UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA

TESIS DE GRADO

ONTOLOGÍA EN EL MODELO ESPACIO VECTORIAL

POSTULANTE: **Univ. SIMON QUISPE COILA**
TUTOR: **Lic. GROVER ALEX ROGRIGUEZ RAMIREZ**
REVISOR: **M.Sc. ELIZABETH LILIAN GARCÍA ESCALANTE**

LA PAZ – BOLIVIA

Dedicado a:

Mis Padres, por haberme dado la vida, y por enseñarme a seguir adelante a pesar de los momentos adversos.

qc.saimon@gmail.com

AGRADECIMIENTOS

Esta tesis, si bien ha requerido de esfuerzo y mucha dedicación por parte de mi persona, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que a continuación citaré y muchas de las cuales han sido un soporte muy importante en diferentes momentos de mi vida.

Primero y antes que nada, dar gracias a **Dios**, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo este periodo de estudio.

Agradecer hoy y siempre a mis padres Manuel y Emiliana por su esfuerzo realizado para que yo pueda culminar mis estudios.

A la M.Sc. Elizabeth García Escalante, por escucharme, por aconsejarme, por guiar mis ideas, por su tiempo dedicado, por la paciencia, por la colaboración brindada durante toda la tesis.

A mi tutor Lic. Grover Rogriguez Ramirez.

Un agradecimiento especial al Ph.D. Guillermo Choque Aspiazu, si bien no participo directamente en el desarrollo de esta tesis, sus enseñanzas como docente en Inteligencia Artificial me mostraron el camino de la investigación científica.

A mis compañeros de carrera Lic. Carlos Calle Medina por brindarme su colaboración, ánimos, amistad desde el primer momento, por ser una persona ejemplo a seguir como profesional y padre de familia con la que puedo contar, al Lic. Roberto Carlos Mendoza por brindarme su apoyo y colaboración en todo momento, y a Ruben Chiara Valencia por estar ahí cuando uno lo necesitaba.

A la Asociación de Investigación en Software Inteligente (AISI), un espacio donde tuve la oportunidad de participar como investigador.

En general quisiera agradecer a todas y cada una de las personas que han vivido conmigo la realización de esta tesis, con sus altos y bajos y que no necesito nombrar porque tanto ellas como yo sabemos que desde los más profundo de mi corazón les agradezco el haberme brindado todo el apoyo, colaboración, ánimo y sobre todo cariño y amistad.

RESUMEN

Desde la creación de los ordenadores éstos han encontrado un amplio uso como herramientas para la creación, almacenamiento y el acceso efectivo a información en diferentes formatos digitales y los sistemas utilizados para buscar y acceder a esta información son los buscadores, buscadores que son desarrollados bajo un modelo de recuperación como el espacio vectorial que basa su búsqueda en una simple comparación léxica entre consulta y palabras que representan a los documentos, dejando de lado documentos que están relacionados semánticamente con la consulta. Una búsqueda semántica es una consulta en la que se tiene en cuenta el contexto, y por tanto el significado, de aquello por lo que se pregunta y no solamente las palabras de la consulta, los modelos de recuperación, deben disponer de medios para conocer el sentido exacto que tiene la palabra en la búsqueda, se admite que las búsquedas semánticas se basan en técnicas para extraer información mediante la utilización de ontologías, el uso de ontologías permite definir formalmente los dominios de interés con la suficiente riqueza expresiva. Con una búsqueda semántica que simula la comprensión de las palabras y, por ende, establece relaciones entre ellas, para realizar búsquedas de interés para el usuario aunque en los documentos devueltos no figuren las palabras o expresiones de la búsqueda. Las búsquedas semánticas son muy superiores a las basadas en palabras clave: uno encuentra documentos de interés que no encontraría buscando con palabras clave.

En este trabajo se incorpora una estructura ontologica en el modelo espacio vectorial en la búsqueda de documentos, con la estructura ontología la consulta es enriquecida semánticamente y partir de esta consulta enriquecida buscar y recuperar documentos que no son tomados en cuenta con una simple búsqueda sintáctica.

Palabras clave: Recuperación de información, modelo espacio vectorial, ontologías, methontology, proceso de desarrollo de ontologías.

Abstract

From the creation of the computers these have found an ample use as tools for the creation, storage and the effective access to information in different digital formats and the systems used to search and agreeing to this information are seekers, seekers that are trained under a model of recuperation like the vectorial space that your search in a simple lexical comparison between consultation and words that they represent the documents bases, brushing aside documents that are related semantic with the consultation. A semantic search is a consultation in which the context is had in account, and therefore significance, of that so that he asks himself and not only the words of the consultation, the models of recuperation, they should have means to know the exact sense that the word has on the prowl, it is admitted that the semantic searches are based on techniques to extract intervening information the utilization of ontologies, the use of ontologies it allows defining the controls of concern with enough expressive wealth formally. With a semantic search that simulates the understanding of the words and, as a consequence, establishes relations between them, in order to accomplish searches of concern for the user although in the return items they do not represent words or expressions of the search. The semantic searches are very superior to the based in words nail down: One finds documents of concern that it would not find searching with key words.

In this work incorporates him an ontological structure in the model vectorial space in the search of documents, with the structure ontology the consultation is enriched semantic and departing from this office enriched to search and recovering documents that are not taken in accord with a simple syntactic search.

Key words: Information retrieval, model vectorial space, ontologies, methontology, process of development of ontologies.

ÍNDICE GENERAL

ÍNDICE ESPECIFICO

CAPITULO 1: PRELIMINARES

CAPITULO 2: MARCO TEÓRICO

Pag.

CAPITULO 3: MARCO APLICATIVO

CAPITULO 4: CONCLUSIONES

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

LISTA DE ABREVIACIONES

API	APPLICATION PROGRAMMING INTERFACE
BIRM	BINARY INDEPENDENCE RETRIEVAL MODEL
CORBA	COMMON OBJECT REQUEST BROKER ARCHITECTURE
CQS	COMPETENCY QUESTIONS
CYC	ENCYCLOPEDIA
DAML+OIL	DARPA AGENT MARKUP LANGUAGE
DARPA	DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
DL	DESCRIPTION LOGIC
DTD	DOCUMENT TYPE DEFINITION
EJB	ENTERPRISE JAVABEANS
FACT	FAST CLASSIFICATION OF TERMINOLOGIES
HTML	HYPERTEXT MARKUP LANGUAGE
IDEF-0	ICAM(INTEGRATED COMPUTER-AIDED MANUFACTURING) DEFINITION METHOD ZERO
IEEE	INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS
JEE	JAVA PLATFORM, ENTERPRISE EDITION
JESS	JAVA EXPERT SYSTEM SHELL
LISP	LIST PROCESSING
NSF	NATIONAL SCIENCE FOUNDATION
OIL	ONTOLOGY INFERENCE LAYER
ORSD	ONTOLOGY REQUIREMENTS SPECIFICATION DOCUMENT
OWL	ONTOLOGY WEB LANGUAGE
RACER	RENAMED ABOX AND CONCEPT EXPRESSION REASONER
RDF	RESOURCE DESCRIPTION FRAMEWORK
RDQL	RDF DATA QUERY LANGUAGE
RMI	REMOTE METHOD INVOCATION
RQL	RDF QUERY LANGUAGE
RUP	RATIONAL UNIFIED PROCESS
SGML	STANDARD GENERALIZED MARKUP LANGUAGE
SHOE	SIMPLE HTML ONTOLOGY EXTENSIONS.
SOAP	SIMPLE OBJECT ACCESS PROTOCOL
SPARQL	SPARQL PROTOCOL AND RDF QUERY LANGUAGE
SQL	STRUCTURED QUERY LANGUAGE
TREC	TEST RETRIEVAL CONFERENCE
URL	UNIFORM RESOURCE IDENTIFIER
W3C	WORLD WIDE WEB CONSORTIUM
XML	EXTENSIBLE MARKUP LANGUAGE
XOL	XML-BASED ONTOLOGY-EXCHANGE LANGUAGE
XP	XTREME PROGRAMMING

PRELIMINARES

1. PRELIMINARES

1.1. INTRODUCCION

Desde la creación de los ordenadores éstos han encontrado un amplio uso como herramientas para la creación, almacenamiento y el acceso efectivo a información en diferentes formatos digitales. Desde los inicio se han creado en un sentido muy amplio, sistemas de información, que facilitan la búsqueda y recuperación a esta información a partir de los llamados modelos clásicos de recuperación de información: booleano, espacio vectorial y probalistico. Un tipo de información son los documentos (libros, revista, tesis, artículos) y los sistemas para almacenarlos, acceder a ellos y recuperarlos de forma efectiva han sido clasificados como *Sistemas de Recuperación de Información (SRI)*.

Estos modelos clásicos, al estimar la relevancia de un documento respecto a una pregunta, se basan en la comparación de los términos que aparecen en ambos. En el modelo booleano, esto es parte fundamental del modelo, ya que la existencia de una o varias de las palabras especificadas en la pregunta es condición necesaria para la recuperación de un documento. Con respecto al modelo probabilístico, la suposición de independencia condicional de la aparición de términos es la que equivale a la no consideración de posibles relaciones entre las palabras. Finalmente, el modelo espacio vectorial el valor de relevancia de un documento solamente se determina en función de los términos que la pregunta y el documento tiene en común; sin considerar que algunas palabras de los documentos puedan tener una similitud semántica con las palabras de la pregunta. En este sentido, este modelo también considera los términos como independientes entre sí.

En este trabajo se propone aplicar ontologías en el modelo espacio vectorial para una búsqueda semántica entre la pregunta y los documentos en la recuperación de información para mejorar la efectividad del resultado del modelo, en repositorios de documentos

1.2. ANTECEDENTES

La *Recuperación de Información (RI)* es el conjunto de tareas mediante las cuales un usuario localiza y accede a los recursos de información que son pertinentes a su necesidad de información [Tramullas, 2004]. La RI se define como el problema de la selección de información, depositada en un medio de almacenamiento, en respuesta a consultas realizadas por un usuario [López, 2006].

En términos generales la RI trata la investigación relacionada con sistemas que facilitan el almacenamiento, la recuperación y el mantenimiento de documentos[1]. Estos sistemas han sido clasificados como *Sistemas de Recuperación de Información (SRI)*. El diseño de un SRI se realiza bajo un modelo de recuperación, donde queda definido cómo se obtienen las representaciones de los documentos y de la consulta, la estrategia para evaluar la relevancia de un documento respecto a una consulta y los métodos para establecer la importancia (orden) de los documentos de salida [Villena, 1997]. Los tres modelos de recuperación comúnmente utilizados son: booleano, espacio vectorial y probabilístico [Martínez, 2002].

Formalmente un modelo de RI se define de la siguiente manera [Baeza, 1999]:

Definición 1 *Un modelo de recuperación de información es una cuaterna <D,Q,F,sim> en la que:*

- *D es un conjunto de vistas lógicas de los documentos.*
- *Q es un conjunto de vistas lógicas de las necesidades de información de los usuarios. Los elementos de Q se denominan "preguntas".*
- *F es un marco que permite modelar los documentos, las preguntas y las relaciones entre ellos.*
- *sim :* $D \times Q \rightarrow R$*, siendo R el conjunto de los números reales, es una función de ordenación que asocia un número real a una pregunta* $q_j \in Q$ *y un documento* $d_i \in D$*. Esta función define un orden sobre los documentos respecto a su relevancia (o similitud) a la pregunta* q_j*.*

En el *modelo booleano* las preguntas están especificadas por expresiones booleanas. Una pregunta q consiste en un conjunto de palabras claves que están conectadas mediante los operadores lógicos *"AND", "OR" y "NOT"*. En el proceso de recuperación, la función de ordenación, *sim*, devolverá un valor de uno para todos los documentos (y sólo para estos), que cumplan la expresión formulada en la pregunta, y un valor de cero para todos los demás. De esta forma si para un documento d_i y una

[1] De forma general, se emplea el término documento para hacer referencia a los entes de información tratados en un SRI [Bilihardt, 2003]

pregunta q el valor de $sim(d_i, q)$ es uno, entonces se estima que d_i es relevante para q y se añade d_i al conjunto de los resultados.

