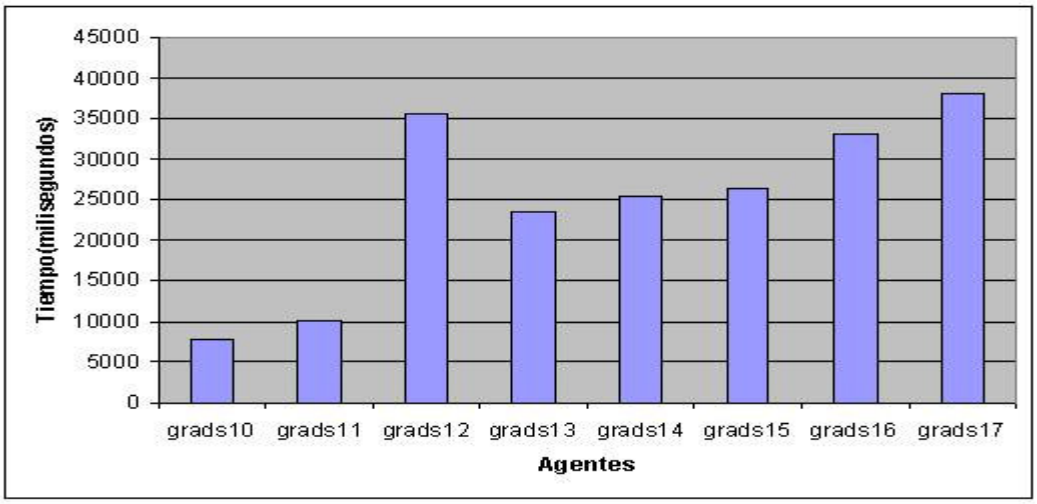


Figura 7.1 Tiempo de Respuesta de las Consultas Realizadas por 8 Agentes (Instancias del Agente UserAgent)

En los resultados mostrados en este escenario, se observa que el agente *grads17* tiene el mayor tiempo promedio en obtener la respuesta de una consulta (alrededor de 38 segundos). Y el agente *grads12* también tiene un tiempo promedio de alrededor de 35 segundos. En el caso del agente *grads12,* el tiempo de respuesta se debe a la mayor cantidad de documentos encontrados que coincidieron con los conceptos buscados por dicho agente.

## 7.2 Segundo Escenario

Al igual que el anterior escenario se configuraron 8 agentes (instancias del agente *UserAgent*) en las computadoras de la oficina de estudiantes graduados. Y sólo se configuró en el servidor *ece* el agente *OntotologyECE* (instancia del agente *OntologyAgent*) que provee información sobre la ontología de documentos. En cada uno de los agentes se configuró un repositorio RDF o base de conocimiento, lo cual permitió almacenar localmente las sentencias RDF producidas por el agente.

Para medir el tiempo de respuesta de las consultas, cada agente envía una consulta a los demás agentes – instancias del agente *UserAgent* – configurados en las computadoras. La búsqueda es realizada en el repositorio RDF de cada uno de los agentes que recibieron la consulta y, una vez procesada la búsqueda el resultado es enviado al agente que originó la consulta, para luego formar un único resultado.

En este escenario se realizaron cuatro pruebas, en la primera prueba se utilizó el concepto *KnowledgeBase* para formar la consulta que es utilizada por cada agente – instancias del agente *UserAgent* – para ser enviada a los demás agentes. De esta forma se realizó las tres restantes pruebas utilizando los conceptos *Knowledge, KnowledgeEngineering y Consumer*. En la Figura 7.2 se muestra la comparación de tiempo promedio de cada uno de los agentes. Los resultados del tiempo promedio de respuesta de cada prueba en los ocho agentes se muestra en la Tabla 7-2.

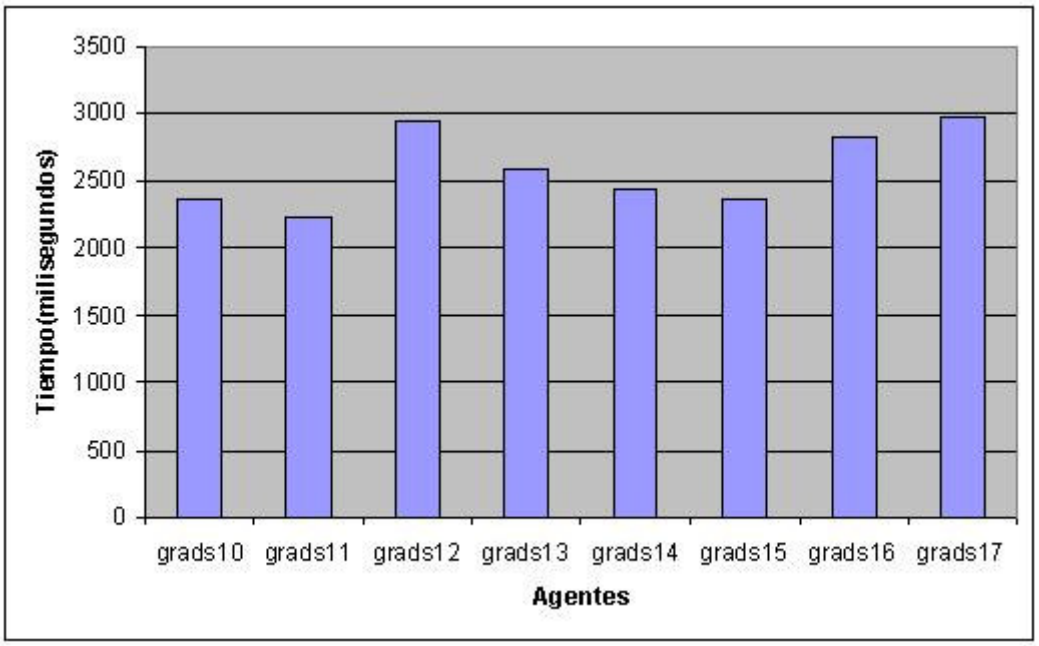


Figura 7.2 Representación Gráfica del Tiempo de Respuesta de las Consultas Realizadas por los Agentes en un Ambiente Distribuido

