



**UNIVERSIDAD
NACIONAL
DE LOJA**

PP TT-N-CIS



Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“Sistema Multiagente basado en un modelo Ontológico para la búsqueda de Objetos de Aprendizaje.”

Tesis previa a la Obtención del título de Ingeniero en Sistemas

Autores:

- Guamán-Morocho, Gabriela-Paola
- Martínez-Pacheco, John-Carlos

Tutor:

- Ing. Jácome-Galarza, Luis-Roberto. Mg. Sc.

Autorizado por:

- Jácome-Galarza, Luis-Roberto, Coordinador de la Carrera CIS

LOJA-ECUADOR

2015

Certificación del Director

Loja, 27 de Julio de 2015

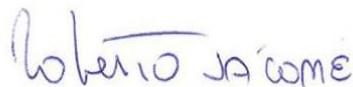
Ing. Luis-Roberto Jácome-Galarza, Mg. Sc.

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, revisado y corregido el trabajo de tesis “**SISTEMA MULTIAGENTE BASADO EN UN MODELO ONTOLÓGICO PARA LA BÚSQUEDA DE OBJETOS DE APRENDIZAJE**”, realizado por los egresados Gabriela Paola Guamán Morocho y John Carlos Martínez Pacheco, previo a la obtención del título de INGENIERO EN SISTEMAS.

En vista de que el mismo reúne los requisitos necesarios, autorizo su presentación y defensa ante el tribunal que se designe para el efecto.



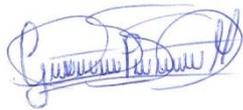
.....
Ing. Luis-Roberto Jácome-Galarza, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACION

Autoría

Nosotros **GABRIELA PAOLA GUAMAN MOROCHO** y **JOHN CARLOS MARTINEZ PACHECO**, declaramos ser autores del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestra tesis en el Repositorio Institucional – Biblioteca Virtual.



Firma:.....

Cédula: 1105137143.

Fecha: Siete de Agosto del 2015.



Firma:.....

Cédula: 1104121346

**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LOS AUTORES,
PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y
PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.**

Nosotros **GABRIELA PAOLA GUAMÁN MOROCHO** y **JOHN CARLOS MARTÍNEZ PACHECO** declaramos ser autores de la tesis titulada: **SISTEMA MULTIAGENTE BASADO EN UN MODELO ONTOLÓGICO PARA LA BÚSQUEDA DE OBJETOS DE APRENDIZAJE**, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, siete días del mes de agosto del dos mil quince.



Firma:.....

Autor: Gabriela Paola Guamán Morocho

Cédula: 1105137143.

Dirección: Loja (El Valle: Vía Oriental de Paso y Génova).

Correo electrónico:

gpguamanm@unl.edu.ec

paogaby-0803@hotmail.com

Teléfono: 073025040 **Celular:** 0980351258



Firma:.....

Autor: John Carlos Martínez Pacheco

Cédula: 1104121346

Dirección: Loja (Esteban Godoy: Germán Pitiur y Héctor Pilco).

Correo electrónico:

jcmartinezp@unl.edu.ec

jcmartinez@hotmail.com

Teléfono: 072546530 **Celular:** 0983732319

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Tribunal de Grado: Ing. Marco Augusto Ocampo Carpio, Mg. Sc.

Ing. Hartman José Torres Asanza, Mg. Sc.

Ing. Carlos Miguel Jaramillo Castro, Mg. Sc.

Dedicatoria

A Dios, a mis padres Cecilia y Georgi por ser ejemplo de perseverancia y haberme dado el mejor legado de un padre a su hijo la educación y a mis hermanos Sofía y Jorge por ser mi alegría e inspiración para ser mejor persona.

Gabriela Laola.

El presente trabajo de titulación está dedicado a Dios, la gloria y el mérito son suyos, él es y será mi fortaleza por siempre, agradecer a mi familia principalmente a mis padres que supieron apoyarme en todo momento, mi Padre que con sus consejos me motivo a seguir adelante en la vida y convertirme principalmente en un hombre de bien, mi Madre que es un sustento en mi vida, a mis hermanos que estarán siempre cuando los necesite brindándome su apoyo en todo lo que puedan, a mi compañera de tesis. Además sin duda alguna dedicar este logro a mi esposa Silvana que me motivo y brindo su ayuda, incluso en aquellos momentos que parecía llegar el fracaso. Principalmente quiero dedicar este logro a lo más hermoso y valioso que me ha regalado Dios y la vida mi hijo bello Gerald, ese motorcito que me inspira a seguir adelante para ser un gran profesional, padre y esposo.

John. Carlos

Agradecimiento

A la Universidad Nacional de Loja, a la Carrera de Ingeniería en Sistemas, por habernos dado la oportunidad de estar día a día en sus aulas y poder alcanzar nuestra meta profesional de la mejor manera.

A los docentes y personal administrativo que siempre estuvieron dispuestos a colaborar con toda la información requerida en el transcurso de nuestra carrera.

Y finalmente agradecemos de manera sincera a nuestro tutor de tesis Ing. Roberto Jácome, por su capacidad para guiar nuestras ideas, enriqueciendo nuestro conocimiento para el avance y finalización de este Trabajo de Titulación.

Índice de Contenidos

Certificación del Director	II
Autoría	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LOS AUTORES ..	IV
Dedicatoria	V
Agradecimiento	VI
Índice de Contenidos	VII
Índice de Tablas	XII
1. Título	1
2. Resumen	2
2.1. Abstract	3
3. Introducción	4
4. Revisión de Literatura	6
4.1. CAPÍTULO 1: Objetos de Aprendizaje	6
4.1.1. Definición	6
4.1.2. Atributos	7
4.1.3. Reutilización de los Objetos de Aprendizaje	8
4.1.4. Recolección de Metadatos	11
4.2. CAPÍTULO 2: Web Semántica.	13
4.2.1. Definición	13
4.2.2. Ontologías.....	14
4.2.2.1. Componentes	14
4.2.2.2. Lenguajes de representación	15
4.2.2.3. Consultas SPARQL	17
4.3. CAPÍTULO 3: Agentes Inteligentes	19
4.3.2. Características	19
4.3.3. Tipos de Agentes.....	20
4.3.4. Arquitecturas de tipos de agentes	21
4.3.5. Sistemas Multi-Agente	23
4.4. CAPÍTULO 4: Trabajos relacionados	24
4.4.1. CiteSeer.....	24

4.4.2.	LETIZIA.....	25
4.4.3.	ABIMETAGIS.....	27
4.5.	CAPÍTULO 5: Herramientas y tecnologías utilizadas	29
4.5.1.	Protegé.....	29
4.5.2.	JENA.....	29
4.5.3.	VIRTUOSO.....	30
4.5.4.	JADE.....	30
4.5.5.	IDE NETBEANS	31
4.5.6.	TWITTER BOOTSTRAP	31
4.5.7.	PRIMEFACES 5	31
4.5.8.	JavaScript	31
4.5.9.	CASCADE STYLE SHEETS (CSS).....	32
5.	MATERIALES Y METODOS.....	33
6.	RESULTADOS.	37
6.1.	FASE I: Revisar el estado del arte en lo referente a Objetos de Aprendizaje, Ontologías y Agentes Inteligentes	37
6.1.1.	Conclusión de la Revisión de Objetos de Aprendizaje.....	37
6.1.2.	Selección de los Estándares de Metadatos	39
6.1.3.	Selección de Repositorios de Objetos de Aprendizaje.....	41
6.1.4.	Análisis de trabajos relacionados.	43
6.1.5.	Analizar las metodologías y herramientas existentes para la construcción de Ontologías y Agentes Inteligentes.....	44
6.1.5.1.	Análisis y selección de Metodología para la Ontología.....	44
6.1.5.2.	Análisis y selección de Herramientas para la Ontología	48
6.1.5.3.	Análisis y selección de Metodología para la construcción de los Agentes Inteligentes.....	50
6.1.5.4.	Análisis y selección de la Herramienta para la construcción de los Agentes Inteligentes.....	52
6.2.	FASE II: Diseñar una Ontología con los parámetros que deben poseer los recursos a ser recuperados.....	55
6.2.1.	Ontología para la representación Objetos de Aprendizaje.....	55
6.2.1.1.	Introducción	55
6.2.1.2.	Especificación de requerimientos de la ontología	55
6.2.1.3.	Conceptualización	58

6.2.1.4.	Formalización.....	66
6.2.1.5.	Implementación.....	70
6.3.	FASE III: Construir un Sistema Multiagente para la búsqueda y recuperación de Objetos de Aprendizaje.....	74
6.3.1.	Conceptualización del Sistema Multi-agente.....	74
6.3.1.5.	Casos de Uso	75
6.3.1.6.	Modelo de Agentes.	76
6.3.1.7.	Modelo Objetivos y tareas.....	80
6.3.1.8.	Modelos de Interacción	82
6.3.1.9.	Modelo de Entorno.....	84
6.3.1.10.	Arquitectura del Sistema Multiagente	86
6.3.2.	Implementación.....	87
6.3.2.1.	Diagrama de componentes	87
6.3.3.2.	Codificación	88
6.4.	FASE IV: Evaluar el funcionamiento de la aplicación en la búsqueda y recuperación de Objetos de Aprendizaje.....	97
6.4.1.	Pruebas de funcionalidad.....	97
6.4.2.	Nivel Técnico.....	101
6.4.3.	Prueba de validación de Usuarios	101
7.	Discusión.....	102
7.1.	Desarrollo de la propuesta alternativa	102
7.2.	Valoración técnica económica ambiental	105
8.	Conclusiones.....	109
9.	Recomendaciones	110
10.	Bibliografía	111
11.	Anexos	114
11.1.	ANEXO 1: Análisis de requerimientos y diseño preliminar con la Metodología ICONIX	114
11.2.	ANEXO 2: Formato RDF/OWL generado desde Protegé.....	135
11.3.	ANEXO 3: Consultas SPARQL.....	141
11.4.	ANEXO 4: Encuesta	145
11.5.	ANEXO 5: Resultados de la encuesta realizada a los estudiantes de la carrera de Ingeniería en Sistemas.....	147

11.6. ANEXO 6: Anteproyecto	152
11.8. ANEXO 7: Artículo sobre el trabajo de titulación	164
11.7. ANEXO 8: Certificado de traducción del resumen	172
11.9. ANEXO 9: Licencia	173

Índice de Figuras

Figura 1. Conceptualización de un Objeto de Aprendizaje.....	6
Figura 2. Petición HTTP mediante el protocolo OAI-PMH.....	12
Figura 3. Web Semántica.....	13
Figura 4. Modelo de datos RDF.....	15
Figura 5. Estándar de Sentencias SPARQL.....	18
Figura 6. Arquitectura Reactiva.....	22
Figura 7. Arquitectura Deliberativa.....	23
Figura 8. Arquitecturas en capas horizontales y verticales.....	24
Figura 9. Interfaz de Letizia.....	27
Figura 10. Interfaz de ABIMETAGIS.....	28
Figura 11. Diagrama de Agentes ABIMETAGIS.....	29
Figura 12. Descripción de la Metodología propuesta.....	37
Figura 13. Estructura de Dublin Core.....	40
Figura 14. Estructura de LOM.....	41
Figura 15. Taxonomía de la Ontología.....	66
Figura 16. Definición de Conceptos.....	67
Figura 17. Definición de Relaciones.....	68
Figura 18. Definición de Propiedades.....	69
Figura 19. Diseño de la Ontología en Protegé.....	70
Figura 20. Fragmento de Código de una Clase OWL.....	71
Figura 21. Grafo generado por el Validador W3C.....	72

Figura 22. Metadatos de un Objeto de Aprendizaje.....	73
Figura 23. Esquema de extracción de metadatos.....	74
Figura 24. Simbología para el modelado de Agentes.....	76
Figura 25. Identificación de los casos de uso.....	77
Figura 26. Agente Administrador.....	79
Figura 27. Agente Servicios Web.....	79
Figura 28. Agente Interfaz.....	80
Figura 29. Agente Buscador.....	80
Figura 30. Diagrama de Objetivos Agente Buscador.....	81
Figura 31. Diagrama de Objetivos Agente Administrador.....	82
Figura 32. Diagrama de Objetivos Agente Servicios Web.....	82
Figura 33. Diagrama de Objetivos Agente Interfaz.....	83
Figura 34. Diagrama de Interacción 1.....	84
Figura 35. Diagrama de Interacción 2.....	84
Figura 36. Diagrama de Interacción 3.....	85
Figura 37. Arquitectura Multiagente para recuperación OAs.....	87
Figura 38. Diagrama de Componentes.....	88
Figura 39. Método processCommand – Agente Interfaz.....	89
Figura 40. Método setup agente Interfaz.....	90
Figura 41. Comportamiento temporizador agente Administrador.....	91
Figura 42. Comportamiento cíclico agente Administrador.....	92
Figura 43. Comportamientos de búsqueda agente Buscador.....	93
Figura 44. Comportamientos paralelos agente WSDC.....	94
Figura 45. Comportamientos paralelos agente WSLOM.....	95
Figura 46. Contenedor de Agentes.....	96
Figura 47. Agente Sniffer.....	97
Figura 48. Tabla de resultados prueba 1.....	99

Figura 49. Tabla de resultados prueba 2.....	100
Figura 50. Gráfica de prueba 1.....	101
Figura 51. Gráfica de prueba 1.....	101

Índice de Tablas

TABLA I. ATRIBUTOS DE UN OBJETO DE APRENDIZAJE.....	7
TABLA II. ESQUEMA DE BASE LOM.....	9
TABLA III. ESQUEMA DE BASE DUBLIN CORE.....	10
TABLA IV. COMPONENTES DE UNA ONTOLOGÍA.....	14
TABLA V. ESPECIFICACION RDF SCHEMA.....	16
TABLA VI. CARACTERÍSTICAS DE UN AGENTE INTELIGENTE.....	20
TABLA VII. TIPOS DE AGENTES.....	21
TABLA VIII. IMPORTANCIA DE LOS OA.....	39
TABLA IX. REPOSITORIOS DE OA SELECCIONADOS.....	42
TABLA X. ANÁLISIS DE TRABAJOS RELACIONADOS.....	44
TABLA XI. METODOLOGÍAS PARA EL DISEÑO DE ONTOLOGÍAS.....	47
TABLA XII. HERRAMIENTAS PARA LA CONSTRUCCIÓN DE ONTOLOGÍAS.....	50
TABLA XIII. METOLOGIAS PARA LA CONSTRUCCIÓN DE AGENTES.....	52
TABLA XIV. HERRAMIENTAS PARA LA CONSTRUCCIÓN DE AGENTES.....	54
TABLA XV. VOCABULARIOS DE LA ONTOLOGIA.....	59
TABLA XVI. GLOSARIO DE TERMINOS DE LA ONTOLOGIA.....	61
TABLA XVII. ACTORES DEL SISTEMA.....	76
TABLA XVIII. RESPONSABILIDADES DE AGENTES.....	79
TABLA XIX. RECURSOS HUMANOS.....	106
TABLA XX. RECURSOS MATERIALES.....	107
TABLA XXI. RECURSOS TÉCNICOS Y TECNOLÓGICOS.....	107
TABLA XXII. PRESUPUESTO DEL PROYECTO.....	108

1. Título

**“SISTEMA MULTIAGENTE BASADO EN UN MODELO ONTOLÓGICO PARA
LA BÚSQUEDA DE OBJETOS DE APRENDIZAJE”**

2. Resumen

El presente trabajo consiste en el desarrollo de un sistema Multiagente de búsqueda de Objetos de Aprendizaje denominado SMAS, que utiliza un modelo Ontológico como dominio de definición de los mismos. En la construcción de este sistema se implanta las bases de la Web Semántica, demostrando su utilidad junto con los Agentes de Software, esta mezcla de tecnologías es considerada como una futura versión de la Web Actual; mediante el uso de una ontología la información se encuentra mejor definida y los Agentes puedan extraer información de ella, en su entorno estos son los encargados de utilizar los Servicios Web de los repositorios disponibles en la Internet, los cuales nos proporciona los metadatos de los Objetos de Aprendizaje y son instanciados en la Ontología para su posterior búsqueda.

Para llevar a cabo este trabajo se empezó con la revisión literaria de Objetos de Aprendizaje, Ontologías y Agentes inteligentes, así como también un análisis de herramientas y metodologías. En base al análisis realizado se determinó la utilización de una metodología híbrida y robusta que permita incluir actividades tanto para el diseño de la Ontología y los Agentes, para lo cual se tomó las fases de ICONIX, Methontology e Ingenias; como resultado se obtuvo un análisis de requerimientos y a partir de estos se construye el modelo del dominio, casos de uso y prototipo de interfaz; para posteriormente desarrollar la fase de análisis y diseño preliminar; en la fase de diseño se realizó la definición de la Ontología y el diseño del entorno de los Agentes tomando las fases de análisis de requerimiento y diseño preliminar se define el comportamiento de estos; en la implantación se realizó el diagrama de componentes y codificó lo especificado en las fases anteriores.

Mediante las pruebas de funcionalidad y usabilidad se verificó el correcto funcionamiento del sistema según los requerimientos planteados inicialmente y con los resultados se constató el desconocimiento que existe sobre los recursos educacionales abiertos, con lo cual se concluye que este software es de gran ayuda ya que permite aumentar la eficacia a la hora de realizar las actividades de búsqueda de Objetos de Aprendizaje. Finalmente, para usar SMAS no se requiere de la adquisición de hardware costoso, éste se encuentra desarrollado con herramientas que son gratuitas; además es una gran solución, puesto que el uso de recursos digitales gratuitos disponibles en la Web incentiva la utilización de información fiable.

2.1. Abstract

This thesis work involves the development of a Multiagent system learning objects search called SMAS, which uses an Ontological model as domain defining them, in the construction of this system the basis of the Semantic Web is implemented, establishing its utility with the Software Agent. This mix of technologies is considered as a future version of the current Web, through the use of a Ontology, make the information be better defined and agents can extract information from it, in the environment these are responsible for using the Web services of available repositories on the Internet; which provides metadata of learning objects and they are instantiated in the Ontology for further search.

To carry out this work it was used the literature review of Learning Objects, Ontologies and Intelligent Agents, as well as an analysis of tools and methodologies to complete this work, so the use of a hybrid and robust methodology is determined to include activities for both the design of Ontology and Agents, in which the phases of ICONIX, Methontology and Ingenias, which as a result a requirements analysis was gotten and to build the domain model, case-use and interface prototypes was obtained based on them the preliminary design phase; in the design phase the ontology definition was made and design environment of Agents taking the phases of requirements analysis and preliminary design the performance was defined, in the implementation a components diagram was done and codified elements in the previous phases.

By using functionality and usability testing the proper functioning of the system was verified according to the requirements initially set and with the results it was possible to verify the lack of knowledge about open educational resources, so it is concluded that this software is helpful it allows verifying effectiveness in performing search activities and learning objects. Finally, to use SMAS does not require the purchase of expensive hardware, this is developed with tools that are free, and it is also a great solution because using free digital resources available on the Web encouraging the use of reliable information.

3. Introducción

En la actualidad la Word Wide Web ha alcanzado un amplio uso en la sociedad, existiendo una abrumadora cantidad de información. En el ámbito educativo existe gran cantidad y diversidad de material que puede contribuir al proceso enseñanza-aprendizaje, pero el usuario se ve expuesto a grandes bancos o sitios de información en los que la mayoría de ellos no corresponden a su interés por la falta de estrategias en la búsqueda. Este trabajo se ha enfocado en la utilización de Objetos de Aprendizaje que es cualquier recurso digital que puede ser utilizado para facilitar el aprendizaje, estos son accesibles desde repositorios vía Web; dichos materiales educativos han sido diseñados y creados con el propósito de maximizar el uso y la reutilización de un mismo material en distintas situaciones educativas, con el fin de generar conocimientos, habilidades y competencias en el ámbito de la educación.

Por lo expuesto anteriormente este trabajo busca estimular la utilización de Objetos de Aprendizaje y centralizar su búsqueda desde diferentes repositorios Web; para lo cual se ha desarrollado un software de búsqueda, utilizando diferentes tecnologías no muy usuales que están en etapa de crecimiento, entre ellas Agentes Inteligentes los cuales ayudan a percibir y responder al entorno en un tiempo razonable, exhibir comportamientos dirigidos por metas e interactuar con otros agentes; además se hace uso de Ontologías, las cuales nos permiten organizar la información por dominios especializados.

Este trabajo comienza con la presente introducción y se organiza como sigue: Revisión literaria en donde se describe los Objetos de Aprendizaje, sus atributos y los estándares bajo los cuales se los cataloga, como también la manera para recuperarlos desde los repositorios a través del cosechado de metadatos; Ontologías que comprenden la descripción como base semántica para la extracción de información; Agentes Inteligentes, trabajos relacionados y la descripción de las herramientas y tecnologías utilizadas. Luego se detalla los resultados de acuerdo a los objetivos planteados con tareas específicas para llegar a cumplir cada uno de ellos, en la Fase I de resultados esta todo el análisis teórico de Objetos de aprendizaje su importancia en el ámbito educacional, los estándares que los describen y repositorios con los cuales se trabajó, así como también un análisis de metodologías y herramientas para el diseño de Ontologías y Agentes Inteligentes; La Fase II que comprende el diseño de la Ontología, la cual describe el dominio de los Objetos de

Aprendizaje y es la base semántica que permitirá la búsqueda de los mismos; la Fase III describe la construcción del Sistema Multiagente el cual se completa con la Ontología diseñada en la fase II, y la Fase IV es la evaluación del Software donde se especifica las pruebas de funcionalidad, usabilidad y de usuarios.

A continuación se describe la discusión en la cual se plasma de forma general los objetivos de acuerdo a los resultados obtenidos. Por último, se plantean algunas conclusiones y recomendaciones.

4. Revisión de Literatura

Con la finalidad de brindar la mayor información respecto a este trabajo, a continuación se presenta algunas temáticas y términos más relevantes que se tomaron en cuenta para la realización del mismo.

4.1. CAPÍTULO 1: Objetos de Aprendizaje

Este capítulo introduce el concepto de Objetos de Aprendizaje, sus principales atributos para ser considerados recursos educativos gratuitos, aspectos notables para su uso y reutilización, metadatos que son información complementaria para la publicación de los mismos y los estándares bajo los cuales son creados que son la base para su recuperación desde Web. Además se describe el proveedor de servicios utilizado para la recuperación de los metadatos, mediante el protocolo OAI-PMH que ofrecen los repositorios de Objetos de Aprendizaje.

4.1.1. Definición

Según la IEEE, un objeto de aprendizaje (OA) es una entidad digital con características de diseño instruccional que puede ser: usado, reutilizado, o referenciado; todo ello con el objetivo de generar conocimientos, habilidades, aptitudes y competencias en el ámbito educativo [1].

Los objetos de aprendizaje son una conceptualización de material educativo gratuito más la utilización de metadatos que son información complementaria que se añade sobre los OA para su publicación, describiendo aspectos sobre su contenido con el único fin de que estos sean recuperados de manera fácil y puedan ser reutilizados [2].

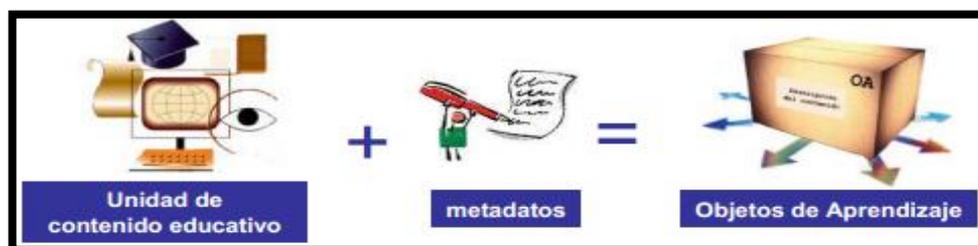


Figura 1. Conceptualización de un Objeto de Aprendizaje.

4.1.2. Atributos

Los Objetos de aprendizaje no pueden ser creados como otro recurso más de información aislada, por ende deben de poseer ciertos atributos que los diferencian de los demás recursos de la web, los cuales se resumen en la siguiente tabla [3]:

TABLA I.
ATRIBUTOS DE UN OBJETO DE APRENDIZAJE

Atributo	Descripción
Accesibilidad	Deben ser indexados para una localización y recuperación más eficiente; empleando estándares de metadatos.
Adaptabilidad	El OA debe de acoplarse a las necesidades de aprendizaje de cada individuo.
Autocontención Conceptual	Capacidad de los OA de poder auto explicarse.
Durabilidad	Un OA debe de procurar contener información que no se convierta en información obsoleta con el paso del tiempo.
Escalabilidad	Los OA deben de adaptarse con facilidad a estructuras más complejas dentro de su dominio de aprendizaje.
Gestión	Facilidad para obtener información concreta y correcta.
Interactividad	Facilidad para generar comunicación entre los sujetos involucrados en el proceso de aprendizaje.
Interoperabilidad	Funcionan independientemente del hardware y software
Portabilidad	Deben de adaptarse a diferentes plataformas sin necesidad de modificaciones.

4.1.3. Reutilización de los Objetos de Aprendizaje

Los objetos de aprendizaje para poder ser accedidos con mayor facilidad por los usuarios deben de contener características fundamentales que los describan; estas características son conocidas como Metadatos. Para la reutilización de los OA se pretende hacer uso de los metadatos, a partir de los cuales se crea una área del dominio del conocimiento mediante una Ontología.

4.1.3.1. Metadatos.

A través de los metadatos se obtiene un primer acercamiento con el objeto, conociendo rápidamente sus principales características. Los metadatos son un conjunto de atributos necesarios para describir un recurso como: el tema, el estilo pedagógico, el formato, el nivel de dificultad, el rango de edades y las restricciones de copyright; permitiendo con ello una fácil ubicación y recuperación [4].

Existen dos estándares de metadatos que describen cada uno de los OA que se crean, proporcionando una búsqueda rápida y efectiva en el repositorio donde se encuentre almacenado, surgen por la necesidad de compartir objetos que provienen de distintas fuentes y organizar su almacenamiento. Entre los estándares de metadatos, los cuales sirven para catalogar los objetos de aprendizaje, se encuentran el IEEE-LOM y Dublin Core [5].

4.1.3.1.1. Estándar IEEE-LOM

Es el estándar por excelencia para metadatos de objetos de aprendizaje y está patrocinado por el comité de Estandarización de Tecnologías Educativas del IEEE. EL principal objetivo de este estándar es facilitar la búsqueda, evaluación, la adquisición, y el uso de recursos educativos, pretendiendo facilitar el intercambio de objetos educacionales.

El Learning Object Metadata (LOM) es una especificación que define un conjunto de etiquetas para describir metadatos de un objeto de aprendizaje. Las etiquetas se estructuran en 9 categorías distintas [6]:

TABLA II.
ESQUEMA DE BASE LOM

Categorías	Descripción	Elementos
1. General	Los metadatos en esta categoría representan información general sobre el material educativo que describe el mismo como un todo.	Identificador, Título, Entrada de catálogo, Lengua, Descripción, Descriptor, Cobertura, Estructura y Nivel de agregación.
2. Ciclo de vida	Esta categoría agrupa metadatos referidos a la historia y estado actual del proceso de producción y mantenimiento por parte de los autores.	Versión, Estatus, Otros colaboradores.
3. Meta-información	Esta categoría agrupa información relativa a los metadatos en sí (de ahí su nombre).	Identificador, Entrada de catálogo, Otros colaboradores, Esquema de metadatos y Lengua
4. Técnica	Categoría que agrupa metadatos relativos a las características y requisitos técnicos del material en sí.	Formato, Tamaño, Ubicación, Requisitos, Comentarios sobre la instalación, Otros requisitos para plataformas y Duración.
5. Uso educativo	Categoría que agrupa metadatos relativos a los usos educativos del material.	Tipo de interactividad, Tipo de recurso de aprendizaje, Nivel de interactividad, Densidad semántica, Usuario principal, Contexto [Nivel educativo], Edad, Dificultad, Tiempo previsto de aprendizaje y Descripción.
6. Derechos	Categoría que agrupa metadatos relativos a los derechos de propiedad e intelectuales del material.	Coste, Copyright y otras restricciones, Descripción
7. Relación	Categoría de metadatos utilizados para establecer relaciones entre el material y otros materiales.	Tipo [naturaleza de la relación con el recurso], Recurso [recurso principal al que se refiere esta].
8. Observaciones	Anotaciones y comentarios sobre el material educativo.	Persona, Fecha y Descripción.
9. Clasificación	Metadatos para la clasificación del material en taxonomías.	Finalidad, Nivel taxón (taxonómico), Descripción, y Descriptor.

4.1.3.1.2. Estándar Dublin Core

El estándar Dublin Core (DC) un modelo de metadatos cuya finalidad es promover un vocabulario extenso y familiarizado. DC, es un sistema de quince descriptores, diseñado específicamente para proporcionar un vocabulario de características capaces de otorgar la información descriptiva básica sobre cualquier recurso bibliográfico [7].

TABLA III
ESQUEMA DE BASE DUBLIN CORE

Contenido	
Título	El nombre dado a un recurso.
Claves	Frases que describen el título o el contenido del recurso.
Descripción	Una descripción textual del recurso.
Fuente	Secuencia de caracteres para identificar unívocamente un trabajo.
Lengua	Idioma del contenido intelectual del recurso.
Relación	Enlazar los recursos relacionados.
Cobertura	Cobertura espacial y/o temporal del contenido intelectual del recurso
Propiedad Intelectual	
Creador	La persona u organización responsable de la creación del contenido intelectual del recurso.
Editor	La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual.
Colaboradores	Una persona u organización que haya tenido una contribución intelectual significativa.
Derechos	Son una referencia sobre derechos de autor.
Instanciación	
Fecha	Fecha en la cual el recurso se puso a disposición del usuario en su forma actual.
Tipo	La categoría del recurso.
Formato	Es el formato de datos de un recurso.
Identificador	Secuencia de caracteres utilizados para identificar unívocamente un recurso.

4.1.4. Recolección de Metadatos

Para acceder a los repositorios de OA, existen mecanismos de cosechado de metadatos que proporciona el protocolo de Iniciativa de Archivos Abiertos para la recolección de Metadatos (OAI-PMH), este protocolo permite realizar el intercambio de información para que desde puntos centralizados (proveedores de servicio) se puedan realizar búsquedas conjuntas sobre los metadatos de todos aquellos repositorios asociados (proveedores de datos) [8].

Este protocolo maneja peticiones Hypertext Transfer Protocol (HTTP), para emitir preguntas y obtener respuestas entre un servidor y un cliente. A continuación de detalla estas peticiones utilizadas para recuperar OA [9]:

- **Identify:** Se utiliza para recuperar información acerca de un repositorio. Parte de la información que se devuelve se requiere como parte del protocolo OAI-PMH. Repositorios pueden también emplear el verbo Identificar para devolver información descriptiva adicional.
- **GetRecord:** Se utiliza para recuperar un registro de metadatos individuales de un repositorio.
- **ListMetadataFormats:** Regresa los formatos que soporta el proveedor como LOM y Dublin Core.
- **ListRecords:** Regresa un conjunto de registros de OAs del repositorio.
- **ListIdentifiers:** Este es una forma abreviada de ListRecords, recuperar sólo los encabezados en lugar de registros.
- **ListSets:** Regresa el listado de conjuntos o clasificaciones por catálogo del repositorio.

En nuestro caso se utilizó la mayoría de estas peticiones, pero la más frecuente fue la petición ListRecords para recolectar los registros en formato XML de cada OA para ser posteriormente almacenado en la Ontología.

El ListRecords tiene argumentos opcionales que permiten la recolección selectiva de registros basados en conjunto de miembros y / o marca de fecha mediante peticiones HTTP [10].



Figura 2. Petición HTTP mediante el protocolo OAI-PMH.

El usuario realiza una petición que es enviada al proveedor de servicios, el cual, por medio del protocolo OAI-PMH, usa transacciones HTTP e idea un proveedor de datos que le regrese los metadatos almacenados en los recursos localizados dentro del repositorio. Estos metadatos son enviados primero al proveedor de datos, quien a su vez los envía al proveedor de servicio, en donde se encuentran instalados los recolectores. Dichos recolectores procesan los metadatos, que arrojarán todos los resultados de la búsqueda en formato XML, para que el usuario, al acceder a los recursos mostrados, pueda ver los documentos que se encuentran en el repositorio.

4.2. CAPÍTULO 2: Web Semántica.

Este capítulo de la memoria muestra los principales conceptos relativos a las Ontologías, los componentes necesarios para representar formalmente la información, así como también los lenguajes Ontológicos RDF, RDFs y OWL y el lenguaje de consultas SPARQL que nos permitirá realizar búsquedas en la Ontología.

4.2.1. Definición

La Web Semántica es una Web extendida y dotada de mayor significado, en la que los internautas podrán encontrar respuestas a sus preguntas de forma rápida y sencilla gracias a una información mejor definida.

Para conseguirlo, se propone describir los recursos de la web con representaciones procesables y entendibles no sólo por las personas, sino también por programas que puedan reemplazar a los seres humanos en tareas rutinarias de búsqueda que requieren gran cantidad de tiempo y esfuerzo por nuestra parte. La semántica de los elementos Web, se deberá codificar mediante lenguajes de metadatos y ontologías. Al dotar de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común. Para conseguir comprender el contenido de los documentos de la Web, la información se debe codificar mediante lenguajes de Metadatos, Ontologías y la utilización de Agentes Inteligentes que resuelven los problemas ocasionados por una Web carente de significado [11].

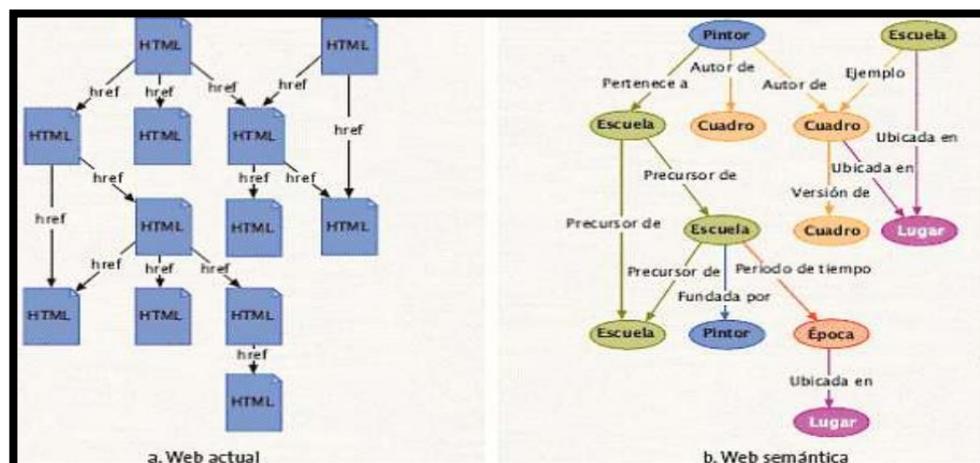


Figura 3. Web Semántica.

4.2.2. Ontologías

Son el medio principal para alcanzar los objetivos planteados por la Web Semántica. Más concretamente, una Ontología está formada por una taxonomía relacional de conceptos y por un conjunto de axiomas o reglas de inferencia mediante los cuales se podrá inferir nuevo conocimiento [12]. Mediante la Ontología se representa el modelo del conocimiento de los Objetos de Aprendizaje para compartir conocimiento utilizando un mismo vocabulario de forma coherente y consistente.

4.2.2.1. Componentes

Las ontologías tienen componentes que se utilizan para representar un determinado dominio, consiguiendo una representación formal de la información de la Web, se pueden procesar dicha información e inferir más información de ella. Entre los componentes esenciales de estos sistemas están: conceptos, relaciones, funciones, instancias y axiomas.

Las ontologías tienen los siguientes componentes que sirven para representar el conocimiento de un dominio [13]:

TABLA IV.
COMPONENTES DE UNA ONTOLOGÍA.

Componente	Definición
Clases	Se utilizan para definir conceptos que se intentan formalizar. Un concepto es un conjunto de instancias con cierto interés para el modelo de dominio, y con unas características comunes que nos permiten agruparlas.
Atributos	Representan características de los conceptos y son descritos por medio de un valor básico
Relaciones	Representan la interacción y enlace entre los conceptos del dominio.
Funciones	Son un tipo concreto de relación donde se idéntica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.
Instancias	Es la representación en el mundo real de un concepto. Todos los atributos de una instancia tienen un valor concreto.

4.2.2.2. Lenguajes de representación

Para que las ontologías sean útiles para la Web Semántica, esto es que puedan ser comunicables, procesables y reutilizables, debe poder ser definida mediante lenguajes formales apropiados. La Web Semántica utiliza fundamentalmente los lenguajes RDF, RDFS y OWL para definir y estructurar los datos de la Web actual, a continuación describe los lenguajes [14].

4.2.2.2.1. RDF

Resource Description Framework (RDF) está diseñado para representar conocimiento y relaciones entre ellos, provee una semántica simple, a la vez que este modelo de datos puede ser representado en sintaxis XML.

El modelo de datos RDF se basa en la idea general de que las cosas que queramos describir tienen ciertas propiedades, y estas propiedades a su vez tienen valores. Esta idea tiene la forma sujeto-predicado-objeto y en terminología RDF se conoce como tripletas.

En la figura 4, se muestra de forma gráfica el modelo de datos RDF donde el recurso "Pintor" tiene una propiedad "Pertenece a" y esta a su vez tiene un valor "Escuela" [15].

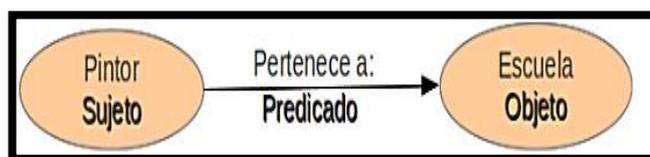


Figura 4. Modelo de datos RDF.

4.2.2.2.2. RDF Schema

Como se ha visto, RDF define un modelo simple que no proporciona mecanismos para la descripción de propiedades, ni la relación entre estas propiedades y otros recursos. La necesidad de una definición de vocabularios para el modelo de datos exige una extensión del vocabulario RDF, siendo RDF Schema (RDFS) una extensión que proporciona las facilidades necesarias para describir las propiedades y las clases de los recursos RDF.

RDFs define una serie de términos que pueden ser usadas en las sentencias RDF, ya que son conceptos especializados de recursos RDF predefinidos con sus propios significados específicos, proporcionando mecanismos para describir grupos de recursos relacionados, y las relaciones entre estos recursos [15].

RDFs proporciona las directrices para distribuir los lenguajes RDF, que a su vez tiene una estructura basada en el lenguaje XML.

TABLA V.
ESPECIFICACION RDF SCHEMA.

Clases	
Nombre	Descripción
rdfs:Resource	Clase a la que pertenecen todos los recursos.
rdf:Property	Clase para las propiedades RDF. Instancia de rdfs:Class.
rdfs:Class	Declaración de recursos como clases para otros recursos.
rdfs:Literal	Clase para los valores literales, cadenas y enteros.
Property	
Nombre	Descripción
rdfs:range	Especifica el rango de una propiedad. Recursos que aparecen como objetos.
rdfs:domain	Especifica el dominio de una propiedad. Dominio: Recursos que aparecen como sujetos.
rdf:type	Propiedad para indicar que un recurso es una instancia de una clase.
rdfs:subClassOf	Permite definir jerarquías de propiedades.
rdfs:label	Propiedad para dar una versión entendible del nombre de un recurso.
rdfs:comment	Propiedad para proporcionar una descripción de un recurso entendible por el usuario.

4.2.2.2.3. OWL

Web Ontology Language (OWL) es un lenguaje de marcado para compartir y publicar ontologías en la web desarrollado por la W3C. Nace de la necesidad de crear un lenguaje de ontologías para compartirlas en la web, y posee más capacidad de interpretación de los datos que XML, RDF o RDFS, y es por eso por lo que está en una capa superior en la estructura de la web semántica. No es un lenguaje independiente de RDFS, es un lenguaje complementario que permite ampliar las capacidades de modelado de RDFS o RDF [16].

4.2.2.3. Consultas SPARQL

SPARQL es el lenguaje estándar de la W3C para consultas de grafos RDFs; permite el acceso y recuperación de información, estructurada y semi-estructurada mediante la consulta de relaciones desconocidas y realizando combinaciones complejas de bases de datos diferentes en una sencilla y única consulta [17].

4.2.2.3.1. Sintaxis SPARQL

SPARQL mantiene una sintaxis similar a SQL, mantiene una estructura declarativa con la típica sintaxis basada en cláusulas [17-18].

- **PREFIX:** Es equivalente al namespace en XML, en lugar de usar un url varias veces se utiliza un prefix.
- **SELECT:** Indica cada elemento a seleccionar: Sujeto, predicado, objeto. Precedido de un símbolo '?' o '*' para representar todos los elementos.
- **FROM:** Indica La tabla a consultar.
- **WHERE:** Siendo la parte más importante de la consulta. Para filtrar tripletas, es posible establecer una parte de la tripleta (sujeto, predicado, objeto) como variable, asignándole un nombre precedido de símbolo '?'.

4.2.2.3.2. Sentencias SPARQL

Las consultas SPARQL se realizan sobre esquemas ontológicos, aunque son muy parecidos al SQL QUERY, SPARQL no realiza consultas a las tablas comunes que encontramos en bases de datos relacionales, sino a recursos RDF, mediante tripletas.