El modelo booleano es el modelo de RI más similar a la recuperación de datos. La especificación de las preguntas es exacta y refleja criterios de búsqueda precisos en lugar de una descripción ambigua de la información buscada. La comparación de la pregunta con los documentos se realiza de forma exacta y no a nivel conceptual o semántico. Eso también implica que los resultados son más susceptibles a posibles errores en la formulación de la pregunta. El resultado es un conjunto de aquellos documentos que cumplen la expresión lógica especificada, ya que se considera que éstos son los documentos relevantes. Por tanto, no se determinan distintos niveles de relevancia y no es posible establecer una ordenación entre los documentos recuperados [Bilihardt, 2003].

Las principales desventajas del modelo booleano son dos. En primer lugar, se trata la relevancia como un criterio binario. Eso, por un lado, impide la posibilidad de facilitar a los usuarios algún criterio adicional acerca de la utilidad de los distintos documentos recuperados, y, por otro lado, no tiene en cuenta aquellos documentos que cumplen los criterios de la pregunta de forma parcial. Estos últimos se consideran simplemente irrelevantes. Este comportamiento es debido al uso de pesos binarios para los términos en las representaciones de los documentos. Una única aparición de cierto término en un documento llega a ser accidental, pero si el mismo término aparece varias veces es más probable que refleje el contenido del documento. En segundo lugar, la traducción de una necesidad de información en una expresión booleana no es siempre fácil, en gran medida porque las expresiones son exactas mientras las necesidades de información muchas veces no lo son.

En el *modelo espacio vectorial*, tanto los documentos como las preguntas de los usuarios se representan mediante vectores en un espacio cuyo conjunto de vectores de base corresponde al conjunto de términos de indexación. Cada documento d_i de una colección D está representado mediante un vector $\vec{d}_i = (w_{i1}, ...w_{in})$ y, de igual forma, cada pregunta q está representada por un vector $q = (w_1, ..., w_n)$. Cada elemento $w_{ij}(w_j)$ refleja la importancia del término t_j en la descripción del documento d_i (la pregunta q). Si $w_{ij} = 0$ el término t_j no ha sido seleccionado para representar el contenido, mientras si $w_{ij} > 0$, el w_{ij} es el peso de t_j en la descripción del documento (pesos negativos no se utilizan) [Bilihardt, 2003].

La idea del modelo espacio vectorial es que la relevancia de un documento a una pregunta se estima a través de la correlación o de la similitud de sus vectores. El proceso de recuperación para una pregunta consiste en calcular las similitudes entre cada uno de los documentos de la colección (sus vectores) y el vector de la pregunta. Después se ordenan los documentos de mayor a menor similitud y se devuelve esta lista ordenada como resultado; es decir, se utiliza la técnica de ordenación.
Las ventajas del modelo espacio vectorial respecto al modelo booleano son: el uso de la relevancia como un criterio de valores continuos, que admite que documentos sean "parcialmente relevantes" y, por tanto, permite el uso de la técnica de ordenación. Otra ventaja del modelo es su gran flexibilidad que consiste en la posibilidad de utilizar distintos esquemas de pesos y distintas funciones de similitud. Como principal inconveniente hay que destacar que la estimación de la relevancia se basa, en su esencia, en una comparación léxica entre preguntas y documento. No se realiza una comparación semántica o conceptual y un documento es considerado más relevante si un mayor número de sus palabras coinciden con las de la pregunta.

El *modelo probabilístico* que se conoce también como BIR[2]. En el modelo BIR la suposición básica es que los términos de indexación están distribuidos de forma desigual entre los documentos relevantes e irrelevantes. Se considera que tanto los documentos como las preguntas están representados mediante conjuntos de términos de indexación; es decir, tanto un documento $\vec{d}_i$ como una pregunta q están representados mediante vectores binarios sobre el conjunto de términos de indexación, $\vec{d}_i = (w_{i1},...w_{in})$ y $\vec{q} = (w_1,...,w_n)$ siendo $w_{ij}, w_j \in \{0;1\}$ para todo j con $1 \le j \le n$.

Dada una pregunta q, un documento $\vec{d}_i$ debe ser recuperado si su probabilidad de pertenecer al conjunto de documentos relevantes para q es mayor que la probabilidad de pertenecer al conjunto de documentos irrelevantes. Con el objetivo de ordenar los documentos de una colección por su probabilidad de ser relevantes, esta regla de decisión da lugar a una función de discriminación que linealiza el ratio [Bilihardt, 2003].

Las limitaciones básicas del modelo probabilístico se resumen en tres puntos: Primero el modelo usa una representación binaria de los documentos y preguntas. No se tiene en cuenta la frecuencia de aparición de los términos en los documentos. Segundo el modelo requiere información sobre los conjuntos de documentos relevantes e irrelevantes para una pregunta. Esta se obtiene, sólo de forma parcial, de la interacción con los usuarios en la recuperación con realimentación sobre relevancia. No

[2] Del ingles Binary Independence Retrieval Model, modelo de recuperación de independencia binaria.

obstante, en muchos escenarios esta interacción no es posible o no es deseable. El tercer punto es la suposición de la independencia condicional de la aparición de los términos en los documentos. En el sentido semántico, esta suposición equivale a considerar que no existen relaciones entre términos, o, en otras palabras, la existencia de ciertos términos en un documento no indica la existencia de otros términos semánticamente relacionados.

A continuación se describen trabajos relacionados con la RI.

En la tesis de Mendoza Roberto "*Agentes móviles para la recuperación de información en bibliotecas digitales*" [Mendoza, 2009] [3], define un modelo de agentes móviles que realiza recuperación semántica de la información basada en descripción Dublín Core de un dominio específico, de acuerdo a la consulta del usuario en un sistema de biblioteca digital distribuido. Utiliza metadatos Dublín Core expresados en el modelo RDF (Resource Description Framework) para describir los recursos digitales de una biblioteca. Con la utilización de esta herramienta semántica mejora la precisión de la búsqueda respecto al modelo vectorial. Lo que no se logra en este trabajo es tener una relación conceptual semántica entre la consulta y los recursos digitales.

Aliaga Grover en su tesis de grado "*Modelo de Recuperación de Información para el Castellano*" [Aliaga, 2008] [4], tiene como objetivo facilitar y optimizar las búsquedas en lenguaje natural sobre colecciones de documentos en el idioma castellano mediante un modelo de RI para el idioma castellano, apoyado en el tratamiento del lenguaje natural y técnicas de RI. Para tal propósito, diseña herramientas para tratamiento lingüístico por separado para luego adecuarlas al modelo booleano y modelo vectorial obteniendo como resultado un modelo híbrido diseñado especialmente para el castellano. Con esta propuesta se logra obtener una precisión de la búsqueda aceptable ante la consulta de usuario.

En la tesis de grado de Alzaru Icaro, "*Alejandría Inteligente: Un experimento Web semántico*" [Alzaru, 2007] [5], pretende probar la utilidad de las herramientas semánticas en la RI y dar respuesta a las interrogante: ¿es posible mejorar los resultados de los buscadores actuales en Internet que se ayudan de métodos estadísticos y comparación de palabras para las operaciones de búsqueda? Para responder esta interrogante propone una arquitectura de utilización de algunas tecnologías semánticas en las

[3] Disponible en: http://bibliotecadigital.umsa.bo:8080/rddu/handle/123456789/983 [Consulta: 11/05/2010]
[4] Disponible en: http://bibliotecadigital.umsa.bo:8080/rddu/handle/123456789/852 [Consulta: 11/05/2010]
[5] Disponible en: http://webdelprofesor.ula.ve/ingenieria/jacinto/tesis/2007-feb-msc-icaro-alzuru.pdf [Consulta: 11/05/2010]

aplicaciones de manejo de información en general (usando Alejandría[6]como caso de prueba). Como resultado se logra una mejora en la recuperación de información obtenida a partir de las búsquedas que realizan los usuarios a través de la interfaz de la aplicación.

En el proyecto "*Sistema Multiagente basado en ontologías para optimizar la recuperación de Información*" realizado por [Vidal et. al., 2005][7], se presenta un prototipo que aplica técnicas emergentes en los campos de Web Semántica y Sistemas Multiagente con el objetivo de entregar resultados concretos que evitan pérdidas de tiempo en dispendiosos procesos de clasificación y análisis de la gran cantidad de información que usualmente entregan los buscadores tradicionales, resultados que si bien algunas veces logran satisfacer los requerimientos de búsqueda, en la mayoría de los casos no poseen la información que el usuario requiere. El resultado de este proyecto muestra que si es posible, en la Web actual, mediante procesos automáticos, recuperar información relevante evitando al usuario invaluables pérdidas de tiempo en procesos de selección y evaluación de documentos. Lo que no se logra es implementar un modelo de representación genérico de las consultas de usuario que permita desligar el sistema de filtrado del dominio modelado, y conseguir así una aplicación genérica que pueda ser usada en cualquier dominio, mejorar el proceso de optimización realimentando el sistema con información proveniente de los usuarios, mediante el uso de técnicas como la realimentación por relevancia que permitan tener en cuenta aspectos como los intereses particulares de los usuarios y el grado de satisfacción del usuario con el sistema y no implementa métodos de aprendizaje de los agentes de tal forma que mejoren su desempeño en la categorización de documentos y consultas a partir de las ejecuciones anteriores.

En el proyecto "Búsqueda semántica en bases documentales gubernamentales" [Teso et. al., 2007][8], se desarrollo un buscador que intenta mejorar y facilitar el acceso de la ciudadanía al Boletín Oficial del Principado de Asturias (BOPA). El problema que se intenta resolver es la barrera léxica entre el vocabulario del BOPA y el léxico de entrada al sistema de los usuarios y el objetivo de este proyecto es aplicar tecnologías semánticas (ontologías y tesauros) para la recuperación de documentos, para superar la barrera léxica entre el vocabulario de los documentos y el léxico de entrada al sistema de los usuarios, añadiendo conocimientos mediante la aplicación de ontologías y tesauros al motor de búsqueda para completar la falta de información del usuario al utilizar buscador. Con la aplicación de

[6] Alejandría es una aplicación informática creada con la idea de gestionar los diferentes tipos de documentos que se manejan en bibliotecas: Monografías, publicaciones seriadas, tesis, etc.
[7] Disponible en: http://www.inf.udec.cl/~revista/ediciones/edicion14/aperez.pdf [Consulta: 11/05/2010].
[8] Disponible en: http://www.unioviedo.es/teso/pdfs/technical-report.pdf [Consulta: 11/05/2010].

estas tecnologías permite al buscador enriquecer semánticamente la consulta del usuario y ejecutar búsquedas mas precisas.

Bilihardt Holger en su tesis doctoral "*Fusión de modelos vectoriales y contextúales para la recuperación de información*"[Bilihardt, 2003][9], tiene el objetivo de desarrollar y evaluar nuevas técnicas para la RI "ad hoc" que proporcionen capacidad de recuperación semántica, y mejoren la efectividad de la recuperación a través de la integración de varios modelos o técnicas. Realiza un análisis de los modelos de RI clásicos donde la recuperación y estimación de la relevancia de un documento para una pregunta se reduce básicamente a una función sobre las palabras que ambos tienen en común. Este hecho implica que los modelos no resuelven el denominado problema del vocabulario, el hecho de que se describan los mismos conceptos, temas y materias con diferentes palabras o la posibilidad de utilizar las mismas palabras para describir temas distintos. Debido a ello, la efectividad de estos modelos está intrínsecamente limitada. Una posible solución a este problema consiste en usar representaciones que reflejan el contenido semántico y conceptual de los documentos y preguntas, más que su contenido léxico. El aporte de su trabajo es un modelo para la recuperación de documentos de texto que tiene en cuenta las relaciones semánticas en el proceso de indexación. Con este modelo de recuperación mejora la efectividad de la búsqueda.

Martínez Francisco en su tesis doctoral "*Propuesta y desarrollo de un modelo para la evaluación de la recuperación de información en Internet*" [Martínez, 2002] [10], realiza un análisis de la efectividad de los SRI, los parámetros que utiliza en la determinación basada en la relevancia de la efectividad de los SRI, son la precisión y la exhaustividad de las operaciones de búsqueda, y logra determinar que los pares de valores de exhaustividad y precisión no ofrecen una medida exacta de la efectividad de la recuperación de información por si solos, y que es necesario aplicar otras medidas de valor simple (generalmente basadas, en estos pares de valores), para medir la efectividad de un SRI. Estas medidas de valor simple deben ser capaces de reflejar la efectividad de la RI.