Tabla 7-2 Tiempo de Respuesta de la Consultas Realizadas por 8 Agentes en un Ambiente Descentralizado

| Num. Prueba | grads10 | grads11 | grads12 | grads13 | grads14 | grads15 | grads16 | grads17 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| 1 | 3251 | 3302 | 4708 | 3484 | 4109 | 3802 | 4404 | 4641 |
| 2 | 2041 | 2041 | 2214 | 2718 | 1688 | 1776 | 2476 | 2359 |
| 3 | 1573 | 1605 | 2214 | 1656 | 1672 | 1761 | 1975 | 2469 |
| 4 | 2632 | 1947 | 2681 | 2468 | 2281 | 2073 | 2413 | 2437 |
| | | | | | | | | |
| Promedio | 2374.25 | 2223.75 | 2954.25 | 2581.5 | 2437.5 | 2353 | 2817 | 2976.5 |

En este escenario, el agente *grads17* presenta el mayor tiempo de respuesta, éste es de 2976.5 milisegundos (alrededor de 3 segundos) y el agente *grads12* tiene un tiempo de respuesta de 2954.25 milisegundos (alrededor de 3 segundos).

## 7.3 Discusión

Al realizar una comparación con los resultados obtenidos en los dos escenarios, se observa que los tiempos promedios de respuesta en un escenario descentralizado es mucho menor que en el escenario centralizado. Uno de los factores que afectó los resultados es la ubicación de los repositorios, en donde tener un repositorio RDF por cada agente es mejor que un solo repositorio RDF donde se almacene todos los metadatos de los documentos.

En este capítulo se muestran los resultados de las pruebas, en donde tener una organización descentralizada de agentes con su propio repositorio RDF, presenta tiempos de respuesta mucho más cortos y más uniformes para todos los agentes frente a una organización centralizada de agentes con un único repositorio RDF.

# CAPÍTULO 8

# CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo de investigación se construyó una ontología de documentos y su representación en el lenguaje semántico RDF-S como una alternativa para compartir y reutilizar el conocimiento de un dominio. También, se investigó y se diseñó una arquitectura basada en agentes para la búsqueda y la recuperación de documentos para la cual se utilizó como caso de estudio los recursos educativos (documentos) utilizados por los estudiantes y los profesores del Departamento de Ingeniería Eléctrica y de Computadoras de la Universidad de Puerto Rico, Recinto Universitario de Mayagüez. En este trabajo, también se realizó una encuesta sobre usos y tipos de recursos de información utilizados por los profesores.

## *8.1 Conclusiones*

Las ontologías proveen una gran ayuda en la identificación de conceptos que son usados frecuentemente en un dominio, así como sus propiedades y sus relaciones. El uso de las ontologías en la representación de los recursos educativos permitió construir una base de conocimiento, que en su aplicación permitiría compartir y reutilizar conocimiento entre usuarios. Se tomó como dominio los documentos utilizados por estudiantes y profesores para la construcción de una ontología, éste consiste de una taxonomía de los documentos, sus atributos y las relaciones entre estos documentos. Con esta ontología se proporciona un vocabulario con el que se podrá compartir y organizar contenidos como

los documentos, el cual ayuda enormemente a procesos como las búsquedas sobre bases de conocimiento al tener un vocabulario en común.

El lenguaje semántico *Resource Description Framework Schema* (RDF-S) provee un conjunto de términos con el que se construyó un vocabulario. Este vocabulario está formado por los elementos que son parte de una ontología, como clases, propiedades y relaciones. Este vocabulario que representa la ontología de documentos será accedido a través de un *namespace* o espacio de nombres. A través de este *namespace* se podrá acceder el vocabulario que es utilizado para describir los recursos educativos y construir los repositorios RDF.

El paradigma de la tecnología de agentes ofrece un potencial amplio para el intercambio de información y la delegación de tareas específicas como la búsqueda y la recuperación de metadatos. Con el uso de los agentes se logró lo siguiente:

- Una comunicación entre los agentes independiente del sistema operativo (*Windows*, *Linux*, *Unix*, etc.), en donde los mensajes se estructuraron bajo un estándar (*Agent Communication Language*).

- El desarrollo de tareas específicas para cada agente, tareas como: actualizar y modificar la estructura de la ontología de documentos y buscar expresiones RDF sobre un documento.

- El desarrollo de agentes en un entorno descentralizado (organización distribuida), en donde cada agente tiene su propio repositorio RDF. Y la implantación de agentes en un entorno centralizado con un solo repositorio RDF.