El estándar establecido por la W3C es el siguiente [18]:

```
SELECT <<nombre_del_recurso>>
WHERE
{
    ?Sujeto ?Predicado ?Objeto
}
```

Figura 5. Estándar de Sentencias SPARQL.

Las consultas SPARQL cubren tres objetivos:

- Extraer información en forma de URIs y literales.
- Extraer subestructuras RDFs.
- Construir nuevas estructura RDF partiendo de resultados de consultas.

4.3. CAPÍTULO 3: Agentes Inteligentes

En este capítulo se aborda las temáticas referentes a Agentes Inteligentes, en las cuales se fundamenta el presente trabajo. Temáticas como tipos de agentes, arquitecturas de agentes, siendo estas necesarias para la construcción del Sistema Multi-agente.

4.3.1. Definición

Un agente es un sistema computacional capaz de actuar de manera autónoma para satisfacer sus objetivos y metas, mientras se encuentra situado persistentemente en su medio ambiente [19]. El uso de Agentes Inteligentes (AI) permite optimizar tareas, entre las que destacan el ahorro de tiempo y efectividad en la búsqueda de información.

4.3.2. Características

Un agente inteligente al tratarse de una entidad autónoma requiere de ciertas características para una correcta ejecución de sus objetivos y metas, por ende debe de contemplar las siguientes características [20]:

TABLA VI.

CARACTERÍSTICAS DE UN AGENTE INTELIGENTE.

Característica	Descripción
Reactividad	Permite percibir su entorno y responder a los cambios en él.
Cooperación y Coordinación	Permite comunicarse con otros agentes para realizar tareas.
Autonomía	Permite operar sin intervención directa de un ente.
Distribución de tareas	Permite establecer límites en la resolución de problemas.
Movilidad	Permite el desplazamiento en una red electrónica.
Paralelismo	Permite la ejecución simultánea de tareas.

4.3.3. Tipos de Agentes

Dentro del estudio de los Agentes Inteligentes es recomendable conocer y analizar los diferentes tipos existentes, para con ello poder definir correctamente el tipo de agente adecuado para la realización del presente trabajo. Existe una amplia clasificación de los agentes Inteligentes, pueden clasificarse dependiendo de su movilidad, de la capacidad para resolver problemas, según sus atributos entre otros.

A continuación se describe una breve clasificación de los tipos de agentes [21-22].

TABLA VII.
TIPOS DE AGENTES.

Agente	Ámbito.	Características.
Según la capacidad de resolver problemas		
Reactivos	Reaccionan en base a estímulo-respuesta.	Simples, inteligencia limitada
Deliberativos	Razonan acerca de sus intenciones y conocimientos	Coordinan con otros agentes
Según los atributos		
Cooperativos	Enfatizan en la autonomía y la cooperación	Tienen fuentes de información, permiten interconexión e interoperabilidad
Interfaz	Enfatizan en la autonomía y aprendizaje	Proporcionan asistencia y se adapta a las preferencias del usuario
Según la movilidad		
Móviles	Recorren redes WAN, interactuando con otros hosts	Recoger información, realizando tareas del usuario
Estáticos	Instalados en un sistema, pueden interactuar con otros agentes	Realizan tareas específicas
Otros		
Información/ Internet	Pueden ser móviles o estáticos, con cambios respectivamente	Manejar, manipular, coleccionar información
Híbridos	Combinación de uno o más agentes	

4.3.4. Arquitecturas de tipos de agentes

Existen diferentes tipos de arquitecturas para la construcción de agentes, los cuales se pueden clasificar de acuerdo a su comportamiento. A continuación se detallara las arquitecturas más utilizadas.

4.3.4.1. Arquitecturas Reactivas

Los agentes actúan de manera estímulo-respuesta. La arquitectura de los agentes reactivos está basada en capas especializadas en cada tipo de estímulos que se pueden estructurar en paralelo, o en cadena si los resultados de una capa son la entrada de la siguiente [23].

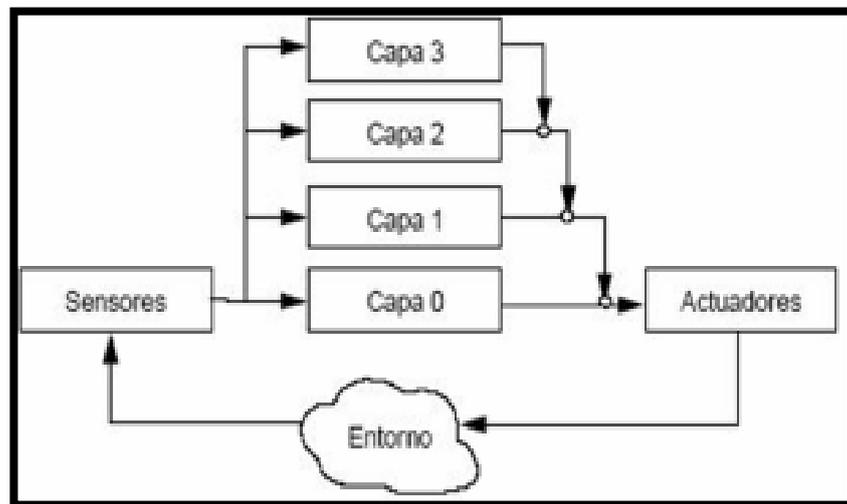


Figura 6. Arquitectura Reactiva.

En la Figura 6 se muestra como se distribuye la arquitectura reactiva, cuando se produce un estímulo en el entorno, los sensores captan el estímulo y lo envían a la capa respectiva que produce una respuesta, la respuesta es enviada a los actuadores los cuales visualizan esta respuesta en el entorno.

4.3.4.2. Arquitectura Deliberativa

Los agentes tienen un enfoque intencional, siguen el modelo Belief, Desire, Intention (BDI) está caracterizada por el hecho de que los agentes que la implementan están dotados de los estados mentales de creencias, deseos e intenciones [23].

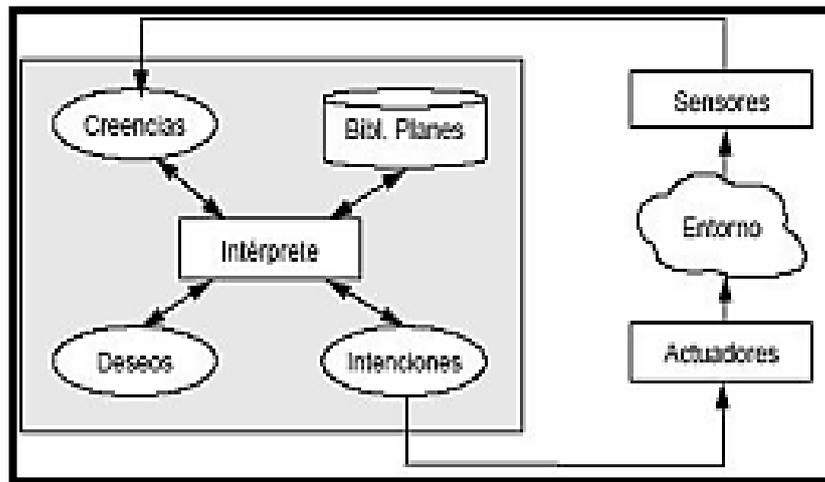


Figura 7. Arquitectura Deliberativa

En la Figura. 7 se observa la arquitectura deliberativa, donde los sensores obtienen la respuesta del entorno, la cual hace cambiar la creencia del agente; esta creencia ejecuta un objetivo o deseos, estos ejecutan ciertos planes para poder cumplir intenciones que son objetivos a corto plazo. Una vez cumplidos los objetivos a corto plazo son enviados a los actuadores.

4.3.4.3. Arquitectura Híbrida

La arquitectura consiste en los subsistemas de percepción y acción, constando con diferentes capas. Por su propia naturaleza estas arquitecturas son propicias para una estructuración por capas, que puede ser:

- Vertical: sólo una capa tiene acceso a los sensores y actuadores.
- Horizontal: todas las capas tienen acceso a los sensores y a los actuadores.

Estas capas pueden ser reactivas y deliberativas, las cuales son independientes de su acción [23].

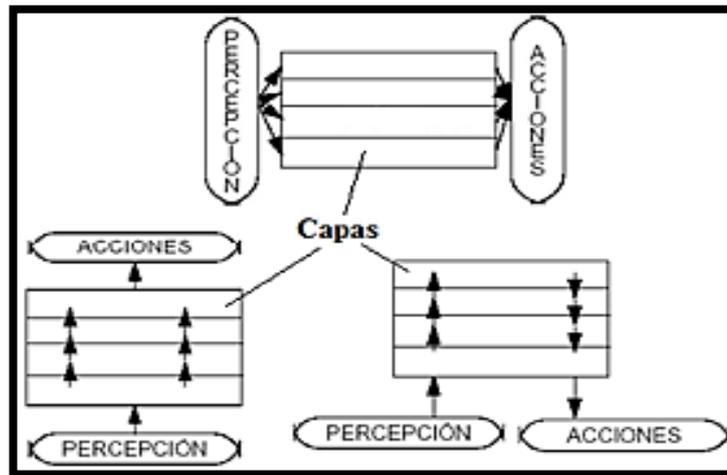


Figura 8. Arquitecturas en capas horizontales y verticales

En la Figura 8 se puede observar dos tipos de arquitecturas híbridas tanto horizontales como verticales y como es el flujo de información entre capas.

4.3.5. Sistemas Multi-Agente

Los Sistemas Multi-agentes (SMA) están compuestos de un conjunto de agentes que operan e interactúan en un ambiente con el fin de resolver un problema particular. Este paradigma presenta una nueva forma de análisis, diseño e implementación de sistemas de software complejos y ha sido utilizado para el desarrollo de sistemas de recomendación [24].

Este paradigma permite abordar sistemas que requieren de un análisis, diseño e implementación complejo, aprovechando las posibilidades que se brindan al resolverlo a través de su distribución del conocimiento, necesario para enfrentar el problema. Las principales características de un SMA son que cada agente tiene la capacidad para resolver una parte del problema con un limitado punto de vista, que no hay un sistema global de control y que la información está descentralizada. Los SMA están siendo ampliamente utilizados en problemas asociados con ambientes de enseñanza/aprendizaje, ya que se pueden descomponer en sub-problemas que son solucionados por agentes independientes que intercambian información e interactúan para conseguir los objetivos de enseñanza [24-25].

4.4. CAPÍTULO 4: Trabajos relacionados

Este apartado presenta los trabajos relacionados al sistema propuesto, tomando especial énfasis a los casos relacionados con Agentes Inteligentes en la búsqueda y recuperación de contenidos, para así hacer notar que la realización de este trabajo es factible de acuerdo a los casos estudiados.

A continuación se describen las características de algunos ejemplos de agentes de información que pueden ser utilizados desde la red.

4.4.1. CiteSeer

CiteSeer es un motor de búsqueda de la biblioteca y de la evolución de la literatura científica digital, que se ha centrado principalmente en la literatura de las ciencias de la computación y la información, tiene como objetivo mejorar la difusión de la literatura científica y de proporcionar mejoras en la funcionalidad, facilidad de uso, la disponibilidad, el costo, la integralidad, la eficiencia y oportunidad en el acceso a los conocimientos científicos y académicos [26].

4.4.1.1. Funcionamiento del Agente.

- Recorre la web y descarga artículos en diferentes formatos que convierte a texto.
- Analiza los artículos para extraer las citas (incluye un algoritmo para identificar los diferentes formatos de cita).
- De cada artículo se extrae la siguiente información: URL del fichero, cabecera, resumen, introducción si existe, citas, contexto de las citas y texto completo.
- Se extrae cada cita y se localiza el título, autor, año, páginas y formato de cita.
- Crea una base de datos con las citas.

4.4.1.2. Ventajas respecto a los índices tradicionales

- Incluye los artículos más recientes, desde el momento en el que están disponibles en web.
- Posibilidad de crear bases de datos actualizadas sin limitarse a un conjunto restringido de publicaciones.

- Funciona de forma totalmente autónoma: reduce los costes de indización manual. Refleja el impacto de un artículo y su contexto (browsing), esto es, cómo un artículo dado es citado en las publicaciones posteriores.
- Inconvenientes respecto a los índices tradicionales: No cubre publicaciones significativas, sólo las disponibles en web.
- No distingue subcampos de forma precisa. Aunque utilizan bases de datos de nombres de autores y revistas (ficheros de autoridades automatizados) para ayudar a identificar los subcampos.

4.4.1.3. Utilidades en investigación científica y técnica

- Localización de artículos por palabras clave, por la información contenida en las citas o mediante un vector de similitud de palabras.
- Recuperación de artículos sobre un ámbito específico de conocimiento siguiendo redes de citas.
- Evaluación y clasificación de artículos, autores y revistas según el número de citas recopiladas.
- Identificación de las tendencias en investigación.

4.4.2. LETIZIA

Letizia es un agente de interfaz de usuario que ayuda a un usuario a navegar por la World Wide Web. A medida que el usuario acciona un navegador Web convencional, el agente da seguimiento del comportamiento de los usuarios y los intentos de anticipar temas de interés haciendo una exploración concurrente [27].

4.4.2.1. Funcionamiento del Agente

- Letizia es un agente autónomo de interfaz que permite una acción cooperativa y continua entre el usuario y un agente de búsqueda en el espacio de la web.
- Registra la URL elegida y lee las páginas para formarse un perfil de interés del usuario.

- Realiza un simple cálculo de frecuencia de palabras claves que usa para analizar páginas, como medida de recuperación de información.
- Letizia siempre va a estar activo, buscando el espacio de la Web cercano a la posición del usuario.
- Usa la propia interfaz de Netscape para presentar sus resultados, en una ventana independiente en que el agente señala páginas que probablemente sean de interés.

4.4.2.2. Interfaz

Consiste en tres ventanas de Netscape. Por defecto, la ventana de la izquierda se reserva para el usuario que realiza la actividad. El usuario puede navegar en esta ventana de manera normal, y puede ignorar completamente cualquier actividad del agente.

Otras dos ventanas menores, situadas del lado derecho una arriba de la otra, están bajo el control de agente.

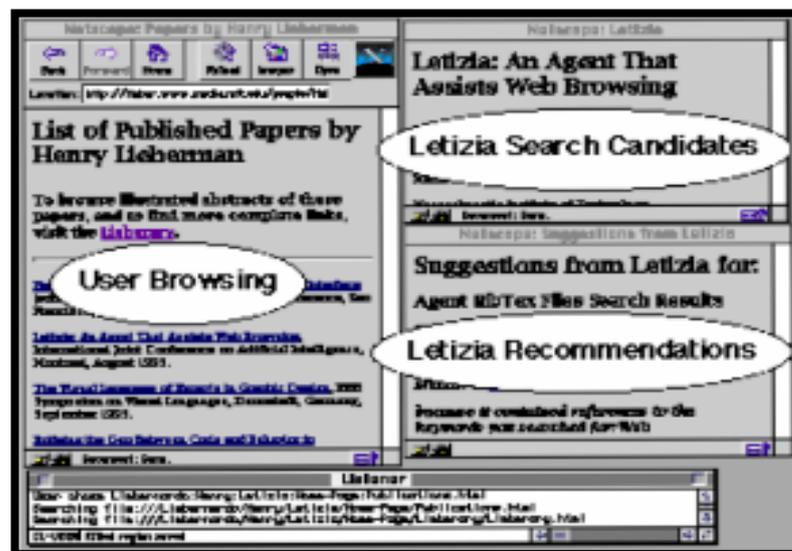


Figura 9. Interfaz de Letizia.

4.4.3.2. Funcionamiento del Sistema Multiagente.

La parte medular de ABI-METAGIS esta sostenida por los agentes, que mediante la ayuda de razonadores y mecanismos de inferencia manejan la lógica del negocio mediante la comprensión de las ontologías creadas para el dominio de la información geográfica. Utiliza una ontología para la comunicación de los agentes, para que ellos puedan entender las consultas y respuestas que se envían. Esto se lo hace básicamente entre el agente interprete y el agente buscador, puesto que ambos agentes usan una ontología, el agente interprete recoge el parámetro de búsqueda para enviar un mensaje al agente buscador quien recibe y busca en el modelo de conocimiento.

4.4.3.3. Diagrama de Agentes.

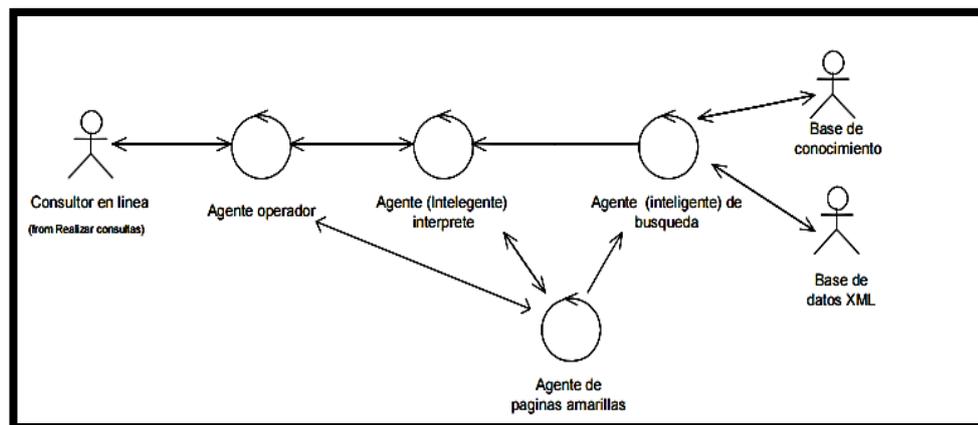


Figura 11. Diagrama de Agentes ABIMETAGIS.

- **Agente Operador:** Responde a una solicitud de búsqueda de un agente interprete disponible para que interactúe con el usuario.
- **Agente Intérprete:**
 1. Solicita una lista de agentes de búsqueda de metadatos.
 2. Solicita una búsqueda de metadatos apegados a ciertos parámetros.
 3. Presenta resultados de búsqueda.
- **Agente de búsqueda:**
 1. Responde a una solicitud de búsqueda de metadatos.
 2. Responde a una solicitud de entrega de un metadato específico

4.5. CAPÍTULO 5: Herramientas y tecnologías utilizadas

En este apartado se detallan las herramientas seleccionadas para la realización del sistema propuesto; considerando para su elección las características y funcionalidades que presenta cada una de las mismas. A continuación se presenta las herramientas seleccionadas.

4.5.1. Protegé

Protegé es una herramienta para el desarrollo de Ontologías y sistemas basados en el conocimiento creada en la Universidad de Stanford. Dentro de la plataforma de Protegé existen dos maneras de modelar ontologías por medio de los editores Protegé-Frames y Protegé-Owl.

Entre sus funcionalidades para Ontologías pueden ser exportadas a distintos formatos incluyendo RDF(S), OWL, XML Schema. Protegé se basa en Java, es extensible lo cual hace que sea una base flexible para un rápido prototipo y desarrollo de aplicaciones [29].

4.5.2. JENA

Es un Framework Java que permite escribir aplicaciones para la Web Semántica, es de libre distribución y permite escribir sobre RDF como también tiene soporte para OWL; además se puede realizar consultas SPARQL que permiten el razonamiento sobre las Ontologías. Su arquitectura tiene [30]:

- Una API para la lectura, procesamiento y escritura de datos RDF en XML, N-Triples (RDF Triple Language).
- Una API para la gestión de ontologías OWL (Web Ontology Language) y RDFS.
- Un motor de inferencia para razonar sobre ontologías RDF y OWL.
- Almacenamiento para una gran cantidad de tripletas RDF .
- Un motor de búsqueda compatible con la última especificación de SPARQL .
- Servidores para permitir que los datos RDF se puedan publicar en otras aplicaciones utilizando diferentes protocolos, incluyendo SPARQL.

4.5.3. VIRTUOSO

Es un tipo de base de datos para entornos de la Web Semántica, es un nivel empresarial de servidor de datos Multi-modelo innovador para las empresas ágiles. La arquitectura híbrida revolucionaria del servidor permite comunicar diferentes funcionalidades de tradicionalmente distintos servidores, dentro de un producto único

Este servidor permite guardar millones de tripletas para realizar consultar SPARQL de manera ágil, guarda directamente los datos RDF en una serie de tablas del motor de bases de datos relacionales. En concreto estos datos son guardados como tripletas por una fila de la tabla, donde no sólo se guardan las tripletas que representan a los datos, sino que también se guarda el grafo al que pertenece dicha triplete [31].

4.5.4. JADE

Java Agent Development (JADE) es un framework de agentes de software que permite crear agentes específicos para satisfacer las necesidades del usuario que lo utiliza; además ofrece una plataforma donde los agentes pueden ser ejecutados.

Algunas de las principales características que convierte a JADE en unos de los principales frameworks para el desarrollo de agentes es el cumplimiento de las especificaciones creadas por la Foundation for Intelligent Physical Agents (FIPA); se encuentra desarrollado en el lenguaje de programación Java y distribuido bajo una licencia LGPL.

El framework JADE ofrece considerables funcionalidades, entre las que destacan [32]:

- Contenedores, que proveen todos los servicios necesarios para albergar y ejecutar los agentes.
- Tiene una identidad única para toda la plataforma.
- Se comunica con otros agentes mediante mensajes asincrónicos, permite: el ahorro de recursos y un rápido cambio entre comportamientos.

4.5.5. IDE NETBEANS

Sirve para poder escribir, compilar, depurar y ejecutar programas. Permite realizar tareas complejas como: Pruebas de servicios web, creación de páginas JSF, creación de manejadores de JPA, etc.

Es un proyecto de código abierto de gran éxito con una gran base de usuarios y una comunidad en constante crecimiento [33].

4.5.6. TWITTER BOOTSTRAP

Es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript adicionales.

La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores.

4.5.7. PRIMEFACES 5

Es un componente para JSF de código abierto que cuenta con un conjunto de componentes que facilitan la creación de las aplicaciones web.

Las principales características de Primefaces son [34]:

- Soporte nativo de Ajax, incluyendo Push/Comet.
- Kit para crear aplicaciones web para móviles.
- Es compatible con otras librerías de componentes, como JBoss RichFaces.
- Uso de javascript no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque <script>).
- Es un proyecto open source, activo y bastante estable entre versiones.

4.5.8. JavaScript

Es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos.

JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias [35].

4.5.9. CASCADE STYLE SHEETS (CSS)

Es un mecanismo simple para añadir estilo a los documentos Web; es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.

El CSS es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores .

5. MATERIALES Y METODOS.

Para el presente Trabajo de Titulación se utilizó diferentes métodos de investigación así como técnicas de recolección de información; las cuales permitieron realizar un estudio de la información recolectada y adquirir los conocimientos necesarios para dar solución al problema de investigación. Para el proceso de desarrollo se tomó en cuenta la utilización de una metodología formal que ayude a obtener resultados de calidad, por lo cual se estimó conveniente utilizar la metodología ICONIX la cual propone un ciclo de vida a seguir. Adicional a ello se tomó en cuenta la utilización de dos metodologías más, una para el diseño de la Ontología y otro para la construcción de los Agentes Inteligentes, ya que el diseño de la Ontología y los Agentes Inteligentes son la parte esencial del Trabajo de Titulación.

5.1. Métodos de investigación

5.1.1. Recolección bibliográfica

El uso de esta técnica nos permitió recolectar información necesaria para poder sustentar la base teórica del Trabajo de Titulación, mediante consultas en fuentes bibliográficas confiables, casos de éxito, artículos científicos, revistas indexadas; entre otras.

5.1.2. Estudio de Casos

Este método nos permitió realizar una exploración e investigación en profundidad de problemas específicos, basados en experiencias ya vividas.

En base al estudio de estos casos podemos afirmar que la realización de Trabajo de Titulación resulto viable y factible de realizar.

5.1.3. Observación activa

Nos permitió palpar de forma directa el entorno en donde se ejecutan los Agentes Inteligentes y así analizar los comportamientos para que funcionen de acuerdo a lo requerido.

5.1.4. Método Analítico:

La utilización de este método nos permitió descomponer en fases el Trabajo de Titulación y de esta forma analizar Objetos de Aprendizaje, Ontologías y Agentes Inteligentes, sus herramientas y tecnologías de manipulación para integrar en un todo.

5.2. Metodologías

Se utilizó una metodología formal para el desarrollo del software ICONIX, la cual se complementó con una metodología para el diseño de la Ontología y otra para la construcción del Sistema Multiagente; las cuales son la parte esencial de todo proyecto ya que permite guiar y organizar las actividades para cumplir los objetivos planteados.

De esta manera de determino la utilización de una metodología híbrida y robusta que permita incluir actividades tanto para el diseño de la Ontología como de los Agentes.

5.2.1. Metodología para el diseño de la Ontología

Methontology, proporciona una guía para todos los aspectos clave del proceso de ingeniería de la ontología; es decir el desarrollo, la reutilización de los recursos ontológicos/no ontológicos, la evolución y el mantenimiento de las ontologías.

A continuación se detalla cómo se desarrolló cada una de las fases de esta metodología:

- **Especificación:** En esta fase determinamos por qué se construye la ontología, cuál será su uso, quiénes serán sus usuarios finales y requerimientos (preguntas que responderá la Ontología).
- **Conceptualización:** Se convierte una percepción informal del dominio en una especificación semi-formal, para lo cual utiliza un conjunto de representaciones, basadas en notaciones tabulares y gráficas, que pueden ser fácilmente comprendidas por los expertos de dominio y los desarrolladores de ontologías. El resultado de esta actividad es el modelo conceptual de la ontología.
- **Formalización:** Se transforma el modelo conceptual en un modelo formal o semi-computable.

- **Implementación:** Se construye los modelos computables en un lenguaje de ontologías (Ontolingua, RDF Schema, OWL, etc.). La mayor parte de las herramientas de ontologías permiten llevar a cabo esta actividad de manera automática.

5.2.2. Metodología para la construcción de Agentes

INGENIAS es una metodología intuitiva que utiliza modelos como mecanismo de especificación, su desarrollo está soportado por herramientas que permiten el procesamiento automático de los modelos generados.

Los modelos desarrollados se detallan a continuación:

- **Modelo de agentes:** Describe agentes particulares y los estados mentales en que se encontrarán en su ciclo de vida.
- **Modelo de objetivos y tareas:** se usa para asociar el estado mental del agente con las tareas que ejecuta.
- **Modelo de interacción:** describe cómo se coordinan y comunican los agentes.
- **Modelo de entorno:** define qué existe alrededor del nuevo sistema.
- **Modelo de Arquitectura:** define la forma en que se agrupan los agentes, la funcionalidad del sistema y qué restricciones existen sobre el comportamiento de los agentes.

5.2.3. Metodología de desarrollo del Software.

La metodología formal para el desarrollo del software fue ICONIX, a continuación se menciona como se ha desarrollado cada una de las fases de esta metodología y como se fusiono en una metodología híbrida y robusta, incluyendo la metodología Methontology e Ingenias:

- **Análisis de requerimientos:** En esta fase se aplicó el estándar IEEE 830 (ver Anexo 1) el cual permitió realizar una especificación de requerimientos de forma clara, para seguidamente construir los diagramas: modelo del dominio, modelo de casos de uso, prototipo de interfaz de usuario.

- **Análisis y diseño preliminar:** En esta fase se construye los diagramas de casos de uso, en los cuales se describió los flujos de la aplicación tanto normales como alternos.
- **Diseño:**
 - **Definición de la Ontología:** Se toma las actividades propuestas por Methontology para la definición de la Ontología que modela el contexto de los Objetos de Aprendizaje.
 - **Conceptualización del Sistema Multiagente:** tomando las fases de análisis de requerimiento, diseño preliminar y la definición de la Ontología mencionadas anteriormente se establece el comportamiento de los Agentes de Software con la metodología Ingenias la cual propone la construcción de algunos modelos.
- **Implementación y pruebas:** En esta fase se realizó el diagrama de componentes y se codificó todo lo que se ha especificado en los diagramas de las fases anteriores, así como también se realizaron las pruebas de funcionalidad y usabilidad.

A continuación se muestra el esquema de la Metodología propuesta:

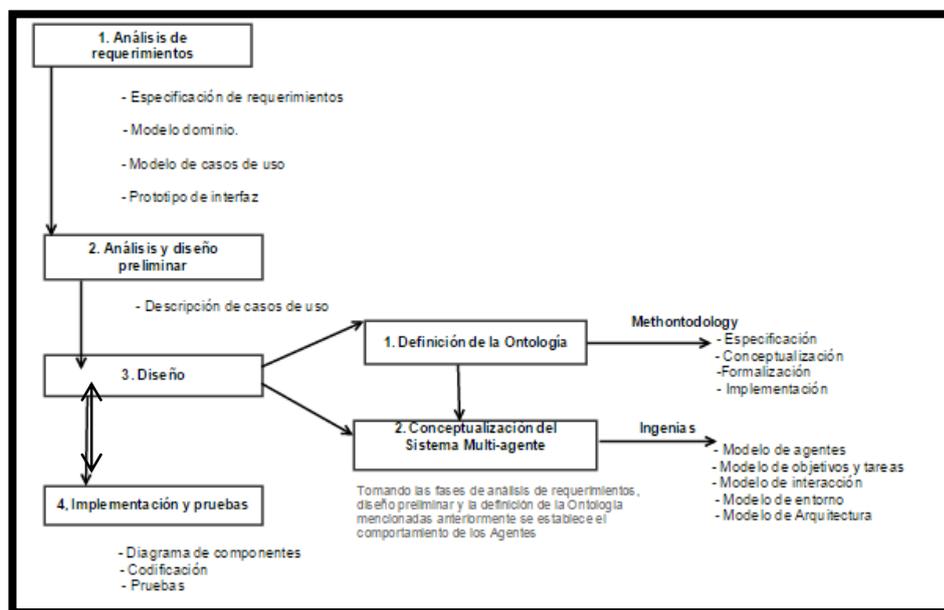


Figura 12. Descripción de la metodología propuesta.

6. RESULTADOS.

Durante el desarrollo del Sistema Multiagente se crearon fases esenciales para su desarrollo y por cada fase se realizaron diferentes actividades. La primera fase fue exclusivamente de revisión literaria, análisis de herramientas y tecnologías de Objetos de Aprendizaje, Ontologías y Agentes Inteligentes; en la segunda fase se diseñó la Ontología que contiene los OAs; la tercera fase consistió en la construcción del Sistema Multiagente complementando con los resultados de la segunda fase; y en la cuarta fase se evalúa el funcionamiento de la aplicación en la búsqueda de Objetos de Aprendizaje.

6.1. FASE I: Revisar el estado del arte en lo referente a Objetos de Aprendizaje, Ontologías y Agentes Inteligentes

La primera fase del presente Trabajo de Titulación resultó exclusivamente investigativa y analítica. A continuación se presenta los resultados obtenidos al revisar la teoría de Objetos de Aprendizaje, Ontologías, Agentes Inteligentes, trabajos relacionados, metodologías y herramientas para su desarrollo, haciendo constar su importancia en la utilización del presente trabajo.

6.1.1. Conclusión de la Revisión de Objetos de Aprendizaje.

De acuerdo a la investigación bibliográfica detallada en el apartado de Revisión Literaria de los OA, se encontraron aspectos importantes que ameritan su investigación, los cuales se mencionan a continuación.

La TABLA VIII, muestra la importancia de los Objetos de Aprendizaje en el ámbito educativo, tanto para los estudiantes, investigadores o autodidactas que desean adquirir conocimiento y auto educarse desde la Web.

TABLA VIII.
IMPORTANCIA DE LOS OA.

	Estudiantes/Investigadores	Docente
Personalización	<ul style="list-style-type: none"> ✓ Puede realizar búsquedas personalizadas de acuerdo a su interés por aprender cierta temática. 	<ul style="list-style-type: none"> ✓ Puede acoplar los OA de acuerdo a lo que quiera impartir a sus estudiantes respetando los derechos de autor. ✓ Recomendar OA como caminos alternativos para reforzar los conocimientos de sus estudiantes.
Accesibilidad/ Disponibilidad	<ul style="list-style-type: none"> ✓ Pueden ser accedidos en cualquier momento para su aprendizaje. 	<ul style="list-style-type: none"> ✓ Pueden ser accedidos en cualquier momento para su utilización.
Reutilización	<ul style="list-style-type: none"> ✓ Permita utilizar OA que han sido verificados con criterios de calidad. 	<ul style="list-style-type: none"> ✓ Reducen el tiempo invertido en el desarrollo del material didáctico de enseñanza.
Flexibilidad	<ul style="list-style-type: none"> ✓ Al ser diseñados en base a modelos pedagógicos, se adaptan con facilidad al proceso de aprendizaje del estudiante. 	<ul style="list-style-type: none"> ✓ Crear Objetos de aprendizaje en cualquier disciplina. ✓ Adaptar el uso de OA a diferentes metodologías de enseñanza-aprendizaje
Durabilidad/ Actualización	<ul style="list-style-type: none"> ✓ Permite obtener información veraz y actualizada sobre un determinado tema de interés. 	<ul style="list-style-type: none"> ✓ Permiten mantenerse actualizado y a su vez rediseñar y adaptar OA a las nuevas tecnologías.
Interoperabilidad	<ul style="list-style-type: none"> ✓ Pueden ser accedidos independientemente del hardware y software, siempre y se disponga de acceso a Internet. 	

6.1.2. Selección de los Estándares de Metadatos

Se analizaron los estándares de metadatos bajo los cuales se recupera los Objetos de Aprendizaje, con lo cual nos encontramos con la existencia de únicamente dos estándares para catalogar los mismos; los cuales tienen diferentes atributos y categorías que son tomados en cuenta para modelar el dominio del conocimiento de los OAs mediante la Ontología. Entre los estándares de Metadatos con los cuales se trabajó son Dublin Core y LOM.

El estándar Dublin Core es un estándar bastante simple, pero no menos eficiente para la catalogación de OA, a continuación en la Figura 13 se detalla cómo fueron tomadas sus categorías [6]:

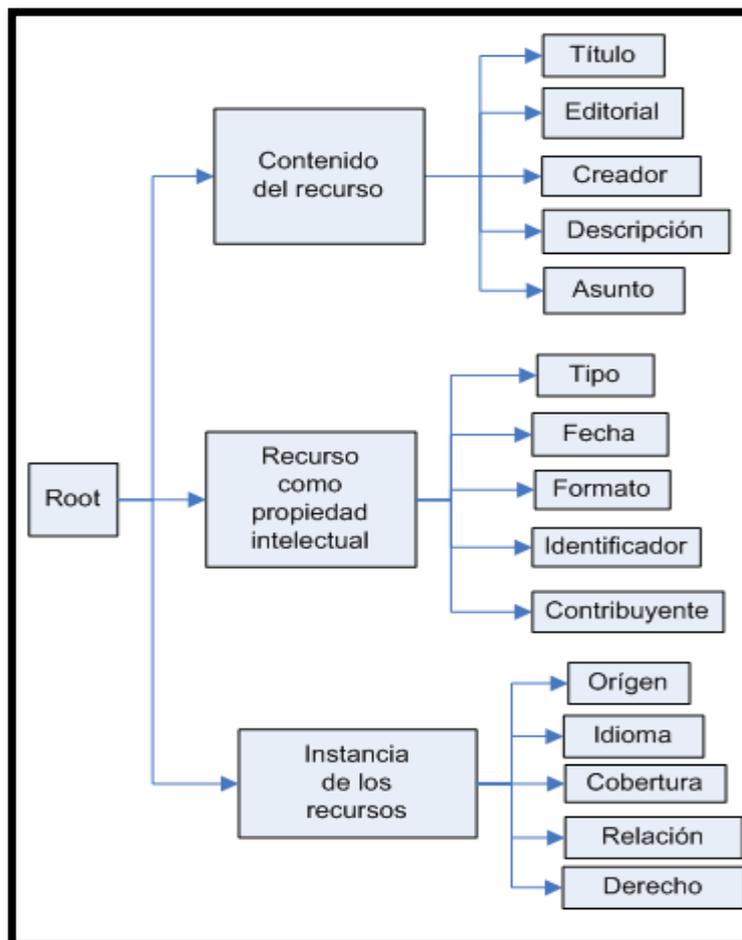


Figura 13. Estructura de Dublin Core.

El estándar LOM, es muy amplio y describe a fondo cada Objeto de Aprendizaje, en la actualidad es el estándar por excelencia considerado por la IEEE. La Figura 14 detallada su estructura para recuperar los metadatos de OA en dicho estándar [7]:

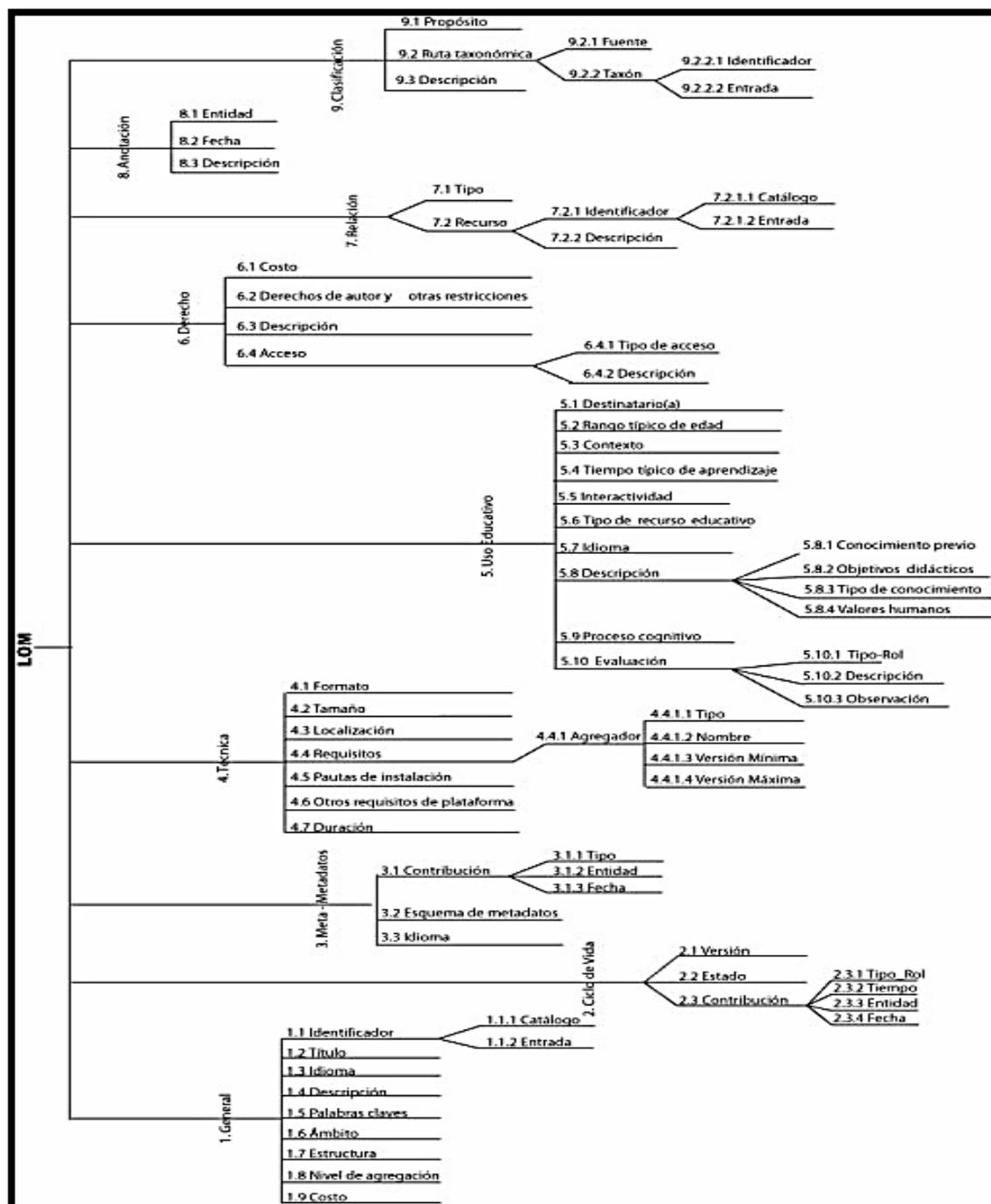


Figura 14. Estructura de LOM.