1.3. PLANTEAMIENTO DEL PROBLEMA

El problema fundamental que surge en la recuperación de información utilizando los modelos clásicos: booleano, espacio vectorial y probalistico, es el problema del vocabulario. Este problema, en realidad, está relacionado con dos aspectos básicos del significado de las palabras: i) la sinonimia: palabras

[9] Disponible en: http://oa.upm.es/218/ [Consulta: 11/05/2010].
[10] Disponible en: http://www.cervantesvirtual.com/FichaObra.html?Ref=10010 [Consulta: 11/05/2010].

distintas que describen el mismo concepto, tema u objeto, y ii) la polisemia: palabras con varios significados distintos. En modelos de recuperacion que se basan en una estricta comparación del conjunto de términos especificados en la pregunta con el conjunto de términos que representa cada uno de los documentos (comparación sintáctica), el predominio en la pregunta de términos para los cuales existen sinónimos lleva a que algunos documentos relevantes no sean recuperados o estén más bajos en la ordenación, porque, aunque sí tratan los conceptos o temas buscados, emplean los términos sinónimos en vez de los términos especificados en la pregunta. El efecto, en sistemas sin ordenación, es que los resultados tendrán unos valores bajos de recuperación y, en sistemas con ordenación, que la precisión para los niveles mayores de recuperación disminuye. Por otro lado, la existencia de términos polisémicos en la pregunta, está relacionada con peores valores de precisión. En sistemas con ordenación, eso se traduce en peores valores de precisión en los niveles de recuperación más bajos; es decir, en que una mayor parte de los primeros documentos de la lista de resultados serán, en realidad, irrelevantes. Eso es debido a que si una pregunta contiene un mayor número de términos con un significado ambiguo entonces el sistema va a estimar muchos documentos irrelevantes como relevantes y a ponerlos al principio de la ordenación, simplemente porque contienen una o varias palabras en común con la pregunta, aunque el significado de estas palabras en los documentos no coincida con su significado intencionado en la pregunta.

En los modelos clásicos, éstos no tienen en cuenta las posibles relaciones entre las palabras de indexación y las palabras de la consulta de un usuario. Dicho de otra forma, estos modelos, al estimar la relevancia de un documento respecto a una pregunta, se basan en la comparación de los términos que aparecen en ambos. En el modelo booleano, esto es parte fundamental del modelo, ya que la existencia de una o varias de las palabras especificadas en la pregunta es condición necesaria para la recuperación de un documento. Con respecto al modelo probabilístico, la suposición de independencia condicional de la aparición de términos es la que equivale a la no consideración de posibles relaciones entre las palabras. Finalmente, el modelo espacio vectorial el valor de relevancia de un documento solamente se determina en función de los términos que la pregunta y el documento tiene en común; sin considerar que algunas palabras de los documentos tengan una similitud semántica con las palabras de la pregunta. En este sentido, este modelo también considera los términos como independientes entre sí.

Con esta similitud en la estimación de las relevancias, y aparte de la posibilidad de especificar la necesidad de información en forma de una expresión lógica en el modelo booleano y la posibilidad de aprovechar la información de un ciclo de búsqueda anterior en el modelo probabilístico, la principal diferencia práctica entre los tres modelos consiste en el cálculo de los pesos que se asignan a cada

término en la comparación de la pregunta con los documentos y es esta la razón por lo que utilizara el modelo espacio vectorial para la búsqueda semántica.

1.3.1. Problema general

El modelo espacio vectorial que basa la búsqueda en una simple comparación de términos en la recuperación de información, solo devuelve documentos que sean léxicamente iguales entre la consulta y los términos de indexación que representan a los documentos, dejando de lado documentos que están relacionados semánticamente con la consulta.

1.3.2. Problemas específicos

- El cálculo de similitud y relevancia de un documento se basa, en una comparación léxica entre términos de una consulta y los términos índice que representa al documento.

- La consulta debe especificarse usando el mismo conjunto de términos índice de los documentos, empleadas para indexar el documento.

1.4. OBJETIVOS

1.4.1. Objetivo General

Aplicar ontologías en el modelo espacio vectorial en la búsqueda y recuperación de documentos relacionados semánticamente con una consulta.

1.4.2 Objetivos Específicos

Los objetivos específicos planteados para el desarrollo de la propuesta son:

1. Relevar información sobre los componentes de un Sistema de Recuperación de Información.
2. Definir el modelo espacio vectorial incorporando una estructura ontologica en el proceso de recuperación de información.
3. Modelar conceptualmente la búsqueda ontologica en el modelo espacio vectorial para la recuperación de información.

4. Desarrollar un prototipo software Sistema de Recuperación de Información en base al modelo espacio vectorial, que interactué con la estructura ontológica y la consulta del usuario para una búsqueda semántica en la recuperación de información.
5. Interpretar resultados.

1.5. HIPOTESIS

"Se sustenta que con la aplicación de ontologías en la búsqueda de información en el modelo espacio vectorial, es posible recuperar información semánticamente relacionada de acuerdo a la consulta del usuario con un grado de precisión[11] superior respecto a una búsqueda sintáctica".

Donde las variables son: **Variable independiente:** ontología en la búsqueda de información.
Variable dependiente: recuperación de información semánticamente relacionada.

1.6. JUSTIFICACION

1.6.1. Científica

En los últimos años se esta aplicando técnicas, herramientas en el desarrollo de sistemas de acceso a la información con objeto de mejorarlos. En concreto, métodos, conceptos y técnicas de Inteligencia Artificial están siendo aplicados en los procesos de obtención de información dando lugar a la aparición de nuevas disciplinas como la Web semántica, minería de datos, recuperación de información en Internet, etc. Por tanto, el estudio y desarrollo de nuevas técnicas para la RI basada en ontologías, se muestra como una línea de investigación a seguir por otros investigadores.

1.6.2. Económica

Este trabajo se encuentra justificado de manera económica, por la necesidad que tienen los usuarios finales de contar con SRI donde exista diferentes entes de información en la cual encontrar información precisa y rápida ante consultas, sea menor en complejidad que sus similares, logrando menores costos en tiempos de búsqueda y acceso a la información.

[11] Precisión, es la proporción entre documentos relevantes recuperados y documentos recuperados.

1.6.3. Social

La proliferación de unidades y fuentes de información, tanto en el ámbito científico, profesional e incluso domestico y noticias, pone de manifiesto la importancia que la sociedad tiene de contar con SRI que respondan a sus necesidades de información sobre temas que son de su interés. Contar con ontológias que represente la información almacenada y este incorporado en un modelo de recuperación, permitirá que los usuarios finales de estos sistemas obtengan información más eficiente.

1.6.4. Técnica

Este trabajo plantea el modelado de una estructura ontológica acoplado al modelo espacio vectorial que enriquezca de manera semántica la necesidad de información usuarios finales. Al final de este trabajo se contara con un modelo de sistema de RI que puede ser aplicado en lugares que cuenten con información en formato texto a ser consultada.

1.7. METODOLOGIA

Para este trabajo se utiliza la metodología de investigación científica deductiva, siguiendo los lineamientos establecidos por Mario Bunge (1997), de la siguiente manera:

a. Observación. Se realiza un relevamiento y teorización de las falencias en la búsqueda y recuperación de información en los modelos clásicos: booleano, espacio vectorial y probabilistico.
b. Planteo de la hipótesis. Se sustenta que con la aplicación de ontologías en la búsqueda de información en el modelo espacio vectorial, es posible recuperar información semánticamente relacionada de acuerdo a la consulta del usuario con un grado de precisión superior respecto a una búsqueda sintáctica
c. Diseño de la aplicación. Se procederá a definir el modelo espacio vectorial incorporando la estructura ontologica y describir un modelo conceptual de búsqueda semántica en el modelo espacio vectorial para la recuperación de información.
d. Casos de prueba. Se mostrara el prototipo de buscador sobre el que se testeara la precisión de la aplicación con un conjunto de casos de prueba. Para la evaluación de resultados se seleccionarán un grupo de consultas que abarquen un amplio conjunto de tópicos y escenarios. La información recuperada será contrastada con las búsquedas tradicionales por palabra clave considerando medidas de evaluación precisión.
e. Conclusiones. Se dará un informe final de la investigación relacionada con las ontologías y la RI.

1.8. HERRAMIENTAS

El prototipo software estará guiado por las directrices señaladas por ICONIX. ICONIX es un proceso intermedio entre XP[12] y el RUP[13], siendo el primero muy útil para software's pequeños y, el segundo, muy útil para software's industriales; por tanto, ICONIX es una mezcla entre la agilidad de XP y la robustez de RUP

Para el desarrollo de la ontología se utilizara el proceso de desarrollo de ontología propuesto en el framework de la metodología Methontology por Gómez [Gómez et. al., 2004]. Como herramienta de diseño de la ontología se utilizara Protege.
Para realizar las consultas de expresiones semánticas se utilizará Jena versión 2.6 que es un API[14] para gestionar ontologias en lenguaje de programación Java, Jena es un framework desarrollado por laboratorios de Hewlett Packard para manipular y consultar ontologias desde una aplicación Java, Jena incluye el motor de consulta de SPARQL[15] en el paquete ARQ. SPARQL es una recomendación del Consorcio de la World Wide Web para crear un lenguaje de consulta dentro de la Web semántica.

1.9. LIMITES Y ALCANCES

Siendo el campo de investigación de RI extensa respecto a modelos de recuperación y el medio en que es almacenada la información, este trabajo tendrá las siguientes limitaciones:

- No se toma en cuenta información almacenada en videos, audio e imágenes en el proceso de recuperación de infamación.

- No se realiza un análisis de la indexación automática y semántica de información para el modelo recuperación propuesto.

- No se utilizara una función de similitud semántica para ordenar los resultados.

[12] Del inglés Xtreme Programming, Programación Extrema.
[13] Del inglés Rational Unified Process, Proceso de Desarrollo Unificado.
[14] Del inglés Application Programming Interface , Interfaz de Programación de Aplicaciones.
[15] Acrónimo recursivo del inglés SPARQL Protocol and RDF Query Language – Protocolo SPARQL y Lenguaje de Consulta RDF.

Los alcances de este trabajo son:

- El modelo de recuperación a estudiar y modelar es el espacio vectorial, el tipo de información a tomar en cuentas será, información almacenada en documentos texto.

- Los documentos y el dominio de la ontología tomados en cuenta para el diseño del prototipo software para la interpretación de resultados aran referencia a la temática de agentes y sistemas multiagentes.

- El índice invertido de la base documental estará formado por las palabras claves de un artículo. Por lo tanto la indexación será semi automática a partir de las palabras claves de un artículo.

- La función de similitud para ordenar los resultados será el producto escalar.

- El prototipo estará diseñado con los procesos: indexación de documentos, evaluación del modelo y la búsqueda en si. La consulta y búsqueda semántica a procesar en la recuperación de información se realizara en base a la temática de los artículos, el resultado de una consulta se mostrara en una lista documentos.

1.10. APORTES

El aporte de la investigación es:

Un Modelo conceptual de búsqueda semántica con el modelo espacio vectorial.

Una estructura ontológica acoplado al modelo espacio vectorial para la recuperación de información.

MARCO TEÓRICO

2. MARCO TEÓRICO

2.1. RECUPERACIÓN DE INFORMACIÓN

La RI[31] es el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes a una necesidad de información [Tramullas, 2004]. La RI se define como el problema de la selección de información, depositada en un medio de almacenamiento, en respuesta a consultas realizadas por un usuario [López, 2006].

En términos generales la RI trata la investigación relacionada con sistemas que facilitan el almacenamiento, la recuperación y el mantenimiento de entes de información. Estos sistemas han sido clasificados como SRI[32]. El punto central de cualquier SRI es proporcionar a los usuarios un acceso fácil y eficiente a la información que les interesa. En realidad todos los aspectos tratados y controlados en un SRI giran alrededor de este problema. La representación de los entes de información, su organización y mantenimiento, y las formas y técnicas de acceso están fuertemente relacionados con la búsqueda de información y deben ser solucionados de una forma que permita que esta búsqueda se realice de forma eficiente y efectiva [Bilihardt, 2003].