Se realizaron pruebas de tiempo de respuesta de las consultas realizadas por los agentes para dos escenarios: centralizado y descentralizado. Estas pruebas fueron realizadas para determinar cuál de los dos escenarios presenta un mejor desempeño en términos del tiempo de respuesta. En términos generales, es posible concluir que una organización descentralizada de los agentes (instancia del agente *UserAgent*) con sus propios repositorios RDF, presenta un mejor rendimiento para realizar consultas, frente a una organización centralizada de agentes con un solo repositorio RDF donde el tiempo promedio de respuesta es bastante alto.

El escenario descentralizado presenta mejores resultados en términos de tiempo de respuesta de consultas y en donde también se aprovecha las características de la Web semántica, como la representación semántica de la información compartida por los agentes. La característica del uso de una ontología para la representar los documentos, permitió tener un vocabulario común para describir los documentos utilizados por los estudiantes y profesores en el salón de clase.

El desarrollo de la arquitectura de un sistema basado en agentes utilizando las tecnologías de la Web semántica, puede abrir un camino en la investigación que pueda conducir a la creación de herramientas que ayuden a los estudiantes y a los profesores en la búsqueda, recuperación, actualización y almacenamiento de recursos educativos.

## 8.2 Trabajos Futuros

Dentro de los principales trabajos futuros que se propone están los siguientes:

- La ampliación y la representación de la ontología de documentos en un lenguaje semántico de mayor expresividad, tal como el *Ontology Web Language* (OWL) o

*Darpa Agent Markup Language and Ontology Inference Language* (DAML-OIL). Ambos lenguajes proveen un vocabulario adicional para dar una mayor expresividad a los elementos que forman parte de una ontología. Utilizando estos lenguajes, se puede lograr una mejor representación semántica y construcción de bases de conocimiento más complejas. También, la búsqueda puede ser más eficaz dentro de contextos más específicos, al tener la información representada en estos lenguajes. De esta forma se podrá abarcar una mayor cantidad de recursos educativos utilizados por los estudiantes.

- Investigar sobre el soporte a inferencias de la información representada a base de la ontología de documentos. La información esta expresada en expresiones RDF, es decir, tienen la forma *sujeto, predicado* y *objeto* en el que se pueden definir reglas para buscar otras expresiones o agregar nuevas expresiones a la base del conocimiento.

- La ampliación de la arquitectura de agentes a dispositivos inalámbricos de mano como PDA, dispositivos *HandHeld* y otros dispositivos portátiles. La importancia de investigar sobre estos dispositivos es por su flexibilidad y su movilidad que utilizados en un entorno educativo pueden obtener grandes ventajas. El mayor reto es tener acceso permanente a las bases de conocimiento desde cualquier punto, ya sea desde el salón de clase, reuniones de grupo en diferentes lugares del campus o fuera de él para hacer tareas u otros tipos de asignaciones.

# BIBLIOGRAFÍA

[Baclawski01] Baclawski, K., M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski y M. Aronson. 2001. Extending UML to Support Ontology Engineering for the Semantic Web. In Fourth International Conference on UML. Toronto.

[Beckett02] Beckett, D., E. Miller y D. Brickley. 2002. Expressing Simple Dublin Core in RDF/XML. http://dublincore.org/documents/2002/07/31/dcmes-xml/.

[Bellifemine99] Bellifemine, F., A. Poggi y G. Rimasa. 1999. JADE-A FIPA- compliant agent framework. Proceeding of PAAM'99, London..

[Berners01] Berners-Lee, T., J. Hendler y O. Lassila. 2001 The Semantic Web: A new Form of Web Content that is Meaningful to Computers will Unleash a Revolution of new Possibilities. http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21.

[Brickley03] Brickley, D. y L. Millar. 2003. FOAF Vocabulary Specification. http://xmlns.com/foaf/0.1/.

[Broekstra01] Broekstra, J., A. Kampman y F. van Harmelen. 2001. Sesame: An Architecture for Storing and Querying RDF Data and Shema Information.

[Dhraief01] Dhraief, H., W. Nejdl, B. Wolf y M. Wolpers. 2001, Open Learning Repositories and Metadata Modeling. International Semantic Web Working Symposium (SWWS), July 30 – August 1. Stanford University, California, USA.

[Duineveld00] Duineveld, A. J., R. Stoter, M.R. Weiden, B. Kenepa y V.R. Benjamins. 2000. WonderTools?: a comparative study of ontological engineering tools. International Journal of Human-Computer Studies, Volume 52, Issue 6, p: 1111-1133.

[Fensel00] Fensel, D. et al. 2000. OIL in a Nutshell. In: Knowledge Acquisition, Modeling, and Management. Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligent, LNAI, Springer-Verlag.

[Fensel01] Fensel, D., F. van Harmelen, I. Horrocks, D.L. Mcguiness y P.F. Patel-Schneider. 2001. OIL: An Ontology Infrastructure for the Semantic Web. IEEE Intelligent Systems, 16(2): 38-45.

[Finin94] Finin, T., D. Mckay y R. McEntire. 1994. KQML-A Language and Protocol for Knowledge and Information Exchange. Technical Report CS-94-02. Computer Science Department. University of Maryland. UMBC, Baltimore MD 21228.

[FIPA96] The Foundation for Intelligent Physical Agents. 1996. http://www.fipa.org.

[FIPA02a] FIPA Agent Management Specification. 2002. http://www.fipa.org/specs/fipa00023/.

[FIPA02b] FIPA Communicative Act Library Specification. 2002. Foundation for Intelligent Physical Agents. http://www.fipa.org/specs/fipa00037/.

[FIPA02c] FIPA ACL Message Structure Specification. 2002. Foundation for Intelligent Physical Agents. http://www.fipa.org/specs/fipa00061/.