6.1.3. Selección de Repositorios de Objetos de Aprendizaje

Los repositorios de OA por su parte son librerías digitales especializadas para el almacenaje de este tipo de entidades y sus metadatos, en la tabla IX se resume los repositorios con los cuales se trabajó, y nos brindaron sus Servicios Web, de los cuales se extrae los metadatos de los Objetos de Aprendizaje:

TABLA IX.
REPOSITORIOS DE OA SELECCIONADOS

Nombre	URL	Descripción	Estándar	Organización
Repositorio Institucional Universidad Nacional Autónoma de México (UNAM).	www.rad.unam.mx	Es un proyecto el cual tiene como propósito apoyar a las dependencias y entidades de la UNAM en la búsqueda de OAs.	Dublin Core	Universidad Nacional Autónoma de México.
Mayor Repositorio del Reino Unido (JORUM).	www.jorum.ac.uk	Es un repositorio que sirve para descubrir y compartir recursos educativos abiertos para la educación superior y más allá y el sector de habilidades.	Dublin Core	Mayor Repositorio del Reino Unido, Europa

Nombre	URL	Descripción	Estándar	Organización
Centro de Recursos para la enseñanza y aprendizaje (CREA).	www.crea.udg.mx	Es un repositorio de materiales educativos orientado al fortalecimiento de procesos de enseñanza-aprendizaje y a la formación de integral de los estudiantes.	Dublin Core	Universidad de Guadalajara, México
Banco Internacional de Objetos de Aprendizaje.	http://objetoseducacionais2.mec.gov.br/	Este repositorio almacena recursos digitales sobre la educación de Brasil.	Dublin Core	TVEscola, Brasil
Repositorio Objetos de aprendizaje de Escuela Superior Politécnica de Litoral.	www.sidweb.espol.edu.ec	Es un repositorio que aloja OA, los cuales son diseñados por estudiantes, docentes e investigadores de la ESPOL para realizar actividades de enseñanza-aprendizaje a través del Internet	LOM	Escuela Superior Politécnica, Ecuador
Fedora Commons.	http://fedora-commons.org/	Repositorio para la Gestión de objetos digitales	Dublin Core	DuraSpace, Europa

6.1.4. Análisis de trabajos relacionados.

Se revisó varios trabajos relacionados que tenían que ver con Agentes y Ontologías los cuales se encargan de buscar información en la Web, estos se encuentran detallados en el Capítulo 4 de la Revisión de Literatura, mediante el estudio de estos trabajos se esclareció dudas respecto al funcionamiento de los Agentes en la búsqueda de información y a tener una perspectiva puntual para enfocarnos en nuestro trabajo. En la Tabla X, se especifica si los Sistemas estudiados tienen búsqueda simple o avanzada, así como si utilizan técnicas de Web semántica y se compara con nuestro Software especificando a que dominio de la información se enfocan sus búsquedas.

TABLA X.

ANÁLISIS DE TRABAJOS RELACIONADOS

Parámetros de Evaluación		Trabajos			
		CiteSeer [26].	LETIZIA [27].	ABIMETAGIS [28].	SMAS (Sistema Propuesto)
Personalización	Búsqueda simple	✓	✓	✓	✓
	Búsqueda avanzada	✓			✓
Técnicas de Web Semántica				✓	✓
Tipo de Aplicación		Web	Escritorio	Web	Web
Dominio	OA				Objetos de Aprendizaje
	Otros	Literatura Científica	Sitios Web	Datos Geográficos	-
Recogida de Información		URL	Metadatos	Metadatos	Metadatos

6.1.5. Analizar las metodologías y herramientas existentes para la construcción de Ontologías y Agentes Inteligentes.

En esta fase se escogió varias metodologías y herramientas, las cuales se analizaron mediante cuadros comparativos, para posteriormente proceder a la selección de la metodología y herramienta más adecuada para nuestro proyecto.

6.1.5.1. Análisis y selección de Metodología para la Ontología

Las metodologías se diferencian principalmente en las fases a seguir para la construcción del modelo ontológico, en los pasos posteriores e incluso en la utilización de herramientas de apoyo para la construcción de una ontología.

Los criterios de evaluación a los cuales serán sometidas cada una de las metodologías seleccionadas, serán los siguientes:

- **Documentación:** Cantidad de documentación disponible acerca de la metodología a ser evaluada, tomando en cuenta los sitios oficiales de la metodología donde existe documentación detallada.
- **Reutilización:** Se refiere a una vez aplicada la metodología, la facilidad de ser reusados sus pasos para la construcción de otro modelo ontológico
- **Basado en estándares:** Indica si la metodología fue construida en base a estándares internacionales que garanticen la calidad de la misma.
- **Herramientas de soporte:** Si cuenta con herramientas desarrolladas que den soporte a dicha metodología.
- **Facilidad de Uso:** En este parámetro se evaluará la sencillez en los pasos para ser aplicados dentro de la metodología, es decir que los pasos estén bien definidos para ser realizados.
- **Ciclo de vida Recomendado (CVR):** Señala si la metodología evaluada cuenta con las actividades del ciclo de vida de desarrollo de la ontología común desde la perspectiva de la ingeniería que involucra las siguientes actividades

- ✓ **Especificación:** Identificar el alcance ontología, actividad muy importante lo que lleva a obtener datos necesarios para ser analizados.
- ✓ **Conceptualización:** El proceso de estructuración del conocimiento del dominio, que ha sido adquirida en un modelo conceptual que proporciona la descripción necesaria sobre el problema y su solución.
- ✓ **Formalización:** En esta actividad, el modelo conceptual se transformará en un modelo semi-formal o formal.
- ✓ **Implementación:** El resultado de esta actividad es la ontología codificada.
- ✓ **Mantenimiento:** El mantenimiento es el proceso de actualización y / o corregir la ontología implementado.
- ✓ **Adquisición de conocimientos:** Es el proceso de adquisición de los conocimientos ontología es decir poblar la Ontología de información.
- ✓ **Evaluación:** La evaluación significa juzgar la calidad de la ontología técnicamente.
- ✓ **Documentación:** Incluye escribir la necesario documentación para facilitar el uso, reutilización y mantenimiento de la ontología.

TABLA XI
COMPARACIÓN DE METODOLOGÍAS PARA LA CONSTRUCCIÓN DE ONTOLOGÍAS

↓ Parámetros de Evaluación		→ Metodología					
		CYC	KACTUS	METHONTOL OGY	NeOn	Grüninger y Fox	Uschold y King
Reutilización		✓	✓	✓			✓
Basado en estándares				✓			
Herramientas de soporte					✓		
Facilidad de Uso				✓	✓		
Documentación disponible		Algunos detalles	Algunos detalles	Suficientes detalles	Suficientes detalles	Detalles insuficientes	Detalles insuficientes
CVR	Especificación	No Propuesto	Propuesto	Propuesto	Propuesto	Propuesto	Propuesto
	Conceptualización	No Propuesto	Parcialmente Propuesto	Propuesto	Propuesto	Parcialmente Propuesto	Parcialmente Propuesto
	Formalización	No Propuesto	Parcialmente No propuesto	Propuesto	Propuesto	Propuesto	Parcialmente Propuesto
	Implementación	Propuesto	Propuesto	Propuesto	Propuesto	No propuesto	Propuesto
	Mantenimiento	No Propuesto	No propuesto	No propuesto	No propuesto	No propuesto	No propuesto
	Adquisición de conocimientos	Propuesto	No propuesto	Propuesto	Propuesto	No propuesto	Propuesto
	Evaluación	No Propuesto	No propuesto	Propuesto	No propuesto	Propuesto	No propuesto
	Documentación	Propuesto	Propuesto	Propuesto	No propuesto	Propuesto	Propuesto

Como se puede observar en la TABLA XI, algunas metodologías se centran en distintos aspectos para el desarrollo de Ontologías. Por ejemplo unas profundizan la especificación y conceptualización, pero al mismo tiempo carecen de la debida atención en la fase de diseño, que es tan importante como las fases anteriores. De la misma manera, algunas metodologías hablan de cubrir distintas fases de desarrollo de la ontología, pero su documentación no informa acerca de las técnicas particulares que deben ser empleadas durante estas fases.

Selección de la Metodología.

En relación con las implicaciones se identificaron los pros y contras de las diferentes metodologías en base a los criterios analizados. CYC, Kactus y METHONTOLOGY se enfocan a la reutilización de las Ontologías, característica relevante para la reingeniería de Software. Al mismo tiempo el uso de estándares en la construcción de ontologías busca la certificación de un producto, para permitir tener una mejor aceptación otorgando confianza al cumplir con normas de calidad, METHONTOLOGY hasta el momento es la que cuenta con el estándar IEEE 1074, pero el uso de los mismos no necesariamente garantiza el éxito de una metodología si no se hace un uso correcto.

La documentación disponible es otro factor que importa mucho, así como también herramientas de soporte, entre las metodologías que no cuentan con sitios oficiales donde se suba documentación actualizada, ni descripción de las mismas se encuentran CYC y Kactus de quienes solo existen algunos detalles de su implementación y las metodologías Grüninger Fox y Uschold King la información es escasa, por lo cual se hace difícil su uso. NeOn y METHONTOLOGY cuentan con herramientas de soporte que dan seguimiento a la implementación de la metodología.

METHONTOLOGY maneja un ciclo de vida recomendado, por lo tanto, es una de las razones que ha sido adoptado para la construcción de un buen número de ontologías de dominio en comparación con otras metodologías.

De acuerdo al análisis realizado la metodología escogida para el desarrollo de la ontología es: Methontology, debido a las siguientes características:

- Es la metodología más difundida para el desarrollo de ontologías, contando suficiente información que detalla cada una de sus fases para la implantación,
- Provee un proceso de desarrollo basado en el ciclo de vida del software; es decir, se basa en el estándar IEEE 1074, el mismo que permite que el software evolucione de acuerdo a un ciclo de actividades tales como: análisis de requerimientos, diseño, implementación, pruebas, entrega y mantenimiento de software.
- Es una metodología que utiliza un proceso iterativo, permitiendo la entrega de software en pequeños incrementos que a su vez permite la mejora del mismo.
- Permite realizar tareas como: control, aseguramiento de calidad, adquisición de conocimiento.

6.1.5.2. Análisis y selección de Herramientas para la Ontología

Para el análisis de las herramientas se tomaron los siguientes parámetros de evaluación:

- **Gestión de Backups:** Permite respaldar el diseño de la Ontología contando con gestión propia de Bases de Datos
- **Interfaz Gráfica:** Si cuenta con interfaz gráfica para el diseño de la Ontología
- **Lenguajes Importación:** Los lenguajes soportados para la importación de Ontologías.
- **Lenguajes de Exportación:** Los lenguajes soportados para la exportación de Ontologías.
- **Almacenamiento:** El modo en que almacena la Ontología

TABLA XII.
COMPARACIÓN DE HERRAMIENTAS PARA LA CONSTRUCCIÓN DE ONTOLOGÍAS

Herramienta → ↓ Parámetros de Evaluación	OntoSaurus	OilEd	WebOnto	OntoEdit	Protégé	WebODE	Ontolingua Server
Gestión de Backups			✓			✓	
Consultas					✓		
Interfaz de Grafica	✓	✓		✓	✓	✓	
Extensibilidad				✓	✓	✓	
Arquitectura	Client/Server	Standalone	Client/Server	Standalone	Standalone	Client/Server & Standalone	Client/Server
Lenguajes Importación	LOOM, IDL, KIF, C++	RDF(S), DAML+OIL	OMCL	RDF(S), DAML+OIL	RDF(S), OWL	RDF(S), DAML+OIL, OWL	Ontolingua, DAML+OIL, CLIPS
Lenguajes de Exportación	LOOM, IDL, KIF, C++	RDF(S), DAML+OIL, OWL	OMCL, Ontolingua, RDF(s), OIL	RDF(S), DAML+OIL, OWL	RDF(S), OWL, CLIPS, Java, html	RDF(S), DAML+OIL, OWL, CLIPS	Ontolingua, DAML+OIL, CLIPS
Almacenamiento	Archivos	Archivos	Archivos	Archivos	Archivos, DBMS(JDBC)	DBMS(JDBC)	Archivos

Existen herramientas para la construcción de ontologías mediante código, otras herramientas que exigen aprendizaje/conocimiento de un lenguaje en específico y hay herramientas de ontologías que son más gráfica (Protégé y IsaViz). Otras proporcionan documentación, la ontología de importación/exportación para diferentes formatos, vista gráfica de ontologías, bibliotecas de ontologías y motores de inferencia adjuntos.

Selección de la herramienta para la Ontología.

El editor de ontologías seleccionado para el desarrollo del presente trabajo es Protégé ya que es una herramienta que resulta acorde con el planteo metodológico. Permite una comparación de la aplicación durante todo el proceso, otro factor para su elección fue la existencia de una gran cantidad de ejemplos de desarrollos de ontologías en otros dominios lo que facilita corroborar el desarrollo de nuestra aplicación. Dentro de las ventajas que colaboraron con el criterio de selección es que dispone de un ambiente visual de diseño y registro de ontologías.

6.1.5.3. Análisis y selección de Metodología para la construcción de los Agentes Inteligentes

Las metodologías escogidas se centran fundamentalmente en las etapas de análisis y diseño en Sistemas Multiagente.

Los criterios que se han valorado se muestran así:

- **Arquitectura agente:** Indica si la metodología es recomendada para las diferentes arquitecturas de Agentes o solo una en específico o ninguna.
- **Herramientas de soporte:** Si cuenta con herramientas desarrolladas que den soporte a dicha metodología.
- **Documentación disponible:** Cantidad de documentación disponible acerca de la metodología a ser evaluada, tomando en cuenta sitios oficiales.
- **Ciclo de Vida:** Señala si la metodología evaluada cuenta con actividades que establecen un ciclo de vida.
- **Enfoque hacia el desarrollo SMA:** Si la metodología es recomendada para la construcción de Sistemas Multiagente.

TABLA XIII.

COMPARACIÓN DE METODOLOGÍAS PARA LA CONSTRUCCIÓN DE AGENTES

<div style="display: flex; align-items: center;"> <div style="text-align: right; padding-right: 10px;"> ↓ Parámetros de Evaluación </div> <div style="text-align: center;"> Metodología → </div> </div>		MaSE	GAIA	TROPOS	INGENIAS
		Arquitectura agente		Independiente	Independiente
Herramientas de soporte		AgentTool	-	-	INGENIAS Development Kit
Documentación disponible		Suficientes detalles	Algunos detalles	Algunos detalles	Suficientes detalles
Ciclo de Vida	Requisitos	✓		✓	✓
	Análisis	✓	✓	✓	✓
	Diseño	✓	✓	✓	✓
	Implementación	✓			✓
	Pruebas				
Enfoque hacia el desarrollo MAS		✓	✓	✓	✓

Es de observar que las metodologías expuestas intentan cubrir fundamentalmente las etapas de análisis y diseño de sistemas Multiagente. El resto de etapas no son consideradas o bien se deja total libertad para su desarrollo. La mayoría de las metodologías cubren con éxito las propiedades y conceptos de los Sistemas Multi-agente. La arquitectura del Agente es otro factor que se debe tomar en cuenta, nuestro caso programaremos agentes reactivos, Tropos especifica que está orientada únicamente a la creación de Agentes Deliberativos (BDI).

Selección de la Metodología

En el resultado comparativo de las metodologías realizado previamente se observa que INGENIAS ha sido la mejor cualificada según los criterios propuestos para nuestro caso concreto de estudio. No se podría afirmar que INGENIAS fuese la metodología óptima para cualquier sistema Multi- agente ya que depende del sistema concreto y de la finalidad del desarrollador. Esta posee las siguientes cualidades:

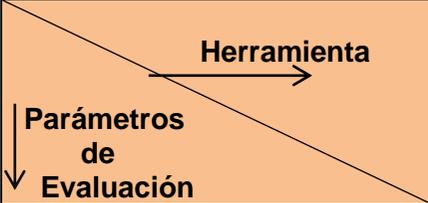
- Especifica la organización de los agentes a diferentes niveles e indicando sus objetivos y tareas.
- Describe los objetivos y tareas de los agentes de forma individual y colectiva.
- Muestra detalladamente el intercambio de mensajes entre agentes (interacciones).
- Incluye diversos ejemplos que sirven de guía a desarrolladores.
- Proporciona una herramienta CASE (IDK (INGENIAS Development Kit)) con editor gráfico y generador de código, el cual incluye un módulo para JADE, uno de los frameworks de desarrollo de agentes más comunes, que emplea Java.

6.1.5.4. Análisis y selección de la Herramienta para la construcción de los Agentes Inteligentes.

Existe variedad de opciones para la elección de herramientas para la construcción de sistemas Multi-agentes; con el paso de los años se han desarrollado varias plataformas, cada una de ellas con ciertas características específicas, por lo cual a continuación se presenta una descripción de las más representativas.

TABLA XIV.

COMPARACIÓN DE HERRAMIENTAS PARA LA CONSTRUCCIÓN DE AI

	AgentBuilder	JACK	JADE	ZEUS
Arquitectura Base	BDI	BDI	Cualquiera	BDI
Tipos de agentes soportados	Cualquiera	Cualquiera	Cualquiera	Deliberativos Colaborativos
Estándar Protocolo mensajes	-----	-----	ACL FIPA	KQML
Lenguajes soportados	-----	JACK	Java	Java
Movilidad código	No detalla	No detalla	Migración débil	No detalla
Licencia	Versión de evaluación libre y limitada	30 días gratis	LGPL	Mozilla public
Interface	GUI	GUI	GUI	GUI
Documentación	Suficientes detalles	Algunos detalles	Suficientes detalles	Detalles insuficientes

Selección de la Herramienta

JADE es la herramienta seleccionada, por la arquitectura de comunicación de esta plataforma. Además es de mencionar que soporta cualquier tipo de agentes, los cuales son implementados utilizando el lenguaje java; esta plataforma es liberada bajo licencia LGPL y ofrece una completa documentación.

6.2. FASE II: Diseñar una Ontología con los parámetros que deben poseer los recursos a ser recuperados.

Siguiendo la metodología planteada para el desarrollo del software, en el Anexo I se puntualiza la fase de Especificación de Requerimientos y Análisis Preliminar; la siguiente fase es el diseño y una de las actividades es la definición de la Ontología utilizando las fases propuestas por la metodología METHONTOLOGY.

6.2.1. Ontología para la representación Objetos de Aprendizaje.

6.2.1.1. Introducción

El desarrollo de Ontologías con Methontology involucra una serie de pasos y actividades a realizarse, las mismas que permiten representar el dominio que se desea modelar. En el presente proyecto se realizará una ontología para integrarse a un Sistema Multiagente de búsqueda de Objetos de Aprendizaje.

6.2.1.2. Especificación de requerimientos de la ontología

6.2.1.2.1. Alcance.

La ontología va enfocada a la búsqueda de Objetos de Aprendizaje proveniente de repositorios disponibles en la Web y por ende servirá de base semántica para la extracción de información. Esta describirá el dominio de los Objetos de Aprendizaje basada en los estándares Dublin Core y LOM, que describen la terminología adecuada para la catalogación de OAs.

6.2.1.2.2. Propósito.

Construir una ontología que describa el dominio del Conocimiento de los Objetos de Aprendizaje que se utilizará como base semántica para la búsqueda Objetos.

6.2.1.2.3. Lenguaje de implementación

La ontología será implementada en el lenguaje RDFs/OWL.

6.2.1.2.4. Posibles usuarios finales.

Investigadores, estudiantes y/o profesionales que deseen consultar un tema de interés.

6.2.1.2.5. Posibles usos

- Buscar información que les ayude a desarrollar o aprender un tema de interés.
- Buscar ayuda en un tema en común.

6.2.1.2.6. Requerimientos

- **Funcionales**

La Implementación de la Ontología será en el Idioma inglés, debido a que se utilizara los estándares de metadatos los cuales se encuentran establecidos en dicho idioma.

- **Cuestiones de competencia**

Las interrogantes que responderá la Ontología de acuerdo al dominio de los OAs basado en los estándares DC y LOM, serán los siguientes:

Preguntas referentes a los repositorios

¿A qué **repositorio** pertenece el Objeto de Aprendizaje?

¿Cuál es **estándar de metadatos** del Objeto de Aprendizaje?

Preguntas referentes a Objetos de Aprendizaje Dublín Core

¿Qué Objetos de Aprendizaje tiene por **título**.....?

¿Qué Objetos de Aprendizaje su **creador** es

¿Qué Objeto de Aprendizaje pertenecen a la **materia** de.....?

¿Qué Objetos de Aprendizaje tienen por **descripción**.....?

¿Qué Objetos de Aprendizaje tienen por **fuentes**.....?

¿Qué Objetos de Aprendizaje son de **tipo**.....?

¿Qué Objetos de Aprendizaje guardan **relación** con.....?

¿Qué Objetos de Aprendizaje tiene la **cobertura** de.....?

Preguntas referentes a Objetos de Aprendizaje LOM

- ¿Qué Objetos de Aprendizaje tiene por **título**.....?
- ¿Qué Objetos de Aprendizaje tienen el **idioma**.....?
- ¿Qué Objetos de Aprendizaje coinciden con las **palabras claves**.....?
- ¿Qué Objetos de Aprendizaje tiene la **descripción** de.....?
- ¿Qué Objetos de Aprendizaje son de la **fecha**.....?
- ¿Qué Objetos de Aprendizaje su **creador** es
- ¿Qué Objetos de Aprendizaje son de **formato**.....?
- ¿Qué Objetos de Aprendizaje están **localizados** en.....?
- ¿Cuáles son las **observaciones de instalación** del Objeto de Aprendizaje?
- ¿Qué **requerimientos de otras plataformas** tiene el Objeto Aprendizaje?
- ¿A qué **tipo de recurso educativo** pertenece el Objeto de Aprendizaje?
- ¿Cuál es el **nivel de interacción del Objeto** de Aprendizaje?
- ¿Cuál es la **densidad semántica** del Objeto de Aprendizaje?
- ¿A qué **rol de usuario final** va dirigido el Objeto de Aprendizaje?
- ¿Cuál es el **contexto** del Objeto de Aprendizaje?
- ¿Cuál es el **rango de edades de los usuarios** a quien va dirigido el Objeto de Aprendizaje?
- ¿Cuál el **grado de dificultad del Objeto** de Aprendizaje?
- ¿Cuál es el **coste** que tiene el Objeto de Aprendizaje?
- ¿Qué **derechos de copyright** y restricciones tiene el Objeto de Aprendizaje?
- ¿Cuál es el **tipo de relación** que tiene dicho Objeto de Aprendizaje con otros?

- **Pre- glosario de términos**

- ✓ Recurso Educativo Abierto
- ✓ Object Learning Metadata
- ✓ Dublin Core
- ✓ Metadatos
- ✓ Repositorio

6.2.1.2.7. Herramientas para el desarrollo de la Ontología

Para el desarrollo de la ontología se utilizaron las siguientes herramientas y librerías:

- **CMAPTOOLS COE v5.0.3:** Para el modelo inicial de la Ontología.
- **PROTEGE v4.3.0:** Modelamiento de la ontología a un lenguaje formal.
- **JENA:** Proporciona soporte para crear un modelo RDFs, OWL. Se utiliza para el razonamiento de la Ontología (Poblar de individuos y Consultas)
- **VIRTUOSO OPENSOURCE:** Para el almacenamiento de Tripletas y la realización de pruebas a la Ontología.

6.2.1.3. Conceptualización

6.2.1.3.1. Vocabularios

Para describir de manera formal los términos de la Ontología se utilizaron los siguientes vocabularios ya establecidos y disponibles en la Web.

TABLA XV.
VOCABULARIOS DE LA ONTOLOGIA.

Vocabularios		
Nombre	Prefix	URI
Asset Description Metadata Schema	adms	http://www.w3.org/ns/adms#
Asset Management Vocabulary	Am	http://open-services.net/ns/asset#
Cmo	Cmo	http://purl.org/twc/ontologies/cmo.owl
Vocabulary related to incident communication	Com	http://purl.org/twc/ontologies/cmo.owl#
Data Catalog Vocabulary	dcat	http://www.w3.org/ns/dcat#
Dublin Core	Dce	http://purl.org/dc/elements/

Vocabularios		
Nombre	Prefix	URI
FRBR-aligned Bibliographic Ontology	fabio	http://purl.org/spar/fabio/
Federal Enterprise Architecture	Fea	http://vocab.data.gov/def/fea/
FOAF Vocabulary Specification	foaf	http://xmlns.com/foaf/spec/
linguistics-ontology	Gold	http://purl.org/linguistics/gold
Lifecycle Schema	Lcy	http://purl.org/vocab/lifecycle/schema
Lemon	lemon	http://lemon-model.net/lemon#
NEPOMUK Contact Ontology	Nco	http://www.semanticdesktop.org/ontologies/2007/03/22/nco
Opus	Opus	http://lstdis.cs.uga.edu/projects/semdis/opus#
The PROV Namespace	prove	http://www.w3.org/ns/prov#
Cube	Qb	http://purl.org/linked-data/cube#
Repository Asset Distribution	radion	http://www.w3.org/ns/radion/
the Scholarly Contributions and Roles Ontology	scoro	http://purl.org/spar/scoro/EducationalRole/
Stac	Stac	http://securitytoolbox.appspot.com/stac#
Upper Mapping and Binding Exchange Layer	umbel	http://umbel.org/umbel/
VCard Ontology	vcard	http://www.w3.org/2006/vcard/ns#
Xhiv	Xhiv	http://www.w3.org/1999/xhtml/vocab/
Sistema Multiagente	smas	http://smasunl.ml/

6.2.1.3.2. Definición del glosario de términos.

El glosario de términos detallado a continuación hace referencia a un conjunto de conceptos que son de interés del dominio de la ontología, se definen los términos que forman parte de esta.

TABLA XVI.

GLOSARIO DE TERMINOS DE LA ONTOLOGIA.

Termino	Tipo	Descripción	Vocabulario
OpenEducational Resouerce	Class	Recurso digital gratuito para el aprendizaje.	smas
LearningObject	Class	Objeto de Aprendizaje para el Aprendizaje.	smas
Metadadata	Class	Un trabajo independiente que proporciona información que describe una o más características de un recurso o entidad.	fabio
Repository	Class	Un repositorio es un sistema o servicio que ofrece instalaciones para el almacenamiento y el mantenimiento de las descripciones de los activos y distribuciones.	radion
Name	Property	Un nombre para algún objeto.	foaf
Country	Property	El país donde se encuentra.	nco:
Standard	Property	Es un proceso, protocolo o técnica utilizada para hacer algo concreto.	smas
Uri	Property	Un identificador de recursos uniforme o URI	smas
DublinCore	Class	Es un modelo de metadatos elaborado.	swrc
IntellectualPropert yRights	Class	Derechos de propiedad intelectual de un recurso.	cmo
TechnicalAspects	Class	Elementos relacionados principalmente con la instanciación del recurso.	fabio
ContentsInformati on	Class	Elementos relacionados principalmente con el contenido del recurso.	xhv

Termino	Tipo	Descripción	Vocabulario
Contributor	Property	Una persona u organización que haya tenido una contribución intelectual significativa	dce
Coverage	Property	Es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso	dce
Creador	Property	La persona u organización responsable de la creación del contenido intelectual del recurso	dce
Date	Property	Una fecha en la cual el recurso se puso a disposición del usuario en su forma actual.	dce
Description	Property	Una descripción textual del recurso. Puede ser un resumen.	dce
Format	Property	Es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso.	dce
Identifier	Property	Secuencia de caracteres usados para identificar unívocamente un trabajo.	dce
Language	Property	Lengua/s del contenido intelectual del recurso.	dce
Publisher	Property	La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual.	dce
Relation	Property	Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos	dce
Right	Property	Son una referencia para una nota sobre derechos de autor	dce
Source	Property	Un recurso relacionado a partir del cual se deriva el recurso descrito	dce
Subject	Property	Describe las materias que abarca el Recurso.	dce
Title	Property	El nombre dado a un recurso, habitualmente por el autor.	dce
Type	Property	La categoría del recurso.	dce
Lom	Class	Estándar de metadatos Learning Object Metadata.	smas

Termino	Tipo	Descripción	Vocabulario
Relation	Class	La relación que guarda con otros elementos.	cmo
Classification	Class	Metadatos para la clasificación del material en taxonomías.	com
Rights	Class	Categoría que agrupa metadatos relativos a los derechos de propiedad e intelectuales del material.	ebucore
Technical	Class	Categoría que agrupa metadatos relativos a las características y requisitos técnicos del material en sí.	fabio
General	Class	Los metadatos en esta categoría representan información general sobre el material educativo que describe el mismo como un todo.	gold
Lifecycle	Class	Esta categoría agrupa metadatos referidos a la historia y estado actual del proceso de producción y mantenimiento del material educativo por parte de los autores.	lcy
Metametadata	Class	Esta categoría agrupa información relativa a los metadatos en sí (de ahí su nombre).	mo
Annotation	Class	Anotaciones y comentarios sobre el material educativo.	owl
Educational	Class	Categoría que agrupa metadatos relativos a los usos educativos del material.	scoro
Identifier	Class	Identificador descriptivo del material educativo. Su valor debe identificar unívocamente el material en su contexto educativo.	adms
Taxon	Class	Taxón es la clase de categorías científicas utilizado	gold
Requirement	Class	Plataforma informática necesaria para utilizar este material.	stac
Contribute	Class	Introduce información acerca de un contribuyente a la producción del material	prov

Termino	Tipo	Descripción	Vocabulario
Size	Property	Tamaño en bytes del material.	am
Keyword	Property	Los temas del recurso. Claves o frases que describen el título o el contenido del recurso.	dcat
Id	Property	Identificador único.	disco
Versión	Property	La edición o versión del material.	doc
Status	Property	El estado de producción del material.	foaf
Context	Property	El entorno educativo típico en el que se usará el material.	lemon
Entry	Property	Describe un catálogo y una entrada en dicho catálogo único.	lemon
Structure	Property	La estructura interna del material.	qb
Catalog	Property	Describe Un catálogo de datos que contiene un conjunto de datos.	schema
Cost	Property	Establece si el recurso es o no de pago.	schema
Duration	Property	Duración (únicamente para material para el que tenga sentido una duración en su reproducción.	schema
Location	Property	Forma de localizar al material (por ejemplo, una URL, o una descripción textual acerca de cómo llevar a cabo dicha localización).	schema
Purpose	Property	El propósito que persigue el material	schema
aggregationLevel	Property	Define la granularidad del material.	smas
copyrightAbdOth Restriccions	Property	Derechos de copia y otras restricciones.	smas
Difficulty	Property	Qué difícil es trabajar a través del recurso para el público objetivo típico.	smas
installationRemarks	Property	Notas de instalación para el recurso.	smas
intendedEndUserRole	Property	Usuario normal del recurso, más dominante primero.	smas
interactivityLevel	Property	Nivel de interactividad entre un usuario final y el recurso.	smas

Termino	Tipo	Descripción	Vocabulario
interactivityType	Property	El tipo de interactividad soportado por el recurso.	smas
Kind	Property	Especifica la clase de material que es.	smas
learningResourceType	Property	Especifica el tipo de material	smas
maximunVersion	Property	La Versión Máxima del Material	smas
minimunVersion	Property	La Versión Mínima del Material	smas
otherPlataformRequirements	Property	Otros requisitos software y hardware.	smas
semanticDensity	Property	Una medida subjetiva de la utilidad educativa del material en comparación con su tamaño y/o duración.	smas
typicalAgeRange	Property	Rango de edades típico de los usuarios a los que va dirigido el material.	smas
typicalLearningTime	Property	Tiempo aproximado que se necesita para trabajar con el recurso.	smas
Role	Property	Para especificar la función o el papel desempeñado en una situación particular por el objeto.	vcard

6.2.1.3.3. Taxonomía de Conceptos y Relaciones

La Figura 15, muestra el modelo conceptual de la ontología, la misma que permite representar OAs en base a los metadatos DC y LOM y relaciones existentes entre los mismos.

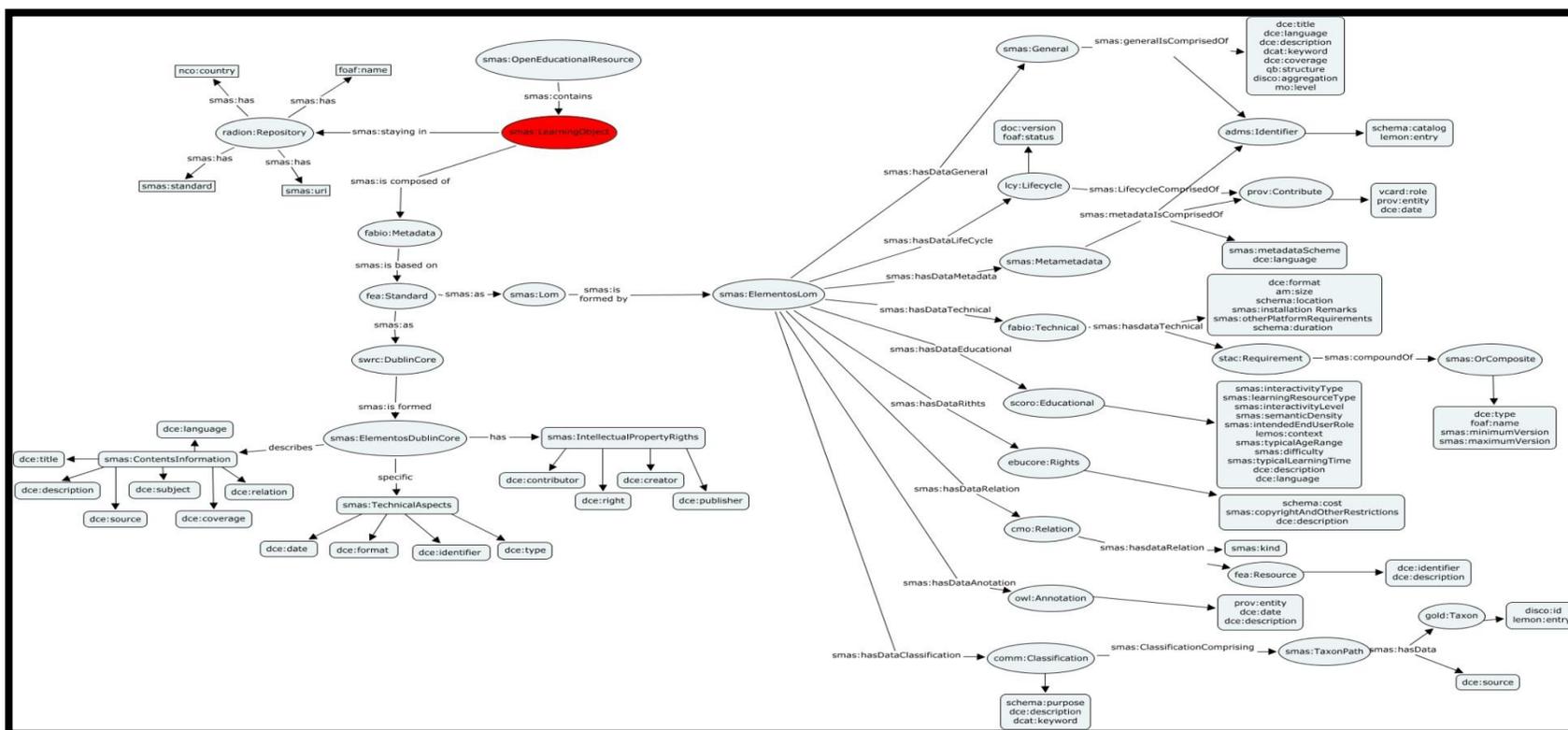


Figura 15. Taxonomía de la Ontología

6.2.1.4. Formalización

Plasmamos la conceptualización de la etapa anterior en un modelo formal o semi-computable, con la utilización de la herramienta Protegé.

6.2.1.4.1. Definición de Conceptos

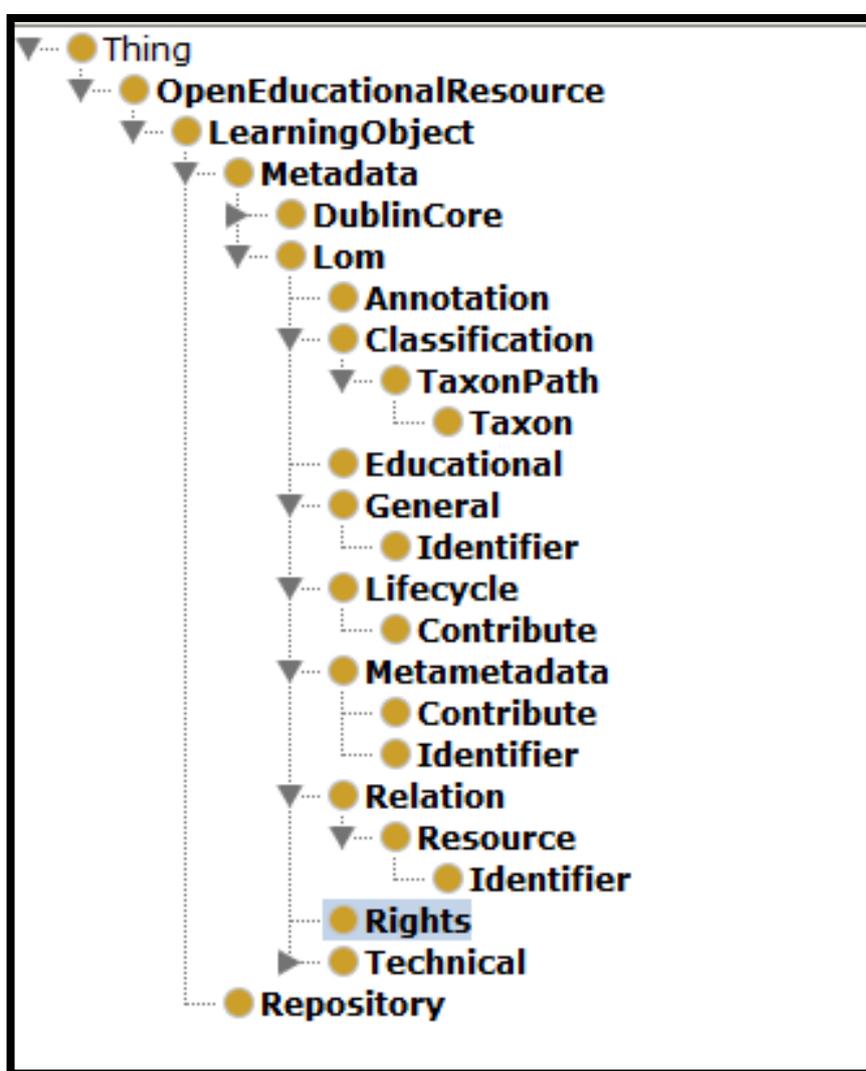


Figura 16. Definición de Conceptos.

6.2.1.4.2. Definición de Relaciones

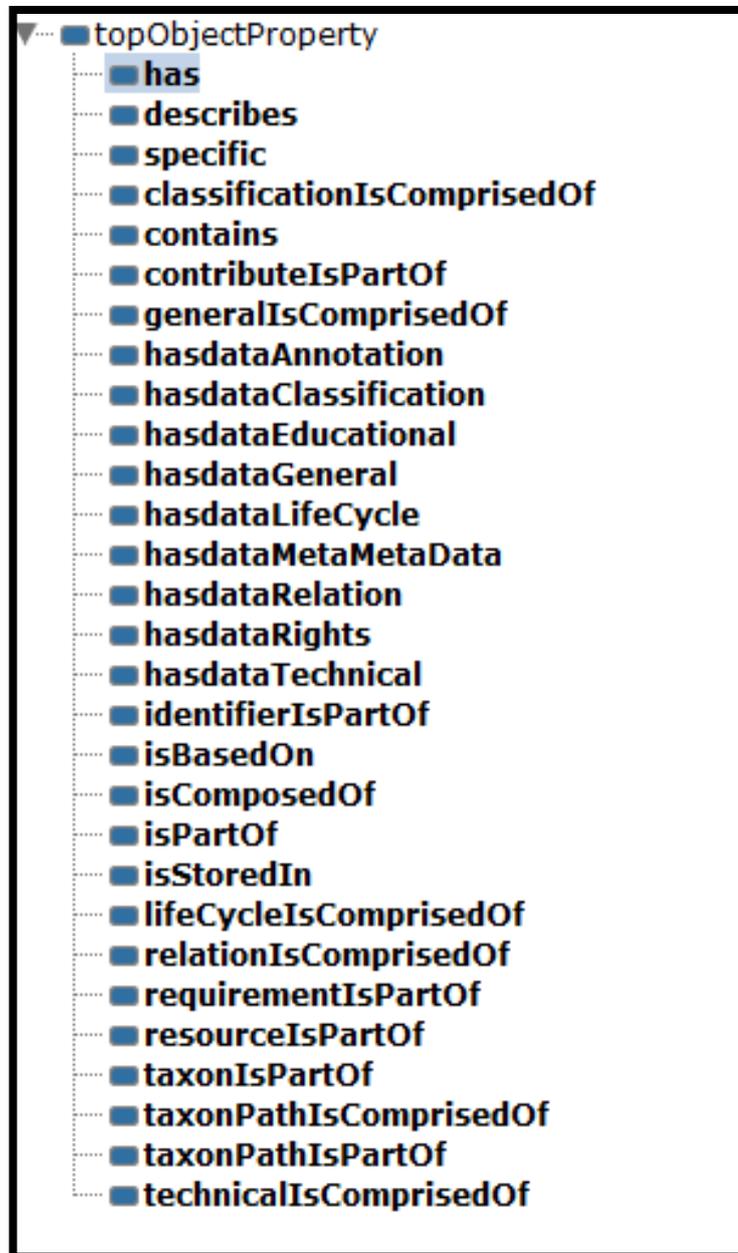


Figura 17. Definición de Relaciones

6.2.1.4.3. Definición de Instancias



Figura 18. Definición de Propiedades

6.2.1.5. Implementación

La transformación del modelo ontológico diseñado en la fase anterior de formalización a un lenguaje formal y computable, resultó de la exportación del diseño en la herramienta Protegé al código RDFs/OWL.