2.1.1. Recuperación de Información en comparación con recuperación de datos

En muchos aspectos la recuperación de información es similar a la recuperación de datos tal como se realiza a través de los SGBD[33]. El primer parecido es el objetivo de los sistemas: la satisfacción de las necesidades de información de usuarios. En ambos paradigmas los sistemas almacenan entes de información y facilitan el acceso a ellas. También en ambos paradigmas los usuarios especifican, a través de algún interfaz, la información que les interesa o que están buscando, y la tarea del sistema consiste en encontrar esta información [Martínez, 2002].

[31] De las siglas Recuperación de Información.
[32] De las siglas Sistemas de Recuperación de Información.
[33] De las siglas Sistema Gestor de Base de Datos.

Las principales diferencias entre ambos modos de recuperación resultan de los distintos tipos de entes de información tratados y la forma en que la se recupera la información. En la recuperación de datos la fuente de información son datos estructurados o hechos con una semántica claramente definida a través de elementos como atributos, entidades y relaciones. Al contrario, en la RI los entes de información normalmente no tienen este nivel de estructuración y su semántica es ambigua. Baeza-Yates expone las diferencias entre ambos tipos de recuperación destacando que los datos se estructuran en tablas y árboles para recuperar exactamente lo que se quiere, los otros entes de información no poseen una estructura clara y no resulta fácil crearla. En la recuperación de datos, se utiliza un lenguaje de consulta como SQL[34], cada pregunta define claramente las condiciones suficientes y necesarias que tienen que cumplir los registros de una base de datos para que sean devueltos como resultado. Por el contrario, en la recuperación de información, las preguntas normalmente son textos en lenguaje natural que describen de forma subjetiva la necesidad de información [Baeza, 1999] [Bilihardt, 2003].

La tabla 2.1 sintetiza las diferencias fundamentales existentes entre la recuperación de de datos y RI

Tabla 2.1: Comparación entre recuperación de datos y RI.
Fuente: [Baeza, 1999]

	Recuperación de datos	**Recuperación de información**
Entes de información	Totalmente estructurados con semántica bien definida (registros).	No estructurados con semántica ambigua (documentos).
Lenguaje de preguntas	Artificial.	Natural.
Especificación de preguntas	Completa; Definición de las condiciones de la búsqueda.	Incompleta; Especificación ambigua de los temas de interés.
Reacción a errores en la pregunta	Susceptible.	No susceptible
Comparación	Exacta no interpretativa.	Parcial; buscando mayor similitud; interpretativa.
Inferencia	Deducción.	Inducción.
Modelo	Determinista.	Probabilística.
Clasificación	Monotética.	Politética.
Resultados	Registros que cumplan la especificación de la pregunta.	Documentos relevantes, que traten los temas y materias especificadas.
Presentación de resultados	Listas de registros.	Ordenación de los documentos por relevancia ("ranking")
Exactitud de los resultados	Se recuperan exactamente todos y sólo aquellos registros que cumplan la especificación.	No se recuperan todo el documento relevante y no necesariamente solo éstos.
Posibilidad de mejorar los resultados	No.	Realimentación sobre relevancia.

[34] De las siglas en ingles Structured Query Language, Lenguaje Estructurado de consulta.

2.1.2. Tareas estudiadas en la recuperación de información

La investigación en el campo de RI ha ido evolucionando. Eso no sólo se refiere a la mejora de los modelos, métodos y técnicas empleadas, sino también a las tareas estudiadas. A continuación se presenta una lista de las tareas, escenarios y enfoques principales que se han estudiado y se están estudiando en el campo de la RI y se describe cada una de ellas de forma breve. El objetivo no es ofrecer una visión rigurosa y completa de los distintos aspectos investigados, sino dar una idea general de la polivalencia de esta disciplina de investigación. Por tanto, la lista no pretende ser completa y las descripciones son breves y superficiales [Bilihardt, 2003].

Desde el punto de vista de la tarea que un SRI realiza se diferencian los sistemas según los siguientes puntos [Bilihardt, 2003]:

a. Búsqueda de documentos o recuperación ("ad hoc"): la búsqueda de documentos es el escenario clásico de la RI. En ella, los sistemas almacenan información acerca de un conjunto de documentos, tradicionalmente de texto. Los usuarios especifican las necesidades de información en forma de preguntas y la tarea del sistema consiste en encontrar aquellos documentos que satisfagan estas necesidades; es decir, los documentos que son relevantes con respecto a las preguntas. Las características de estos sistemas son que las colecciones de documentos son relativamente estáticas y las preguntas de los usuarios son variables. Sistemas de búsqueda de artículos o referencias bibliográficas son ejemplos de este tipo de tarea.

b. Filtrado de información ("information filtering"): el filtrado de documentos se refiere a un escenario en el que las necesidades de información definen un interés fijo de los usuarios en posibles documentos que aparecen de forma continua. Para reflejar el carácter estático, estos intereses se denominan normalmente perfiles de usuarios, en vez de preguntas. Los sistemas de filtrado reciben un flujo continuo de nuevos documentos y la tarea consiste en filtrar aquellos documentos que son relevantes respecto a los perfiles de usuarios; es decir, los sistemas tienen que decidir para cada documento si éste llega a interesar a los distintos usuarios o no. Por tanto, al contrario de lo que ocurre en la recuperación "ad hoc", las necesidades de información son estáticas mientras que los documentos son variables. Un ejemplo para este tipo de tarea es el filtrado de noticias.

c. Contestación a preguntas: esta tarea se aleja del enfoque clásico de RI al tratar de encontrar respuestas a preguntas concretas, en vez de devolver uno o varios documentos relevantes. Las preguntas especifican hechos y, normalmente; requieren respuestas breves y los sistemas deben encontrar estas respuestas en un conjunto de documentos de texto. Por tanto, la tarea no consiste

sólo en encontrar documentos relevantes, que contengan las respuestas, sino también en extraer de ellos la información buscada.

d. Clasificación y agrupación de documentos: la clasificación y la agrupación ("clus-tering") de documentos no están directamente relacionados con una tarea de usuario. La clasificación se usa primordialmente para asignar los documentos a una serie de conceptos que muchas veces están ordenados en jerarquías. De este modo se crean estructuras navegables, en las que los usuarios recupera y vea conjuntos de documentos que traten conceptos o temas predefinidos.

2.1.3. Modelos de Recuperación de Información (RI)

Formalmente se define un modelo de RI de la siguiente manera [Baeza, 1999]:

Definición 1 *Un modelo de recuperación de información es una cuaterna <D,Q,F,sim> en la que:*

- *D es un conjunto de vistas lógicas de los documentos.*
- *Q es un conjunto de vistas lógicas de las necesidades de información de los usuarios. Los elementos de Q se denominan "preguntas".*
- *F es un marco que permite modelar los documentos, las preguntas y las relaciones entre ellos.*
- *sim :* $D \times Q \rightarrow R$ *, siendo R el conjunto de los números reales, es una función de ordenación que asocia un número real a una pregunta* $q_j \in Q$ *y un documento* $d_i \in D$ *. Esta función define un orden sobre los documentos respecto a su relevancia (o similitud) a la pregunta* q_j*.*

Definición 2 *Sea* $D = \{d_1,...,d_m\}$ *un conjunto de vistas lógicas de documentos y Q un conjunto de vistas lógicas de preguntas. Además, sea* $T = \{t_1,...,t_n\}$ *el conjunto de términos de indexación para el conjunto D y R el conjunto de números reales.*

El conjunto D se denomina colección de documentos, m es el número de documentos en la colección y n es el número de términos de indexación utilizados en las vistas lógicas de D.

Por razones de formalización, se representa cada documento $d_i \in D$ *y cada pregunta* $q \in Q$ *mediante un vector sobre el conjunto de los términos de indexación,* $\vec{dtf}_i = (tf_{i1},...,tf_{in})$ *y* $\vec{qtf} = (tf_1,...,tf_n)$*, respectivamente, siendo* tf_{ij} *y* $tf_i \in R$ *las frecuencias de aparición del término* t_j *en el documento* d_i *y en la pregunta q. Estos vectores se denominan vectores de frecuencias de términos o simplemente vectores de frecuencias.*

El diseño de un SRI se realiza bajo un modelo, donde queda definido cómo se obtienen las representaciones de los documentos y de la consulta, la estrategia para evaluar la relevancia de un documento respecto a una consulta y los métodos para establecer la importancia (orden) de los documentos de salida [Villena, 1997].

Existen varias propuestas de clasificación de modelos, una de las síntesis más completas la realiza Dominich en cinco grupos (tabla 2.2) [Dominich, 2000]:

Tabla 2.2: Clasificación de los modelos de recuperación de información.
Fuente: [Dominich, 2000]

Modelo	Descripción
Modelos clásicos	Incluye los tres más citados: booleano, espacio vectorial y probabilístico.
Modelos alternativos	Están basados en la Lógica Fuzzy
Modelos lógicos	Basados en la Lógica Formal. La recuperación de información es un proceso inferencial.
Modelos basados en la interactividad	Incluyen posibilidades de expansión del alcance de la búsqueda y hacen uso de retroalimentación por la relevancia de los documentos recuperados.
Modelos basados en la Inteligencia Artificial	Bases de conocimiento, redes neuronales, algoritmos genéticos y procesamiento del lenguaje natural.

Los tres modelos comúnmente citados son: booleano, espacio vectorial y probabilística [Martínez, 2002]. Los tres modelos corresponden a tres paradigmas diferentes. En el modelo booleano, los documentos y preguntas son representados mediante conjuntos de palabras y el proceso de recuperación consiste en realizar operaciones sobre estos conjuntos. Por tanto, el modelo emplea la teoría de conjuntos. El modelo espacio vectorial es un modelo algebraico, utiliza vectores como forma de representación y realiza operaciones algebraicas sobre estos vectores. Por último, en el modelo probabilístico, el proceso de recuperación está basado en la teoría de probabilidades [Bilihardt, 2003]. A continuación se presenta un descripción de los modelo booleano y probabilistico.

a) Modelo booleano: el modelo booleano es uno de los modelos más simples e intuitivos. El modelo booleano está basado en la teoría de conjuntos. Se considera que los documentos están representados mediante conjuntos de palabras claves que son sub conjuntos del conjunto de términos de indexación. Utilizando la definición 2 de la página 19, estos conjuntos se representan mediante vectores binarios n-dimensionales. Cada documento $d_i \in D$ se representa con un vector $\vec{d}_i = (w_{i1},...w_{in})$ sobre el conjunto de términos de indexación *T*.

Por otra parte, las preguntas están especificadas por expresiones booleanas. Una pregunta q consiste en un conjunto de palabras claves que están conectadas mediante los operadores lógicos *"AND"*, *"OR"* y *"NOT"*. En el proceso de recuperación, la función de ordenación sim, devolverá un valor de *1* para todos los documentos (y sólo para estos), que cumplan la expresión formulada en la pregunta, y un valor de *0* para todos los demás. De esta forma si para un documento d_i y una pregunta q el valor de $sim(d_i, q)$ es *1*, entonces se estima que d_i di es relevante para q y se añade d_i al conjunto de los resultados.

El modelo booleano es el modelo de RI más similar a la recuperación de datos. La especificación de las preguntas es exacta y refleja criterios de búsqueda precisos en lugar de una descripción ambigua de la información buscada. La comparación de la pregunta con los documentos se realiza de forma exacta y no a nivel conceptual o semántico. Eso también implica que los resultados son más susceptibles a posibles errores en la formulación de la pregunta. El resultado es un conjunto de aquellos documentos que cumplen la expresión lógica especificada, ya que se considera que éstos son los documentos relevantes. Por tanto, no se determinan distintos niveles de relevancia y no es posible establecer una ordenación entre los documentos recuperados. Las principales desventajas de este modelo son dos. En primer lugar, se trata la relevancia como un criterio binario. Eso, por un lado, impide la posibilidad de facilitar a los usuarios algún criterio adicional acerca de la utilidad de los distintos documentos recuperados, y, por otro lado, no tiene en cuenta aquellos documentos que cumplen los criterios de la pregunta de forma parcial. Estos últimos se consideran simplemente irrelevantes. En segundo lugar, la traducción de una necesidad de información en una expresión booleana no es siempre fácil, en gran medida porque las expresiones son exactas mientras las necesidades de información muchas veces no lo son. La principal ventaja del modelo booleano es la sencillez y, usando la estructura de ficheros invertidos, se implementa de forma eficiente.

b) *Modelo probabilístico:* este modelo, que se conoce también como BIR[35]. En el modelo BIR la suposición básica es que los términos de indexación están distribuidos de forma desigual entre los documentos relevantes e irrelevantes. Se considera que tanto los documentos como las preguntas están representados mediante conjuntos de términos de indexación; es decir, tanto un documento d_i como una pregunta q están representados mediante vectores binarios sobre el conjunto de términos de indexación, $\vec{d}_i = (w_{i1}, ...w_{in})$ y $\vec{q} = (w_1, ..., w_n)$ siendo $w_{ij}, w_j \in \{0;1\}$ para todo j con $1 \le j \le n$.