[Fischer01] Fischer, G. y J. Ostwald. 2001. Knowledge management: problems, promises, and challenges. IEEE Intelligent Systems, vol. 16,No. 1.

[Fridman01] Fridman, N. y D. McGuinness. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.

[Gomez02] Gómez-Pérez, A. y O. Corcho. 2002. Ontology Languages for the Semantic Web. Intelligent Systems, IEEE, Volume: 17, Issue: 1, Pages: 54-60.

[Hendler99] Hendler, J. 1999. Is There an Intelligent Agent in Your Future?. http://www.nature.com/nature/webmatters/agents/agents.html.

[Hermans00] Hermans, B. 2000. Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the society & a prediction of (near -) future developments.

[Iannella98] Iannella, R. 1998. An Idiot's Guide to the Resource Description Framework. The New Review of Information Networking, Volume 4. http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/.

[Lassila99] Lassila, O. y R. Webick. 1999. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation. www.w3.org/TR/PR-rdf-syntax.

[Manola04] Manola, F. y E. Miller. 2004. RDF Primer. W3C Recommendation. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[McBride00] McBride, B. 2000. Jena: Implementing the RDF Model and Syntax Specification. Technical report. Hewlett Packard Laboratories. Bristol. http://www.hpl.hp.com/semweb/index.html."

[McGuinness04] McGuinness, D. y F. van Harmelen. 2004. OWL Web Ontology Language Overview. http://www.w3.org/TR/2004/REC-owl-features-20040210.

[Nejdl01] Nejdl, W., H. Dhraief. y M. Wolpers. 2001. O-Telos-RDF: A Resource Description Format with Enhenced Meta-Modeling Functionalities based on O-Telos. Technical Report. Uni. of Hannover.

[Noy01] Noy, N.F., M. Sintek, S. Decaer, M. Crubezy, R.W. Fergerson y M.A. Musen. 2001. Creating Semantic Web Contents with Protégé-2000. IEEE Intelligent Systems 16(2):60-71.

[Nwana96] Nwana, H. S. 1996. Software Agents: an overview. Knowledge Engineering Review, 1(3):205-244.

[Ramesh93] Patil, R., R. Fikes, P. Patel-Schneider, D. Mckay, T. Finin, T. Gruber, y R. Neches. 1993. The Darpa knowledge Sharing Effort: Progress Report. KR'92 Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference.

[Roberto99] Roberto, V., V. Della-Mea, L. Di-Gaspero, y A. Conti. 1999. MANTHA: Agent-based Management of Hypermedia Documents. Multimedia Computing and Systems. IEEE International Conference on, vol: 2, 7-11, pp:814-818.

[Seaborne04] Seaborne, A. 2004. RDQL – A Query Language for RDF. W3C. HP Labs Bristol. http://www.w3.org/Submission/RDQL/.

[Sean01] Sean-Bechhofer, C.G. 2001. Towards Annotation using DAML+OIL. Group Department of Computer Science. University of Manchester.

[Staab01] Staab, S., A. Maedche y A. Handschuh. 2001. An Annotation Framework for the Semantic Web. In Proceedings of the First Workshop on Multimedia Annotation. Tokyo, Japan.

[Sato02] Sato, H., Y. Abe y A. Kanai. 2002. Hyperclip: A Tool for Gathering and Sharing Meta-Data on Users' Activities by using Peer-to-Peer Technology. International Workshop Real World RDF and Semantic Web Applications.

[Steinacker01] Steinacker, A., A. Ghavam y R. Steinmetz. 2001. Metadata Standards for Web-Based Resources. Multimedia, IEEE, vol: 8, issue: 1, pp: 70-17.

[Vega04] Vega-Riveros, J. F. 2004. Research Challenges for Knowledge Systems in Higher Education. Submitted to IEEE Frontiers in Education FIE.

[Venkatsubramani02] Venkatsubramani, S. y RKVS. Raman. 2002. Annotation Tool for the Semantic Web. International Workshop Real World RDF and Semantic Web Applications.

[Wilson97] Wilson, B. G. 1997. Reflections on Constructivism and Instructional Design. In Instructional Development Paradigms, Englewood Cliffs NJ: Educational Technology Publications.

# Apéndice A

## Encuesta Realizada a los Profesores

### *Cuestionario Utilizado para la Encuesta*

Universidad de Puerto Rico

Recinto Universitario de Mayagüez

Proyecto:

Diseño de un modelo para Gerencia de Conocimiento en el Departamento de Ingeniería Eléctrica y de Computadoras.

Objetivo:

Identificar las diferentes técnicas y herramientas que utilizan los profesores del departamento de Ingeniería Eléctrica y de Computadoras, en apoyo a su proceso de enseñanza.

Cuestionario

Por favor, responda las siguientes preguntas. Usted puede seleccionar una o varias alternativas en cada pregunta. Para cada una de las alternativas seleccione el nivel de referencia según se indica a continuación:

(3): Muy frecuente

(2): Ocasionalmente

(1): Rara vez

(0): Nunca

1. Indique qué elementos utiliza para la preparación de sus clases:

(3) (2) (1) (0) Libros

(3) (2) (1) (0) Revistas (artículos técnicos y artículos de conferencia)

(3) (2) (1) (0) Apuntes anteriores del mismo curso

(3) (2) (1) (0) Páginas Web

(3) (2) (1) (0) Otros

Si es otros, por favor indique cuáles:

.....

.....

.....