El código completo generado de la ontología se lo puede ver en el Anexo 2.

```
<owl:ObjectProperty rdf:about="&smas;classificationIsComprisedOf">
  <rdfs:range rdf:resource="&smas;TaxonPath"/>
  <rdfs:domain rdf:resource="&com;Classification"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;contains">
  <rdfs:range rdf:resource="&smas;LearningObject"/>
  <rdfs:domain rdf:resource="&smas;OpenEducationalResource"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;contributeIsPartOf">
  <rdfs:range rdf:resource="&mo;Metametadata"/>
  <rdfs:range rdf:resource="&loy;Lifecycle"/>
  <owl:inverseOf rdf:resource="&smas;lifeCycleIsComprisedOf"/>
  <rdfs:domain rdf:resource="&prov;Contribute"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;generalIsComprisedOf">
  <rdfs:domain rdf:resource="&gold;General"/>
  <rdfs:domain rdf:resource="&mo;Metametadata"/>
  <rdfs:domain rdf:resource="&fea;Resource"/>
  <rdfs:range rdf:resource="&adms;Identifier"/>
</owl:ObjectProperty>
```

Figura 20. Fragmento de Código de una Clase OWL.

6.2.1.5.1. Validador del Código formal.

Al haber obtenido la Ontología en el Lenguaje RDFs/OWL, mediante la herramienta del estándar 3WC, disponible en: www.w3.org/RDF/Validator/, se puede validar el código y esto nos debe generar un grafo de la Ontología si su sintaxis es correcta.

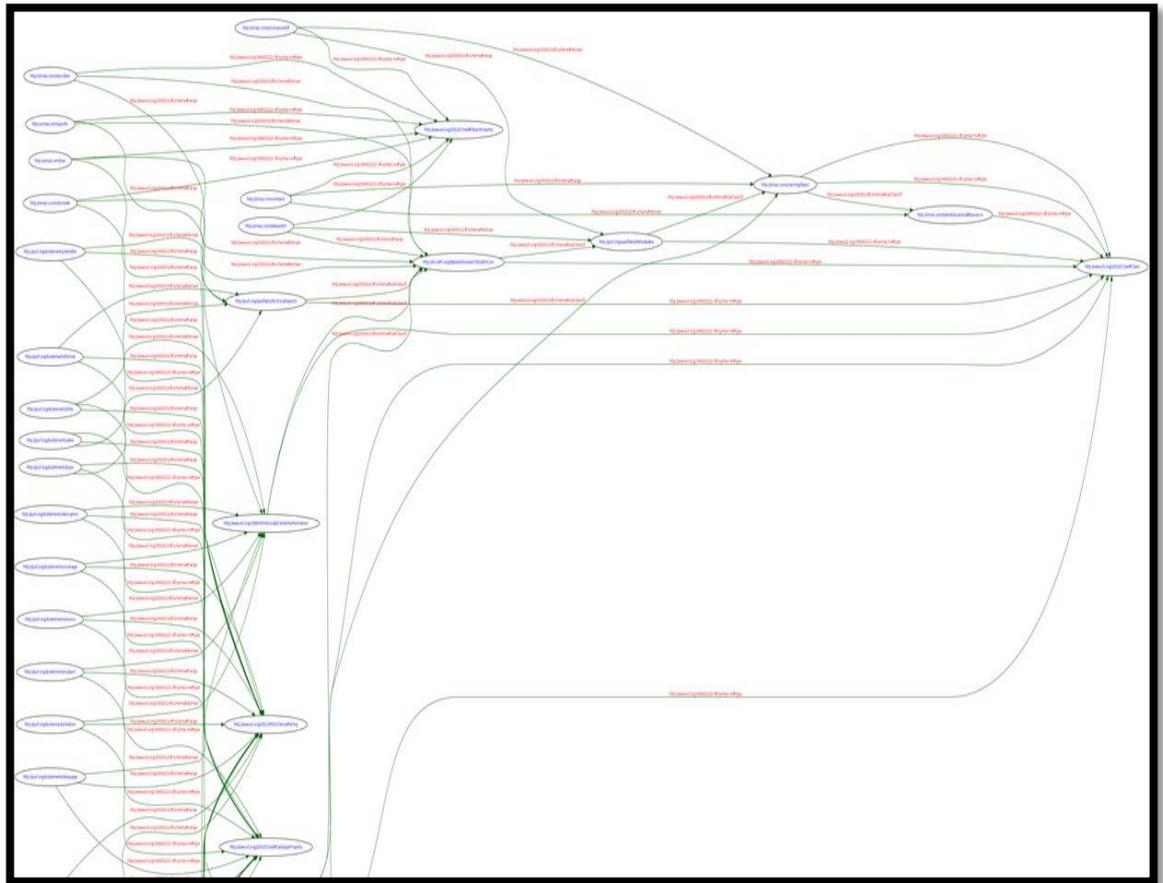


Figura 21. Grafo generado por el Validador W3C.

6.2.1.5.2. Instanciación de la Ontología

Luego de haber validado el modelo Ontológico, se procede a utilizar para el razonamiento de la ontología la librería JENA, la cual nos permite poblar la Ontología y realizar consultas SPARQL desde el lenguaje de programación Java.

Los datos para instanciar la ontología se obtienen mediante el cohechado de metadatos de los Objetos de Aprendizaje, el cual nos entrega en formato XML desde los repositorios, y luego estos son leídos uno a uno y almacenados en la Ontología.

Los metadatos de Objetos de Aprendizaje se recuperan en XML como se muestra en la Figura 22.

```

<metadata>
  <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/"
    instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/
    <dc:creator>Ortíz González, Alejandra</dc:creator>
    <dc:date>2009-10-19T16:57:56Z</dc:date>
    <dc:date>2009-10-19T16:57:56Z</dc:date>
    <dc:date>2009-10-19T16:57:56Z</dc:date>
    <dc:identifier>http://www.crea.udg.mx/handle/123456789/248</dc:identifier>
  <dc:description>
    El usuario será capaz de definir la propuesta de Gardner, identificar las Inteligencias Múltiples presentes
    Modelo, adicionalmente las actividades de aprendizaje aquí establecidas le permiten aplicar los conocimientos
    la inteligencia y capacidades propias con las descritas por Gardner.
  </dc:description>
  <dc:description>
    La Teoría de las Inteligencias Múltiples introduce la idea de categorías mentales amplias y diferenciadas,
    inteligencia única y monolítica
  </dc:description>
  <dc:language>es</dc:language>
  <dc:subject>Howard Gardner</dc:subject>
  <dc:subject>Modelo de Gardner</dc:subject>
  <dc:subject>Teoría de Howard Gardner</dc:subject>
  <dc:subject>Teoría de las Inteligencias Múltiples</dc:subject>
  <dc:subject>Inteligencia Lingüística</dc:subject>
  <dc:subject>Inteligencia Lógico Matemática</dc:subject>
  <dc:subject>Inteligencia Corporal Kinestésica</dc:subject>
  <dc:subject>Inteligencia Intrapersonal</dc:subject>
  <dc:subject>Inteligencia Interpersonal</dc:subject>
  <dc:subject>Inteligencia Espacial</dc:subject>
  <dc:subject>Inteligencia Naturalista</dc:subject>
  <dc:subject>Inteligencia Musical</dc:subject>
  <dc:subject>Psicología Pedagógica</dc:subject>
  <dc:subject>Psicopedagogía</dc:subject>
  <dc:subject>Pedagogía</dc:subject>
  <dc:title>Inteligencias Múltiples</dc:title>
</oai_dc:dc>
</metadata>

```

Figura 22. Metadatos de un Objeto de Aprendizaje

En la Figura 23 se observa el esquema mediante el cual se instancia la Ontología de Objetos de Aprendizaje:

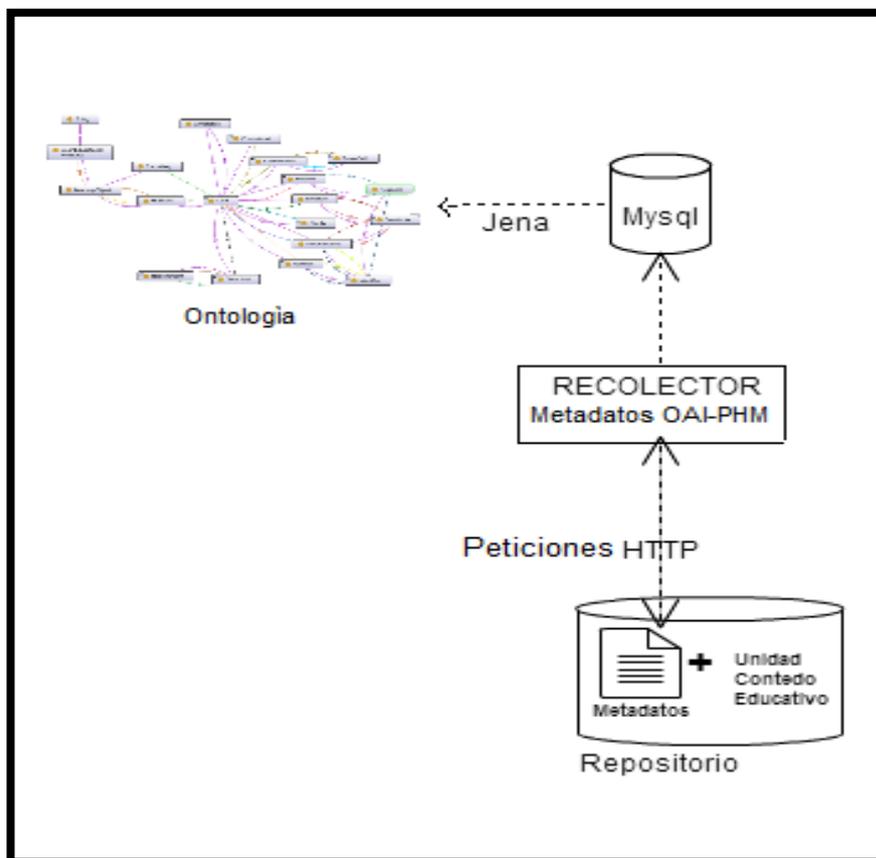


Figura 23. Esquema de extracción de metadatos.

6.2.1.5.3. Consultas SPARQL.

Lo final fue comprobar que la Ontología cumple con los requerimientos planteados mediante consultas SPARQL (Ver Anexo 3).

6.3. FASE III: Construir un Sistema Multiagente para la búsqueda y recuperación de Objetos de Aprendizaje.

En esta fase se detalla el desarrollo del software a partir del empleo de Agentes de Software y el uso de la ontología detallada en la Fase 2, haciendo uso de la metodología Ingenias.

6.3.1. Conceptualización del Sistema Multi-agente.

6.3.1.1. Problema a resolver

Se desea modelar mediante un Sistema Multiagente la búsqueda de Objetos de Aprendizaje, los OAs resultantes serán consultados en una Ontología, los agentes se encargan de gestionar los Servicios Web y poblar la Ontología, y también serán los encargos de gestionar las búsquedas y presentar los resultados.

6.3.1.2. Dominio de la Aplicación

La aplicación tendrá como dominio los repositorios de Objetos de Aprendizaje.

6.3.1.3. Función Principal

Buscar Objetos de Aprendizaje.

6.3.1.4. Simbología

Los modelos requeridos para la construcción de un sistema Multiagente, que se consideran dentro de la metodología INGENIAS son los siguientes:

- Modelo de agentes
- Modelo de objetivos y tareas
- Modelo de interacción
- Modelo de entorno
- Modelo de Arquitectura

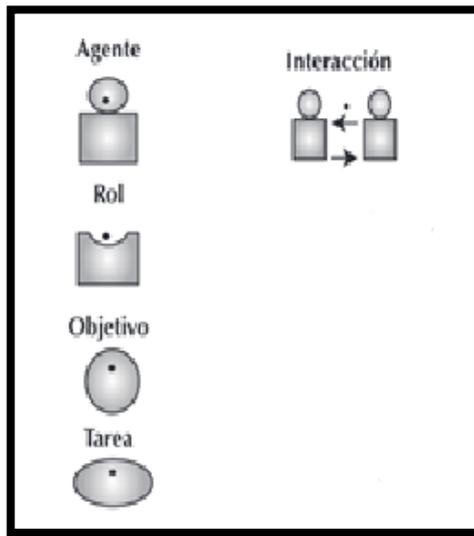


Figura 24. Simbología para el modelado de Agentes

6.3.1.5. Casos de Uso

- Actores

TABLA XVII.
ACTORES DEL SISTEMA

Actor	Rol
Administrador	Es la persona encargada de administrar el sistema.
Investigador	Cualquier persona que accede al sistema para realizar una consulta.

- **Identificación de Casos de Uso.**

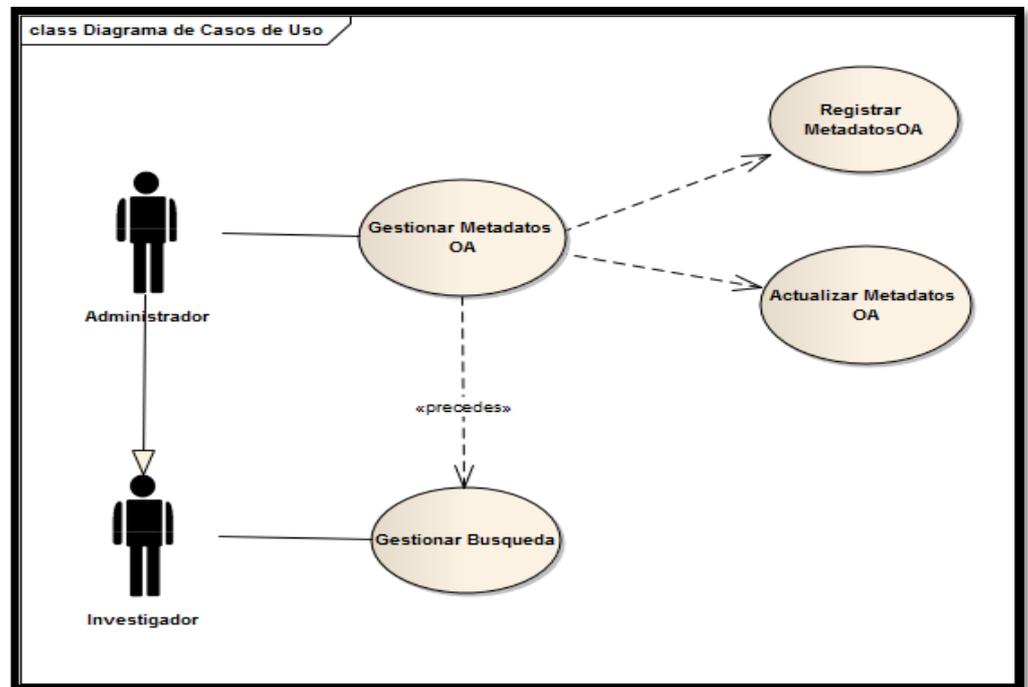


Figura 25. Identificación de los casos de uso

La descripción de los Casos de Uso y los requerimientos que satisfacen se encuentran detallados en el Anexo 1, donde se encuentra el análisis de requerimientos y análisis preliminar.

6.3.1.6. Modelo de Agentes.

Mediante estos modelos se identifica y define los Agentes necesarios en la implementación del sistema propuesto.

A continuación se presentan los modelos de los agentes: Administrador, Interfaz, Búsqueda y Servicios Web.

TABLA XVIII.
RESPONSABILIDADES DE AGENTES

Nombre Agente	Responsabilidades
Agente Administrador	<ul style="list-style-type: none"> • Inicializar todos los Agentes Inteligentes (Búsqueda, Interfaz y Servicios Web)
Agente de Búsqueda	<ul style="list-style-type: none"> • Busca Objetos de Aprendizaje sobre la Ontología. • Entregar resultados al Agente Interfaz
Agente Interfaz	<ul style="list-style-type: none"> • Cumplir peticiones de Interfaz. • Entregar resultados.
Agente de Servicios Web	<ul style="list-style-type: none"> • Llamar al cosechador de metadatos. • Buscar los Servicios Web de los repositorios y recuperar los metadatos.

- **Agente Administrador**

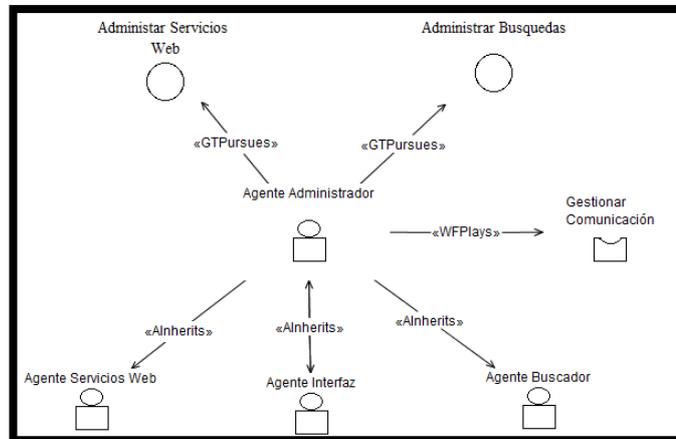


Figura 26. Agente Administrador

- **Agentes Servicios Web**

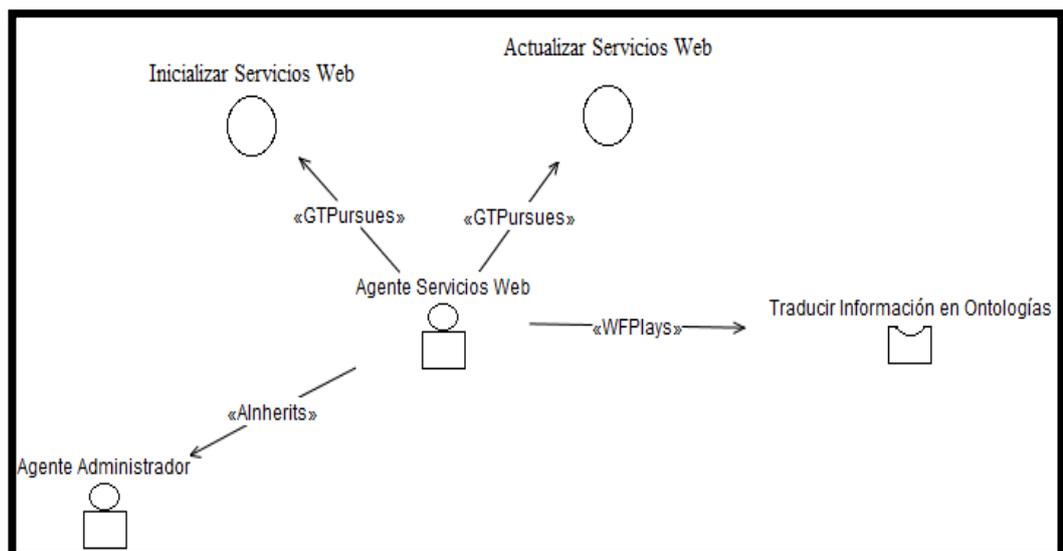


Figura 27. Agente Servicios Web

- **Agentes Interfaz**

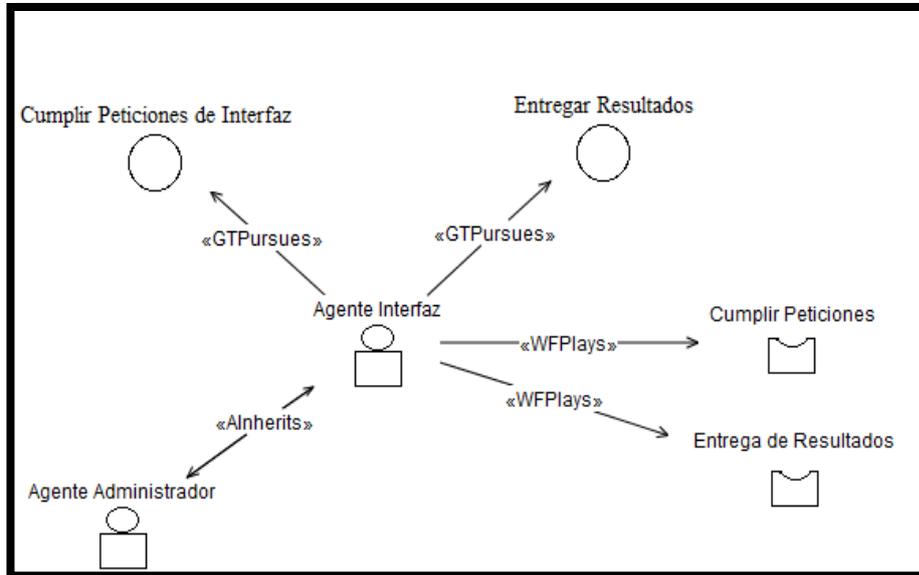


Figura 28. Agente Interfaz

- **Agentes Buscador**

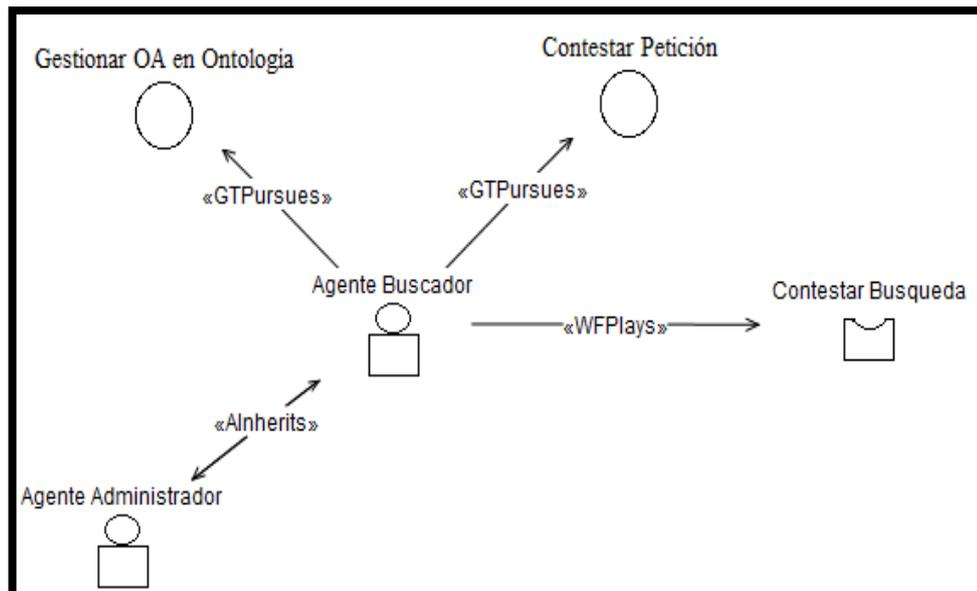


Figura 29. Agente Buscador

6.3.1.7. Modelo Objetivos y tareas

Los agentes identificados en el modelo anterior deben cumplir objetivos y tareas para garantizar la funcionalidad de la aplicación; a continuación se detalla cada uno de estos modelos.

- **Objetivos y tareas del Agente Buscador**

Los objetivos y tareas que persigue el Agente Buscador son de recibir consultas y dirigirse a la Ontología para extraer los Objetos de Aprendizaje y a su vez contestar con los resultados al Agente Interfaz, para que presente los resultados obtenidos al usuario.

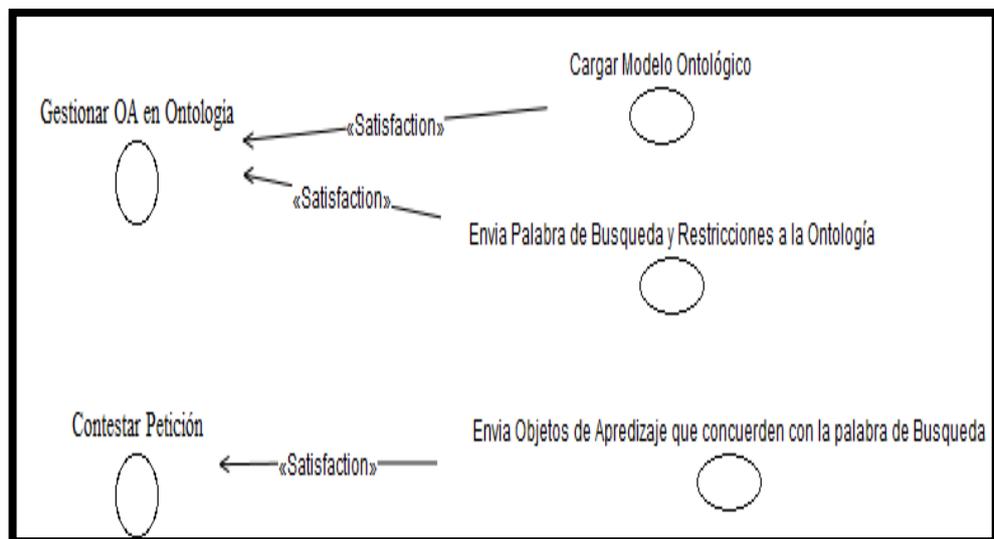


Figura 30. Diagrama de Objetivos Agente Buscador

- **Objetivos y tareas del Agente Administrador**

Este agente persigue el Objetivo de Administrar los Servicios Web y comunicarse con el Agente Buscador y mantener siempre la comunicación entre el Agente Interfaz, búsqueda y Servicios Web.

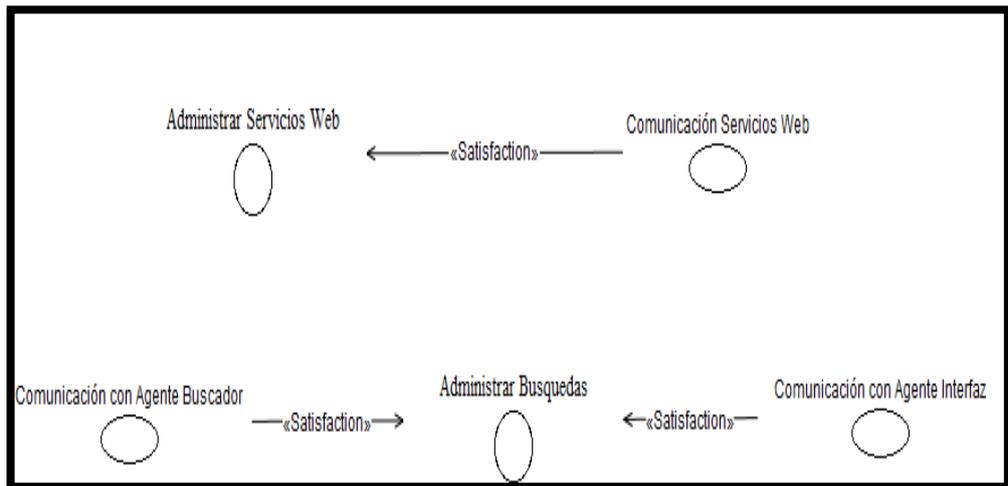


Figura 31. Diagrama de Objetivos Agente Administrador.

- **Objetivos y tareas del Agente Servicios Web**

La gestión y uso de los servicios Web es uno de los propósitos perseguidos por el agente Servicios Web y a su vez se debe de encargar de que se tener en cuenta las condiciones establecidas por el repositorio a la hora de ejecutar los servicios correspondientes.

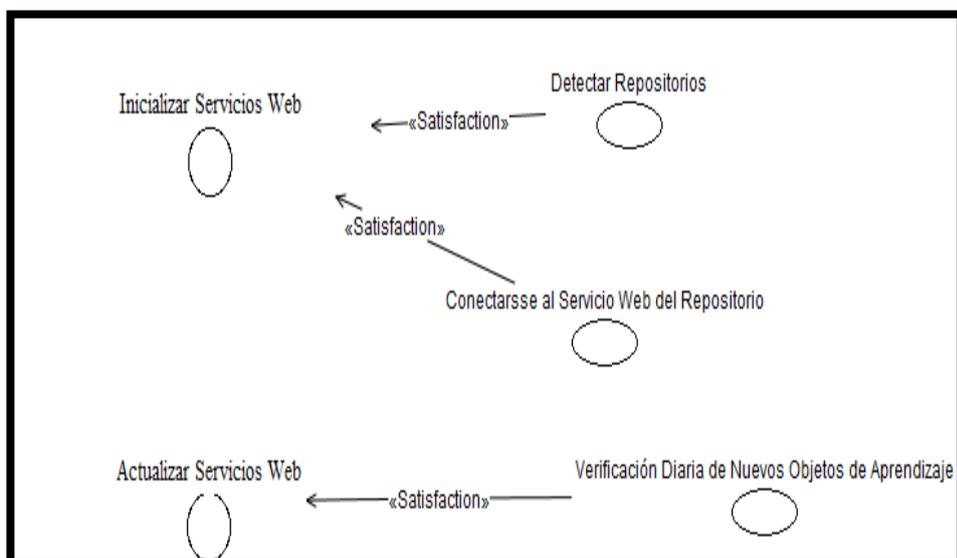


Figura 32. Diagrama de Objetivos Agente Servicios Web

- **Objetivos y tareas de Agente Interfaz**

Persigue dos objetivos principales, cumplir peticiones de interfaz y entregar resultados para que el investigador pueda establecer y satisfacer sus necesidades, complementariamente este agente se comunica con el agente Administrador y Búsqueda.



Figura 33. Diagrama de Objetivos Agente Interfaz

6.3.1.8. Modelos de Interacción

En el sistema a desarrollar es necesaria la interacción entre los diferentes agentes, a continuación se procede a mostrar los modelos de interacción:

- **Interacción Agente Interfaz - Agente Administrador**

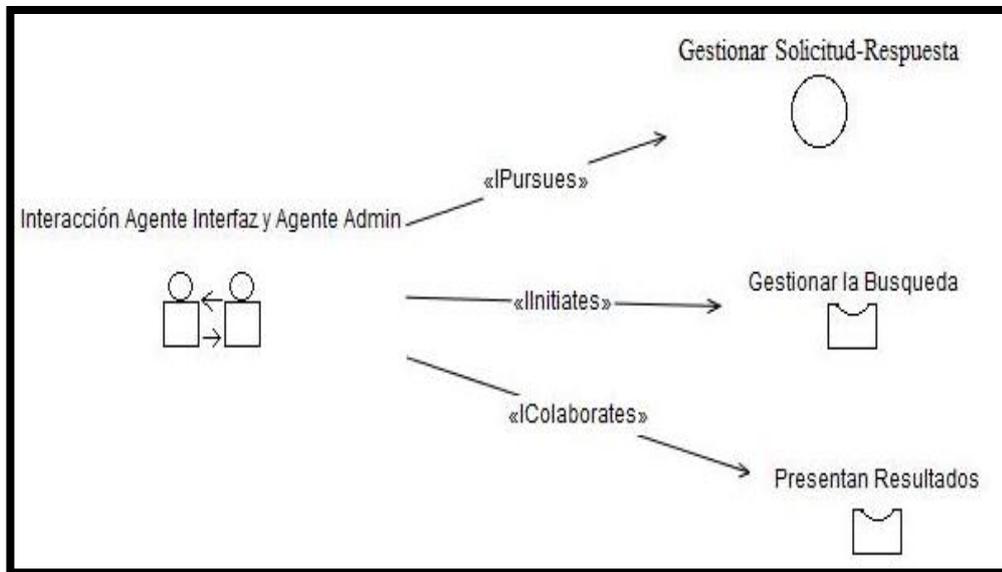


Figura 34. Diagrama de Interacción 1

- **Interacción Agente Administrador - Agente Servicios Web**

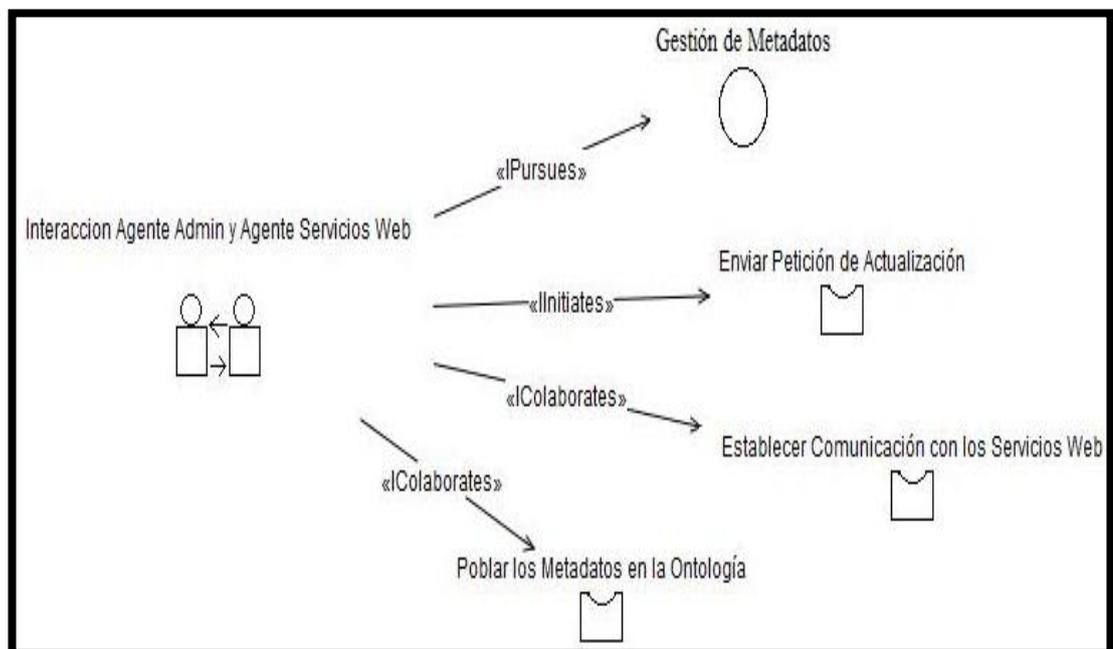


Figura 35. Diagrama de Interacción 2

- Interacción Agente Administrador - Agente Buscador

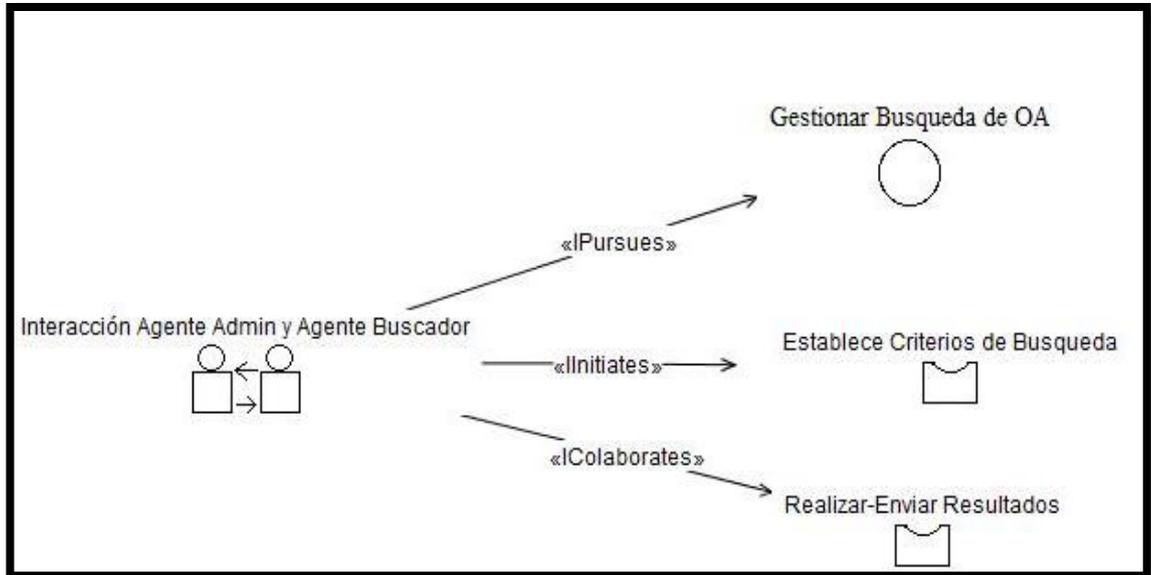
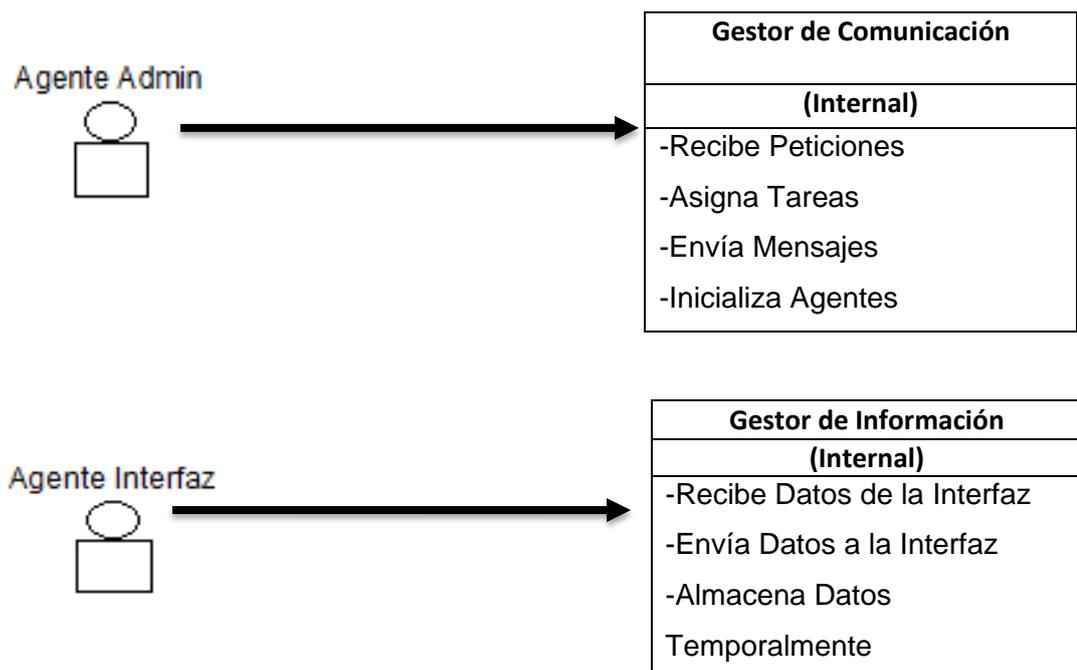
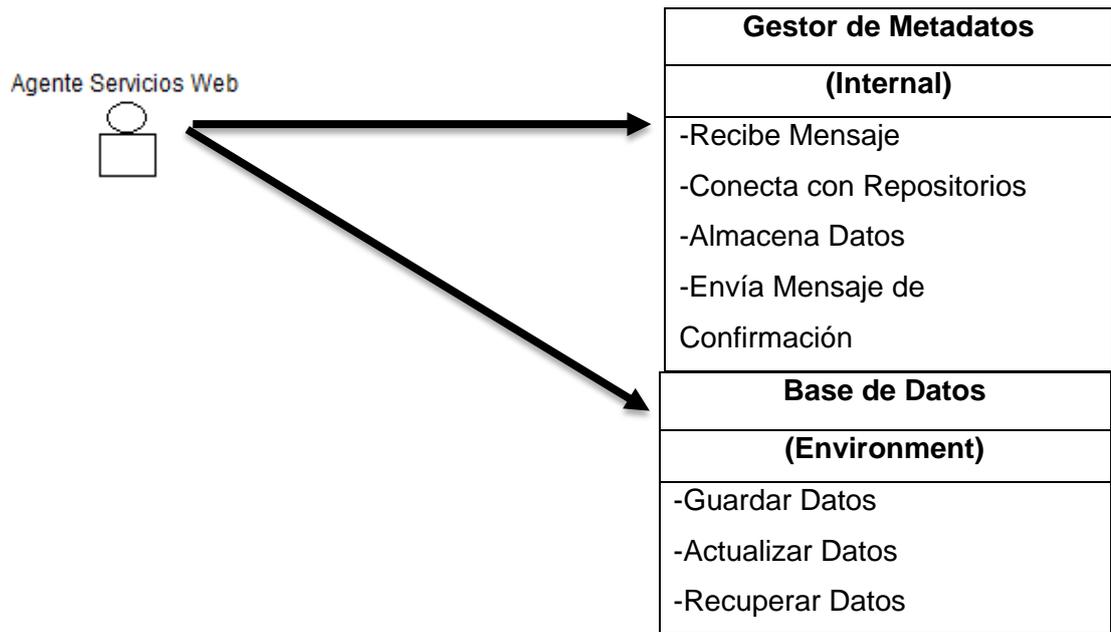
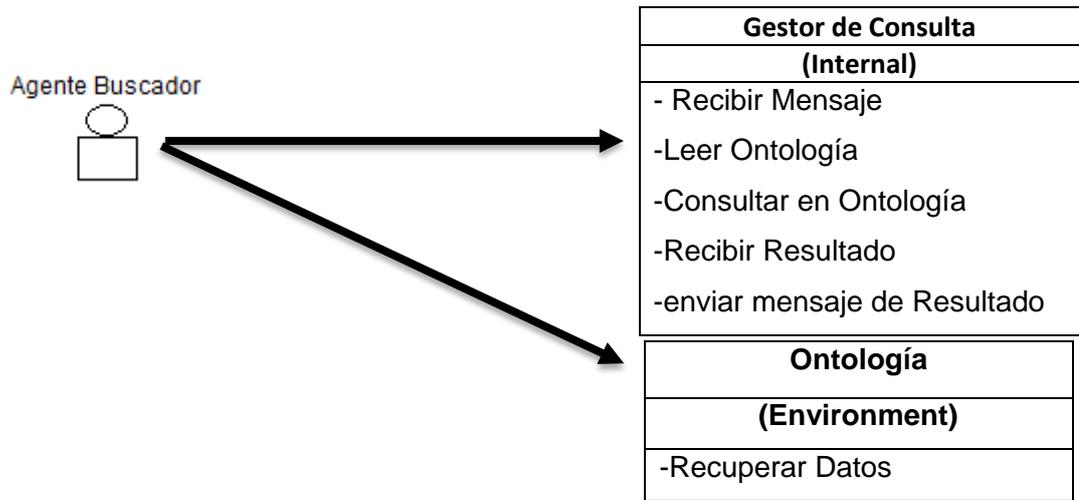


Figura 36. Diagrama de Interacción 3

6.3.1.9. Modelo de Entorno

Haciendo uso de estos modelos es posible definir el tipo de recursos y aplicaciones a utilizar para el funcionamiento del sistema.