[35] Del ingles Binary Independence Retrieval Model", modelo de recuperación de independencia binaria.

Dada una pregunta q, un documento d_i debe ser recuperado si su probabilidad de pertenecer al conjunto de documentos relevantes para q es mayor que la probabilidad de pertenecer al conjunto de documentos irrelevantes; es decir, si $P(R \mid d_i) > P(\overline{R} \mid d_i)$. Con el objetivo de ordenar los documentos de una colección por su probabilidad de ser relevantes, esta regla de decisión da lugar a una función de discriminación que linealiza la ratio entre $P(R \mid d_i)$ y $P(\overline{R} \mid d_i)$:

$$s(d_i) = log \frac{P(R|d_i)}{P(\overline{R}|d_i)} \qquad (2.1)$$

La ecuación 2.1 es la transformación logística de $P(R \mid d_i)$. Como esta transformación es monótona, se obtiene una función de ordenación s, que cumple con el PRP, es decir, permite ordenar los documentos de mayor a menor probabilidad de ser relevantes para q.

Las limitaciones básicas del modelo probabilística son: no se tiene en cuenta la frecuencia de aparición de los términos en los documentos, otra limitación es la suposición de la independencia condicional de la aparición de los términos en los documentos. En el sentido semántico, esta suposición equivale a considerar que no existen relaciones entre términos, o, en otras palabras, la existencia de ciertos términos en un documento no indica la existencia de otros términos semánticamente relacionados.

Los modelos booleano y probabilistico no son objeto del análisis que se presentas en esta tesis, por lo que se ha optado por una descripción breve y general de estos modelos. No obstante el modelo espacio vectorial que es el tema de investigación en este trabajo, es analizado y descrito a continuación.

2.2. MODELO DE RECUPERACIÓN ESPACIO VECTORIAL

Saltón fue el primero en proponer los SRI basados en Espacio Vectorial a finales de los 60 dentro del marco del proyecto SMART [Saltón, 1983]. Este modelo es aplicable, tanto cuando la indexación de los documentos es manual, mediante la asignación de palabras claves, como cuando es automática por texto completo. Este modelo permite la definición y el uso de pesos para los términos de indexación e implementa la técnica de ordenación por relevancia. Con ello, el resultado de un proceso de recuperación es una lista de documentos ordenados de mayor a menor relevancia respecto a la pregunta.

El objetivo de los SRI es, dada una colección de documentos y una consulta formulada por un usuario en un cierto momento, proporcionar el subconjunto de documentos que es relevante para la consulta del usuario. Para alcanzar este objetivo existen dos procesos muy distintos e importantes: la introducción de la información (preparación de las estructuras de búsqueda), y la búsqueda en sí. El

primer proceso se llama "indexación", o "indización", este nombre viene del resultado de obtener una serie de índices de búsqueda. El segundo proceso es el de búsqueda, que se apoyará en los índices generados en el primer proceso [López, 2006].

2.2.1. Proceso de indexación en el modelo espacio vectorial

En el modelo espacio vectorial, cada documento es representado mediante un vector de *n* elementos, siendo *n* igual al número de términos de indexación que existen en el documento. Existe un vector para cada documento, y, en cada vector, un elemento para cada término o palabra susceptible de aparecer en el documento. Cada uno de esos elementos es ocupado con un valor numérico, dado que una palabra dada es más o menos significativa, este valor se conoce con el nombre de peso del término en el documento, este vector para cada documento es la representación de términos índice que identifica a un documento, estos términos son extraídos con el proceso de la figura 2.1, este proceso se divide en los siguiente sub procesos [Becerra, 2008] [Tula y Medeot, 2007]:

a) Análisis de texto: en esta etapa se eliminan y normalizan los términos del documento a ser indexado, se descartan las palabras que constituyen palabras vacías "stopwords" (la, las, y, los, en) y se eliminan los símbolos ortográficos.

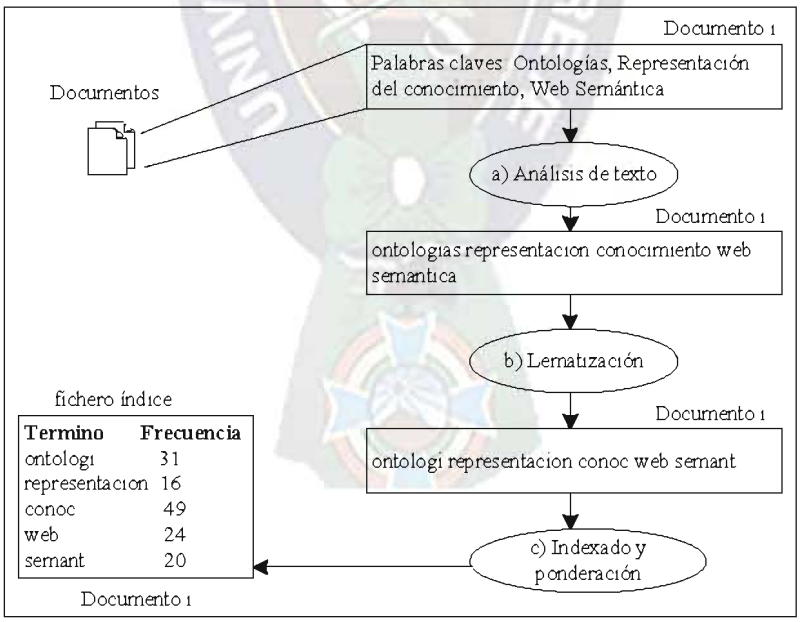

Figura 2.1: Proceso de indexación.
Fuente: Modificado [Becerra, 2008]

b) Lematización: se reducen palabras a su raíz gramatical (stemming) para evitar redundancia en el índice y lograr optimización del mismo.

c) Indexado y ponderación: se construye un fichero índice de palabras, con punteros a los documentos en que dicha palabra tiene una o más ocurrencias, esta ocurrencia es el peso de cada término en un documento.

El fichero índice es una estructura de almacenamiento que facilita la búsqueda de palabras en documentos. Los aspectos más importantes a tener en cuenta al decidir la estructura de datos para la fichero índice son: capacidad de representar el contenido de los documentos, es decir, los conceptos o términos y sus relaciones con los documentos, la estructura de datos que se emplea debe ser capaz de reflejar toda la información de los documentos que es necesaria para realizar el proceso de recuperación, las estructuras de almacenamiento deben ser eficientes en el aprovechamiento de la memoria secundaria y principal y rapidez de acceso a la información. Algunas de las propuestas de estructuras de almacenamiento basados en índices de palabras son: i) almacenamiento basado en vectores, ii) almacenamiento basado en listas y iii) ficheros invertidos [Bilihardt, 2003].

i) Almacenamiento basado en vectores: la estructura consta de tres ficheros: 1) un diccionario, 2) un fichero de los documentos, y 3) un fichero de índice. El diccionario contiene el conjunto de términos de indexación. Cada término tiene asignado un identificador único. Este identificador corresponde a la posición del término dentro de los vectores que representan los documentos. El diccionario también se incluye información sobre los términos que es necesaria o útil en el proceso de recuperación. En la figura 2.2 se aprecia un ejemplo para una colección de dos documentos.

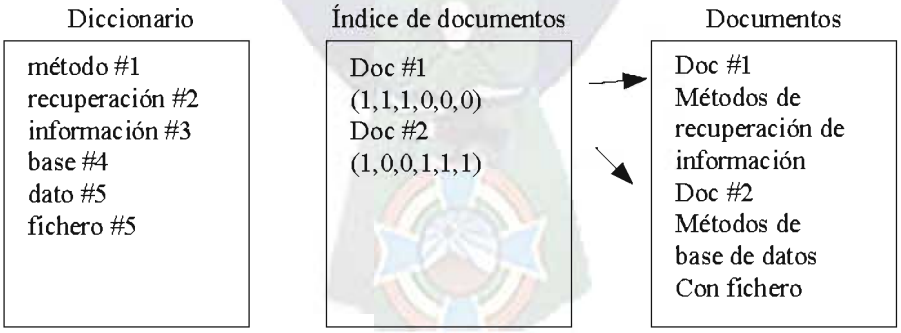

Figura 2.2: Estructura almacenamiento basado en vectores.
Fuente: [Bilihardt, 2003].

ii) Almacenamiento basado en listas: en la estructura basada en listas se almacenan los conjuntos que representan los documentos en forma de una lista ordenada de pares <identificador del término, peso>. De esta forma, se resuelve el almacenamiento más eficiente de vectores dispersos con pocos elementos distintos de cero. La figura 2.3 presenta un ejemplo.

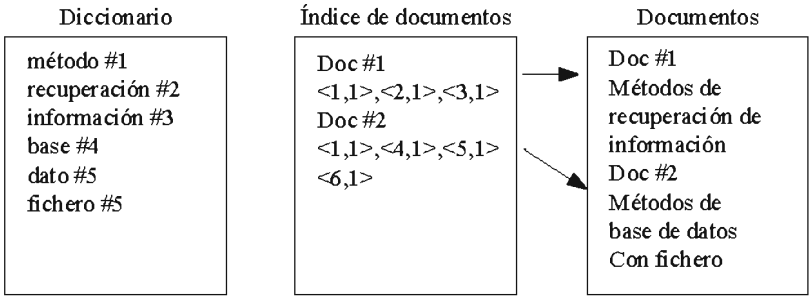

Figura 2.3: Estructura basada en listas.
Fuente: [Bilihardt, 2003].

iii) Ficheros invertidos: también llamado índice invertido, un fichero invertido está compuesto por tres ficheros: 1) un diccionario, 2) un fichero de documentos, y 3) un fichero con las listas invertidas (el índice). La figura 2.4 presenta la estructura de un fichero invertido.

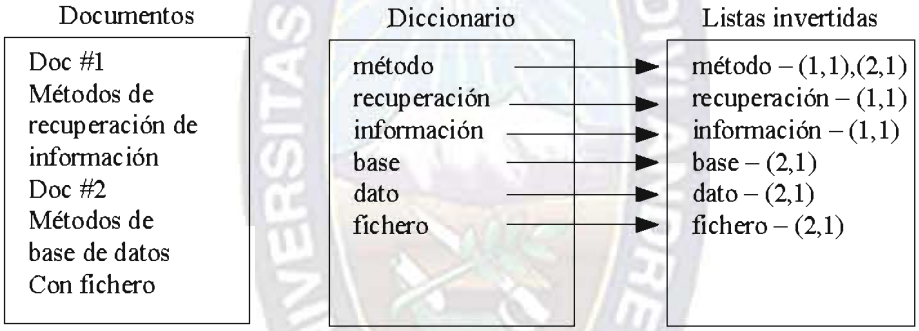

Figura 2.4: Estructura basada en ficheros invertidos.
Fuente: [Bilihardt, 2003].

La idea principal del índice invertido consiste en el almacenamiento de la información a través de los términos. Es decir, en lugar de almacenar los términos asignados a cada uno de los documentos como en los casos anteriores, el fichero de índices está estructurado por los términos y para cada término se almacena una lista de los documentos a los que pertenece. En la lista se guarda, aparte de los identificadores de los documentos, los pesos que el término tiene en sus representaciones.

Una vez generado el vector del documento, se organiza esta información en una asociación termino-documento, utilizando las estructuras de datos necesarias para generar el conjunto completo de índices inversos de los documentos.