2. Indique qué elementos utiliza para el diseño de los exámenes y otras formas de evaluación en los cursos que enseña:

(3) (2) (1) (0) Libros

(3) (2) (1) (0) Revistas (artículos técnicos y artículos de conferencia)

(3) (2) (1) (0) Exámenes anteriores propios

(3) (2) (1) (0) Exámenes anteriores de otros profesores

(3) (2) (1) (0) Otros

Si es otros, por favor indique cuáles:

.....

.....

.....

3. ¿Qué técnicas de presentación utiliza en sus sesiones de clases?

(3) (2) (1) (0) Presentación de diapositivas o transparencias

(3) (2) (1) (0) Escritura en la pizarra

(3) (2) (1) (0) Demostraciones multimedia (incluye herramientas como MS Power Point, MS Word y similares)

(3) (2) (1) (0) Presentación de videos o películas

(3) (2) (1) (0) Modelos de computadora, simulación y otros programados

(3) (2) (1) (0) Otros

Si es otros, por favor indique cuáles:

.....

.....

.....

4. ¿Qué tipo de trabajo asigna usted en sus cursos?

(3) (2) (1) (0) Asignaciones de lectura

(3) (2) (1) (0) Asignaciones de programación

(3) (2) (1) (0) Presentación de exposiciones

(3) (2) (1) (0) Presentación de seminarios

(3) (2) (1) (0) Experimentos de laboratorio

(3) (2) (1) (0) Trabajos prácticos

(3) (2) (1) (0) Ejercicios del texto u otras fuentes

(3) (2) (1) (0) Otros

Si es otros, por favor indique cuáles:

…….

…….

…….

5. ¿Qué fuentes de información espera que utilicen los estudiantes para sus asignaciones y estudio?

(3) (2) (1) (0) Libros

(3) (2) (1) (0) Revistas (artículos técnicos y artículos de conferencia)

(3) (2) (1) (0) Paginas Web (Internet)

(3) (2) (1) (0) Exámenes de años anteriores

(3) (2) (1) (0) Trabajos de otros compañeros de años anteriores

(3) (2) (1) (0) Trabajos previos propios

(3) (2) (1) (0) Otros

Si es otros, por favor indique cuáles:

…….

…….

…….

6. ¿Comparte experiencias sobre el curso con otros colegas? Si su respuesta es afirmativa, indique cómo comparte estas experiencias.

(3) (2) (1) (0) Conversaciones informales

(3) (2) (1) (0) Correo electrónico

(3) (2) (1) (0) Reuniones de comité

(3) (2) (1) (0) Eventos y revistas

(3) (2) (1) (0) Técnicas sobre educación

(3) (2) (1) (0) Otros

Si es otros, por favor indique cuáles:

…….

…….

…….

…….

7. Por favor, indique las técnicas que utiliza para comunicarse con sus estudiantes:

(3) (2) (1) (0) Correo electrónico

(3) (2) (1) (0) Página Web

(3) (2) (1) (0) Horario de oficina y clase

(3) (2) (1) (0) Otros

Si es otros, por favor indique cuáles:

…….

…….

…….

## *Resultados de la Encuesta*

En el proceso de la investigación de realizó una encuesta para identificar las diferentes técnicas y herramientas que utilizan los profesores del departamento de Ingeniería Eléctrica y de Computadoras. También, permitió identificar los documentos que comúnmente utilizan los profesores en el salón de clase.

A continuación, se muestran las preguntas y los resultados de cada una de ellas. Se tomaron cuatro indicadores de medida:

3 Muy frecuentemente
2 Ocasionalmente
1 Rara vez
0 Nunca

1. **Indique qué elementos utiliza para la preparación de sus clases:**

Item 1 Libros
Item 2 Revistas (artículos técnicos y artículos de conferencia)
Item 3 Apuntes anteriores del mismo curso
Item 4 Páginas Web
Item 5 Otros

En la siguiente gráfica se muestra los resultados obtenidos en esta pregunta.

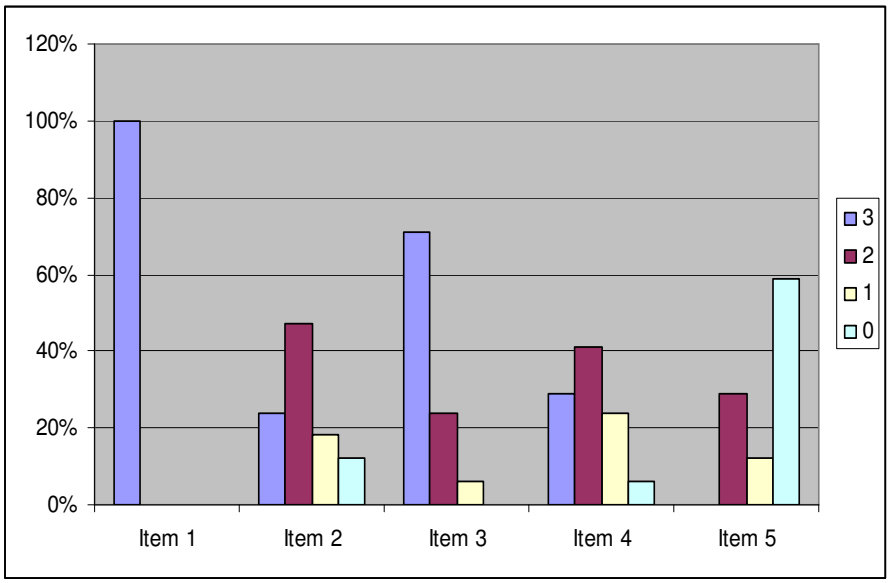


Figura A-1 Elementos utilizados en la preparación de sus clases

En la Figura A-1 se muestra los resultados obtenidos sobre los elementos utilizados por los profesores en la preparación de sus clases. En el Item 5 Otros, se obtuvo los siguientes elementos: planos de proyectos reales, informes técnicos, trabajo de investigación, programas de simulación, catálogos, etc.