6.3.1.10. Arquitectura del Sistema Multiagente

En base a los modelos definidos y diagramas producidos en las fases anteriores, se ha desarrollado un sistema basado en conocimiento que consume los servicios Web de repositorios de Objetos de Aprendizaje disponibles en la Web, los cuales proporcionan metadatos. La arquitectura general del sistema se presenta de forma gráfica en la Figura 37.

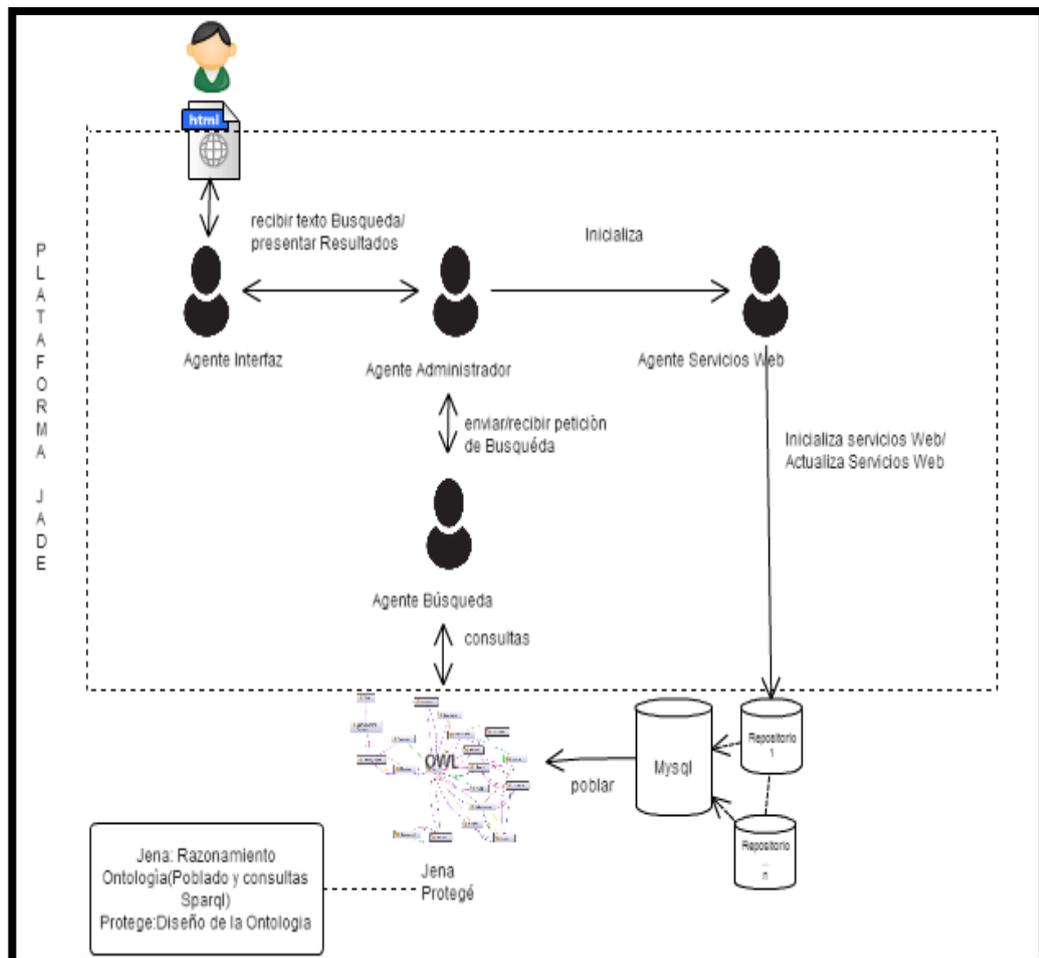


Figura 37. Arquitectura Multiagente para recuperación OAs.

6.3.2. Implementación

6.3.2.1. Diagrama de componentes

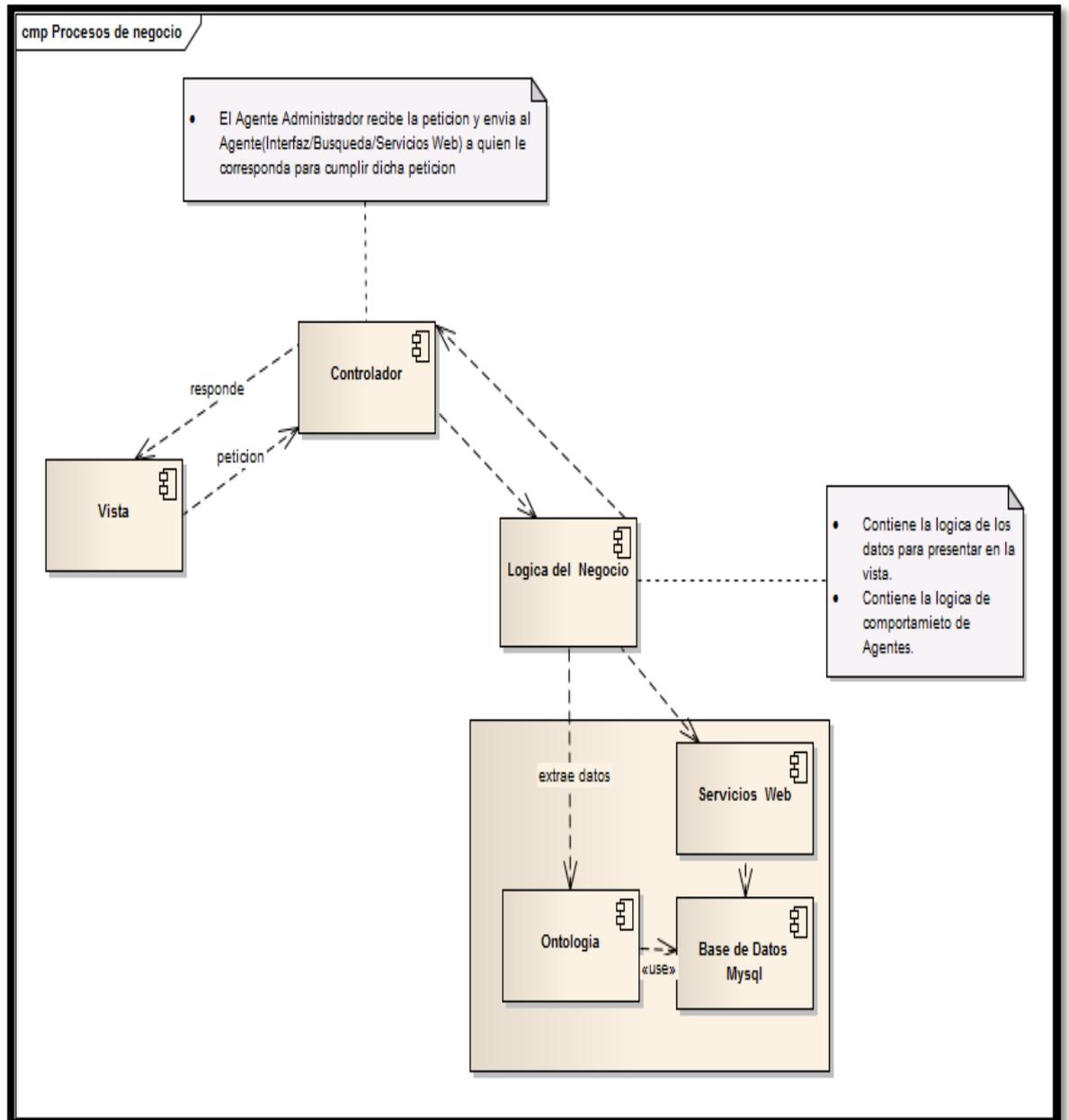


Figura 38. Diagramas de Componentes

6.3.3.2. Codificación

En base a los modelados y la arquitectura realizados en el apartado anterior se procede a la codificación de los agentes. La codificación de estos agentes se hace haciendo uso del lenguaje de programación Java y con la ayuda de la librería Jade junto a su plataforma e interfaz gráfica que facilita la visualización de los agentes.

6.3.3.2.1. Agente Interfaz

El agente interfaz extiende de la clase GatewayAgent, al extender de esta clase debe de incluir el método protegido processCommand el cual permite recibir el objeto que es enviado desde el controlador, mismo que contiene toda la información necesaria para realizar las consultas. A continuación se describe los métodos y comportamientos que integran este agente.

- **processCommand**

Este método se encarga de recibir un objeto como parámetro; dentro del método se transforma este objeto recibido en un objeto de la clase LOM, seguidamente se procesa el objeto y se envía un mensaje al agente administrador con el objeto recibido.

Se debe considerar las excepciones que se producen dentro del código de este método y además considerar que los objetos a utilizar deben de ser serializables.

```
ObjetoAprendizaje resultado = new ObjetoAprendizaje();

@Override
protected void processCommand(java.lang.Object u) {
    if (u instanceof ObjetoAprendizaje) {
        resultado = (ObjetoAprendizaje) u;
    }
    ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
    msg.addReceiver(new AID("AgenteAd", AID.ISLOCALNAME));
    try {
        msg.setContentObject(resultado);
    } catch (Exception e) {
        System.out.println("No se pudo enviar");
    }
    send(msg);
}
```

Figura 39. Método processCommand – Agente Interfaz.

- **setup**

El método setup es un método obligatorio dentro de todo agente; dentro de este método se definen los comportamientos o se agregan comportamientos ya existentes.

EL setup de este agente permite el retorno del objeto de aprendizaje con todas las coincidencias encontradas durante la búsqueda para posteriormente ser mostradas al usuario mediante la interfaz.

```
addBehaviour(  
    new CyclicBehaviour(this) {  
  
        @Override  
        public void action() {  
            ACLMessage recibido = receive();  
            if ((recibido != null) && (resultado != null)) {  
                try {  
                    ObjetoAprendizaje result = (ObjetoAprendizaje) recibido.getContentObject();  
                    resultado.setResultadoDC(result.getResultadoDC());  
                    resultado.setResultadoLom(result.getResultadoLom());  
                } catch (UnreadableException e) {  
                    System.out.println("no se pudo");  
                }  
  
                resultado.setPalabraClave(null);  
                resultado.setRestriccion1(null);  
                resultado.setRestriccion2(null);  
                resultado.setArchivoDC(null);  
                resultado.setArchivoLOM(null);  
                releaseCommand(resultado);  
  
            } else {  
                block();  
            }  
        }  
    }  
);
```

Figura 40. Método setup – Agente Interfaz.

6.3.3.2.2. Agente Administrador

El agente administrador es el encargado de gestionar el envío de mensajes y de inicializar los métodos de búsqueda y gestión de la ontología.

Para el cumplimiento de sus tareas implementa los siguientes métodos y comportamientos:

- **Comportamiento TickerBehaviour**

Se trata de un comportamiento temporizado que se ejecuta diariamente y a su vez crea los agentes encargados de la gestión del cosechado de metadatos.

A continuación se muestra el código empelado en este comportamiento:

```
addBehaviour(new TickerBehaviour(this, 86400000) {
    @Override
    protected void onTick() {
        OneShotBehaviour lom = new OneShotBehaviour(myAgent) {
            @Override
            public void action() {
                AgentContainer c = getContainerController();
                try {
                    AgentController a = c.createNewAgent("AgenteLom", "agentes.AgenteWSLom", null);
                    a.start();
                } catch (StaleProxyException e) {
                }
            }
        };
        OneShotBehaviour dc = new OneShotBehaviour(myAgent) {
            @Override
            public void action() {
                AgentContainer c = getContainerController();
                try {
                    AgentController a = c.createNewAgent("AgenteDc", "agentes.AgenteWSDc", null); //
                    a.start();
                } catch (StaleProxyException e) {
                }
            }
        };
        ParallelBehaviour p = new ParallelBehaviour(myAgent, ParallelBehaviour.WHEN_ALL);
    }
};
```

Figura 41. Comportamiento temporizador – Agente Administrador.

- **Comportamiento CyclicBehaviour**

Para poder gestionar la parte de la búsqueda se ha estimado conveniente la implementación de un método cíclico, el cual se encontrara siempre disponible respondiendo a las solicitudes de los usuarios; este comportamiento se encarga de pasar los parámetros de búsqueda al **agente buscador** para que sea este el que se encargue de encontrar las coincidencias y hacérselo llegar nuevamente al agente.

```

addBehaviour(new CyclicBehaviour(this) {
    int cont = 1;

    @Override

    public void action() {

        ACLMessage refresh = blockingReceive();
        if (refresh != null) {
            try {
                ObjetoAprendizaje result = (ObjetoAprendizaje) refresh.getContentObject();
                AgentContainer c = getContainerController();
                try {
                    AgentController a = c.createNewAgent("AgenteB", "agentes.AgenteBuscador", null);
                    a.start();
                } catch (StaleProxyException e) {

                }

                ACLMessage busqueda = new ACLMessage(ACLMessage.REQUEST);
                busqueda.addReceiver(new AID("AgenteB", AID.ISLOCALNAME));
                busqueda.setContentObject(result);
                send(busqueda);

            } catch (Exception e) {

            }
        } else {

        }
    }
}

```

Figura 42. Comportamiento cíclico – Agente Administrador.

6.3.3.2.3. Agente Buscador

El agente buscador se encarga netamente de llamar a los métodos de búsqueda implementados, los mismos que permiten obtener las coincidencias de metadatos que se encuentran cargados en la ontología; para realizar una búsqueda con un mejor tiempo de respuesta el agente se encarga de ejecutarlos en forma paralela y posteriormente enviar el objeto con todos los resultados para que puedan ser visualizados. Adicional a ello emplea un condicional para poder saber cuándo debe de realizar un búsqueda simple o una búsqueda avanzada de objetos de aprendizaje.

```
public void setup() {
    addBehaviour(new CyclicBehaviour(this) {
        @Override
        public void action() {
            ACLMessage msg = receive();
            if (msg != null) {
                try {
                    ObjetoAprendizaje oa = new ObjetoAprendizaje();
                    oa = (ObjetoAprendizaje) msg.getContentObject();
                    String clave = oa.getPalabraClave();
                    ObjetoAprendizaje res = new ObjetoAprendizaje();
                    consultarOntologia consulta = new consultarOntologia();
                    if (oa.getRestriccion1() != null) {
                        String rest1 = oa.getRestriccion1();
                        String rest2 = oa.getRestriccion2();
                        res.setResultadoLom(consulta.BusquedaAvanzadaLOM(clave, oa.getArchivoLOM(), rest1, rest2));
                        res.setResultadoDC(consulta.BusquedaAvanzadaDC(clave, oa.getArchivoDC(), rest1, rest2));
                        ACLMessage msj = new ACLMessage(ACLMessage.REQUEST);
                        msj.addReceiver(new AID("ControlContainer-1", AID.ISLOCALNAME));
                        try {
                            msj.setContentObject(res);
                        } catch (IOException e) {
                            System.out.println("NO SE PUDO LLENAR EL MSJ CORRECTAMENTE" + e);
                        }
                    }
                    send(msj);
                } else {
                    res.setResultadoDC(consulta.BusquedaSimpleDC(clave, oa.getArchivoDC()));
                    res.setResultadoLom(consulta.BusquedaSimpleLOM(clave, oa.getArchivoLOM()));
                }
            }
        }
    });
}
```

Figura 43. Comportamientos de búsqueda – Agente Buscador.

6.3.3.2.4. Agente Web Servicios Dublin Core

El agente permite la actualización de los metadatos o de ser necesaria una nueva inicialización de los mismos.

Para ello hacer uso de los métodos relacionados a los servicios web. Para reducir el tiempo necesario en el desarrollo de esta función se implementa con la ayuda de comportamientos paralelos.

A continuación se da a conocer la parte de código encargada de realizar esta actualización.

```
@Override
public void action() {
    try {
        ServiciosWebDublinCore swDublinCore = new ServiciosWebDublinCore();
        System.out.println("crea");
        swDublinCore.actualizar(4, "http://148.202.105.131/dspace-oai/request");
    } catch (Exception e) {

    }
}

OneShotBehaviour fedora = new OneShotBehaviour(this) {
    @Override
    public void action() {
        try {
            ServiciosWebDublinCore swDublinCore = new ServiciosWebDublinCore();
            System.out.println("fedora");
            swDublinCore.actualizar(5, "http://stacks.cdc.gov/fedora/oai");
        } catch (Exception e) {

        }
    }
};

ParallelBehaviour p = new ParallelBehaviour(this, ParallelBehaviour.WHEN_ALL);
p.addSubBehaviour(rad);
p.addSubBehaviour(jorum);
p.addSubBehaviour(crea);
p.addSubBehaviour(fedora);
addBehaviour(p);
```

Figura 44. Comportamientos paralelos – Agente WSDC.

6.3.3.2.5. Agente Web Servicios Lom

AL igual que el agente anterior este se encarga de la actualización de los metadatos que se encuentren bajo el estándar LOM o de ser necesaria una inicialización de dichos servicios. Para lograr esto hace uso de comportamientos paralelos y de los métodos implementados para los servicios web.

A continuación se da a conocer las líneas de código necesarias para la ejecución de esta tarea:

```

OneShotBehaviour laclo = new OneShotBehaviour(this) {
    @Override
    public void action() {
        try {

            ServiciosWebLOM p = new ServiciosWebLOM();
            p.inicializar(1, "http://laflor.laclo.org/admin/services/oai");
        } catch (Exception e) {

        }

    }
};

ParallelBehaviour p = new ParallelBehaviour(this, ParallelBehaviour.WHEN_ALL);
p.addSubBehaviour(laclo);

addBehaviour(p);
}

protected void takeDown() {
    try {
        DFService.deregister(this);
    } catch (Exception e) {

    }
}

```

Figura 45. Comportamientos paralelos – Agente WSLOM.

6.3.3.2.6. Ejecución de los agentes inteligentes

Los agentes inteligentes necesitan de una plataforma para poder ser ejecutados, la librería jade a más de permitir la creación de agentes nos ofrece una plataforma para poder ejecutarlos.

Con la realización de las respectivas configuraciones en las variables de entorno podemos tener acceso a la interfaz gráfica de la plataforma antes mencionada. A continuación se muestra los agentes que intervienen en el funcionamiento del sistema SMAS.

6.3.3.2.7. Plataforma y Contenedor de los agentes inteligentes

Para la visualización de la plataforma jade se debe de ejecutar el siguiente comando desde la consola:

java jade.Boot -gui

Al ejecutar el comando se procede a levantar y mostrar la plataforma jade.

Durante la ejecución del sistema SMAS se procede a la creación de un nuevo contenedor donde se albergaran los agentes y por ende se inicializa la creación de los agentes inteligentes.

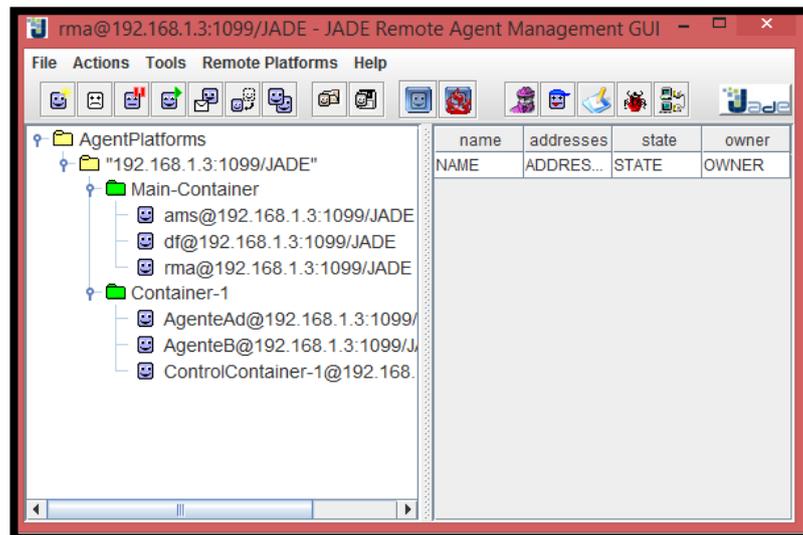


Figura 46. Contenedor de Agentes

6.3.3.2.8. Envió de mensajes entre agentes

La librería jade permite realizar una inspección de los agentes para conocer su estado e incluso el paso de mensajes que sucede entre los mensajes; para ello hace uso de un tipo especial de agentes el agente Sniffer.

A continuación se muestra una captura de pantalla donde se ve el intercambio de información entre los agentes que participan al momento de realizar búsquedas en el sistema.

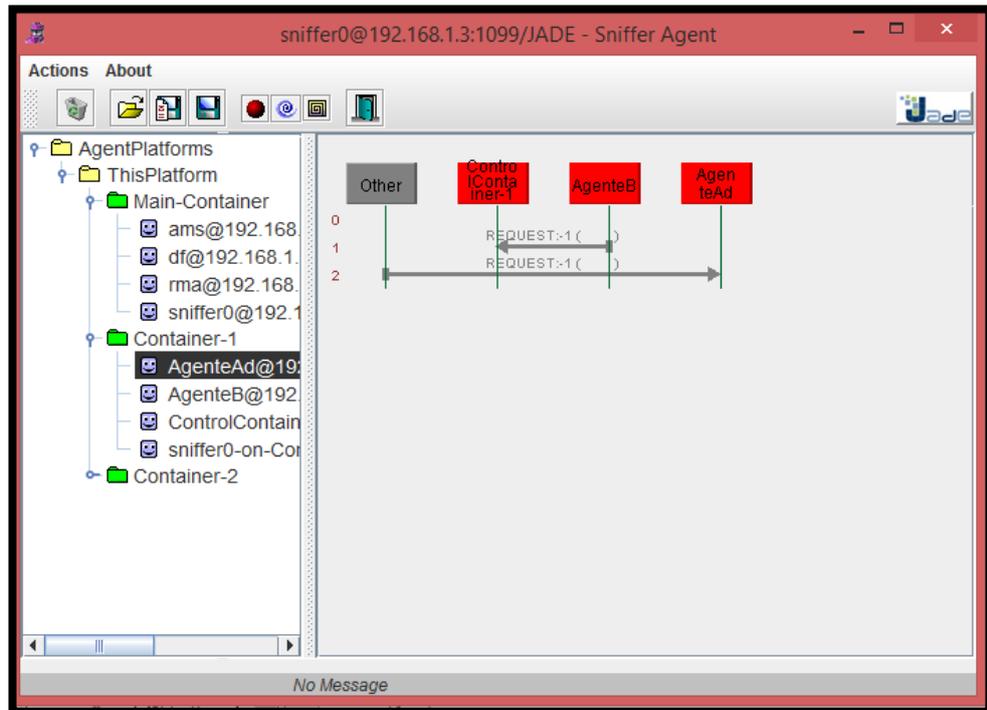


Figura 47. Agente Sniffer

6.4. FASE IV: Evaluar el funcionamiento de la aplicación en la búsqueda y recuperación de Objetos de Aprendizaje.

La forma de evaluar el desempeño del producto final, fue con la utilización de la herramienta JMeter para pruebas de carga y funcionalidad; luego al ser desplegada en un servidor, en cual se corrigieron errores técnico y lógicos y se puso a disposición de usuarios para su evaluación final.

6.4.1. Pruebas de funcionalidad

Para realizar las pruebas de funcionalidad y rendimiento de SMAS se ha empleado la herramienta JMeter, esta herramienta permite simular la concurrencia de un número de usuarios haciendo diversas peticiones durante un tiempo determinado; y en base a esta simulación obtener resultados como el tiempo exacto de respuesta de la aplicación.

Para la evaluación de los resultados se hizo uso de 2 componentes que incluye la herramienta JMeter en su versión 2.11:

Aggregate Graph: obtiene de forma precisa los datos de la mediana y la línea al 90%:

- **URL:** etiqueta de la muestra.
- **#Muestras:** cantidad de thread utilizados para la URL.
- **Media:** tiempo promedio en milisegundos para un conjunto de resultados.
- **Mediana:** tiempo promedio del percentil 50.
- **Línea del 90%:** máximo tiempo utilizado por el 90% de la muestra.
- **Min:** tiempo mínimo de la muestra de una determinada URL.
- **Max:** tiempo máximo de la muestra de una determinada URL.
- **%Error:** porcentaje de requerimientos con errores.
- **Rendimiento:** rendimiento medio en los requerimientos por segundo/minuto/hora.
- **KB/sec:** rendimiento medido en Kbytes por segundo.

Gráfico de resultados: se muestra de forma gráfica la media, mediana, dispersión y el rendimiento de la aplicación.

A continuación se indicará los resultados obtenidos con la herramienta JMeter en la cual se ha definido 2 pruebas de 100 y 200 números de hilos cada uno, los cuales simulan 100 y 200 usuarios accediendo respectivamente al servidor.

6.4.1.1. Prueba Nro. 1

Esta prueba fue realizada simulando las peticiones HTTP de 100 usuarios, cada 100 segundos. A continuación se indican los resultados obtenidos.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec
SistemaMulti...	100	245	198	350	389	909	192	1228	0,00%	1,0/sec	6,5
SistemaMulti...	100	300	218	434	602	1246	205	1510	0,00%	1,0/sec	9,8
SistemaMulti...	100	282	234	382	559	905	203	970	0,00%	1,0/sec	10,3
Total	500	359	230	603	782	1361	192	3556	0,00%	5,0/sec	49,9

Figura 48. Tabla de resultados prueba 1.

Como puede observarse en la Figura 48, el tiempo promedio para acceder a una página es 0,359 segundos, realizándose un total de 500 requerimientos al servidor.

El tiempo total utilizado para los 100 usuarios se puede calcular con la siguiente fórmula:

$$\text{Tiempo Total (TT)} = \# \text{muestras} * \text{media}$$

$$\text{TT} = 500 * 359 = 179500 \text{ milisegundos}$$

El tiempo promedio total requerido por cada usuario, se puede calcular de la siguiente manera:

$$\text{Tiempo promedio (TP)} = ((\text{Tiempo Total} / 1000) / 60) / \text{cantidad de usuarios}$$

$$\text{TP} = ((179500 / 1000) / 60) / 100$$

$$\text{TP} = 2,9916 / 100 = 0,0299 \text{ minutos}$$

6.4.1.2. Prueba Nro.2

Para esta prueba se configuro 200 usuarios cada 200 segundos. A continuación en la Figura 49, se indican los resultados obtenidos.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec
SistemaMulti...	200	416	222	839	1055	2019	190	2930	0,00%	1,0/sec	6,5
SistemaMulti...	200	582	402	1109	1377	2145	207	2432	0,00%	1,0/sec	9,7
SistemaMulti...	200	511	396	973	1124	1243	205	1687	0,00%	1,0/sec	10,2
Total	1000	571	413	1070	1326	2432	190	9664	0,00%	4,9/sec	49,9

Figura 49. Tabla de resultados prueba 2

Como puede observarse en la Figura 49, el tiempo promedio para acceder a una página es 571 segundos, realizándose un total de 1000 requerimientos al servidor.

El tiempo total utilizado para los 200 usuarios se puede calcular con la siguiente fórmula:

$$\text{Tiempo Total (TT)} = \# \text{muestras} * \text{media}$$

$$TT = 1000 * 571 = 571000 \text{ milisegundos}$$

El tiempo promedio total requerido por cada usuario, se puede calcular de la siguiente manera:

$$\text{Tiempo promedio (TP)} = ((\text{Tiempo Total} / 1000) / 60) / \text{cantidad de usuarios}$$

$$TP = ((571000 / 1000) / 60) / 200$$

$$TP = 9,52 / 200 = 0,0476 \text{ minutos}$$

6.4.1.3. Gráficas obtenidas en JMeter

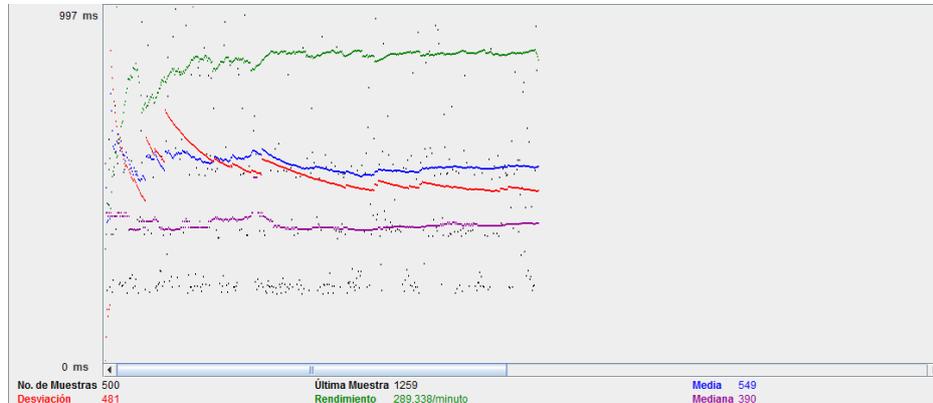


Figura 50. Gráfica de prueba 1

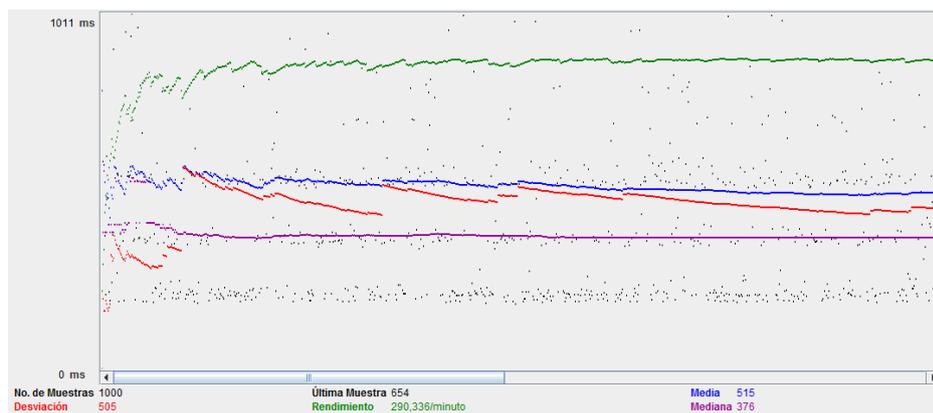


Figura 51. Gráfica de prueba 2

Como podemos observar los tiempos obtenidos son relativamente bajos lo que nos permite afirmar que el servidor soportara la conexión e varios usuarios simultáneamente sin presentar retrasos ni caídas del mismo. Esta característica es relevante para el sistema pues al tratarse de un buscador recibirá un gran número de peticiones de usuarios.

6.4.2. Nivel Técnico.

Las herramientas utilizadas para el desarrollo del software tanto de Ontologías como agentes presentaron un rendimiento positivo, ya que el computador utilizado para desarrollar la aplicación supera notablemente los requerimientos óptimos, además el servidor Red Hat Cloud que se utiliza para el alojamiento de la aplicación en la nube, a pesar de su gratuidad de uso nos confiere un espacio de almacenamiento de 2GB lo cual es suficiente para el software y supera las expectativas, también nos brinda flexibilidad y sobretodo permiten personalizar un entorno adecuado a nuestras necesidades razón por la cual no se ha presentado ningún inconveniente en lo que concierne a las tecnologías utilizadas.

6.4.3. Prueba de validación de Usuarios

La prueba de validación de usuarios se llevó al cabo con ayuda de estudiantes de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, quienes utilizaron SMAS y luego se les aplicó una encuesta, se puede observar detalladamente los resultados y análisis de la encuesta realizada en el Anexo 5, la cual permitió validar la navegabilidad y usabilidad del software.

7. Discusión

7.1. Desarrollo de la propuesta alternativa

El presente trabajo de titulación se basa en cuatro objetivos, los cuales permitieron dar cumplimiento con el objetivo general, a continuación se describe cada una de las fases:

7.1.1. Revisar el estado del arte en lo referente a Objetos de Aprendizaje, Ontologías y Agentes Inteligentes para la búsqueda de información.

Para dar cumplimiento al primer objetivo se realizó el estado de arte sobre Objetos de Aprendizaje, Ontologías y Agentes Inteligentes, donde se pudo conocer los principales estándares de catalogación de OA, metadatos, repositorios y servicios Web; en lo que respecta a Ontologías los aspectos que se tomaron en cuenta fueron componentes, Lenguajes ontológicos y consultas; Y Agentes Inteligentes se estudió los principales tipos de Agentes y arquitecturas, luego de conocer la teoría se realizó un análisis de herramientas, tecnologías y metodologías mediante tablas comparativas para elegir las más adecuadas para la construcción de SMAS.

La metodología escogida para el presente trabajo de titulación fue ICOIX, la cual ayudó a llevar de forma organizada el desarrollo de software y esta se complementó con la metodología Methontology para el diseño de la Ontología y otra para la construcción del Sistema Multiagente Ingenias formando una metodología híbrida, la cual nos permitió guiar y organizar las actividades para cumplir los objetivos planteados.

7.1.2. Diseñar una Ontología basada en los estándares establecidos para la representación de objetos de aprendizaje.

Para el desarrollo de la ontología fue necesario el estudio y el análisis comparativo de metodologías ontológicas que guían el proceso de creación de ontologías la cual se lo hizo en el primer Objetivo. Mediante Methontology se pudo modelar el dominio del problema de la misma, para lo cual se tomaron en cuenta los estándares de

catalogación de Objetos de Aprendizaje LOM y Dublin Core, el primero permite representar la catalogación de Objetos de aprendizaje de manera más detallada, mientras que el otro permite la representación de Objetos de Aprendizaje de manera sencilla y corta, esto son los dos únicos estándares que manejan los repositorios en los cuales se encuentran alojados los OA.

Para el razonamiento de la Ontología se lo hizo a través del motor de inferencias Jena, para las cuales el criterio de búsqueda debe ser escrito de la misma forma como se encuentran representadas las instancias dentro de los archivos de las ontologías.

Construida la ontología es necesario instanciar, proceso que actualmente se realiza de manera automática en base a los recursos almacenados, para lo cual se hace mediante un recolector de metadatos que usa peticiones HTTP recupera los metadatos de OA y los almacena en MySQL, y luego estos son cargados con el motor de inferencia Jena como individuos de la Ontología y para las consultas se lo hizo con el lenguaje SPARQL.

7.1.3. Construir un Sistema Multiagente para la búsqueda de Objetos de Aprendizaje.

Para el desarrollo y cumplimiento de este objetivo se empleó la metodología Ingenias, la cual permite detallar el desarrollo del software a partir del empleo de Agentes Inteligentes y el uso de ontologías. En la arquitectura modelada se determinó la utilización de los siguientes agentes: Agente Administrado encargado de inicializar todos los Agentes Inteligentes (Búsqueda, Interfaz y Servicios Web); Agente de Búsqueda encargado de buscar Objetos de Aprendizaje sobre la Ontología y la entregar resultados al Agente Interfaz; Agente Interfaz encargado de cumplir peticiones de Interfaz y entregar resultados; Agente de Servicios Web encargado de llamar al cosechador de metadatos. Y buscar los Servicios Web de los repositorios y recuperar los metadatos. La implementación de esta arquitectura permitió que el Sistema Multiagente busque Objetos de Aprendizaje, los OAs resultantes serán consultados en la Ontología, los agentes se encargan de gestionar los Servicios Web y poblar la Ontología, y también serán los encargos de gestionar las búsquedas y presentar los resultados.

7.1.4. Evaluar el funcionamiento de la aplicación en la búsqueda de Objetos de Aprendizaje.

El desarrollo de este objetivo permitió garantizar la funcionalidad, eficiencia y adaptabilidad del software mediante la utilización de pruebas de rendimiento y usabilidad a nuestros posibles usuarios.

Para las pruebas de rendimiento y funcionalidad se utilizó el software gratuito JMeter, en el cual se generó un caso de prueba a través de la navegación de usuarios en las páginas en donde JMeter actúa como proxy para capturar los pasos que se realiza cuando se navega, además con esta herramienta se obtiene diferentes resultados como gráficas que indican el rendimiento de la aplicación ante las solicitudes realizadas por los usuarios.

Las pruebas de usabilidad permitieron validar la navegabilidad y accesibilidad del software, para lo cual se partió de encuestas realizadas a estudiantes de la Carrera de Ingeniería en Sistemas, quienes luego de haber hecho uso de SMAS y contestar la encuesta quienes nos supieron indicar su apreciación sobre este sistema.

7.2. Valoración técnica económica ambiental

7.2.1. Valoración técnica económica

Desde el punto de vista técnico y económico es factible y necesario, puesto que permitirá incentivar la utilización de recursos educativos de calidad y fiables dado que son creados bajos estándares y estos son accedidos gratuitamente a partir de repositorios de la Web.

La tabla XIX se indica los recursos humanos que hace referencia al personal que intervino en el desarrollo del software, siendo los investigadores los postulantes de este Trabajo de Titulación, además en el transcurso del desarrollo, se contó con las tutorías del director de tesis.

TABLA XIX.
RECURSOS HUMANOS

Equipo Trabajo	Tiempo (Horas)	Precio/ Hora (\$)	Valor Total (\$)
Investigadores (2)	1200	\$ 7,00	\$ 8400.00
Director de Tesis	-	-	-
SUBTOTAL (\$)			\$ 5600.00

En la Tabla XX se hace una descripción detallada de los recursos materiales necesarios para presentar los avances, borradores y el informe final del Trabajo de Titulación con sus respectivos manuales de usuario y programador.

TABLA XX.
RECURSOS MATERIALES

Recursos	Cantidad	Valor unitario	Valor Total
Resma de papel A4	2	\$ 4.00	\$ 8.00
Botellas de Tinta Continua	4	\$ 5	\$ 20.00
Copias	200	\$ 0.02	\$ 4.00
Anillados	3	\$ 2.00	\$ 6.00
Empastados	3	\$ 8.00	\$ 24.00
Internet	800 horas	\$ 0.60	\$ 480
CDs	6	\$ 0.50	\$ 3.00
Suministros de oficina (cuadernos, lápiz, borradores, perfiles, entre otros).	-	\$ 10.00	\$ 10.00
SUBTOTAL (\$)			\$ 550

En la tabla XXI se detalla los recursos técnicos y tecnológicos que se emplearon en el desarrollo del presente trabajo de titulación, entre los cuales está el Hardware necesario para llevar a cabo el desarrollo y software, se puede observar que algunos fueron adquiridos de forma gratuita.

TABLA XXI.
RECURSOS TÉCNICOS Y TECNOLÓGICOS.

Recursos	Cantidad	Valor unitario	Valor Total
HARDWARE			
Portátil DELL DELL INSPIRON	1	\$ 1200.00	\$ 1200.00
Portátil HP Pavilion	1	\$ 1150.00	\$ 1150.00
USB	2	\$ 10.00	\$ 20.00
Impresora	1	\$ 240	\$ 240

SOFTWARE			
Enterprise Architect	Gratuito	\$ 0.00	\$ 0.00
Protegé	Gratuito	\$ 0.00	\$ 0.00
Virtuoso	Gratuito	\$ 0.00	\$ 0.00
MySQL	Gratuito	\$ 0.00	\$ 0.00
Servidor JBoss	Gratuito	\$ 0.00	\$ 0.00
Servidor de Hosting	Gratuito	\$ 0.00	\$ 0.00
SUBTOTAL (\$)			\$ 2610.00

En la Tabla XXII. Se aprecia la suma total de todos los recursos tanto humanos como materiales y técnicos y tecnológicos usados en el proyecto, que da una aproximación del costo real.

TABLA XXII.
PRESUPUESTO DEL PROYECTO

Recursos	Subtotal
Recursos humanos	\$ 8400.00
Recursos materiales	\$ 550.00
Recursos técnicos y tecnológicos	\$ 2610.00
Total	\$ 11560.00

7.2.2. Valoración ambiental

Al tratarse de un trabajo que pretende impulsar la utilización de Objetos de Aprendizaje disponibles en la Web, representa un ahorro significativo en cuanto a la utilización de recursos materiales provenientes del medio ambiente como es el papel para la difusión de información ya que se aprovechara recursos gratuitos de la Web.

8. Conclusiones

Este apartado de la memoria expone las principales conclusiones extraídas a partir de la misma, revisando la principal propuesta del trabajo y los resultados obtenidos.

- Los resultados obtenidos en este proyecto ponen en evidencia que los agentes de software son actualmente un paradigma que cuenta con un nivel aceptable tanto de investigación como de desarrollo, es así que en el presente trabajo se demuestra la aplicabilidad de los sistemas Multi-agente en ámbito búsqueda de recursos educativos.
- Los metadatos constituyen un pilar fundamental dentro de la semántica ya que a través de ellos se puede identificar un recurso dentro de la web, pese a estas descripciones que los metadatos otorgan, existe una carencia de semántica en los mismos, por esta razón el uso de ontologías que permite adicionar relaciones, axiomas y reglas de inferencia para definir un recurso dentro de la web y poder realizar búsquedas.
- El ingreso de metadatos dentro de una estructura ontológica, al realizarse de forma manual resulta tedioso, confuso y hasta costoso, es por ello que se debe buscar herramientas para instanciar la Ontología de individuos de manera automática.
- El aporte fundamental de nuestro enfoque está en la integración de modelos y estrategias de dos áreas como son: agentes de software y recuperación de información de Ontologías para producir un sistema de interacciones dialógicas robusto y fácilmente utilizable por usuarios inexpertos, la inclusión de semántica en los Objetos de Aprendizaje y su importancia en el contexto educativo.
- La tecnología de agentes en general y la de los Sistemas Multiagente en particular ofrecen características muy apropiadas para el desarrollo de aplicaciones distribuidas y complejas en entornos donde se disponga de numerosos componentes con diferentes niveles de experiencia e interés conflictivos, los agentes añaden a este tipo de sistemas propiedades como la autonomía y comportamiento dirigido por objetivos que permiten el desarrollo de numerosas actividades sin la necesidad de internación humana.