Por otro lado, formalmente un documento se considera como un vector que expresa la relación del documento con cada término mediante $\vec{d}_i = (w_{i1},...w_{in})$. Es decir, el vector identifica en qué grado w_{in}

satisface al documento d_i. Por lo tanto, un documento se identificará mediante una colección de términos $t_1, t_2, \ldots t_m$, asignando un peso, o importancia, del término. Por tanto, la relación (base documental) documentos, termino y peso se representa como una ordenación, o matriz, de términos donde cada fila de la matriz representa un documento y cada columna representa la asignación de un término específico a los documentos en cuestión (véase figura 2.5) [Zazo et. al., 2002].

$$\begin{array}{c} \\ d_1 \\ \ldots \\ d_i \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & \ldots\ldots & & t_m \end{array} \\ \begin{bmatrix} w_{11} & \ldots & w_{1m} \\ \ldots & \ldots & \ldots & \ldots \\ w_{1i} & \ldots & w_{im} \end{bmatrix} \end{array}$$

Figura 2.5: Representación matemática de una base documental.
Fuente: Modificación [Luque, 2005]

2.2.2. Proceso de búsqueda en el modelo espacio vectorial

En el modelo del espacio vectorial, el proceso de búsqueda se realiza a partir de la consulta del usuario, cada consulta es representada mediante un vector de las mismas características que las de los documentos (variando los valores numéricos de cada elemento en función de las palabras que forman parte de la consulta). Esto permite calcular fácilmente una función de similaridad dada entre el vector de una consulta y los de cada uno de los documentos. El resultado de dicho cálculo mide la semejanza entre la consulta y cada uno de los documentos, de manera que, aquéllos que, en teoría, se ajustan más a la consulta formulada, producen un índice más alto de similaridad. Naturalmente, se asume que la consulta se formula en lenguaje natural y el resultado de la consulta consiste en una lista de documentos ordenada en orden decreciente en función de su similaridad con la consulta. Este proceso se muestra en la figura 2.6, donde existen sub procesos que se describe a continuación [Tula y Medeot, 2007] [Becerra, 2008]:

a) Vector consulta: una consulta esta expresada en lenguaje natural, y se ve como un documento cuyo contenido son las palabras de la consulta. Este apunte es muy importante porque permite utilizar operaciones de texto para eliminar las palabras vacías y lematizar el contenido de la consulta. Y por ultimo se genera el correspondiente vector asociado a la consulta de la forma:

$$\vec{q} = (w(t_1, q), \ldots, w(t_k, q))$$

donde k es el numero de términos de indexación, y el peso $w(t_k, q)$ indica la relevancia, manual o automática, del termino de indexación t_k en la consulta q. La consulta no tiene por que tener todos los

términos de indexación $t_1,...,t_k$. La razón de generar el vector con dimensión k es realizar un proceso de normalización de vectores, de forma que todos los vectores tengan el mismo modulo, para definir una relación de semejanza donde los términos estén equilibrados y sean comparables.

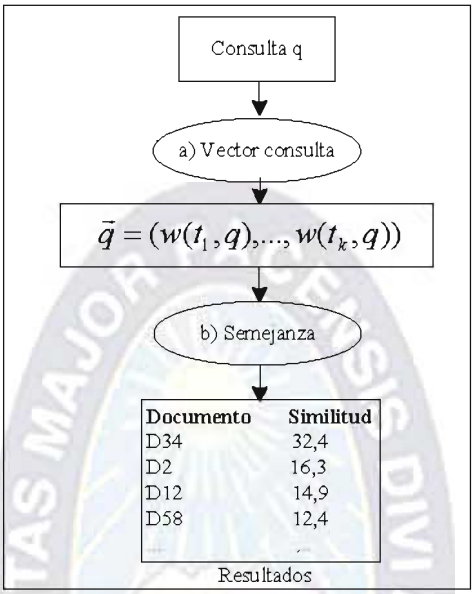

Figura 2.6: Proceso de búsqueda.
Fuente: Modificado [Becerra, 2008]

b) Semejanza: una vez generado el vector que representa la consulta, también denominado ecuación de búsqueda, se implementa el proceso de consulta o recuperación de documentos, definiendo una función de semejanza. Esta función debe permitir recuperar aquellos documentos que sean relevantes a la consulta. Una posible medida de similitud es el producto escalar:

$$sim(d_i,q) = \vec{d}_i \times \vec{q} = \sum_{j=1}^{n} (\vec{d}_i)^*(\vec{q})$$

donde cada documento d_i de una colección D está representado mediante un vector $\vec{d}_i = (w_{i1},..w_{in})$ y, de igual forma, cada pregunta q está representada por un vector $\vec{q} = (w(t_1,q),...,w(t_k,q))$.

La relevancia de un documento a una consulta se estima a través de la correlación o de la similitud de sus vectores. Por eso, las funciones que cuantifican estas correlaciones se suelen llamar funciones o medidas de similitud. El proceso de recuperación para una consulta consiste en calcular las similitudes entre cada uno de los documentos de la colección (sus vectores) y el vector de consulta. Después se ordenan los documentos de mayor a menor similitud y se devuelve esta lista ordenada como resultado.

2.2.3. Evaluación de un modelo de recuperación de información

La evaluación de un modelo de recuperación permite tener una idea bastante exacta de su comportamiento en términos cualitativos (la eficiencia del sistema) y en términos cuantitativos (la efectividad del modelo). Los aspectos considerados en la eficiencia son el tiempo de ejecución, la memoria utilizada en el espacio fisico. Dentro de la evaluación de la efectividad hay dos enfoques, uno que trata de medir objetivamente como la respuesta es adecuada a una consulta y el orientado a los usuarios, donde se trata de medir la satisfacción del que ha realizado la consulta. Sin embargo el primero es más utilizado porque no requiere la intervención activa o pasiva del usuario y es el único posible cuando no se tiene usuarios, cuando se esta instalando por primera vez el sistema de recuperación de información y tomando decisiones estratégicas. Se evalúa la relevancia de los documentos, porque es una medida más objetiva. La utilidad del documento depende del usuario, por lo que cambian según la situación. La relevancia se presta menos a estos cambios, y es además el objetivo que persiguen por defecto los modelos de recuperación [Gómez, 2003].

En general hay dos formas básicas de evaluar la relevancia de los resultados: manual y procedimiento de Polling [Gómez, 2003]:

a. Manual: consiste en la exploración de los documentos uno a uno para saber si se adecuan o no como respuesta a una pregunta. Muchas veces establecer la relevancia de un documento para una pregunta determinada resulta difícil y los especialistas no se ponen de acuerdo, por ello, es conveniente que los juicios los haga más de un especialista. El principal problema que presenta este método, es que en colecciones muy grandes, hay que invertir gran cantidad de tiempo, lo que supone mucho dinero para realizar esta operación y esto no siempre es posible.

b. Procedimiento de Polling: para determinar cuáles son los documentos relevantes, se recurre al "polling". Mediante este sistema se lanza un conjunto de consultas predefinidas sobre el modelo de recuperación. Como resultado se obtiene un número elevado de documentos que el modelo ha considerado como relevantes. Este conjunto de documentos pasa a ser evaluado manualmente por expertos. El resto de los documentos se considera directamente como no relevante. Este procedimiento es el que se viene utilizando en las TREC[36] desde 1994.

[36] De la siglas en ingles Test Retrieval Conference.

Independientemente del procedimiento elegido para crear la colección de pruebas, manual o por polling, el resultado será un conjunto de consultas, un conjunto de documentos sobre los que lanzar las consultas y un conjunto de juicios de relevancia.

En este trabajo para calcular la efectividad de la búsqueda ontologica se aplicara la evaluación de su comportamiento entorno a la relevancia de los documentos que proporcionan como respuesta a una petición de un usuario.

2.2.4. Medidas de evaluación de un modelo de recuperación precisión y exhaustividad

Otro aspecto importante en la evaluación de un modelo de recuperación es establecer una serie de medidas que permitan evaluar la información recuperada. Estas medidas tienen como objetivo sintetizar el comportamiento de un sistema de forma que se aprecie con facilidad la calidad de los resultados y proporcionan una interpretación estadística estandarizada de la forma en que un modelo actúa [Zazo et. al, 2002].

Justo después de que un modelo de recuperación haya resuelto una consulta de un usuario, se divide el conjunto de los documentos de la colección sobre la que dicho modelo actúa en dos grandes grupos, el de los documentos recuperados y el de los documentos no recuperados. Al mismo tiempo, se divide los documentos entorno al criterio de relevancia, con lo que tendríamos otros dos grupos, el de documentos relevantes y el de documentos no relevantes [Bilihardt, 2003] [Gómez, 2003]. Esto mismo se aprecia en la figura 2.7:

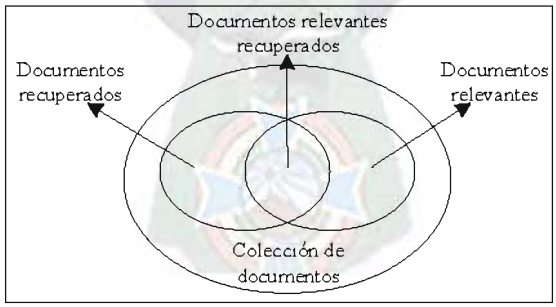

Figura 2.7: Conjuntos de documentos respecto a su relevancia a una consulta.
Fuente: [Gómez, 2003]

Otra forma más clara de representar este fenómeno es con una pequeña matriz que sintetice los estados posibles de un documento cualquiera de la colección:

Tabla 2.3: Conjuntos de documentos respecto a su relevancia a una pregunta
Fuente: [Bilihardt, 2003].

	Relevantes	**No Relevantes**
Recuperados	a	b
No recuperados	c	d

La efectividad óptima de un modelo de recuperación se obtiene cuando los dos criterios siguientes se cumplen [Zazo et. al, 2002]:

- Se han encontrado todos los documentos relevantes.
- No se han recuperado documentos no relevantes.

Estos dos criterios son los que definen las dos medidas básicas y más utilizadas para la evaluación de la efectividad de un modelo: precisión y exhaustividad (recall) [Gómez, 2003]:

a) La precisión: la precisión es la proporción de material recuperado relevante, del total de los documentos recuperados. En esta medida, se evalúa directamente la correlación de la consulta con la colección de documentos e indirectamente sirve para ver cómo es de completo el algoritmo de indexación. La precisión se calcula aplicando la siguiente formula:

$$precisión = \frac{Documentos\ Relevantes\ Recuperados}{Documentos\ Recuperados}$$

Utilizando la tabla 2.3, la precisión es:

$$precisión = \frac{a}{a+b}$$

La recuperación perfecta es en la que únicamente se recuperan los documentos relevantes y por lo tanto la precisión tiene un valor de uno. Esta medida está relacionada con dos conceptos, el de ruido y el de silencio informativo. De este modo, cuanto más se acerque el valor de la precisión a cero, mayor será el número de documentos recuperados que no le sirvan al usuario y por lo tanto el ruido que encontrará será mayor. Sin embargo, una alta precisión suele traer aparejada la probabilidad de que el silencio informativo sea mayor.

b) La exhaustividad: la exhaustividad, es el otro concepto más utilizado en la evaluación de los sistemas de recuperación. Muchos autores, por influencia del término inglés la denominan recall o rellamada. Es la proporción de material relevante recuperado, del total de los documentos que son relevantes en la colección de documentos, independientemente de que éstos se recuperen o no. Esta

medida es en teoría inversamente proporcional a la precisión. La exhaustividad se calcula aplicando la siguiente formula:

$$exhaustividad = \frac{Documentos\ Relevantes\ Recuperados}{Documentos\ Relevantes}$$

Utilizando la tabla 2.3, la exhaustividad es:

$$exhaustividad = \frac{a}{a+c}$$

Si el resultado de este cálculo tiene como valor 1, se tendrá la exhaustividad máxima, ya que se ha encontrado todo lo relevante que había en la colección, por lo tanto no se tendrá silencio informativo. Sin embargo es posible que tenga ruido, dado que no todos los documentos recuperados tienen por qué ser relevantes.

2.2.5. Medidas promedio de la Exhaustividad-Precisión

Salton [Salton, 1983] entiende que los cálculos *Exhaustividad-Precisión* (*E-P* en adelante), deben realizarse documento a documento recuperado, es decir, no son iguales el par de valores E-P en el primer documento que en el segundo, al final de este par de cálculos se propone calcular las medidas *E-P* en términos de promedio, esto significa el promedio *E-P* que el usuario esperaría la realización de la búsqueda por parte del modelo, en la tabla 2.4 se muestra las formulas para el calculo de estas medidas [Martínez, 2004].