**2. Indique qué elementos utiliza para el diseño de los exámenes y otras formas de evaluación en los cursos que enseña.**

Item 1 Libros
Item 2 Revistas (artículos técnicos y artículos de conferencia)
Item 2 Exámenes anteriores propios
Item 3 Exámenes anteriores de otros profesores
Item 4 Otros

A continuación, se muestra una gráfica de los resultados obtenidos en esta pregunta.

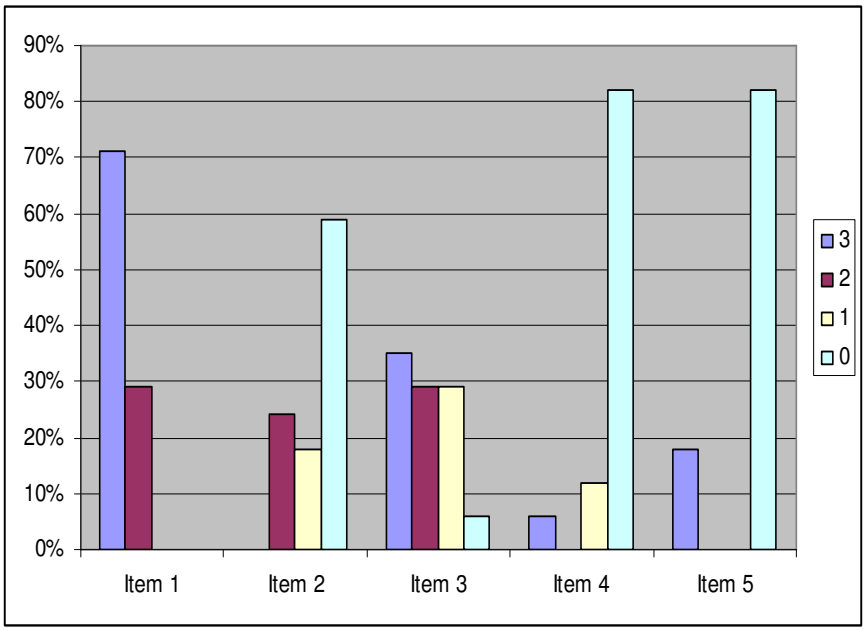


Figura A-2 Elementos utilizados en el diseño de exámenes

En la Figura A-2 se muestra los elementos utilizados en el diseño de exámenes. Adicionalmente en el Item 4 Otros, se obtuvo la siguiente información: diseño de ejericios propios, presentación del tema según hecha en el salón de clase, etc.

**3. ¿Qué técnicas de presentación utiliza en sus sesiones de clases?**

Item 1 Presentación de diapositivas o transparencias
Item 2 Escritura en la pizarra

Item 3 Demostraciones multimedia (incluye herramientas como MS Power Point, MS Word y similares)
Item 4 Presentación de videos o películas
Item 5 Modelos de computadora, simulación y otros programados
Item 6 Otros

A continuación, se muestra en una gráfica los resultados obtenidos en esta pregunta.

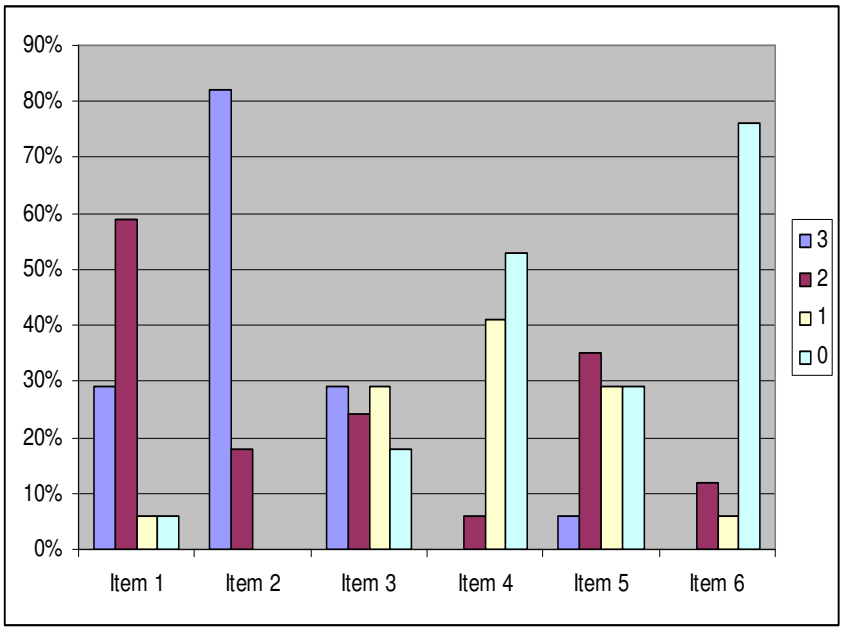


Figura A-3 Técnicas de Presentación

En la Figura A-3 se muestra las diferentes técnicas utilizadas por los profesores. En el Item 6 Otros, se obtuvo la siguiente información: equipos llevados a clase, planos, prácticas de laboratorio, etc.