9. Recomendaciones

Luego de haber realizado el presente trabajo de titulación se recomienda:

- Realizar un análisis correcto de herramientas y tecnología que ayuden en el desarrollo del software y que permitan tomar decisiones de mejoramiento continuo, con el fin de obtener un sistema de calidad.
- Al ser este sistema desarrollado en el Ámbito de Recursos Educativos Abiertos, se lo puede utilizar como base para nuevos proyectos de gestión de información fiable en el ámbito educativo.
- Para agregar contenido semántico a la Web se hace necesario que quienes crean el contenido en la Web, tengan la cultura del uso de metadatos para que los buscadores, que indexan en función de metadatos, nos den el resultado esperado.
- Se recomienda en caso de desarrollar un software donde se utilice diferentes tecnologías como Ontologías y Agentes de software emplear una metodología híbrida que permita incluir actividades tanto para el diseño de la Ontología y otra para construir el entorno de los Agentes ya que no existe ninguna metodología definida en la actualidad.
- Durante la codificación de Agentes es recomendable hacer uso de la interfaz gráfica que ofrece la plataforma JADE puesto que esta nos permite ir inspeccionando la comunicación y envío de mensajes entre agente para poder tomar un control y aplicar los correctivos necesario.
- Al consumir los servicios Web de un repositorio de Objetos de Aprendizaje se debe revisar que este no tengan Objetos de Aprendizajes obsoletos o dañados ya que algunos repositorios son creados y no son debidamente mantenidos.
- Se recomienda que el Sistema desarrollado se adjunte a los servicios informáticos que ofrece la Universidad Nacional de Loja mediante su portal web.

10. Bibliografía

[1] M. Rodríguez, “*Un modelo para la organización semiautomática de contenido educativo desde repositorios abiertos de objetos de aprendizaje*”, Revista Electrónica de Investigación Educativa, Vol. 16, Núm. 1, 2014.

[2] C. López, “*Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*”, España, 2010.

[3] P. Rodríguez, G. Isaza and N. Duque, “*Búsqueda personalizada en Repositorios de Objetos de Aprendizaje a partir del perfil del estudiante*”, Colombia, 2012.

[4] R. Nappa and B. Pandiella, “*ESTUDIO Y APLICACIÓN DE OBJETOS DE APRENDIZAJE A TRAVÉS DEL USO DE RECURSOS EDUCATIVOS ABIERTOS*”, Rev. Electrónica de Tecnología Educativa, Núm. 39, marzo 2012.

[5] Y. Toll, L. Ruiz and Y. Casañola, “*LA CALIDAD DE LOS OBJETOS DE APRENDIZAJE PRODUCIDOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*”, Rev. Electrónica de Tecnología Educativa, Núm. 36, junio 2012.

[6] D. Menéndez, H. Víctor, M. Castellanos, C. Vidal and A. Segura, “*Un Modelo de Calidad de Objetos de Aprendizaje basado en la Semántica de sus Metadatos*”, Universidad Autónoma de Yucatán, México, 2012.

[7] (2012) Dublin Core Homepage. Accessed: Septiembre 2014. [Online]. Available: <http://dublincore.org/documents/dces/>

[8] V. Tabares, N. Duque, J. Moreno; D. Ovalle and R. María, “*Evaluación de la calidad de metadatos en repositorios digitales de objetos de aprendizaje*”, Rev. Interam. Bibliot., vol.36 no.3, 2013.

[9] C. López Guzmán, F. García, P. Pernías, “*Desarrollo de repositorios de objetos de aprendizaje a través de la reutilización de los metadatos de una colección digital*”, RED, 2012.

[10] (2014) Open Archives Protocol Homepage. Accessed: Septiembre 2014. [Online]. Available: <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>

- [11] R. Arenas, “*Web Semántica*”, Universidad Carlos III Leganés, Madrid, Available: <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/04.pdf>
- [12] E. Peis, Y. Hassan, “*Ontologías, metadatos y agentes: recuperación semántica de la información*”, Universidad de Granada, España, Available: <http://www.yusef.es/jotri2003.pdf>.
- [13] A. Lozano, “*Ontologías en la Web Semántica*”, Universidad de Extremadura. España, Available: http://www.anobium.es/docs/gc_fichas/doc/68ERfhjkmv.pdf
- [14] K. Abab and F. Chica, “*Enriquecimiento Semántico de Guías de Programación Electrónica*”, Univ. De Cuenca, Cuenca, 2014.
- [15] M. Rubio, “*Propuesta de Ontología OWL para la representación de la oferta de estudios Universitarios en España.*”, Univ. De Murcia, España, 2014
- [16] (2014) OWL Homepage. Accessed: Noviembre 2014. [Online]. Available: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- [17] M. Valverde, “Publicación de datos enlazados observando los principios de Linked”, Univ. Técnica Particular de Loja, Ecuador, 2014.
- [18] L. Gómez, A. Romero and G. Orellana, “Publicación de datos enlazados observando los principios de Linked Data”, Univ. Técnica Particular de Loja, Ecuador, 2012.
- [19] Guerra H. Alejandro, “Agentes y Sistemas Multi-Agentes Conceptos básicos”, Universidad de Veracruz, México, 2011.
- [20] A. López, R. Sánchez J. y J. Pérez A., “Agentes de información”, Universidad Complutense de Madrid, España, 2005.
- [21] J. Martínez U., “Agentes Inteligentes aplicados a la gestión de información para la investigación”, Accessed Diciembre 2014. [Online]. Available: <http://eprints.ucm.es/5830/1/1999-ConferenciaAI-Celia.pdf>
- [22] P. M. Rodríguez, Tabares M. Valentina, Ovalle C. Demetrio y Duque M. Néstor, “Modelo Multi-agente para recomendación híbrida de objetos de aprendizaje”, Colombia, 2013.

- [23] J. Corchato, "Modelos y Arquitecturas de Agente", Accessed Diciembre 2014. [Online]. Available: [http://bisite.usal.es/archivos/c1%20\(1\).pdf](http://bisite.usal.es/archivos/c1%20(1).pdf)
- [24] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley Longman Publishing, Boston. 1999
- [25] (2013). Programación JADE, Accessed Diciembre 2014. [Online]. Available: <https://programacionjade.wikispaces.com/Ontolog%C3%ADas>
- [26] (2014). CiteSeer Homepage. Accessed: Diciembre 2014. [Online]. Available: <http://csxstatic.ist.psu.edu/about>
- [27] (2014). Letizia Homepage. Accessed: Diciembre 2014. [Online]. Available: <http://web.media.mit.edu/~lieber/Lieberary/Letizia/Letizia-Intro.html>
- [28] J. P. Tandazo, E. L. Naranjo, "Desarrollo de un agente buscador inteligente de metadatos geográficos para la INISIG", Facultad de Sistemas Tesis, Escuela Politécnica Nacional, Guayaquil, Ecuador, 2012.
- [29] (2014) Protegé Homepage. Accessed: Diciembre 2014. [Online]. Available: <http://protege.stanford.edu/>
- [30] R. Alcaraz. "Aplicacions pel web semàntic amb Jena". Noviembre 2012. [Online]. Available: <http://www.cobdc.net/programarilliure/aplicacions-pel-web-semantic-amb-jena/>
- [31] (2014) OpenLink Virtuoso. [Online]. Available: <http://virtuoso.openlinksw.com/>
- [32] (2014) Java Agent Development Framework (JADE). [Online]. Available: <http://jade.tilab.com/>
- [33] (2014) Netbeans. [Online]. Available: <https://netbeans.org/features/index.html>
- [34] (2014) PRIMEFACES. [Online]. Available: <http://blog.primefaces.org/>
- [35] (2015) JavaScript. [Online]. Available: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>

11. Anexos

11.1. Anexo 1: Análisis de requerimientos y diseño preliminar con la Metodología ICONIX

1. Análisis de Requerimientos.

1.1. Fase de Especificación

Para determinar estos factores se ha seguido la guía propuesta por el estándar IEEE-830.

1.1.1. Introducción

En este apartado se va a proporcionar una apreciación global de la especificación de los requerimientos de software, que permitirá conocer los requerimientos, características, funcionamiento y perspectivas de la aplicación.

1.1.1.1. Propósito

Establecer las bases del desarrollo de Software, que describirá cuáles son los requisitos que debe cumplir.

1.1.1.2. Ámbito del Sistema

El sistema se le va a asignar el nombre de "SMAS", tiene como objetivo principal la búsqueda de Objetos de Aprendizaje alojados en repositorios en la Web, utilizando técnicas de Inteligencia Artificial y Web semántica.

1.1.1.3. Personal Involucrado

Para el desarrollo del presente Trabajo de Titulación, intervendrá el siguiente personal:

TABLA XXIII.
DATOS PERSONALES PERSONA 1.

Nombres	Gabriela Paola
Apellidos:	Guamán Morocho
Dirección:	El Valle, Vía Oriental de Paso
Teléfono:	0980351258
E-mail:	gpguamanm@unl.edu.ec
Instrucción:	Superior.
Rol/Cargo:	Analista/ Desarrollador

TABLA XXIV.
DATOS PERSONALES PERSONA 2.

Nombres	John Carlos
Apellidos:	Martínez Pacheco
Dirección:	Esteban Godoy, Héctor Pilco y German Pitiur
Teléfono:	0980351258
E-mail:	jcmartinezp@unl.edu.ec
Instrucción:	Superior.
Rol/Cargo:	Analista/ Desarrollador

1.1.1.4. Definiciones, Acrónimos y Abreviaturas.

Definiciones.

TABLA XXV.
DEFINICIONES.

Administrador	Encargado de administrar el sistema.
Metadatos	Son etiquetas descriptivas que describen un recurso digital.
Investigador	Cualquier persona que accede al sistema para realizar una consulta.
Proveedor	Es cualquier repositorios de Objetos de Aprendizaje disponible en la Web

Acrónimos.

TABLA XXVI.
ACRÓNIMOS.

ERS	Especificación de requerimientos de Software.
OA	Objeto de Aprendizaje
DC	Estándar Dublin Core
LOM	Estándar Learning Object Metada

Referencias

TABLA XXVII.
REFERENCIAS

Referencia	Título	Ruta	Fecha	Autor
1	Especificación de Requisitos según el Estándar de IEEE 830-1998	https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/iee830.pdf	22/10/2008	G. Méndez.

1.1.2. Descripción

1.1.2.1. Perspectiva

En un futuro esta software podría formar parte de un sistema más grande en el que se tendrían en cuenta la Búsqueda de Objetos de Aprendizaje, por ejemplo en un entorno E-Learning.

1.1.2.2. Funciones.

A continuación, se describe con más detalle las tareas, y cómo serán soportadas por el software.

El Administrador:

- Buscar /registrar/editar repositorios de Objetos de Aprendizaje.
- Conectarse con los Servicios de Web de los Repositorios de Objetos de Aprendizaje.
- Poblar la Ontología con Objetos de Aprendizaje (Esta función también lo harán los Agentes de manera automática).
- Inicializar/Actualizar los servicios web de los Repositorios (Esta función también lo harán los Agentes de manera automática).

El Investigador:

- Buscar Objetos de Aprendizaje.

1.1.2.3. Características de los Usuarios

Administrador.

- Universitario con conocimientos Básicos de manejo de sistemas.

Investigador.

- Cualquier persona con conocimientos básicos de manejo de Internet.

1.1.2.4. Restricciones

Para desarrollar la propuesta se usara el lenguaje java específicamente JEE6, que consta JSF + Primefaces + Bootstrap para la interfaz. Los datos serán almacenados en la base de datos relacional MySQL y luego estos datos serán los encargados de Poblar la Ontología y los Agentes Inteligentes serán los encargados de la Administración del Sistema como inicializar los Servicios Web de los Repositorios, así como también mantenerlos actualizados y también los encargados de gestionar las búsquedas en la Ontología.

- Plataforma de desarrollo: Java Enterprise Edition 6
- Lenguajes de programación: Java.
- Para el diseño de la Ontología: Protegé
- Para el Razonamiento de la Ontología: Jena y Sparql.
- Para la programación de Agentes Inteligentes: Jade
- Implementación JSF para vista y diseño Web: Primefaces y Bootstrap v3.1.
- Servidor de aplicaciones: JBoss v2.

1.1.2.5. Restricciones de Diseño.

La interfaz del usuario deberá contener los campos necesarios para que el usuario o administrador del sistema agregue la información que requiera. La interfaz del usuario deberá ser amigable y fácil de manejar, de esta manera el usuario podrá interactuar con el sistema sin que existan problemas. En la pantalla principal existirá un mensaje de bienvenida con el logotipo del software, e incluirá información sobre este proyecto ya que los Objetos de Aprendizaje son recursos digitales pocos conocidos.

1.1.3. Requisitos.

TABLA XXVIII.
REQUERIMIENTOS FUNCIONALES

Referencia	Nombre	Tipo
RF001	El Sistema permitirá al Investigador la Búsqueda de Objetos de Aprendizaje.	Requisito
RF002	El Sistema permitirá al Investigador la Búsqueda Avanzada (Por Título, Autor o Materia) de Objetos de Aprendizaje.	Requisito
RF003	El Sistema presentará los resultados de la búsqueda al Investigador en rejillas de varios elementos agrupados para una mejor visualización.	Requisito
RF004	El Sistema presentará los resultados de la búsqueda al Investigador clasificados por estándar DC o LOM.	Requisito
RF005	El Sistema permitirá al Investigador visualizar todos los Objetos de Aprendizaje en el estándar DC y LOM	Requisito
RF006	El Sistema permitirá visualizar al Investigador todos los proveedores de Objetos de Aprendizaje.	Requisito
RF007	El Sistema permitirá al Administrador poblar la Ontología de Objetos de Aprendizaje.	Requisito
RF008	El Sistema permitirá al Administrador registrar nuevos proveedores de Objetos de Aprendizaje.	Requisito
RF009	El Sistema permitirá al Administrador actualizar nuevos proveedores de Objetos de Aprendizaje.	
RF010	El Sistema permitirá al Administrador inicializar los Servicios Web de los proveedores de Objetos de Aprendizaje.	Requisito
RF011	El Sistema permitirá al Administrador actualizar los Servicios Web de los proveedores de Objetos de Aprendizaje.	Requisito
RF012	El Sistema permitirá al Administrador iniciar sesión mediante el ingreso de usuario y contraseña	Requisito

Requerimientos no funcionales.

El sistema constará con los siguientes requerimientos no funcionales:

TABLA XXIX.

REQUERIMIENTOS NO FUNCIONALES

Referencia	Nombre	Tipo
RNF001	Interfaz amigable para el usuario que garantiza el éxito de usabilidad del prototipo de software a desarrollar	Requisito
RNF002	Disponibilidad al momento de hacer uso de los servicios brindados por el software.	Requisito
RNF003	El sistema emitirá mensajes de error en el caso de ingresar información errónea.	Requisito
RNF004	Brindar información de ayuda a los usuarios para el mejor manejo del software.	Requisito
RNF005	Documentación necesaria, manuales de usuario y del programador comprensible que garantizaran el mantenimiento del software.	Requisito
RNF006	Manejo eficiente de los diferentes roles existentes en el software, mostrando únicamente la información correspondiente a los permisos otorgados.	Requisito

1.2. Modelado del dominio.

A continuación se presenta un diagrama inicial en cual se describe dominio de los OA.

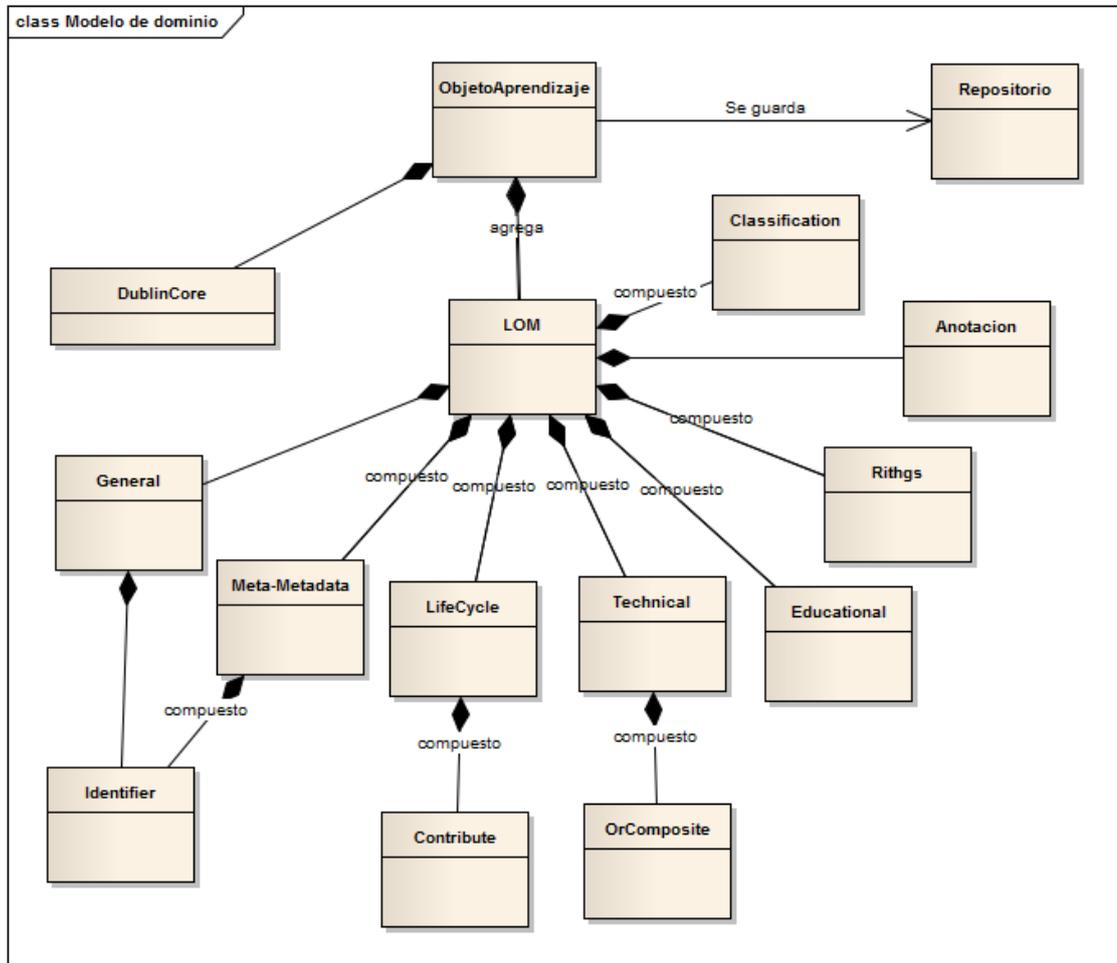


Figura 52. Diagrama de clases inicial.

1.3. Modelado de Casos de Uso.

Para el modelado de los casos de uso, se ha identificado los actores que intervienen en el sistema, a continuación se listan los casos de uso que se obtuvieron mediante la especificación de requerimientos según el estándar IEEE 830.

1.3.1. Identificación de los actores del Sistema

TABLA XXX.

IDENTIFICACIÓN DE LOS ACTORES DEL SISTEMA

Actor	Rol
Administrador	Es la persona encargada de administrar el sistema.
Investigador	Cualquier persona que accede al sistema para realizar una consulta.

1.3.2. Identificación de Casos de Uso

TABLA XXXI.

IDENTIFICACIÓN DE CASOS DE USO

Caso de Uso	Actor	Ref. Requerimiento	Código
Gestionar Objetos Aprendizaje	Administrador	RF012	CU001
Registrar/Actualizar OA en la Ontología	Administrador	RF006, RF007, RF008, RF009	CU002
Gestionar Proveedores	Administrador	RF010, RF011	CU003
Gestionar Búsqueda	Investigador	RF001, RF002, RF003, RF004, RF005, RF006	CU004

1.3.3. Diagrama de Casos de Uso

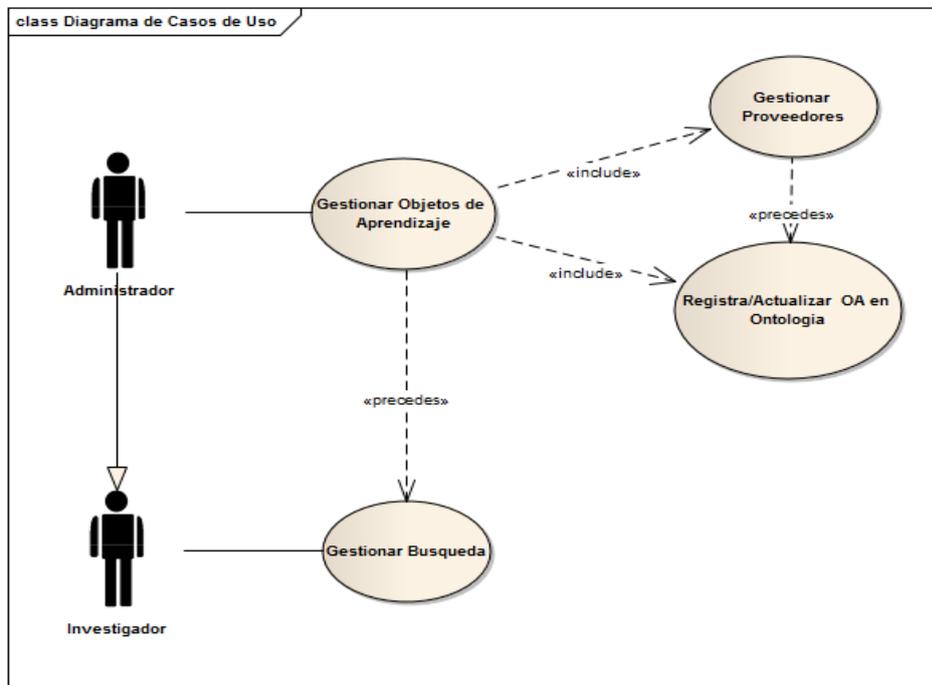


Figura 53. Diagrama de casos de uso.

1.3.4. Prototipo de la Interfaz Gráfica

Para un mayor entendimiento de los casos de uso, se ha considerado hacer un prototipo rápido de las pantallas de los casos de uso con la herramienta de diseño Web Cacao.

- **Gestionar Búsqueda**

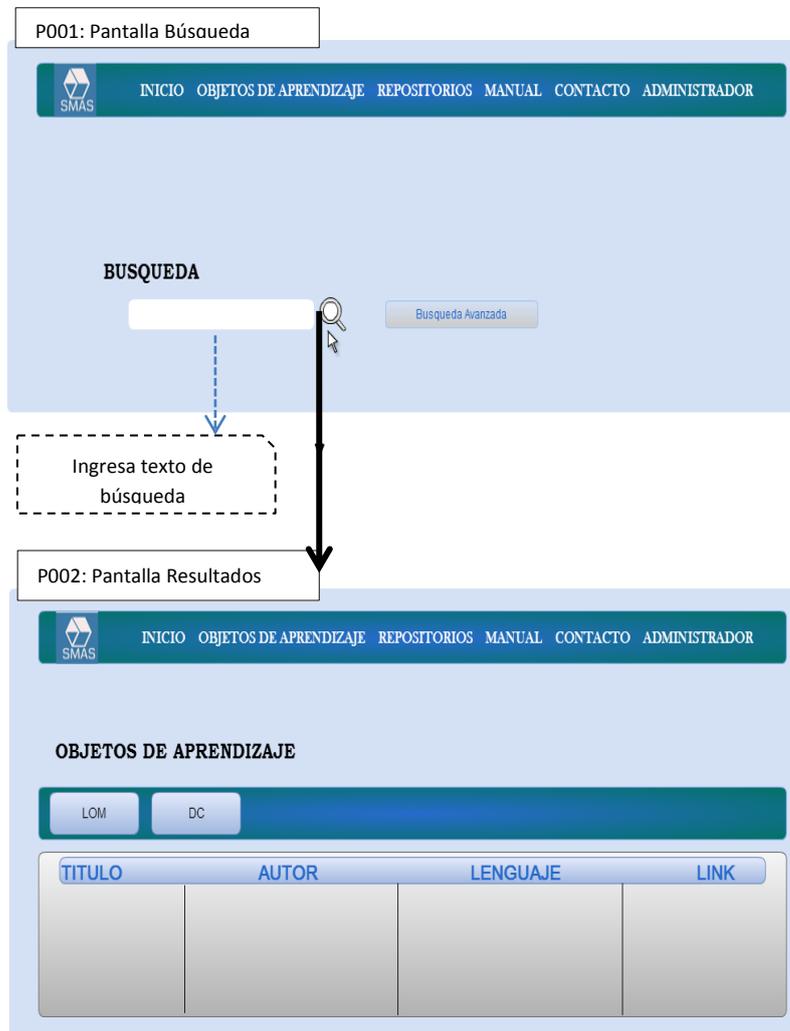


Figura 54. Prototipo 1 del CU004

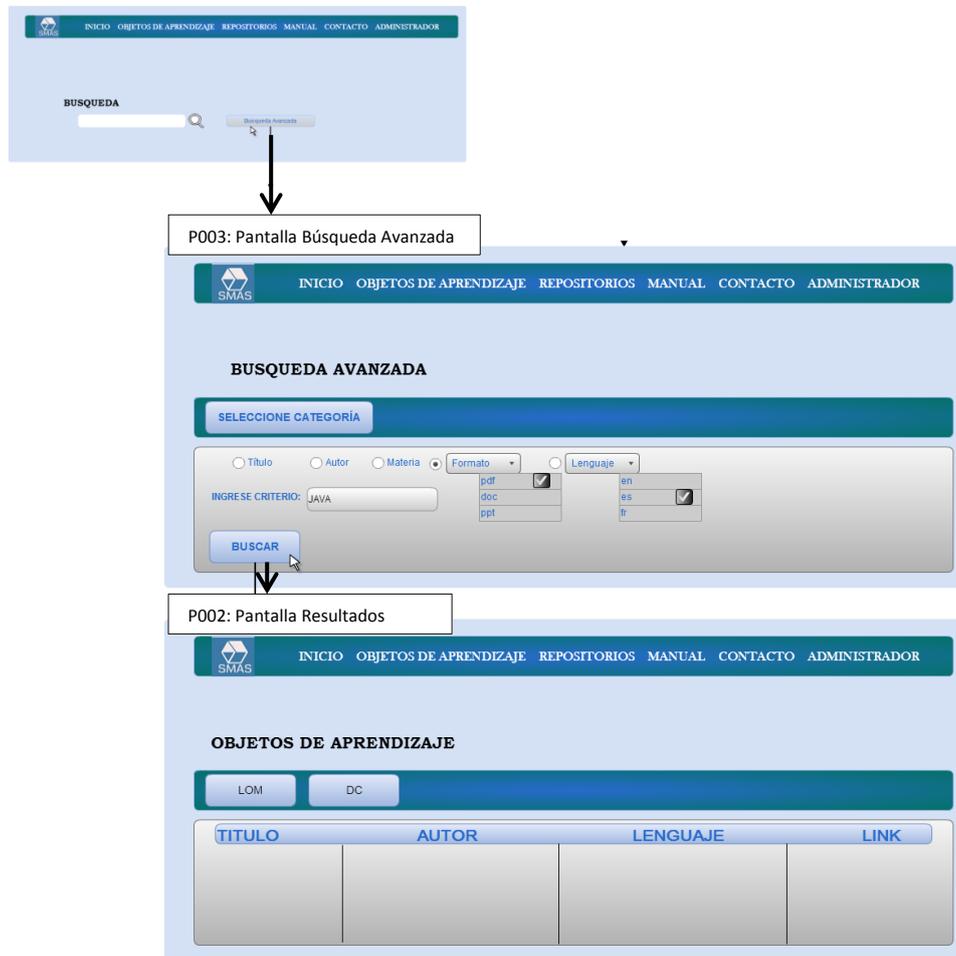


Figura 55. Prototipo 2 del CU004

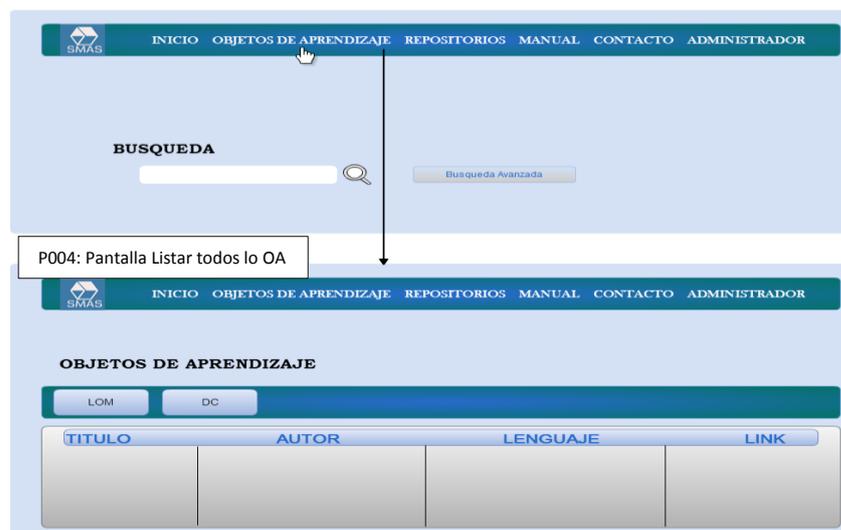


Figura 56. Prototipo 3 del CU004

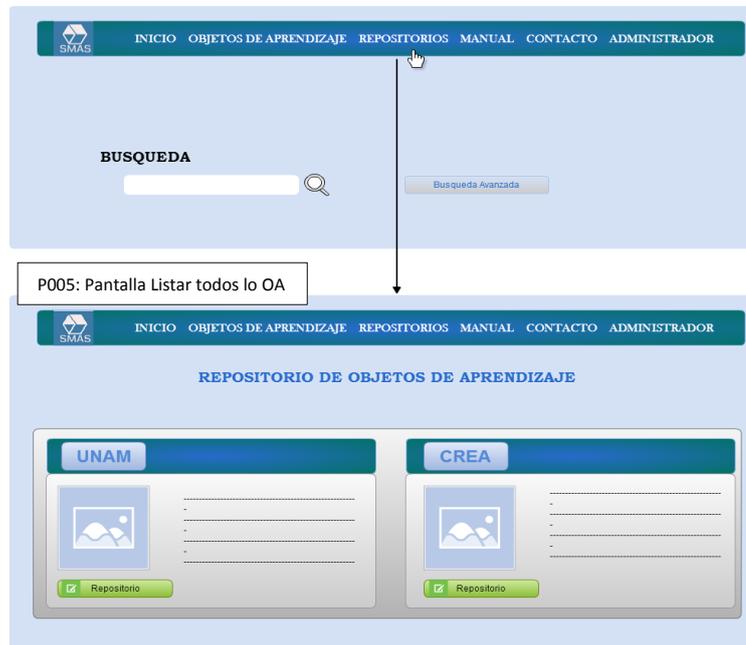


Figura 57. Prototipo 4 del CU004

- **Gestionar Objetos de Aprendizaje**

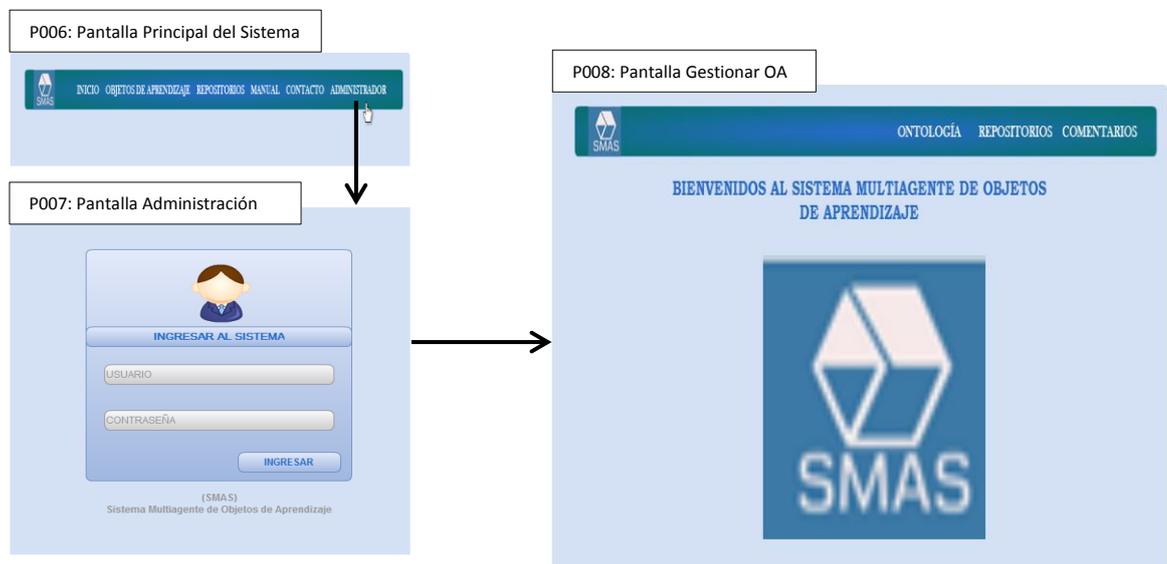


Figura 58. Prototipo 1 del CU001

- **Registrar/Actualizar OA en la Ontología.**



Figura 59. Prototipo 2 del CU002

- **Gestionar Proveedores**

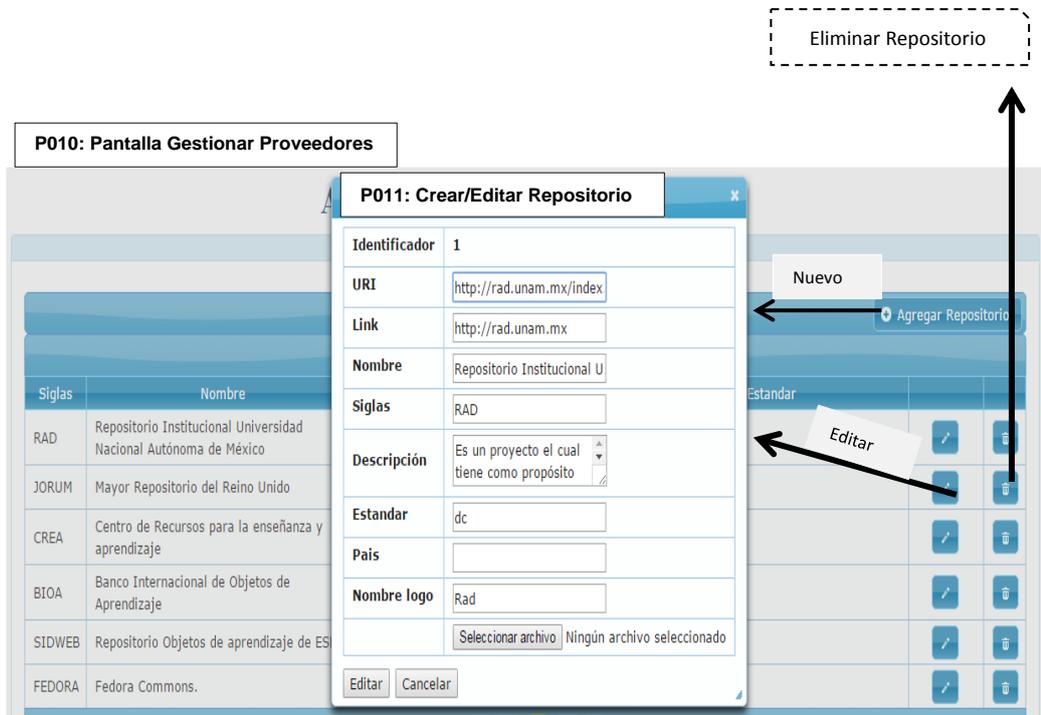


Figura 60. Prototipo del CU003

2. Análisis y Diseño Preliminar.

A continuación se detalla el reconocimiento y la identificación de objetos que participan en cada caso de uso.

TABLA XXXII.

DESCRIPCION DEL CASOS DE USO CU001

Caso de Uso: Gestionar Objetos de Aprendizaje							
Código:	CU001	Tipo:	Primario-Sistema	Actor:	Administrador	Pantalla:	P006, P007, P008
Resumen:	El Administrador ingresa al sistema y podrá gestionar proveedores y registrar/actualizar Objetos de Aprendizaje en la Ontología.						
PreCondiciones:	<ul style="list-style-type: none"> Haber ingresado al sistema. 						
PostCondiciones	<ul style="list-style-type: none"> Haber gestionado proveedor o registrado/Actualizado Ontología de Objetos de Aprendizaje o ninguna 						
Curso Normal de Eventos							
<ol style="list-style-type: none"> El Administrador presiona el botón [Administrador] de la pantalla Principal del Sistema. El Sistema invoca y presenta la pantalla de Administración. El Administrador visualiza la pantalla de Administración y registra los datos (usuario y contraseña). El Sistema verifica si los datos son correctos y presenta la pantalla Gestionar OA. El Administrador visualiza las opciones Ontología, Repositorios y comentarios y puede realizar cualquiera de ellas. El Sistema guarda cualquier cambio realizado por el Administrador. El Administrador cierra sesión en el Sistema El caso de uso finaliza. 							

Curso Alterno de Eventos
<p>A. Campos obligatorios sin llenar o erróneos.</p> <p>A.1. El sistema verifica los campos obligatorios y presenta un mensaje “Datos Incorrectos” en la pantalla.</p>
<p>B. Gestionar Proveedores</p> <p>B.1. Invoca al CU002 (Ver DESCRIPCION DEL CASOS DE USO CU002)</p>
<p>C. Registrar/Actualizar OA en Ontología</p> <p>C.1. Invoca al CU003 (Ver DESCRIPCION DEL CASOS DE USO CU002)</p>

TABLA XXXIII.

DESCRIPCION DEL CASOS DE USO CU002

Caso de Uso: Gestionar Proveedores							
Código:	CU002	Tipo:	Primario/ Esencial	Actor:	Administrador	Pantalla:	P010, P011
Resumen:	El Administrador busca y registra los proveedores de Objetos de Aprendizaje.						
Objetivos:	<ul style="list-style-type: none"> Registrar proveedores de Objetos de Aprendizaje 						
PreCondiciones:	<ul style="list-style-type: none"> Haberse iniciado sesión en el sistema Haber contactado con el Repositorio y tener los permisos necesarios para acceder a sus Servicios Web. 						
PosCondiciones:	<ul style="list-style-type: none"> El proveedor de Objetos de Aprendizaje queda registrados/Actualizado/Eliminado 						

Curso Normal de Eventos

1. El Administrador ingresa a la pantalla de **Repositorios** y presiona el botón [Nuevo].
2. El sistema presenta la pantalla con los campos a llenar.
3. El Administrador ingresa los datos del repositorio (nombre, estándar de metadatos, endpoint, url, siglas del repositorio).
4. El sistema guarda y actualiza la tabla Repositorios y presenta un mensaje "Repositorio guardado".
5. El caso de uso finaliza.

Curso Alternativo de Eventos

A. Campos obligatorios sin llenar o erróneos.

- A.1. El sistema verifica los campos obligatorios y presenta un mensaje "Datos Incorrectos" en la pantalla.

B. Editar Proveedor

- B. 1. El Administrador selecciona el repositorio a editar.
- B.2. Se invoca al Curso normal de Eventos desde el paso (3)

C. Eliminar repositorio

- C.1. El Administrador selecciona el repositorio y presionar el botón [Eliminar].
- C.3. El sistema presenta un mensaje "Repositorio Eliminado"

TABLA XXXIV.

DESCRIPCION DEL CASOS DE USO CU003

Caso de Uso: REGISTRAR/ACTUALIZAR OA EN ONTOLOGIA							
Código:	CU003	Tipo:	Primario/ Esencial	Actor:	Administrador/ Sistema(Agent es de Software)	Pantalla:	P007, P008
Resumen:	El Administrador se encarga de poblar la Odontología desde sus proveedores y mantener actualizada la Ontología.						
Objetivos:	<ul style="list-style-type: none"> • Poblar Objetos de Aprendizaje 						
PreCondiciones:	<ul style="list-style-type: none"> • Tener registrados proveedores de Objetos de Aprendizaje 						
PosCondiciones:	<ul style="list-style-type: none"> • La Ontología poblado/actualizada de Objetos de Aprendizaje. 						
Curso Normal de Eventos							
<p>1. El Administrador ingresa a la pantalla de Ontología y presiona el botón [Poblar Ontología].</p> <p>3. El sistema llena la Ontología de individuos y presenta el mensaje “Ontología poblada con éxito”.</p> <p>5. El caso de uso finaliza.</p>							
Curso Alterno de Eventos							
A. No se puedo poblar la Ontología.							
A.1. El sistema presenta un mensaje “Error al Poblar la Ontología” en la pantalla.							
B. Actualizar Ontología							
B. 1. El Sistema automáticamente busca los proveedores de OA y llena la Ontología de nuevos individuos.							

TABLA XXXV.