Tabla 2.4: Formulación de las Medidas Promedio E-P.
Fuente: [Martínez, 2004]

$Exhaustividad = \frac{1}{N} * \sum_{i=1} (RecRel_i / RecRel_i + NoRecRel_i)$ $Precisión = \frac{1}{N} * \sum_{i=1} (RecRel_i / RecRel_i + RecNoRel_i)$
RecRel: documentos recuperados relevantes — *NoRecRel: documentos no recuperados relevantes*
RecNoRel: documentos recuperados no relevantes — *NoRecNoRel: documentos no recuperados no relevante*

En el anexo B se presenta un ejemplo del calculo *E-P* para una colección de documentos.

2.2.6. Búsqueda semántica

Una búsqueda semántica es una consulta en la que se tiene en cuenta el contexto, y por tanto el significado, de aquello por lo que se pregunta (y no solamente las palabras de la consulta). Una búsqueda semántica con las palabras "agente" e "inteligente" devolvería documentos relacionados sobre aprendizaje, razonamiento conocimiento y comunicación, aunque estos conceptos no aparecieran en los términos de búsqueda, porque identificaría los conceptos que estructuran la búsqueda. Se admite que las búsquedas semánticas se basan en técnicas para extraer información mediante la utilización de ontologías. El uso de ontologías permite definir formalmente los dominios de interés con la suficiente riqueza expresiva para que los usuarios especifiquen sus búsquedas con bastante detalle, ya sea antes de ejecutar la consulta o durante su ejecución. Esta búsqueda semántica se realiza mediante un buscador semántico. En términos técnicos un buscador semántico es una aplicación que comprende las búsquedas de los usuarios y los textos de los documentos mediante el uso de algoritmos que simulan comprensión o entendimiento, y que a partir de éstos proporciona resultados correctos sin que el usuario tenga que abrir el documento e inspeccionarlo por sí mismo. Un buscador de este tipo reconoce el contexto correcto para las palabras o sentencias de búsqueda. Google o Yahoo no son buscadores semánticos, pues se basan fundamentalmente en algoritmos que generan estadísticas a partir de palabras y enlaces, y no en algoritmos cognitivos que capturen el conocimiento implícito en las palabras y su contexto [Abián, 2009]. Un buscador semántico, debe disponer de medios para conocer el sentido exacto que tiene la palabra en la búsqueda. Un buscador semántico que utilice internamente ontologías deberá disponer de herramientas para determinar a qué se refiere el usuario cuando hace una consulta, para ello una opción seria escoger el significado más probable, preguntar al usuario para que elija entre varias opciones o usar las demás palabras de la búsqueda para inferir el significado exacto, los algoritmos semánticos simulan la comprensión de las palabras y, por ende, establecen relaciones entre ellas, para realizar búsquedas de interés para el usuario aunque en los documentos devueltos no figuren las palabras o expresiones de la búsqueda. Las búsquedas semánticas son muy superiores a las basadas en palabras clave, uno encuentra documentos de interés que no encontraría buscando con palabras clave.

2.3 ONTOLOGÍAS

En filosofía, el término ontología se define como "*la parte de la metafísica que trata del ser en general y de sus propiedades trascendentales*", en Inteligencia Artificial en cambio tiene diferentes connotaciones. Existen diferentes autores que ofrecen su propia interpretación de este concepto, la

definición mas consolidada es la propuesta por Gruber [Gruber, 1993], como "*una especificación explícita de una conceptualización",* Borst [Borst, 1997] modifica ligeramente la definición de Gruber: "l*as ontologías son definidas como una especificación formal de una conceptualización compartida*".

Studer, Benjamins y Fensel [Studer, 1998] combinaron y agregaron expresividad a las definiciones de Gruber y Borst de la siguiente manera:

> *"una ontología es una especificación explícita y formal sobre una conceptualización consensuada". La interpretación de esta definición es que las ontologías definen sus conceptos, propiedades, relaciones, funciones, restricciones y axiomas de forma "explícita" en algún lenguaje de implementación capaz de contener este conocimiento. El término "conceptualización" se refiere a un modelo abstracto de algún fenómeno en el mundo. El conocimiento de las ontologías es establecido para ser usado de forma "consensuada" y compartida por diferentes sistemas, que deberán comprometerse con el vocabulario utilizado en la ontología. El término "formal" se refiere a que la ontología debe implementarse en algún lenguaje computable por la máquina.*

Desde la perspectiva de las aplicaciones que las utilizan, existen tres definiciones diferentes de ontología [Lozano, 2002]: a) "una ontología es un conjunto de términos estructurados jerárquicamente que describen un dominio". La ontología será el esqueleto sobre el cual se construye luego la base de conocimientos. La característica principal que se acentúa en esta definición es que las ontologías proporcionan la estructura taxonómica de un dominio, la cuál será especializada con los conocimientos específicos necesitados por la aplicación. b) "una ontología proporciona los significados que describen explícitamente la conceptualización del conocimiento representado en una base de conocimientos". Mediante un proceso de abstracción, el modelo conceptual de la ontología se construye a partir de los conocimientos procedentes de bases de conocimientos existentes. De esta forma, los conocimientos especificados en la ontología son utilizados más fácilmente para construir otras bases de conocimientos en el mismo dominio, o en dominios similares y c) una ontología es "un sistema de conceptos/vocabulario usados como primitivas para construir sistemas artificiales". En este sentido, el propósito en la construcción de ontologías es capturar los conocimientos para que sean empleados en diferentes sistemas, independientemente de la tarea que pretendan resolver.

Estas tres definiciones tienen en común la idea de que las ontologías proporcionan la conceptualización explícita de los términos de un dominio, que sirve como soporte para la implementación de bases de conocimientos preparadas para ser utilizadas por aplicaciones, y que resuelven diferentes tareas.

Las ontologías son ampliamente usadas en la Ingeniería de Conocimiento, la Inteligencia Artificial y la Informática, en aplicaciones relacionadas a la gestión de conocimiento, procesamiento del lenguaje natural, el comercio electrónico, en la integración inteligente de información, recuperación de información, diseño de base de datos y en campos emergentes nuevos como la Web Semántica [Gomez et al, 2004].

2.3.1. Elementos de una ontología

Las ontologías proporcionan un vocabulario común de un área y definen, a diferentes niveles de formalismo, el significado de los términos y relaciones entre ellos. Una ontología consta de un conjunto no vacío de conceptos identificados como entidades relevantes en el dominio a modelar, un conjunto de atributos que describen los conceptos y que son propios o heredados en una especialización, un conjunto de relaciones entre dichos conceptos, un conjunto de funciones, un conjunto de axiomas que vinculan elementos de la ontología en condiciones que deben cumplirse siempre y un conjunto de instancias. A continuación se describe cada uno de estos elementos [Martín, 2004].

- Un *concepto* es un objeto acerca del cual algo se asevere, y por tanto llegaría al ser un objeto físico tangible, u objetos intangibles como por ejemplo la descripción de una tarea, función, acción, estrategia, entre otros. Cada concepto tiene un término asociado como nombre y un conjunto de atributos que lo identifican.
- Los *atributos* representan la estructura interna de los conceptos. Atendiendo a su origen, los atributos se clasifican en específicos y heredados. Los específicos son los propios del concepto al que pertenecen, mientras que los heredados vienen dados por las relaciones taxonómicas en las que el concepto desempeña el rol de hijo y, por tanto, hereda los atributos del padre. Los atributos se caracterizan por el dominio en el cual toman un valor.
- Las *relaciones* representan el tipo de interacción entre los conceptos de un dominio, son formalmente definidas como subconjuntos del producto cartesiano de n conjuntos, esto es R: *C1 x C2 x ... x Cn*. Algunas relaciones tienen un significado especial, a saber: las relaciones binarias de especialización (Es un) y de composición (es parte de). En general los modelos ontológicos definen la relación taxonómica *es un* como irreflexiva, transitiva y asimétrica.

- Las *funciones* son un caso especial de relaciones donde el enésimo elemento de la relación es único para los *n-1* anteriores. Formalmente las funciones se definen como *F: C1 x C2 x....x Cn-1 x Cn.* Ejemplos de funciones son las relaciones Madre-de.
- Los *axiomas* se usan para modelar verdades que se cumplen siempre en la realidad modelada. Los axiomas definidos en una ontología son estructurales o no estructurales.
- Un axioma estructural establece condiciones relacionadas a las jerarquías de la ontología, conceptos y atributos definidos. Los axiomas no estructurales establecen relaciones entre atributos de un concepto, y son específicos de cada dominio.
- Las *instancias* son las ocurrencias en el mundo real de los conceptos. En una instancia, todos los atributos del concepto tienen asignado un valor concreto.

2.3.2. Tipos de ontologías

En el trabajo de Fernández [Fernández, 2002] y Poli [Poli, 2000] se encuentra una descripción de los tipos de ontologías clasificados por el tipo de conocimiento contenido y por motivación de la ontología, a continuación se presenta un resumen de estas dos clasificaciones.

Tabla 2.5: Tipos de ontologías.
Fuente: Fernández [Fernández, 2002] [Poli, 2000].

Clasificación	Tipos de ontologías	Descripción	Fuente
Por el conocimiento contenido	Ontologías terminológicas lingüísticas	Especifican los términos usados para representar conocimiento en el dominio.	[Fernández, 2002]
	Ontologías de información	Especifican la estructura de los registros de la base de datos.	[Fernández, 2002]
	Ontologías para modelar conocimiento	Especifican conceptualizaciones de conocimiento.	[Fernández, 2002]
		Permiten explicar las conceptualizaciones que subyacen de los formalismos de representación de conocimiento.	[Poli, 2000]
	Ontologías del dominio	Contienen todos los conceptos asociados a un dominio particular.	[Fernández, 2002]
		Se refieren a la estructuración detallada de un contexto de análisis con respecto a los sub dominios que lo componen.	[Poli, 2000]
	Ontologías de tarea	Establecen la forma en la cual se usa el conocimiento del dominio para realizar tareas específicas.	[Fernández, 2002]
	Ontologías generales	Contienen descripciones generales sobre objetos, eventos, relaciones temporales, relaciones causales, modelos de comportamiento y funcionalidades.	[Fernández, 2002]
		Ontologías fundamentales que se aplican a todos los niveles ontológicos.	[Poli, 2000]

Por motivación	Ontologías de aplicación	Están ligadas al desarrollo de una aplicación concreta. Tales ontologías cubren los aspectos relacionados con aplicaciones particulares.	[Poli, 2000]
	Ontologías categóricas	Estudian las diversas formas en las que una categoría tiene diversos niveles ontológicos, determinando la posible presencia de una teoría general que subsume sus concretizaciones.	[Poli, 2000]
	Ontologías genéricas	Parecen ligadas a corpus lingüísticos y léxicos conceptuales. Esto significa que cada término debería ser accesible por defecto únicamente en su sentido genérico.	[Poli, 2000]
	Ontología regional	Analiza las categorías y sus conexiones de interdependencia para cada nivel ontológico.	[Poli, 2000]
	Ontología aplicada	Estas ontologías son la aplicación concreta de entorno ontológico a un objeto específico.	[Poli, 2000]

2.3.3. Métodos para el desarrollo de ontologías

Existen varias metodologías para el desarrollo de ontologías, con diverso grado de dificultad y especificación en su aplicación. Todas ellas exponen los procedimientos y las herramientas que son usadas para el desarrollo de ontologías. A continuación se presenta un resumen de los métodos Cyc, Uschold y King's, Grüninger y Fox's, KACTUS y Methontology presentadas en el libro Gómez y colegas [Gómez et al, 2004].

a) Método Cyc. Cyc[37] es un proyecto de inteligencia artificial que intenta ensamblar una ontología comprensiva y una base de datos de conocimiento general con el fin de permitir a las aplicaciones de inteligencia artificial realizar razonamientos del tipo humano. Cyc contiene una multitud de reglas simples (como "el agua causa humedad" y "la humedad pudre la comida"). Un ordenador llega a concluir a partir del motor de inferencia de Cyc que el agua pudre la comida (al menos en exceso). La base de datos contiene aproximadamente 100.000 conceptos y 1.000.000 de declaraciones que abarcan aserciones definidas por humanos, reglas o ideas del sentido común.

La metodología Cyc consiste en varios pasos. En primer lugar hay que extraer manualmente el conocimiento común que está implícito en diferentes fuentes. A continuación, una vez que se tenga suficiente conocimiento en la ontología, se adquiriere nuevo conocimiento común usando herramientas de procesamiento de lenguaje natural o aprendizaje computacional. Así se construyó la ontología Cyc.