**4. ¿Qué tipo de trabajo asigna usted en sus cursos?**

Item 1 Asignaciones de lectura
Item 2 Asignaciones de programación
Item 3 Presentación de exposiciones
Item 4 Presentación de seminarios
Item 5 Experimentos de laboratorio
Item 6 Trabajos prácticos
Item 7 Ejercicios del texto u otras fuentes

Item 8 Otros

A continuación, se muestra resultados obtenidos en esta pregunta.

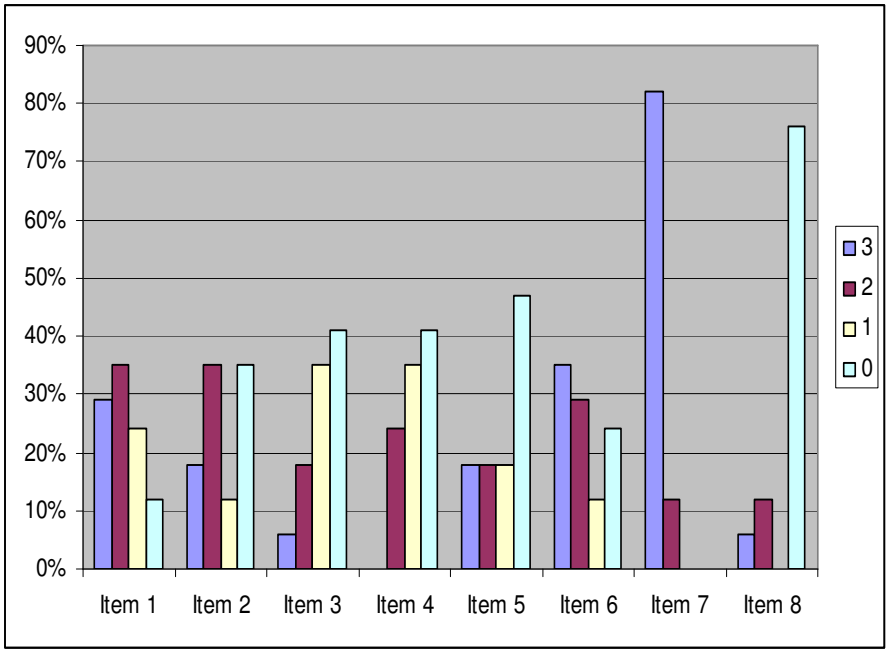


Figura A-4 Trabajos asignados en los cursos

En la Figura A-4 se muestra los trabajos que normalmente asignan los profesores en sus cursos. En el Item 8 Otros, se obtuvo la siguiente información: ejercicios mostrados en páginas webs, trabajos de repaso de asignaturas anteriores, diseños de sistemas electrónicos, etc.

**5. ¿Qué fuentes de información espera que utilicen los estudiantes para sus asignaciones y estudio?**

Item 1 Libros
Item 2 Revistas (artículos técnicos y artículos de conferencia)
Item 3 Páginas Web (Internet)
Item 4 Exámenes de años anteriores
Item 5 Trabajos de otros compañeros de años anteriores

Item 6 Trabajos previos propios
Item 7 Otros

A continuación, se muestra en una gráfica los resultados obtenidos en esta pregunta.

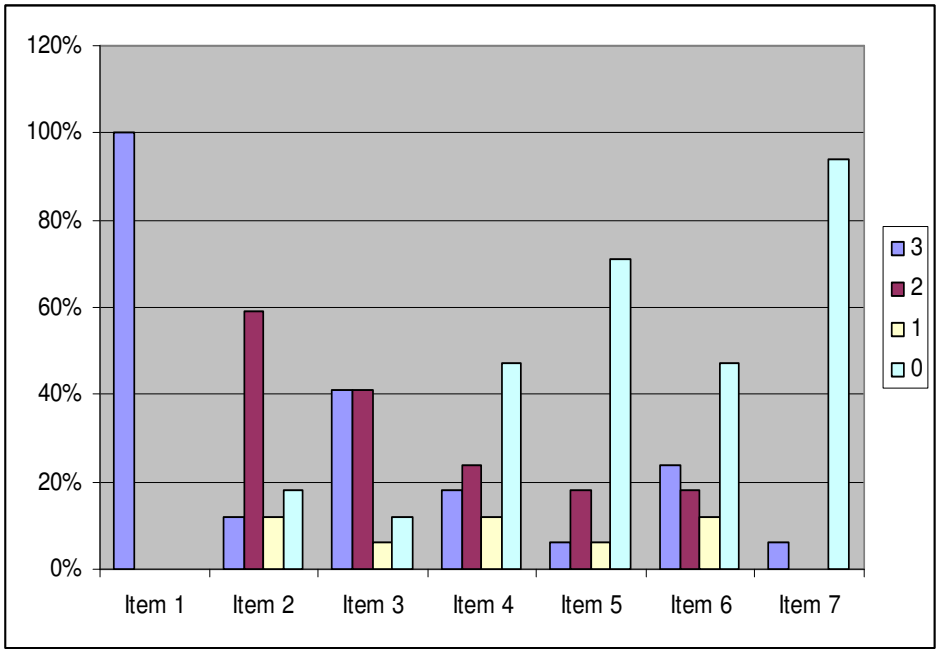


Figura A-5 Fuentes de información que recurren los estudiantes

En la Figura A-5 se muestra las fuentes de información que utilizan los estudiantes para realizar sus trabajos. En el Item 7 Otros, se obtuvo la siguiente información: apuntes de conferencias.