DESCRIPCION DEL CASOS DE USO CU004

Caso de Uso: GESTIONAR BUSQUEDA							
Código:	CU004	Tipo:	Primario/ Esencial	Actor:	Administrador/ Sistema(Agentes de Software)	Pantalla:	P001, P002, P003, P004, P005
Resumen:	El Investigador busca Objetos de Aprendizaje y el sistema recupera los resultados.						
Objetivos:	<ul style="list-style-type: none"> • Buscar Objetos de Aprendizaje 						
Pre-Condiciones:	<ul style="list-style-type: none"> • Haber ingresado a la pantalla de Búsqueda 						
Curso Normal de Eventos							
<p>1. El Investigador ingresa un texto en la pantalla Búsqueda y presionar el botón [Buscar].</p> <p>3. El sistema presenta los resultados encontrados en la pantalla Resultados.</p> <p>4. El caso de uso finaliza.</p>							
Curso Alterno de Eventos							
A. Texto de búsqueda muy extenso.							
A.1. El sistema presenta un mensaje "Limite superado de caracteres" en la pantalla.							
B. Texto de Búsqueda vacío							
B.1. El sistema presenta un mensaje "Campo obligatorio vacío" en la pantalla.							

Curso Alterno de Eventos

C. Búsqueda Avanzada

C.1. 1. El Investigador presiona el botón [Búsqueda Avanzada].

C.1.2. El Sistema presenta la pantalla de **Búsqueda Avanzada**.

C.1.2. El investigador escoge el criterio (título, autor, materia) por el cual desea buscar e ingresa un texto de búsqueda.

C.1. 1. El sistema presenta los resultados encontrados.

C.1. 1. El caso de uso finaliza.

D. Visualizar todos los Objetos de Aprendizaje

D.1. 1. El Investigador presiona el botón [Objetos de Aprendizaje] en la pantalla **Principal del Sistema**.

D.1.2. El sistema presenta en la pantalla **Listar todos OA** con todos los disponibles

D.1. 1. El caso de uso finaliza.

E. Visualizar proveedores de Objetos de Aprendizaje

E.1. 1. El Investigador ingresa a la pantalla de Repositorios.

E.1.2. El sistema presenta todos los repositorios disponibles

E.1. 1. El caso de uso finaliza.

3. Diseño Detallado

En esta fase se realizó la actualización del modelo del dominio.

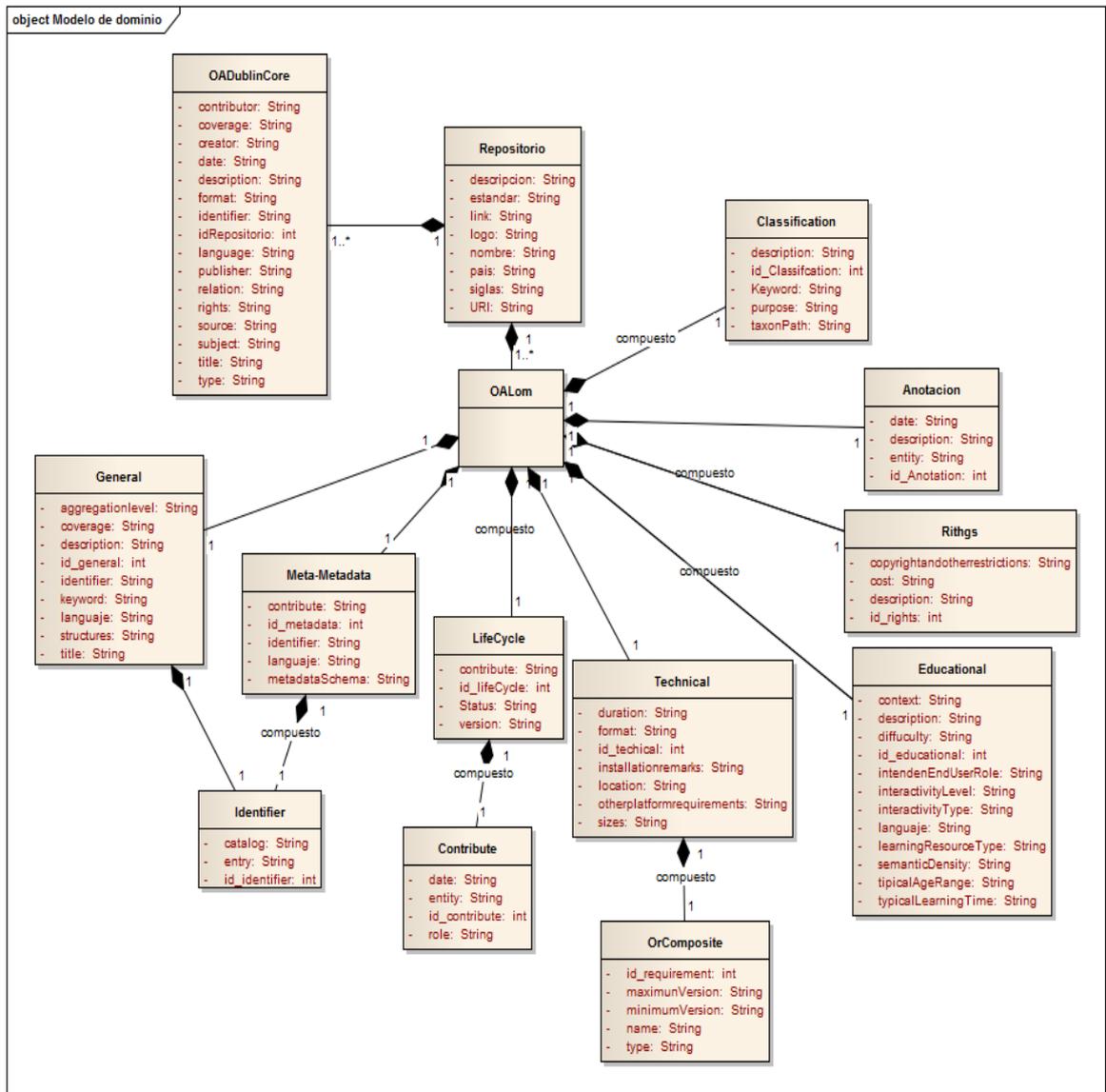


Figura 61. Modelo del Dominio Final

11.2. Anexo 2: Formato RDF/OWL generado desde Protegé

```
<rdf:RDF xmlns="&semantic;ontologies/2014/11/ontologiatesis#"
  xml:base="&semantic;ontologies/2014/11/ontologiatesis"
  xmlns:radion="http://www.w3.org/ns/radion/"
  xmlns:com="http://vocab.resc.info/communication#"
  xmlns:nco="http://www.semanticdesktop.org/ontologies/nco/"
  xmlns:ebucore="http://www.ebu.ch/metadata/ebucore/index.html#"
  xmlns:scoro="http://purl.org/spar/scoro/EducationalRole/"
  xmlns:cmo="http://purl.org/twc/ontologies/cmo.owl#"
  xmlns:adms="http://www.w3.org/ns/adms#"
  xmlns:schema="http://schema.org/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcat="http://www.w3.org/ns/dcat#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:disco="http://rdf-vocabulary.ddialliance.org/discovery#"
  xmlns:dicom="http://purl.org/healthcarevocab/v1/"
  xmlns:swrc="http://lov.okfn.org/dataset/lov/search/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:gold="http://purl.org/linguistics/gold/"
  xmlns:opus="http://lsdis.cs.uga.edu/projects/semdis/opus#"
  xmlns:prov="http://www.w3.org/ns/prov#"
  xmlns:mo="http://purl.org/ontology/mo/"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:lcy="http://purl.org/vocab/lifecycle/schema#"
  xmlns:dce="http://purl.org/dc/elements/"
  xmlns:foaf="http://xmlns.com/foaf/spec/"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:semantic="http://www.semanticweb.org/"
  xmlns:fabio="http://purl.org/spar/fabio/"
  xmlns:qb="http://purl.org/linked-data/cube#"
  xmlns:vcard="http://www.w3.org/2006/vcard/ns#"
  xmlns:umbel="http://umbel.org/umbel/"
  xmlns:fea="http://vocab.data.gov/def/fea/"
  xmlns:am="http://open-services.net/ns/asset#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:doc="http://www.w3.org/2000/10/swap/pim/doc#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:stac="http://securitytoolbox.appspot.com/stac#"
  xmlns:smas="http://smas.com/"
  xmlns:xhv="http://www.w3.org/1999/xhtml/vocab/"
  xmlns:lemon="http://lemon-model.net/lemon#">
<owl:Ontology rdf:about="&semantic;ontologiatesis"/>

<!--
////////////////////////////////////
//
// Object Properties
//
////////////////////////////////////
-->
<owl:ObjectProperty rdf:about="&smas;classificationIsComprisedOf">
  <rdfs:range rdf:resource="&smas;TaxonPath"/>
  <rdfs:domain rdf:resource="&com;Classification"/>
```

```

</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;contains">
  <rdfs:range rdf:resource="&smas;LearningObject"/>
  <rdfs:domain rdf:resource="&smas;OpenEducationalResource"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;contributeIsPartOf">
  <rdfs:range rdf:resource="&mo;Metametadata"/>
  <rdfs:range rdf:resource="&lcy;Lifecycle"/>
  <owl:inverseOf rdf:resource="&smas;lifeCycleIsComprisedOf"/>
  <rdfs:domain rdf:resource="&prov;Contribute"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;generalIsComprisedOf">
  <rdfs:domain rdf:resource="&gold;General"/>
  <rdfs:domain rdf:resource="&mo;Metametadata"/>
  <rdfs:domain rdf:resource="&fea;Resource"/>
  <rdfs:range rdf:resource="&adms;Identifier"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;hasdataClassification">
  <rdfs:domain rdf:resource="&smas;Lom"/>
  <rdfs:range rdf:resource="&com;Classification"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;hasdataEducational">
  <rdfs:range rdf:resource="&scoro;Educational"/>
  <rdfs:domain rdf:resource="&smas;Lom"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;hasdataGeneral">
  <rdfs:range rdf:resource="&gold;General"/>
  <rdfs:domain rdf:resource="&smas;Lom"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;hasdataLifeCycle">
  <rdfs:range rdf:resource="&lcy;Lifecycle"/>
  <rdfs:domain rdf:resource="&smas;Lom"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;hasdataTechnical">
  <rdfs:range rdf:resource="&fabio;Technical"/>
  <rdfs:domain rdf:resource="&smas;Lom"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;isBasedOn">
  <rdfs:domain rdf:resource="&fabio;Metadata"/>
  <rdfs:range rdf:resource="&smas;Lom"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;isComprisedOf">
  <rdfs:range rdf:resource="&fabio;Metadata"/>
  <rdfs:domain rdf:resource="&smas;LearningObject"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;isPartOf">
  <rdfs:domain rdf:resource="&fabio;Metadata"/>

```

```

        <rdfs:range rdf:resource="&smas;LearningObject"/>
        <owl:inverseOf rdf:resource="&smas;isComposedOf"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;isStoredIn">
    <rdfs:domain rdf:resource="&swrc;DublinCore"/>
    <rdfs:domain rdf:resource="&smas;Lom"/>
    <rdfs:range rdf:resource="&radion;Repository"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;lifeCycleIsComprisedOf">
    <rdfs:domain rdf:resource="&mo;Metametadata"/>
    <rdfs:domain rdf:resource="&lcy;Lifecycle"/>
    <rdfs:range rdf:resource="&prov;Contribute"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;relationIsComprisedOf">
    <rdfs:domain rdf:resource="&cmo;Relation"/>
    <rdfs:range rdf:resource="&fea;Resource"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&smas;technicalIsComprisedOf">
    <rdfs:domain rdf:resource="&fabio;Technical"/>
    <rdfs:range rdf:resource="&stac;Requirement"/>
</owl:ObjectProperty>

<owl:ObjectProperty
rdf:about="&semantic;ontologiatesis#describes">
    <rdfs:domain rdf:resource="&swrc;DublinCore"/>
    <rdfs:range rdf:resource="&xhv;ContentsInformation"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&semantic;ontologiatesis#has">
    <rdfs:domain rdf:resource="&swrc;DublinCore"/>
    <rdfs:range rdf:resource="&cmo;IntellectualPropertyRights"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&semantic;ontologiatesis#specific">
    <rdfs:domain rdf:resource="&swrc;DublinCore"/>
    <rdfs:range rdf:resource="&fabio;TechnicalAspects"/>
</owl:ObjectProperty>

<!--
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
-->
<owl:DatatypeProperty rdf:about="&lemon;context">
    <rdfs:domain rdf:resource="&scoro;Educational"/>
    <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&lemon;entry">
    <rdfs:domain rdf:resource="&gold;Taxon"/>
    <rdfs:range rdf:resource="&xsd;string"/>
    <rdfs:domain rdf:resource="&adms;Identifier"/>

```

```

</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&am;size">
  <rdfs:domain rdf:resource="&fabio;Technical"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&dce;coverage">
  <rdfs:domain rdf:resource="&gold;General"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&dce;date">
  <rdfs:range rdf:resource="&xsd;string"/>
  <rdfs:domain rdf:resource="&owl;Annotation"/>
  <rdfs:domain rdf:resource="&prov;Contribute"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&dce;description">
  <rdfs:domain rdf:resource="&gold;General"/>
  <rdfs:domain rdf:resource="&scoro;Educational"/>
  <rdfs:domain rdf:resource="&fea;Resource"/>
  <rdfs:domain rdf:resource="&com;Classification"/>
  <rdfs:domain rdf:resource="&ebucore;Rights"/>
  <rdfs:range rdf:resource="&xsd;string"/>
  <rdfs:domain rdf:resource="&owl;Annotation"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&dce;format">
  <rdfs:domain rdf:resource="&fabio;Technical"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&dce;language">
  <rdfs:domain rdf:resource="&gold;General"/>
  <rdfs:domain rdf:resource="&scoro;Educational"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&dce;title">
  <rdfs:domain rdf:resource="&gold;General"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&schema;location">
  <rdfs:domain rdf:resource="&fabio;Technical"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&schema;purpose">
  <rdfs:domain rdf:resource="&com;Classification"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&smas;names">
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

```

```

<owl:DatatypeProperty rdf:about="&foaf;name">
  <rdfs:domain rdf:resource="&stac;Requirement"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->

<owl:Class rdf:about="&swrc;DublinCore">
  <rdfs:subClassOf rdf:resource="&fabio;Metadata"/>
</owl:Class>

<owl:Class rdf:about="&gold;General">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

<owl:Class rdf:about="&gold;Taxon">
  <rdfs:subClassOf rdf:resource="&smas;TaxonPath"/>
</owl:Class>

<owl:Class rdf:about="&mo;Metametadata">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

<owl:Class rdf:about="&fabio;Metadata">
  <rdfs:subClassOf rdf:resource="&smas;LearningObject"/>
</owl:Class>

<owl:Class rdf:about="&fabio;Technical">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

<owl:Class rdf:about="&fabio;TechnicalAspects">
  <rdfs:subClassOf rdf:resource="&swrc;DublinCore"/>
</owl:Class>

<owl:Class rdf:about="&scoro;Educational">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

<owl:Class rdf:about="&cmo;IntellectualPropertyRights">
  <rdfs:subClassOf rdf:resource="&swrc;DublinCore"/>
</owl:Class>

<owl:Class rdf:about="&cmo;Relation">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

<owl:Class rdf:about="&lcy;Lifecycle">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

```

```

<owl:Class rdf:about="&stac;Requirement">
  <rdfs:subClassOf rdf:resource="&fabio;Technical"/>
</owl:Class>

<owl:Class rdf:about="&smas;LearningObject">
  <rdfs:subClassOf
rdf:resource="&smas;OpenEducationalResource"/>
</owl:Class>

<owl:Class rdf:about="&smas;OpenEducationalResource"/>

<owl:Class rdf:about="&smas;TaxonPath">
  <rdfs:subClassOf rdf:resource="&com;Classification"/>
</owl:Class>

<owl:Class rdf:about="&com;Classification">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

<owl:Class rdf:about="&ebucore;Rights">
  <rdfs:subClassOf rdf:resource="&smas;Lom"/>
</owl:Class>

<owl:Class rdf:about="&xhiv;ContentsInformation">
  <rdfs:subClassOf rdf:resource="&swrc;DublinCore"/>
</owl:Class>

<owl:Class rdf:about="&adms;Identifier">
  <rdfs:subClassOf rdf:resource="&gold;General"/>
  <rdfs:subClassOf rdf:resource="&mo;Metametadata"/>
  <rdfs:subClassOf rdf:resource="&fea;Resource"/>
</owl:Class>

<owl:Class rdf:about="&prov;Contribute">
  <rdfs:subClassOf rdf:resource="&mo;Metametadata"/>
  <rdfs:subClassOf rdf:resource="&lcy;Lifecycle"/>
</owl:Class>

<owl:Class rdf:about="&radion;Repository">
  <rdfs:subClassOf rdf:resource="&smas;LearningObject"/>
</owl:Class>
</rdf:RDF>

```

11.3. Anexo 3: Consultas SPARQL

Para evaluar el funcionamiento de la Ontología se realizó algunas consultas con la ayuda de virtuoso, las cuales fueron exitosas y respondieron de acuerdo a lo planteado en los requerimientos de la Ontología.

Objetos de Aprendizaje Dublin Core

¿Qué Objetos de Aprendizaje tiene por **título 'Java'**?

Consulta SPARQL

```
PREFIX smas: <http://smas.com/>
PREFIX dce: <http://purl.org/dc/elements/>
SELECT DISTINCT *
WHERE {
    ?DublinCore smas:describes ?ContentsInformation.
    ?ContentsInformation dce:title ?title .
FILTER(regex(?title, 'java', 'i'))
}
```

Resultado

DublinCore	ContentsInformation	title
http://lov.okfn.org/dataset/lov/search/http://dspace.jorum.ac.uk/xmlui/handle/10949/1319	http://www.w3.org/1999/xhtml/vocab/http://dspace.jorum.ac.uk/xmlui/handle/10949/1319	"Distributed Web Systems - Java Server Pages"
http://lov.okfn.org/dataset/lov/search/http://dspace.jorum.ac.uk/xmlui/handle/10949/1320	http://www.w3.org/1999/xhtml/vocab/http://dspace.jorum.ac.uk/xmlui/handle/10949/1320	"Distributed Web Systems - Java Server Pages"
http://lov.okfn.org/dataset/lov/search/http://dspace.jorum.ac.uk/xmlui/handle/10949/1321	http://www.w3.org/1999/xhtml/vocab/http://dspace.jorum.ac.uk/xmlui/handle/10949/1321	"Distributed Web Systems - Java Server Pages"
http://lov.okfn.org/dataset/lov/search/http://dspace.jorum.ac.uk/xmlui/handle/10949/1340	http://www.w3.org/1999/xhtml/vocab/http://dspace.jorum.ac.uk/xmlui/handle/10949/1340	"Distributed Web Systems - Using Java beans with JSP"

Figura 62. Resultado 1 Consulta SPARQL.

¿Qué Objeto de Aprendizaje pertenecen a la **materia** de física?

Consulta SPARQL

```
PREFIX smas: <http://smas.com/>
PREFIX dce: <http://purl.org/dc/elements/>
SELECT DISTINCT *
WHERE {
    ?DublinCore smas:describes ?ContentsInformation.
    ?ContentsInformation dce:subject ?subject.
FILTER(regex(?subject, 'fisica', 'i'))
}
```

Resultado

ContentsInformation	title	subject
http://www.w3.org/1999/xhtml/vocab/http://objetoseducacionais2.mec.gov.br/handle/mec/9275	"How to verify Biot-Savart Law?"	"Modalidade: Educação de Jovens e Adultos - 2º ciclo: Ciências Naturais: Visões de mundo, Física, Eletromagnetismo, Eletricidade, Educação Superior: Ciências Exatas e da Terra: Física: Eletricidade e Magnetismo: Campos e Partículas Carregadas, Educação Básica: Ensino Médio: Física: Oscilações, ondas, óptica e radiação"
http://www.w3.org/1999/xhtml/vocab/http://objetoseducacionais2.mec.gov.br/handle/mec/9277	"physics of Rainbow"	"Óptica, Física, Modalidade: Educação de Jovens e Adultos - 2º ciclo: Ciências Naturais: Visões de mundo, Luz, Educação Básica: Ensino Médio: Física: Oscilações, ondas, óptica e radiação"
http://www.w3.org/1999/xhtml/vocab/http://objetoseducacionais2.mec.gov.br/handle/mec/9279	"Multimeter"	"Modalidade: Educação de Jovens e Adultos - 2º ciclo: Ciências Naturais: Visões de mundo, Física, Eletricidade, Eletromagnetismo, Educação Superior: Ciências Exatas e da Terra: Física: Instrumentação Específica de Uso Geral em Física, Educação Básica: Ensino Médio: Física: Equipamentos elétricos e telecomunicações"
http://www.w3.org/1999/xhtml/vocab/http://objetoseducacionais2.mec.gov.br/handle/mec/7996	"Space and Time in Special Relativity"	"Física, Luz, Educação Básica: Ensino Médio: Física: Universo, terra e vida"

Figura 63. Resultado 2 Consulta SPARQL.

¿Qué Objetos de Aprendizaje su **creador** es Francisco?

Consulta SPARQL

```
PREFIX smas: <http://smas.com/>
PREFIX dce: <http://purl.org/dc/elements/>
SELECT DISTINCT *
WHERE {
    ?DublinCore smas:has ?IntellectualPropertyRights.
    ?IntellectualPropertyRights dce:creador ?creador.
FILTER(regex(?creador, 'Francisco', 'i'))
}
```

Resultado

DublinCore	IntellectualPropertyRights	creador
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27898?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27898?proyecto=Irekani	"Jesus Iglesias, Francisco Botello"
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27899?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27899?proyecto=Irekani	"Francisco Botello"
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27903?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27903?proyecto=Irekani	"Jesus Iglesias, Francisco Botello"
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27926?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27926?proyecto=Irekani	"Ricardo Bolanos, Francisco Botello"
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27929?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27929?proyecto=Irekani	"Jesus Iglesias, Francisco Botello"
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27936?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27936?proyecto=Irekani	"Francisco Botello"
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27938?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27938?proyecto=Irekani	"Ricardo Bolanos, Francisco Botello"
http://lov.okfn.org/dataset/lov/search/http://unibio.unam.mx/irekani/handle/123456789/27941?proyecto=Irekani	http://purl.org/twc/ontologies/cmo.owl#http://unibio.unam.mx/irekani/handle/123456789/27941?proyecto=Irekani	"Francisco Botello"

Figura 64. Resultado 3 consulta SPARQL.

¿Qué Objetos de Aprendizaje tienen por **descripción** Inteligencias Múltiples?

Consulta SPARQL

```

PREFIX smas: <http://smas.com/>
PREFIX dce: <http://purl.org/dc/elements/>
SELECT DISTINCT *
WHERE {
    ?DublinCore smas:describes ?ContentsInformation.
    ?ContentsInformation dce:description ?description.
FILTER(regex(?description, ' Inteligencias Múltiples ', 'i'))
}

```

Resultado

DublinCore	ContentsInformation	description
http://lov.okfn.org/dataset/lov/search/http://www.crea.udg.mx/handle/123456789/247	http://www.w3.org/1999/xhtml/vocab/http://www.crea.udg.mx/handle/123456789/247	"El planteamiento Central de la Teoría de Gardner con respecto a las siete Inteligencias Múltiples (lingüística, musical, lógico-matemática, espacial, cinestésico-corporal, intrapersonal e interpersonal) se refiere a la manera en que los individuos aprenden con capacidades y aptitudes distintas respondiendo con mayor eficacia en un entorno que se adapta al tipo de inteligencia que posee"
http://lov.okfn.org/dataset/lov/search/http://www.crea.udg.mx/handle/123456789/248	http://www.w3.org/1999/xhtml/vocab/http://www.crea.udg.mx/handle/123456789/248	"La Teoría de las Inteligencias Múltiples introduce la idea de categorías mentales amplias y diferenciadas, al tiempo de que desestima la vieja idea de inteligencia única y monolítica"

Figura 65. Resultado 4 consulta SPARQL.

¿Qué Objetos de Aprendizaje son de **tipo imagen**?

Consulta SPARQL

```

PREFIX smas: <http://smas.com/>
PREFIX dce: <http://purl.org/dc/elements/>
SELECT DISTINCT *
WHERE {
    ?DublinCore smas:specific ?TechnicalAspects.
    ?TechnicalAspects dce:type ?type.
FILTER(regex(?type, ' imagen', 'i'))
}
    
```

Resultado

ContentsInformation	title	TechnicalAspects	type
http://www.w3.org/1999/xhtml/vocab/http://unibio.unam.mx/irekani/handle/123456789/27884?proyecto=Irekani	"Urocyon cinereoargenteus"	http://purl.org/spar/fabio/http://objetoseducacionais2.mec.gov.br/handle/mec/10097	"Imagem"
http://www.w3.org/1999/xhtml/vocab/http://unibio.unam.mx/irekani/handle/123456789/27884?proyecto=Irekani	"Urocyon cinereoargenteus"	http://purl.org/spar/fabio/http://objetoseducacionais2.mec.gov.br/handle/mec/10098	"Imagem"
http://www.w3.org/1999/xhtml/vocab/http://unibio.unam.mx/irekani/handle/123456789/27884?proyecto=Irekani	"Urocyon cinereoargenteus"	http://purl.org/spar/fabio/http://objetoseducacionais2.mec.gov.br/handle/mec/10099	"Imagem"
http://www.w3.org/1999/xhtml/vocab/http://unibio.unam.mx/irekani/handle/123456789/27884?proyecto=Irekani	"Urocyon cinereoargenteus"	http://purl.org/spar/fabio/http://objetoseducacionais2.mec.gov.br/handle/mec/10100	"Imagem"
http://www.w3.org/1999/xhtml/vocab/http://unibio.unam.mx/irekani/handle/123456789/27884?proyecto=Irekani	"Urocyon cinereoargenteus"	http://purl.org/spar/fabio/http://objetoseducacionais2.mec.gov.br/handle/mec/10101	"Imagem"
http://www.w3.org/1999/xhtml/vocab/http://unibio.unam.mx/irekani/handle/123456789/27884?proyecto=Irekani	"Urocyon cinereoargenteus"	http://purl.org/spar/fabio/http://objetoseducacionais2.mec.gov.br/handle/mec/10102	"Imagem"
http://www.w3.org/1999/xhtml/vocab/http://unibio.unam.mx/irekani/handle/123456789/27884?proyecto=Irekani	"Urocyon cinereoargenteus"	http://purl.org/spar/fabio/http://objetoseducacionais2.mec.gov.br/handle/mec/10103	"Imagem"

Figura 66. Resultado 4 consulta SPARQL.

11.4. Anexo 4: Encuesta

UNIVERSIDAD NACIONAL DE LOJA
Área de la Energía, las Industrias y los Recursos Naturales No Renovables
CARRERA DE INGENIERÍA EN SISTEMAS
Encuesta

La finalidad de la presente encuesta tiene por objetivo evaluar el Trabajo de Titulación denominado "Sistema Multiagente basado en un modelo Ontológico para la búsqueda de Objetos de Aprendizaje", luego de haber usado nuestro Software, su opinión es muy importante para nosotros, por lo tanto se pide sinceridad y responsabilidad al momento de responder la encuesta.

Objeto de Aprendizaje: Son recursos digitales creados con el objetivo de generar conocimiento, habilidades, aptitudes y competencias en el ámbito educativo

3.2. ¿Tiene usted conocimiento de la existencia de algún otro software de búsqueda de Objetos de Aprendizaje?

Si ()

No ()

3.3. ¿Cómo considera la interfaz del software SMAS?

Agradable ()

Poco agradable ()

Nada agradable ()

3.4. El nivel de usabilidad del SMAS le pareció:

Fácil ()

Mediana ()

Difícil ()

4. De acuerdo a la búsqueda realizada esta fue:

Excelente ()

Buena ()

No encontró resultados ()

5. El tiempo de respuesta de su búsqueda fue:

Excelente ()

Bueno ()

Malo ()

6. ¿Considera usted que este Software es un apoyo para el proceso de Enseñanza/Aprendizaje en al Ámbito Educativo?

Si ()

No ()

GRACIAS POR SU COLABORACIÓN

11.5. Anexo 5: Resultados de la encuesta realizada a los estudiantes de la carrera de Ingeniería en Sistemas

La encuesta fue dirigida a estudiantes de la Carrera de Ingeniería en Sistemas de los cuales nos respondieron 18 personas.

Pregunta 1

¿Tiene usted conocimiento de la existencia de algún otro software de búsqueda de Objetos de Aprendizaje?

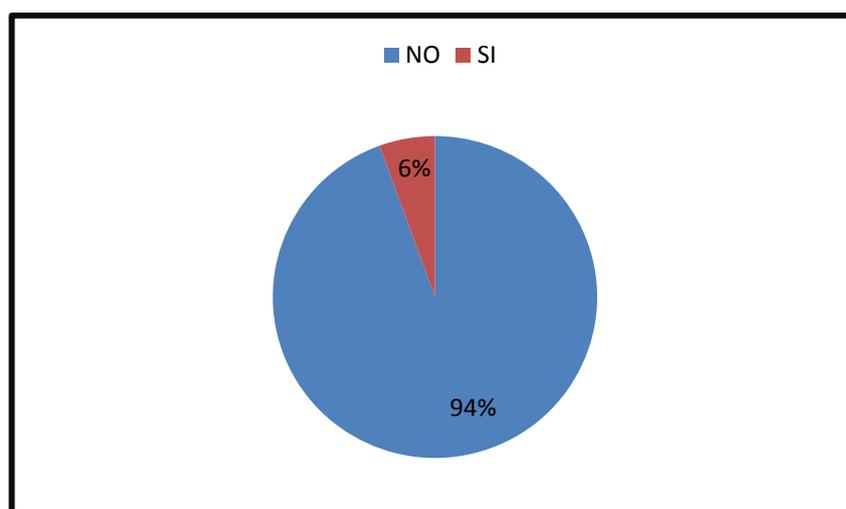


Figura 67. Análisis del resultado pregunta 1.

Interpretación y Análisis.

El 94.4 % de los estudiantes encuestados afirman no conocer ningún software que se encargue de la búsqueda de Objetos de Aprendizaje, mientras que el 5.6 % responde que tiene conocimiento de la existencia de un Software que busca Objetos de Aprendizaje. Aquí se puede evidenciar que la mayoría de los estudiantes no conocen de la disponibilidad de los Recursos Educativos Abiertos que existen disponibles en la Web.

Pregunta 2

¿Cómo considera la interfaz del software propuesto denominado SMAS?

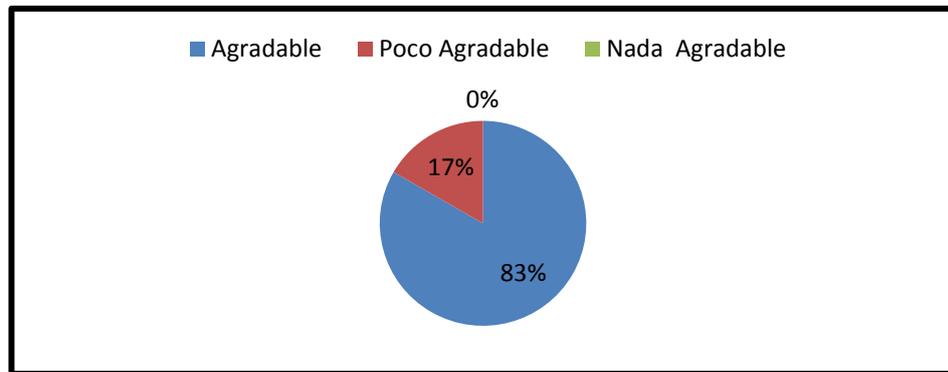


Figura 68. Análisis del resultado pregunta 2.

Interpretación y Análisis.

El 83.3 % de los estudiantes encuestados consideran que la interfaz de SMAS es agradable para su uso, mientras que el 16.7 % responde que encuentra poco agradable la interfaz del Software, y a ninguno de los encuestados le pareció que la interfaz es nada agradable. A una minoría de estudiantes le parece que la interfaz no es agradable a su gusto por lo cual se puede concluir que la interfaz resulto agradable a la mayoría los usuarios.

Pregunta 3

El nivel de usabilidad de SMAS le pareció:

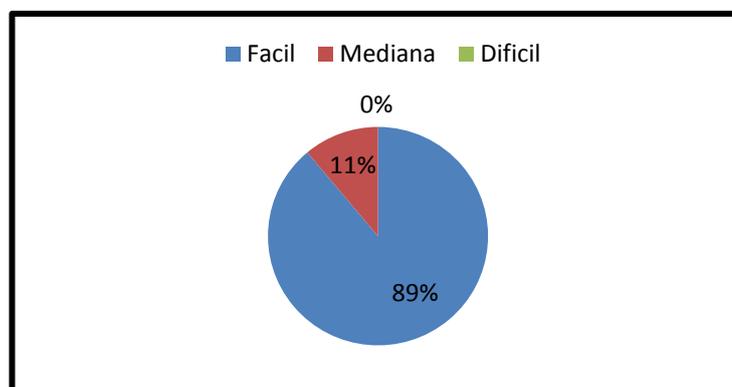


Figura 69. Análisis del resultado pregunta 3.

Interpretación y Análisis.

El 83.9 % de los estudiantes encuestados consideran que la usabilidad de SMAS es fácil de entender, mientras que el 11.1 % responde que encuentra la usabilidad de SMAS mediana y un 0% considera que la usabilidad es difícil. Se puede concluir que SMAS es comprendido, aprendido, usado de manera eficiente de acuerdo a los resultados.

Pregunta 4

De acuerdo a la búsqueda realizada en SMAS esta fue:

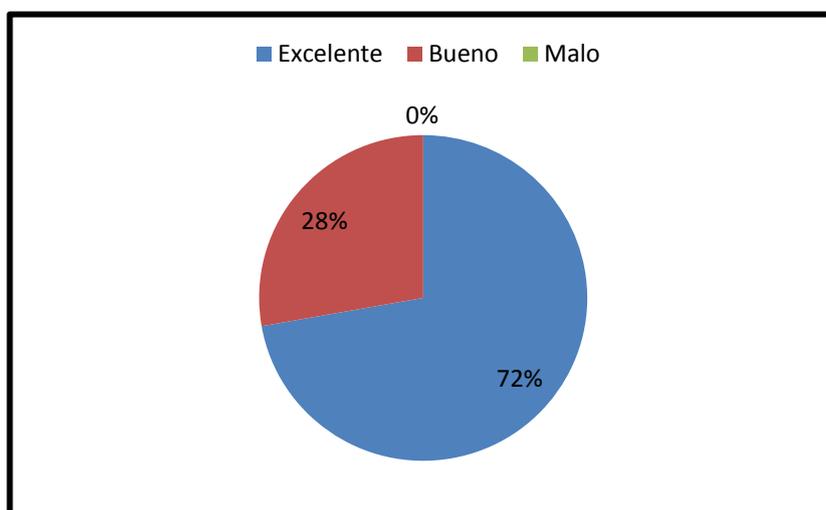


Figura 70. Análisis del resultado pregunta 4.

Interpretación y Análisis.

El 27.8 % de los estudiantes encuestados consideran que luego de haber utilizado SMAS y realizado búsquedas en el mismo esta fue excelente, mientras que el 66.7 % responde su búsqueda en SMAS fue buena, y el 5.6 % considera que de acuerdo a su búsqueda no encontró ningún resultado en el mismo. Se puede concluir que SMAS si es ayuda a encontrar temas de interés en la mayoría de los casos ya que una mínima parte de los encuestados responde que no encontró resultados en su búsqueda.

Pregunta 5

El tiempo de respuesta de su búsqueda fue:

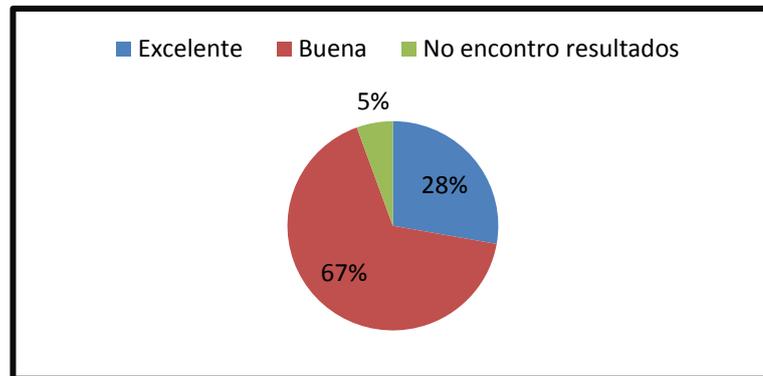


Figura 71. Análisis del resultado pregunta 5.

Interpretación y Análisis.

El 72.2 % de los estudiantes encuestados consideran que luego de haber realizado su búsqueda el tiempo de respuesta fue excelente, mientras que el 27.8 % respondió que el tiempo de respuesta fue bueno y el 0% respondió que el tiempo de búsqueda fue malo. Se puede concluir que el tiempo de respuesta al presentar los resultados de la búsqueda es rápido.

Pregunta 6

¿Considera usted que este Software es un apoyo para el proceso de Enseñanza/Aprendizaje en al Ámbito Educativo?

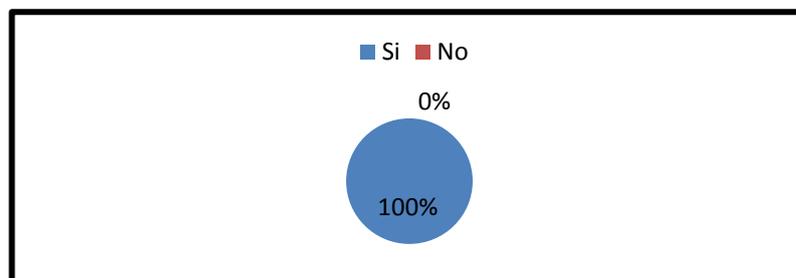


Figura 72. Análisis del resultado pregunta 6.

Interpretación y Análisis.

El 100 % de los estudiantes encuestados consideran que SMAS es un sistema que de ayuda para la Búsqueda de Objetos de Aprendizaje en el Ámbito Educativo. Por lo cual se puede concluir que la implementación de sistemas que recuperen información fiable ayudaría al proceso de enseñanza/aprendizaje

11.6. Anexo 6: Anteproyecto



**UNIVERSIDAD NACIONAL
DE LOJA**



PFM-CIS-XA-012

Área de la Energía las Industrias y los Recursos Naturales No Renovables

Carrera de Ingeniería en Sistemas

“Sistema Multiagente basado en un modelo Ontológico para la Búsqueda de Objetos de Aprendizaje”

ANTEPROYECTO DE TRABAJO DE TITULACIÓN

Autores:

- Guamán-Morocho, Gabriela-Paola
- Martínez-Pacheco, John-Carlos

Tutor:

- Paz-Arias, Henry-Patricio

Loja-Ecuador
2014

Índice

A. Tema	5
B. Problemática	5
1. Situación Problemática.	5
2. Problema de Investigación	6
C. Justificación	6
D. Objetivos	7
1. Objetivo General	7
2. Objetivos Específicos	7
	8
E. Alcance	
F. Metodología	10
G. Cronograma	11
H. Presupuesto y Financiamiento	12
I. Bibliografía	14
J. Anexos	15
1. Anexo 1: Licencia.	15

Índice de figuras

1. Cronograma de Trabajo	13
------------------------------------	----

Índice de tablas

1. Presupuesto estimado Recursos	12
2. Presupuesto estimado Recursos Técnicos.	12
3. Presupuesto estimado Recursos Humanos.	13
4. Presupuesto estimado del Trabajo de Titulación.	13

A. Tema

“Sistema Multiagente basado en un modelo Ontológico para la Búsqueda de Objetos de Aprendizaje”

B. Problemática

1. Situación Problemática

Las tecnologías de la información y comunicación han adquirido gran relevancia, principalmente a partir del amplio uso del Internet, siendo el educativo uno de sus más importantes campos de acción, constituyéndose en una importante herramienta de investigación que permite la interacción a un doble nivel: entre personas y con los contenidos, ello facilita que pueda desarrollarse más fácilmente un proceso de aprendizaje cooperativo centrado en la búsqueda, tratamiento, procesamiento y presentación de la información [1].

Dentro de este marco se han realizado diversas acciones para proporcionar recursos educativos gratuitos en la Web, uno de estos son los Objetos de Aprendizaje (OA), los cuales posibilitan la compartición de recursos digitales que pueden ser usados como soporte para el aprendizaje; dichos materiales educativos han sido diseñados y creados con el propósito de maximizar el uso y reutilización de un mismo material en distintas situaciones educativas, pudiendo ser actualizados o referenciados con el fin de generar conocimientos, habilidades y competencias en el ámbito de la educación [2].

En el ámbito mundial se ha destacado la iniciativa de almacenamiento de Objetos de Aprendizaje en los denominados Bancos, entre los que sobresalen: JoRUM, Merlot y Aproa, los cuales proporciona un repositorio de Objetos de Aprendizaje principalmente orientados para su utilización en la Educación Virtual [3]. En Ecuador podemos mencionar el proyecto denominado “Revolución Educativa”, el cual se ha propuesto mejorar el aprendizaje fomentando el uso de objetos de aprendizaje, en la televisión, la radio, el cine y el impreso en el aula de clase [4].

Debido a la gran cantidad de información disponible en Internet, muchas de las veces la recuperación de Objetos de Aprendizaje, ha generado problemas para ubicar y seleccionar recursos de calidad, ya que al no tener elementos que

describan su contenido, el usuario de Internet se ve expuesto a grandes bancos o sitios de información, en los que la mayoría de ellos no corresponden con su interés por la falta de estrategias en la búsqueda y recuperación de Objetos de Aprendizaje [5].

Entre las estrategias y tecnologías para búsqueda y recuperación de información tenemos la Web Semántica, Linked Data, Ontologías, Agentes Inteligentes, entre otras. En esta área está surgiendo el uso de Agentes Inteligentes, que son herramientas que permiten localizar información de forma automática, solo necesitan que se defina un perfil de búsqueda y donde se deben lanzar (bases de datos, sitios web y otros.) y, automáticamente presenta un informe sobre la nueva información que va surgiendo, generando una ventaja motor que es la rapidez en los resultados, también surge la necesidad de utilizar Ontologías para la comunicación entre agentes, las mismas que permiten definir los lenguajes con los que el agente intercambia mensajes, permitiendo resolver problemas semánticos y logrando la interoperabilidad de sistemas y plataformas.[6].

En base a lo expuesto anteriormente se puede establecer algunos problemas relacionados con la búsqueda y recuperación de los Objetos de Aprendizaje, mencionados a continuación [7, 8]:

- La no existencia de estrategias para la reutilización de los Objetos de Aprendizaje en contextos educativos en general.
- Ausencia de buscadores especializados.
- La escasa calidad de la información recuperada por los buscadores Web, debido a que la estrategia de búsqueda ha sido demasiado específica o que las palabras clave utilizadas no son las adecuadas para definir la búsqueda.
- Documentos recuperados que no son relevantes, porque la estrategia de búsqueda se ha definido demasiado genérica.
- Imposibilidad para acceder de forma directa e instantánea a los Objetos de Aprendizaje por campos concretos.

2. Problema de Investigación

En base a la situación problemática actual, la realización del presente Trabajo de titulación pretende dar solución al siguiente problema:

La falta de búsqueda centralizada de objetos de aprendizaje albergados en diferentes repositorios web, dificulta la búsqueda ocasionando pérdidas considerables de tiempo por parte de los usuarios.

C. Justificación

La realización del presente Trabajo de Titulación denominado “Sistema Multiagente basado en un modelo Ontológico para la Búsqueda de Objetos de Aprendizaje”, pretende dar solución a un tema de suma actualidad, contemporáneo a los cambios que se analizan, en particular en hacer eco a las iniciativas a favor de la libre distribución y acceso a Objetos de Aprendizaje, creemos que es necesario generar conocimiento sobre las formas en que la tecnología puede mejorar el acceso a la información.

El Trabajo de Titulación se justifica académicamente ya que mediante la resolución de problemas específicos, el aporte académico que se generará, resultará beneficioso para poder ampliar los conocimientos e incentivar a los profesionales en formación a la experimentación de tecnologías inusuales en nuestro entorno, debido principalmente al desconocimiento de las mismas. Además impulsará a la interrelación de tecnologías obteniendo nuevos sistemas que sirvan de aporte en el ámbito académico y social mejorando la calidad de vida de las personas.

El aporte científico consistirá en describir las características relevantes que se dan en el proceso de la búsqueda y recuperación de Objetos de Aprendizaje desde la Web utilizando una arquitectura de Agentes Inteligentes, considerando siempre el uso de técnicas adecuadas y buenas practicas. También se justifica ambientalmente, ya que al tratarse de un trabajo que pretende impulsar la utilización de Objetos de Aprendizaje, representa un ahorro significativo en cuanto a la utilización de recursos materiales provenientes del medio ambiente como es el papel para la difusión de información ya que se aprovechara recursos gratuitos de la Web.

Al tratarse un proyecto netamente académico el presupuesto necesario para poder llevar a cabo su elaboración ser cubierto por los responsables de la realización del Trabajo de Titulación; con el fin de apoyar al avance de la Carrera de Ingeniería en Sistemas y por ende a la Universidad Nacional de Loja.

Por lo expuesto anteriormente y fundamentándose en casos de éxito y revisión bibliográfica se considera que el Trabajo de Titulación a realizar queda plenamente justificado desde los ámbitos económico, científico, académico, social y ambiental.

D. Objetivos

1. Objetivo General

Construir un Sistema Multiagente basado en un modelo Ontológico para la búsqueda de Objetos de Aprendizaje desde la Web.

2. Objetivos Específicos

- Revisar el estado del arte en lo referente a Objetos de Aprendizaje, Ontologías y Agentes Inteligentes para la búsqueda de información.
- Diseñar una Ontología basada en los estándares establecidos para la representación de objetos de aprendizaje.
- Construir un Sistema Multiagente para la búsqueda de Objetos de Aprendizaje.
- Evaluar el funcionamiento de la aplicación en la búsqueda de Objetos de Aprendizaje

E. Alcance

Para el desarrollo del presente Trabajo de Titulación se ha estimado y delimitado las siguientes actividades acorde a cada uno de los objetivos planteados:

1. Revisar el estado del arte en lo referente a Objetos de Aprendizaje, Ontologías y Agentes Inteligentes para la búsqueda de información.

- Revisar literatura acerca de los Objetos de Aprendizaje.
- Revisar literatura acerca de Ontologías.
- Revisar metodologías para la construcción de Ontologías.
- Revisar literatura acerca de Agentes Inteligentes.
- Revisar arquitecturas de agentes inteligentes.
- Revisar tipos de agentes inteligentes.
- Revisar metodologías para la construcción de Sistemas Multiagente.
- Revisar trabajos relacionados de Agentes Inteligentes en la búsqueda de información
- Analizar las herramientas existentes para la construcción de Agentes Inteligentes y Ontologías.

2. Diseñar una Ontología basada en los estándares establecidos para la representación de objetos de aprendizaje.

- Especificación de requerimiento de la Ontología.
- Conceptualización de los estándares establecidos para la representación de Objetos de Aprendizaje.
- Construcción de la Ontología con Software Computacional.

3. Construir un Sistema Multiagente para la búsqueda de Objetos de Aprendizaje.

- Análisis de las especificaciones que deben poseer los Agentes Inteligentes para la búsqueda de Objetos de Aprendizaje.
- Diseñar los agentes inteligentes necesarios para la búsqueda de Objetos de Aprendizaje.
- Desarrollar los agentes inteligentes para la búsqueda.
- Implantación del Sistema Multiagente de búsqueda de Objetos de Aprendizaje.

4. Evaluar el funcionamiento de la aplicación en la búsqueda de Objetos de Aprendizaje.
 - Detección de errores técnicos y lógicos.
 - Corrección de fallos.
 - Evaluar el funcionamiento del Sistema Multiagente para la búsqueda de Objetos de Aprendizaje
5. Terminación del Trabajo de Titulación.
 - Documentación de la memoria final
 - Elaborar un Artículo Científico según las normas IEEE

F. Metodología

Para el desarrollo del Trabajo de Titulación se utilizarán distintos métodos de investigación y técnicas de recolección de información bibliográfica, para poder cumplir con los objetivos planteados.

Estudio de casos.-Este método nos permitirá realizar una exploración e investigación en profundidad de problemas específicos, basados en experiencias ya vividas y contadas por los casos de éxito. En base al estudio de casos de éxito podemos afirmar que la realización del Trabajo de Titulación resulta viable y factible de realizar.

Experimentación.- Técnica mediante la cual se podrá realizar las diferentes pruebas que garanticen el correcto funcionamiento de nuestra solución, en base a un escenario donde se ponga a prueba cada una de las aplicaciones y servicios que se puedan implementar.

Técnica de Recolección Bibliográfica.- Esta técnica nos permitirá extraer la información necesaria para poder sustentar la base teórica del Trabajo de Titulación, mediante consultas en fuentes bibliográficas confiables, casos de éxito, artículos científicos, revistas indexadas; entre otras.

G. Cronograma

	Nombre	Inicio	Terminado	# Horas
1	☐ Fase 1: Revisar el estado del arte en lo referente a Objetos de Aprendizaje, Ontologías y Agentes Inteligentes	15/12/14 08:00 AM	30/12/14 05:00 PM	100 h
2	Revisar literatura acerca de los Objetos de Aprendizaje.	15/12/14 08:00 AM	16/12/14 05:00 PM	10h
3	☐ Revisar literatura acerca de Ontologías.	16/12/14 08:00 AM	17/12/14 05:00 PM	16 h
4	Revisar metodologías para la construcción de Ontologías.	16/12/14 08:00 AM	17/12/14 05:00 PM	16 h
5	☐ Revisar literatura acerca de Agentes Inteligentes.	18/12/14 08:00 AM	24/12/14 05:00 PM	74 h
6	Revisar arquitecturas de agentes inteligentes.	18/12/14 08:00 AM	19/12/14 05:00 PM	16 h
7	Revisar tipos de agentes inteligentes.	19/12/14 08:00 AM	22/12/14 05:00 PM	16 h
8	Revisar metodologías para la construcción de Sistemas Multiagente.	22/12/14 08:00 AM	24/12/14 05:00 PM	08 h
9	Revisar trabajos relacionados de Agentes Inteligentes en la búsqueda y recuperación de información.	24/12/14 08:00 AM	26/12/14 05:00 PM	16 h
10	Analizar las herramientas existentes para la construcción de Agentes Inteligentes y Ontologías.	26/12/14 08:00 AM	30/12/14 05:00 PM	18 h
11	☐ Fase 2: Diseñar una Ontología basada en los estándares establecidos para la representación de objetos de aprendizaje	30/12/14 08:00 AM	26/01/15 05:00 PM	70 h
12	Especificación de requerimiento de la Ontología.	30/12/14 08:00 AM	02/01/15 05:00 PM	25 h
13	Conceptualización de los estándares establecidos para la representación de Objetos de Aprendizaje.	02/01/15 08:00 AM	06/01/15 05:00 PM	15 h
14	Construcción de la Ontología.	07/01/15 08:00 AM	26/01/15 05:00 PM	30 h
15	☐ Fase 3: Construir un Sistema Multiagente para la búsqueda de Objetos de Aprendizaje.	26/01/15 08:00 AM	03/04/15 05:00 PM	125 h
16	Análisis de las especificaciones que deben poseer los Agentes Inteligentes para la búsqueda de Objetos de Aprendizaje.	26/01/15 08:00 AM	30/01/15 05:00 PM	25 h
17	Diseñar los agentes inteligentes necesarios para la búsqueda de Objetos de Aprendizaje.	02/02/15 08:00 AM	27/02/15 05:00 PM	25 h
18	Desarrollar los agentes inteligentes para la búsqueda.	02/03/15 08:00 AM	27/03/15 05:00 PM	60 h
19	Implantación del Sistema Multiagente de búsqueda de Objetos de Aprendizaje.	30/03/15 08:00 AM	03/04/15 05:00 PM	15 h
20	☐ Fase 4: Evaluar el funcionamiento de la aplicación en la búsqueda de Objetos de Aprendizaje.	06/04/15 08:00 AM	01/05/15 05:00 PM	50 h
21	Detección de errores técnicos y lógicos.	06/04/15 08:00 AM	10/04/15 05:00 PM	15 h
22	Corrección de fallos.	13/04/15 08:00 AM	24/04/15 05:00 PM	15 h
23	Evaluar el funcionamiento del Sistema Multiagente para la búsqueda de Objetos de Aprendizaje.	27/04/15 08:00 AM	01/05/15 05:00 PM	20 h
24	☐ Fase 5: Terminación del Trabajo de Titulación.	04/05/15 08:00 AM	22/05/15 05:00 PM	55 h
25	Elaborar un artículo científico acorde a las normas IEEE.	04/05/15 08:00 AM	08/05/15 05:00 PM	15 h
26	Documentación del Trabajo de Titulación.	11/05/15 08:00 AM	22/05/15 05:00 PM	40 h
			TOTAL	400 h

Figura 1: Cronograma de Trabajo

H. Presupuesto y Financiamiento

Tabla 1: Presupuesto estimado Recursos Materiales.

Descripción	Cantidad	V. Unitario (\$)	V. Total (\$)
Internet	500	0.80	400
Impresiones	500	0.10	50
Copias	200	0.02	4
Varios	-	-	45
Total (\$):			499

Tabla 2: Presupuesto estimado Recursos Técnicos.

Descripción	Cantidad	V. Inicial	Depre	V. Total (\$)
HARDWARE				
Computador	2	1200	480	960
Flash Memory	1	10	-	10
Disco duro	1	50	-	50
SOFTWARE				
Latex	2	0	0	0
Open Project	2	0	0	0
Total (\$):				1120

Tabla 3: Presupuesto estimado Recursos Humanos.

Rol	N. Horas	V. Unitario (\$)	V. Total (\$)
Investigador 1	200	7	1400
Investigador 2	200	7	1400
Tutor	-	-	-
Total (\$):			2800

Tabla 4: Presupuesto estimado del Trabajo de Titulación.

Recurso	Subtotal (\$)
R. Humanos	2800
R. Técnicos	1120
R. Materiales	499
Imprevistos	500
Total (\$):	4919

La elaboración del presupuesto estimado para el Trabajo de Titulación, ha sido realizada considerando las normativas expuestas en la Norma Ecuatoriana de Contabilidad (NEC No 14) referente a los Costos de Investigación y Desarrollo [11]. Además se ha considerado la Aplicación de la Ley de Régimen Tributario Interno sobre depreciaciones de equipos [12].

El financiamiento para la realización del presente Trabajo de Titulación, al tratarse de un proyecto académico, será asumido en su totalidad por los responsables de su realización.

I. Bibliografía

Referencias

[1] Torricelli M. Raúl G, Lee T. Francisco C y López P. Jorge L, “Repositorio de Recursos Educativos del Ministerio de Educación Superior: REMES”, Ministerio de Educación Superior, Cuba, 2011.

[2] Serrano I. María “Objetos de Aprendizaje”, Revista e-Formadores, Disponible en: http://red.ilce.edu.mx/sitios/revista/e_formadores_oto_10/articulos/angeles_serrano_nov10.pdf

[3] Callejas C. Mauro, Hernández N. Edwin J, Pinzón V. Josué N , “Objetos de aprendizaje, un estado del arte”, Vol.7 No. 1, 2011.

[4] Ministerio de Educación Nacional, “Revolución Educativa”, Disponible en: http://www.mineduacion.gov.co/1621/propertyvalues-31326_tablero_pdf.pdf.

[5] Mora V. Francisco, “Objetos de Aprendizaje: Importancia de su uso en la Educación Virtual”, Volumen 3, Número 1, Mayo 2012.

- [6] José L. Berrocal, Carlos G. Figueroa, Ángel F. Zazo y Emilio Rodríguez, “ Inteligentes: Recuperación autónoma de información en la Web”, Disponible en: <http://redc.revistas.csic.es/index.php/redc/article/viewFile/130/184>.
- [7] Ferreira Ana, Atkinson John, “Un Modelo de Agente de Búsqueda y Filtrado de Información Inteligente”, Universidad de Tarapaca, Chile.
- [8] López C, “Los Repositorios de Objetos de Aprendizaje como soporte a un entorno eLearning”, Tesina doctoral, Universidad de Salamanca, 2013.
- [9] Martín G. and Dawson C., “El Proyecto de Fin de Carrera en Ingeniería en Sistemas: Una Guía para el estudiante”, Prentice HallEducacion, 1era edición, 2002.
- [10] Hernández R., and Fernández C. and Baptista María, “Metodología de la investigación”, 5ta Edición, McGRAW-HILL, México, 2010.
- [11] Federación Nacional de Contadores del Ecuador, C ostos de Investigación y desarrollo (NEC No 14)”, reforma 2001.
- [12] Servicio de Rentas Internas, *Reglamento de Aplicación de la Ley de Régimen Tributario Interno*, (2010), Disponible en: <http://www.sri.gob.ec/web/guest/depreciacion-acelarada-de-activos-fijos>

J. Anexos

1. Anexo 1: Licencia



Sistema Multiagente basado en un modelo Ontológico para la Búsqueda de Objetos de Aprendizaje, by Martínez John y Guamán-Gabriela, se distribuye bajo una [Licencia Creative Commons Atribucion-NoComercial-SinDerivar 4.0 Internacional](#).

11.7. Anexo 7: Artículo sobre el trabajo de titulación

Sistema Multiagente basado en un modelo Ontológico para la búsqueda de Objetos de Aprendizaje.

G. Guamán, J. Martínez, L. Jácome.

Abstract— This article summarizes the elements taken into account for the analysis and design and implementation of a system Multiagent Learning Object Search, which uses an ontological domain model as defining them and which information is extracted in the construction of this system the basis of the Semantic web is implemented, demonstrating their usefulness with Software Agents, these in their environment are responsible for using web Services repositories available on the Internet, and also to keep them updated; which provides metadata and learning objects are instantiated in the Ontology.

Keywords: Multiagent, Learning Object, ontological.

Resumen:

El presente artículo resume los elementos tomados en cuenta para realizar el análisis y diseño e implementación de un sistema Multiagente de búsqueda de Objetos de Aprendizaje, que utiliza un modelo Ontológico como dominio de definición de los mismos, en la construcción de este sistema se implanta las bases de la Web Semántica, demostrando su utilidad junto con Agentes de Software, estos en su entorno son los encargados de utilizar los Servicios Web de los repositorios disponibles en la Internet, los cuales nos proporciona los metadatos de los Objetos de Aprendizaje y son instanciados en la Ontología.

I. INTRODUCCIÓN

En la actualidad la Word Wide Web ha alcanzado un amplio uso en la sociedad, existiendo una abrumadora cantidad de información, pero el usuario se ve expuesto a grandes bancos o sitios de información en los que la mayoría de ellos no corresponden a su interés por la falta de estrategias en la búsqueda. Este trabajo se ha enfocado en la utilización de Objetos de Aprendizaje que es cualquier recurso digital que puede ser utilizado para facilitar el aprendizaje, estos son accesibles desde repositorios vía Web.

Por lo expuesto anteriormente este trabajo busca estimular la utilización de Objetos de Aprendizaje, y centralizar su búsqueda desde diferentes repositorios Web, para lo cual se ha desarrollado un software de búsqueda, utilizando diferentes tecnologías no muy usuales que están en etapa de crecimiento, entre ellas Agentes Inteligentes los cuales ayudan a percibir y responder al entorno en un tiempo razonable, exhibir comportamientos dirigidos por metas, e interactuar con otros agentes, y las Ontologías nos permiten

organizar la información por dominios especializados, ambas tecnologías se completan los Agentes Inteligentes se encargan de inferir conocimiento a partir del dominio específico bien definidos de los Objetos de Aprendizaje.

La organización del trabajo es la siguiente: en la Sección II se explica las características de las herramientas utilizadas, seguidamente la Sección III describe los materiales y métodos. La Sección IV muestra detalles de los resultados. Finalmente, se puede encontrar algunas conclusiones sobre el presente trabajo.

II. CARACTERÍSTICAS DE LAS HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.

Protege: Modelamiento y diseño de la ontología [1].

Jena: Proporciona soporte para crear un modelo RDFs/OWL. Se utiliza para el razonamiento de la Ontología (Poblar de individuos y Consultas Sparql) [2].

JADE: Java Agent Development Framework es un framework de agentes de software, que permite crear agentes específicos para satisfacer las necesidades del usuario que lo utiliza [3].

Virtuoso OpenSource: Para el almacenamiento de Tripletas y la realización de pruebas a la Ontología [4].

Primefaces: Framework de aplicaciones para manejar las peticiones del cliente y cuenta con un conjunto de componentes que facilitan la creación de las aplicaciones web [5].

Bootstrap: Permite que la pantalla se adapte automáticamente al tamaño de un ordenador o de cualquier dispositivo móvil, esto se denomina diseño adaptativo o Responsive Design [6].

III. MATERIALES Y MÉTODOS

Para el desarrollo del proyecto se utilizó diferentes métodos de investigación, como recolección bibliográfica, estudio de casos y observación activa lo que permitió realizar un estudio de la información recolectada y adquirir los conocimientos necesarios para dar solución al problema de investigación. También se utilizó una metodología formal para el desarrollo del software ICONIX, la cual se complementó con una metodología para el diseño de la Ontología e Ingenias para la construcción del Sistema Multiagente, las cuales son la parte esencial de todo proyecto ya que permite guiar y organizar las actividades para cumplir los objetivos planteados.

De esta manera se determinó la utilización de una metodología híbrida y robusta que permita incluir actividades tanto para el diseño de la Ontología como de Agentes utilizando ICONIX, Methontology e Ingenias.

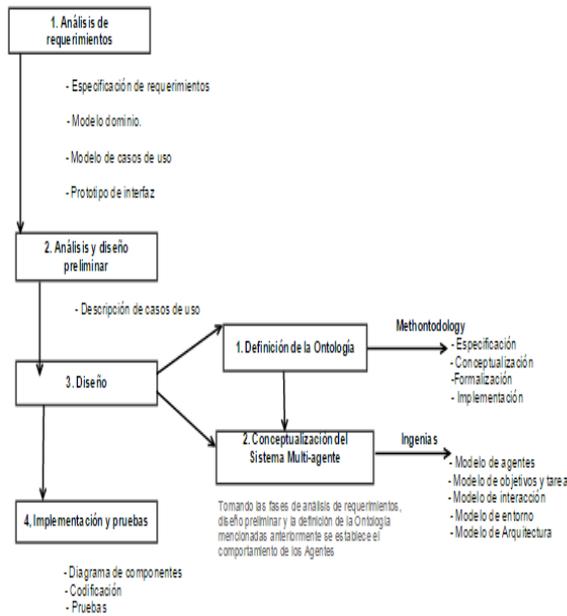


Fig. 1 Descripción de la metodología propuesta.

IV. RESULTADOS.

A. Revisar el estado del arte en lo referente a Objetos de Aprendizaje, Ontologías y Agentes Inteligentes

Al revisar la literatura de los Objetos de Aprendizaje (OA), se seleccionó los principales estándares de catalogación de metadatos los proporcionan una descripción de los mismos para una búsqueda rápida y efectiva en el repositorio donde se encuentre.

Entre los estándares de metadatos, los cuales sirven para catalogar los objetos de aprendizaje, se encuentran el IEEE-LOM y Dublin Core [7-8].

El estándar Dublin Core es un estándar bastante simple, pero no menos eficiente para la catalogación de OA, en la Figura 2 se detalla cómo fueron tomadas sus categorías.

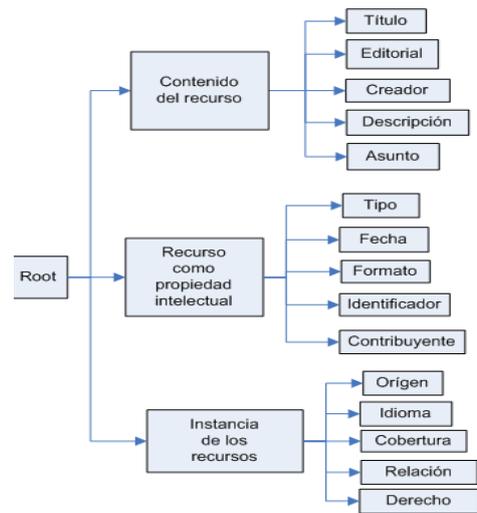


Fig. 2 Estructura de Dublin Core.

El estándar LOM, es muy amplio y describe a fondo cada OA, en la actualidad es el estándar por excelencia considerado por la IEEE.

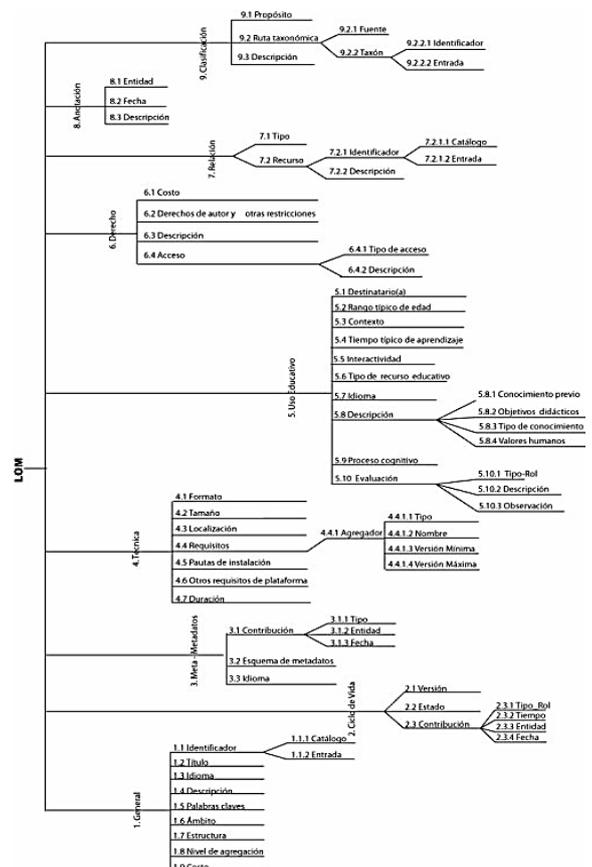


Fig. 3 Estructura de Dublin Core.

Entre los repositorios tomados en cuenta tenemos los siguientes:

TABLA I.
REPOSITARIOS DE OBJETOS DE APRENDIZAJE.

Nombre	URL
Repositorio Institucional Universidad Nacional Autónoma de México (UNAM).	www.rad.unam.mx
Mayor Repositorio del Reino Unido (JORUM).	www.jorum.ac.uk
Centro de Recursos para la enseñanza y aprendizaje (CREA).	www.crea.udg.mx
Banco Internacional de Objetos de Aprendizaje.	http://objetoseducacionais2.mec.gov.br/
Repositorio Objetos de aprendizaje de Escuela Superior Politécnica de Literal.	www.sidweb.espol.edu.ec
Fedora Commons.	http://fedora-commons.org/

Seguidamente se revisó Ontologías, los aspectos que se tomaron en cuenta fueron componentes, Lenguajes ontológicos y consultas, en la cual se concluyó realizar una Ontología del Dominio que describa los Objetos de Aprendizaje utilizando el lenguaje ontológico RDFS/OWL; En la revisión literaria de Agentes Inteligentes se estudió los principales tipos de Agentes y arquitecturas, donde se determinó utilizar una arquitectura

reactiva ya que el razonamiento sobre la búsqueda de OA la realiza la Ontología.

La metodología escogida para el presente trabajo fue ICONIX, la cual ayudó a llevar de forma organizada el desarrollo de software y la cual se complementó con la metodología Methontology para el diseño de la Ontología y otra para la construcción del Sistema Multiagente Ingenias, las cuales permitieron guiar y organizar las actividades para cumplir los objetivos planteados.

B. Análisis de Requerimientos y Diseño preliminar.

Para el modelado de los casos de uso, se identificó dos actores: Administrador e Investigador que intervienen en el sistema.

En la Fig. 4 se muestran el diagrama de casos de uso, y su interacción con los actores tanto Investigador como Administrador.

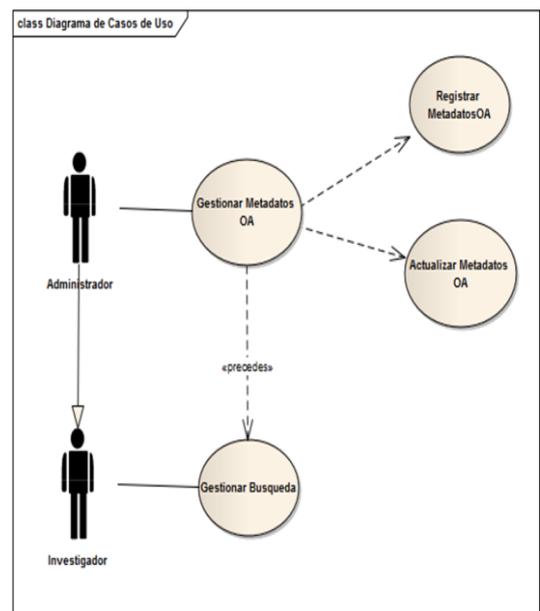


Fig. 4 Diagrama de Casos de Uso.

En la Fig. 5 se muestra las clases identificadas mediante el análisis realizado.

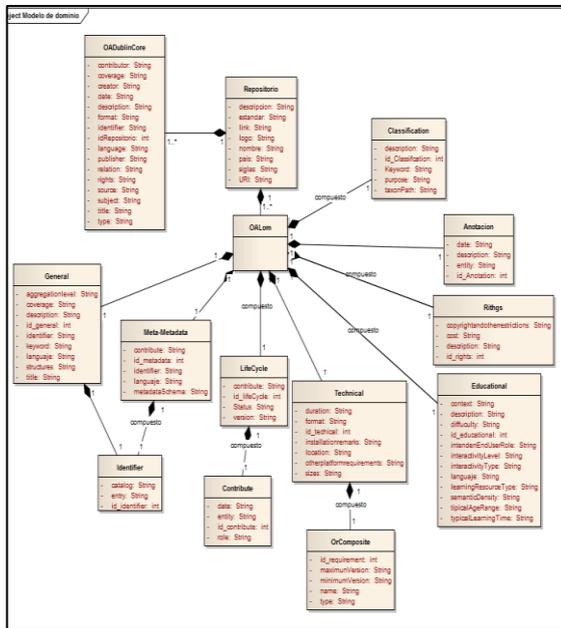


Fig. 5 Modelo del Dominio.

C. Diseñar una Ontología con los parámetros que deben poseer los recursos a ser recuperados

Para modelar la Ontología se utilizaron los estándares LOM y Dublin Core. Para el razonamiento de la Ontología se hace uso del motor de inferencias Jena, para las cuales el criterio de búsqueda deber ser escrito de la misma forma cómo se encuentran representadas las instancias dentro de la Ontología.

Construida la ontología es necesario instanciar, proceso que actualmente se realiza de manera automática en base a los recursos

almacenados, para lo cual se hace mediante un recolector de metadatos que utiliza peticiones HTTP para recuperar los metadatos de OA y los almacena en MySQL, y luego se llena de individuos la Ontología.

La taxonomía de conceptos propuesta para la ontología se puede observar e la Fig. 6 y 7.

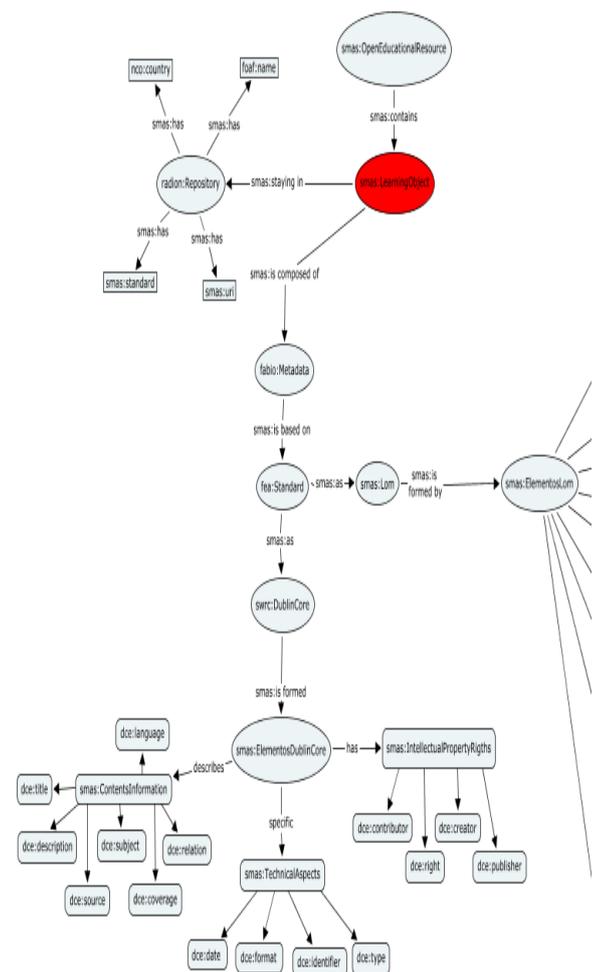


Fig. 6 Taxonomía 1 de la Ontología

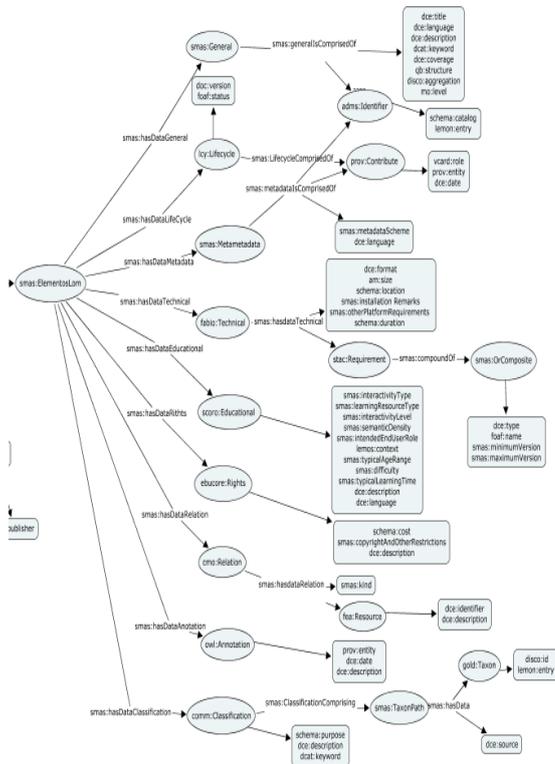


Fig. 7 Taxonomía 2 de la Ontología

D. Construir un Sistema Multiagente para la búsqueda y recuperación de Objetos de Aprendizaje.

La arquitectura general del sistema se presenta de forma gráfica en la Fig. 8.

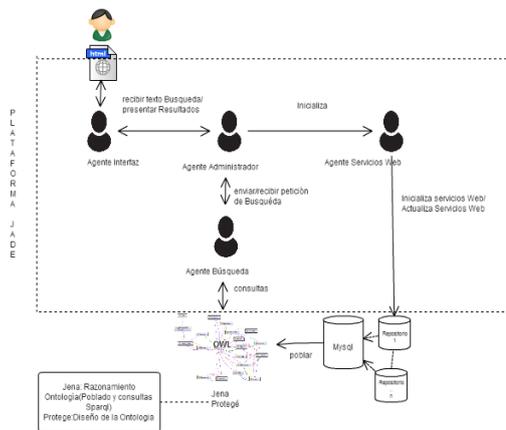


Fig. 8 Arquitectura del Sistema Multiagente

En la arquitectura modelada se determinó la utilización de los siguientes agentes: Agente

Administrador encargado de inicializar todos los Agentes Inteligentes (Búsqueda, Interfaz y Servicios Web); Agente de Búsqueda encargado de buscar Objetos de Aprendizaje sobre la Ontología y la entregar resultados al Agente Interfaz; Agente Interfaz encargado de cumplir peticiones de Interfaz y entregar resultados; Agente de Servicios Web encargado de llamar al cosechador de metadatos. Y buscar los Servicios Web de los repositorios y recuperar los metadatos. La implementación de esta arquitectura permitió que el Sistema Multiagente busque Objetos de Aprendizaje, los OAs resultantes serán consultados en la Ontología, los agentes se encargan de gestionar los Servicios Web y poblar la Ontología, y también serán los encargos de gestionar las búsquedas y presentar los resultados.

La implementación del Sistema Multi-agente se realizó siguiendo la lógica que se muestra en el diagrama de componentes de la Fig. 9.

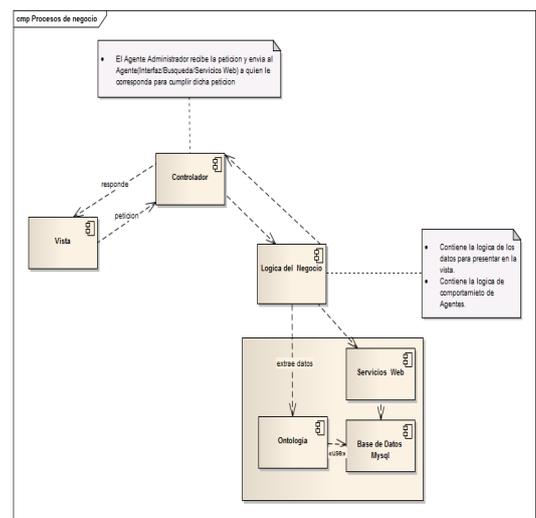


Fig. 9 Diagrama de componentes

E. Evaluar el funcionamiento de la aplicación en la búsqueda y recuperación de Objetos de Aprendizaje.



Figura 10. Página de inicio de SMAS.

Para concluir se realizaron pruebas de usabilidad las cuales permitieron conocer el impacto que el sistema tuvo frente al usuario, de esta forma se evaluó cada módulo con la finalidad de obtener la respuesta de la Aplicación Web. Para conocimiento del software realizado puede visitar la dirección Web: www.smasunl.ml

V. CONCLUSIONES.

- Los resultados obtenidos en este proyecto ponen en evidencia que los agentes de software son actualmente un paradigma que cuenta con un nivel aceptable tanto de investigación como de desarrollo, es así que en el presente trabajo se demuestra la aplicabilidad de los sistemas Multi-agente en ámbito búsqueda de recursos educativos.
- Los metadatos constituyen un pilar fundamental dentro de la semántica ya que a través de ellos se puede identificar un recurso dentro de la web, pese a estas

descripciones que los metadatos otorgan, existe una carencia de semántica en los mismos, por esta razón el uso de ontologías que permite adicionar relaciones, axiomas y reglas de inferencia para definir un recurso dentro de la web y poder realizar búsquedas.

- El ingreso de metadatos dentro de una estructura ontológica, al realizarse de forma manual resulta tedioso, confuso y hasta costoso, es por ello que se debe buscar herramientas para instanciar la Ontología de individuos de manera automática.
- El aporte fundamental de nuestro enfoque está en la integración de modelos y estrategias de dos áreas como son: agentes de software y recuperación de información de Ontologías para producir un sistema de interacciones dialógicas robusto y fácilmente utilizable por usuarios inexpertos, la inclusión de semántica en los Objetos de Aprendizaje y su importancia en el contexto educativo.
- La tecnología de agentes en general y la de los Sistemas Multiagente en particular ofrecen características muy apropiadas para el desarrollo de aplicaciones distribuidas y complejas en entornos donde se disponga de numerosos componentes con diferentes niveles de experiencia e interés conflictivos, los agentes añaden a este tipo de sistemas propiedades como la autonomía y comportamiento dirigido por Objetivos

que permiten el desarrollo de numerosas actividades sin la necesidad de intervención humana.

REFERENCIAS

- [1] (2014) Protegé Homepage. Accessed: Diciembre 2014. [Online]. Available: <http://protege.stanford.edu/>
- [2] R. Alcaraz. "Aplicacions pel web semàntic amb Jena". Noviembre 2012. [Online]. Available: <http://www.cobdc.net/programarilliure/aplicacions-pel-web-semantic-amb-jena/>
- [3] (2014) OpenLink Virtuoso. [Online]. Available: <http://virtuoso.openlinksw.com/>
- [4] (2014) Java Agent Development Framework (JADE). [Online]. Available: <http://jade.tilab.com/>
- [5] (2014) Netbeans. [Online]. Available: <https://netbeans.org/features/index.html>
- [6] (2014) PRIMEFACES. [Online]. Available: <http://blog.primefaces.org/>
- [7] (2012) Dublin Core Homepage. Accessed: Septiembre 2014. [Online]. Available: <http://dublincore.org/documents/dces/>
- [8] V. Tabares, N. Duque, J. Moreno; D. Ovalle and R. Maria, "Evaluación de la calidad de metadatos en repositorios digitales de objetos de aprendizaje", Rev. Interam. Bibliot., vol.36 no.3, 2013.



Gabriela Guamán, egresada de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, condecorada en análisis y desarrollo de software con Java, mantenimiento y reparación de computadoras. Áreas de interés: desarrollo de software e Inteligencia Artificial. Provincia de Loja, Ciudad Loja, Ecuador, 2015.



John Martínez, egresado de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, condecorada en análisis y desarrollo de software con Java. Áreas de interés desarrollo de software e Inteligencia Artificial. Provincia de Loja, Ciudad Loja, Ecuador, 2015.



Luis Jácome, Ingeniero en Sistemas Informáticos y Computación en la Universidad Técnica Particular de Loja, Magíster en Telemática en la Universidad de Cuenca, estudiante de Maestría en Ingeniería Computacional y Sistemas Inteligentes de la Universidad del País Vasco, Área de interés: Desarrollo de Software e Inteligencia Artificial. Provincia de Loja, Ciudad Loja, Ecuador, 2015.

11.8. Anexo 8: Certificado de traducción del resumen



Lic. Jenny Sanmartín
DOCENTE DE FINE-TUNED ENGLISH

CERTIFICA:

Que el documento aquí compuesto está correctamente traducido del idioma español al idioma inglés del resumen para la tesis titulada: "SISTEMA MULTIAGENTE BASADO EN UN MODELO ONTOLÓGICO PARA LA BÚSQUEDA DE OBJETOS DE APRENDIZAJE", sin haber cambiado, aumentado o disminuido su sentido en ninguna línea o párrafo del mismo.

Loja, 27 de Julio del 2015


Jenny Sanmartín
DOCENTE



11.9. Anexo 9: Licencia



Sistema Multiagente basado en un modelo Ontológico para la Búsqueda de Objetos de Aprendizaje, by Martínez-John y Guamán-Gabriela, se distribuye bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivar 4.0 Internacional.