**6. ¿Comparte experiencias sobre el curso con otros colegas? Si su respuesta es afirmativa, indique cómo comparte estas experiencias.**

Item 1 Conversaciones informales
Item 2 Correo electrónico
Item 3 Reuniones de comité
Item 4 Eventos y revistas
Item 5 Técnicas sobre educación
Item 6 Otros

A continuación, se muestra en una gráfica los resultados obtenidos en esta pregunta.

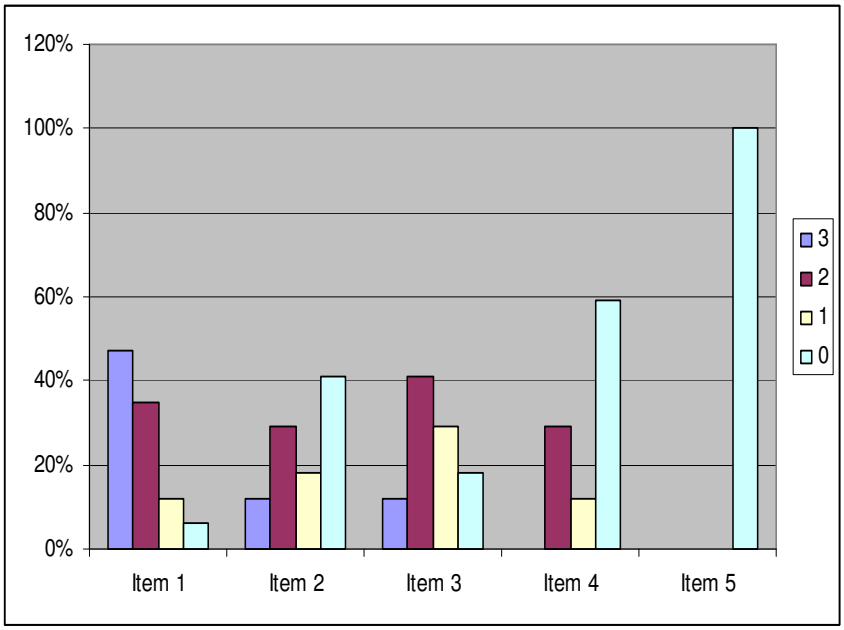


Figura A-6 Mecanismos usados para compartir experiencias

En la Figura A-6 se muestra los mecanismos que recurren los profesores para compartir sus experiencias. No se obtuvo información adicional en el Item 5 Otros.

**7. Por favor, indique las técnicas que utiliza para comunicarse con sus estudiantes:**

Item 1 Correo electrónico
Item 2 Página Web
Item 3 Horario de oficina y clase
Item 4 Otros

A continuación, se muestra en una gráfica los resultados obtenidos en esta pregunta.

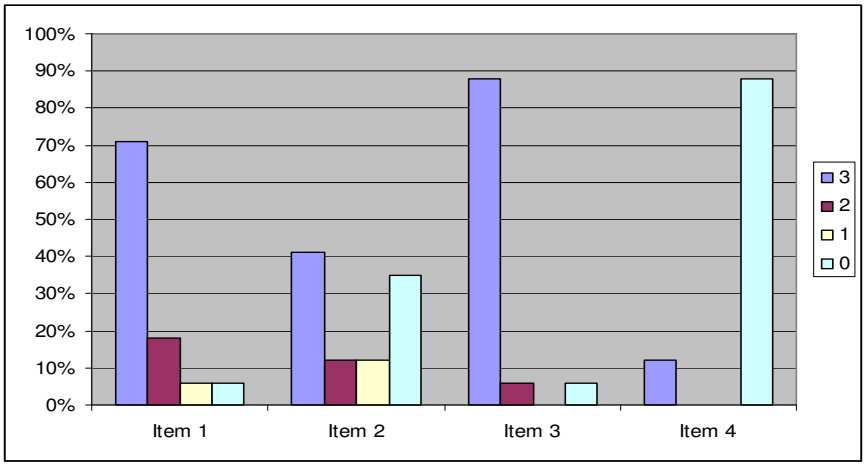


Figura A-7 Técnicas utilizadas para comunicarse con los estudiantes

En la Figura A-7 se muestran los resultados obtenidos acerca de la técnicas utilizadas por los profesores para comunicarse con los estudiantes. Adicionalmente en el Item 4 Otros, se obtuvo la siguiente información: en el mismo salón de clases y herramientas de *software* para el envio de mensajes instantaneos.

.

# Apéndice B

## Acrónimos

| | |
|---|---|
| ACL-FIPA | *Agent Communication Language – FIPA* |
| DAML+OIL | *Darpa Agent Markup Language + Ontology Inference Layer* |
| DCMI | *Dublin Core Metadata Initiative* |
| FIPA | *The Foundation for Intelligent Physical Agents* |
| HTML | *Hypertext Markup Language* |
| JADE | *Java Agent Development Framework* |
| KQML | *Knowledge Query and Manipulation Language* |
| OIL | *Ontology Inference Layer* |
| oSEMA | *Ontology Semantic Agent* |
| OWL | *Ontology Web Language* |
| RDF | *Resource Description Framework* |
| RDF-S | *Resource Description Framework Schema* |
| RDQL | *RDF Data Query Language* |
| WWW | *World Wide Web* |
| W3C | *World Wide Web Consortium* |