[37] Del inglés ***encyclopedia***

Esta metodología recomienda los siguientes pasos: codificación manual de conocimiento implícito y explícito extraído de diferentes fuentes, codificación de conocimiento usando herramientas software y delegación de la mayor parte de la codificación en las herramientas.

b) Metodología de construcción de ontologías de Uschold y King. Esta metodología propone algunos pasos generales para desarrollar ontologías: (1) identificar el propósito; (2) capturar los conceptos y relaciones entre estos conceptos y los términos utilizados para referirse a estos conceptos y relaciones; (3) codificar la ontología. La ontología es documentada y evaluada, y se utilizan otras ontologías para crear la nueva. Esta metodología recomienda los siguientes pasos: identificar propósito, capturar la ontología, codificación, integrar ontologías existentes, evaluación y documentación [Hernández, 2007].

c) Metodología de construcción de ontologías de Grüninger y Fox. En esta metodología el primer paso es identificar intuitivamente las aplicaciones posibles en las que se usará la ontología. Posteriormente, se usa un conjunto de preguntas en lenguaje natural, llamadas cuestiones de competencia, para determinar el ámbito de la ontología. Se usan estas preguntas para extraer los conceptos principales, sus propiedades, relaciones y axiomas, los cuales se definen formalmente en Prolog. Por consiguiente, ésta es una metodología muy formal que se aprovecha de la robustez de la lógica clásica y que es usada como guía para transformar escenarios informales en modelos computables. Esta metodología, recomienda los siguientes pasos: escenarios motivantes, cuestiones informales de competencia, terminología formal, cuestiones formales de competencia, axiomas formales y teoremas de completitud [Gómez et al, 2004]

c) Metodología KACTUS. En esta metodología se construye la ontología sobre una base de conocimiento por medio de un proceso de abstracción. Cuantas más aplicaciones se construyen, las ontologías se convierten en más generales y se alejan más de una base de conocimiento. En otras palabras, se propone comenzar por construir una base de conocimiento para una aplicación específica. A continuación, cuando se necesita una nueva base de conocimiento en un dominio parecido, se generaliza la primera base de conocimiento en una ontología y se adapta para las dos aplicaciones, y así sucesivamente. De esta forma, la ontología representaría el conocimiento consensuado necesario para todas las aplicaciones. Esta metodología ha sido utilizada para construir una ontología para diagnosticar fallos, y recomienda seguir los siguientes pasos: especificación de la aplicación, diseño preliminar basado en categorías ontológicas top-level relevantes, refinamiento y estructuración de la ontología [Gómez et al, 2004].

d) Metodología Methontology. Es una metodología para construir ontologías tanto partiendo desde cero como rehusando otras ontologías, o a través de un proceso de reingeniería. Este entorno permite la construcción de ontologías a nivel de conocimiento, e incluye: (1) identificación del proceso de desarrollo de la ontología donde se incluyen las principales actividades (evaluación, gestión de configuración, conceptualización, integración, implementación); (2) un ciclo de vida basado en prototipos evolucionados; y (3) la metodología propiamente dicha, que especifica los pasos a ejecutar en cada actividad, las técnicas usadas, los productos a obtener y cómo deben ser evaluados. Esta metodología ha sido usada en la construcción de múltiples ontologías, como una ontología química, ontologías hardware y software. Se proponen los siguientes pasos: especificación, conceptualización, formalización, implementación, mantenimiento [Gómez et al, 2004] [Hernández, 2007]. La metodología Methontology se describirá con más detalles mas adelante dentro del proceso de desarrollo de ontologías.

2.3.4. Lenguajes para el desarrollo de ontologías

En estos últimos años, muchas organizaciones, grupos de investigación, y comunidades de Internet están desarrollando lenguajes para representar el conocimiento y hacer posible compartir y re-usar este conocimiento, estos lenguajes son importantes en la representación de los elementos de una ontología. Un esquema de evolución de los lenguajes de marcas a los lenguajes ontologicos se muestra en la tabla 2.6:

Tabla 2.6: Esquema de evolución de los lenguajes ontologicos.
Fuente: [Samper, 2005] [González, 2004].

Lenguaje	Descripción
HTML	HTML[38] es un lenguaje de marcas que se emplea en un documento HTML para visualizar la apariencia de una página Web por medio de un navegador.
XML	Es un lenguaje que no solamente describe la forma de visualizar los datos del documento, sino que describe el contenido del documento. Este lenguaje permite únicamente extraer datos, pero no su contenido semántico.
XML-Schema	Define una serie de reglas, atributos, tipos de datos que restringe el contenido de un documento XML.
SHOE	SHOE[39] es un lenguaje de representación del conocimiento basado en HTML, que permite añadir a las páginas Web conocimiento entendible por programas. Es un conjunto de HTML que añade las etiquetas necesarias para embeber datos semánticos en páginas Web. Es el lenguaje de ontologías más primitivo y menos funcional.
XOL	XOL[40] está diseñado para proporcionar un formato para el intercambio de definiciones de ontología entre un conjunto de partes interesadas. La sintaxis de XOL se basa en un enfoque de etiquetas XML para describir cualquier ontología.
RDF	Es un modelo de datos para objetos (recursos) y relaciones entre ellos, proporcionando una semántica sencilla para este modelo y pudiendo representarse con la sintaxis XML.

[38] Acrónimo del ingles HyperText Markup Language, Lenguaje de Marcado de Hipertexto.
[39] Del ingles Simple HTML Ontology Extensions.
[40] Del ingles XML-based ontology-exchange language, Lenguaje de Intercambio de Ontologías basado en XML

RDF-Schema	Es un vocabulario para describir propiedades y clases de los recursos RDF, con una semántica para jerarquías de generalización de dichas propiedades y clases.
OIL	OIL[41] Es la evolución de SHOE, y utiliza la sintaxis de XML y está definido como una extensión de RDF-Schema, incrementa la riqueza semántica en los lenguajes de representación de conocimiento para la descripción de los significados de los términos involucrados.
DAML+OIL	DAML[42]+OIL, es un lenguaje inspirado por la lógica descriptiva, añade más vocabulario para la descripción de propiedades y clases: entre otras, relaciones entre clases, cardinalidad, igualdad, nuevas propiedades y características de propiedades.
OWL	OWL[43], es una extensión del RDF-Schema. OWL proporciona un vocabulario para la definición de clases, sus propiedades y las relaciones entre las clases. Sin embargo, permite al usuario expresar relaciones de mayor riqueza semántica, dotando de una mayor capacidad de inferencia al procesamiento de un documento OWL. OWL se divide en tres niveles: a) OWL Lite la versión más simple permite la jerarquía de clasificación y las restricciones simples, b) OWL DL tiene todo el vocabulario OWL completo. Las limitaciones son que las clases no son instancias ni tipos y los tipos no son ni instancias ni clases. No permite restricciones de cardinalidad en propiedades transitivas y c) OWL Full esta versión también incluye todo el vocabulario de OWL pero en este caso no hay limitaciones para explotar todo su potencial.

2.3.5. Herramientas para el desarrollo de ontologías

En los últimos años, han aparecido un gran número de entornos para la construcción y uso de ontologías. El uso de herramientas es importante tanto para el proceso de desarrollo de ontologías como para el uso de las ontologías en aplicaciones tales como comercio electrónico, gestión de conocimiento y la Web Semántica. Estas herramientas se organizan en las siguientes categorías: desarrollo de ontologías, evaluación de ontologías, combinación e integración de ontologías, herramientas de anotación basadas en ontologías, almacenamiento y consulta de ontologías, aprendizaje sobre ontologías, gestión de ontologías, razonadores y lenguajes de consulta ontologicos [Navarro y Sarrios, 2007].

a) Herramientas de desarrollo de ontologías: este grupo incluye herramientas, entornos y suites que son usados para construir una nueva ontología desde cero o rehusar ontologías existentes. Aparte de la funcionalidad de edición y navegación, estas herramientas incluyen normalmente documentación, importación y exportación de ontologías, librerías y motores de inferencia adjuntos (ver tabla 2.7).

[41] Del acrónimo ingles Ontology Inference Layer, Capa de Inferencia de Ontologías.
[42] Del ingles DARPA Agent Markup Language.
[43] Acrónimo del ingles Ontology Web Language, Lenguaje Web de Ontologías.

Tabla 2.7: Herramientas de desarrollo de ontologías.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
Ontolingua	Proporciona un entorno distribuido y colaborativo para la creación, edición, modificación, navegación y utilización de ontologías mediante la Web.
OntoStudio	Es una herramienta de edición, desarrollo y mantenimiento de ontologías que utiliza medios gráficos. Permite almacenarlas en una base de datos relacional. Permite la representación semántica de lenguajes conceptuales y estructuras mediante conceptos, jerarquías de conceptos, relaciones y axiomas.
Protégé	Es una herramienta para construir modelos de dominio y aplicaciones basadas en conocimiento con ontologías. En su núcleo, Protege implementa un conjunto de estructuras de modelado de conocimiento y actividades que ayudan a la creación, visualización, y manipulación de ontologías en varios formatos de representación.
Swoop	Es un editor de ontologías inspirado en hipermedia, que emplea la filosofía de uso y diseño de los navegadores Web. Algunas de sus características: Complementos para diferentes presentaciones de sintaxis OWL (sintaxis abstracta, turtle, RDF/XML, validación), búsqueda semántica.
WebODE	Es una herramienta para modelar el conocimiento usando ontologías. Algunas de las características de esta herramienta son: Soporte para múltiples usuarios, conceptualización guiada en interfaz, personalizable mediante plantillas, chequeo completo de la consistencia, edición de taxonomías mediante interfaces basados en formularios y editores gráficos.
WebOnto	Fue diseñado para facilitar la navegación, creación y edición cooperativa de ontologías sin sufrir problemas de interfaz. Las características principales son: Gestión gráfica de ontologías, generación automática de instancias a partir de definiciones de clases, inspección de elementos, teniendo en cuenta la herencia de propiedades y el chequeo de consistencia.

b) Herramientas de evaluación de ontologías: aparecieron como herramientas de ayuda que aseguran que tanto las ontologías como las tecnologías asociadas tenían un nivel de calidad. Asegurar la calidad es extremadamente importante para evitar problemas en la integración de ontologías y en las aplicaciones industriales con tecnologías basadas en ontologías (ver tabla 2.8).

Tabla 2.8: Herramientas de evaluación de ontologías.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
ONE-T de Ontolingua	Permite la verificación de ontologías almacenadas y disponibles en cualquier servidor Ontolingua. Detecta los siguientes tipos de errores de inconsistencia en taxonomías de conceptos: Errores de circularidad, errores de partición, redundancia, errores gramaticales y definición formal idéntica de clases.
OntoClean en WebODE	Es una herramienta de edición, desarrollo y mantenimiento de ontologías que utiliza medios gráficos. Permite almacenarlas en una base de datos relacional. Permite la representación semántica de lenguajes conceptuales y estructuras mediante conceptos, jerarquías de conceptos, relaciones y axiomas.

c) Herramientas de combinación e integración de ontologías: este tipo de herramientas aparecieron para solucionar el problema de combinar o integrar diferentes ontologías del mismo dominio (ver tabla 2.9).

Tabla 2.9: Herramientas de combinación e integración de ontologías.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
Chimaera	Es un entorno de combinación y diagnóstico de ontologías basado en navegador Web. Aunque inicialmente fue pensado para integrarse con Ontolingua, no es un requisito, se utiliza con cualquier editor. Facilita la combinación permitiendo al usuario subir ontologías a su espacio de trabajo y posteriormente sugiriendo candidatos potenciales para combinación basándose en el número de propiedades. Esto genera una lista de resolución de nombres que se utiliza como guía para la tarea de combinación.
PROMPT	Es un módulo extensión de Protege. PROMPT dirige al usuario a través del proceso de combinación, identificando posibles puntos de integración, y haciendo sugerencias relativas a las próximas operaciones a realizar, que conflictos se necesitan resolver, y como resolver. Además de proporcionar sugerencias al usuario, PROMPT identifica conflictos de los siguientes tipos: Conflictos de nombres, referencias nulas, redundancias en jerarquías de clase y restricciones sobre valores de propiedades que violan la herencia de clase.
ODEMerge	Es una herramienta para combinar ontologías que está integrada en WebODE. Esta metodología propone los siguientes pasos: Transformación de los formatos de las ontologías a ser combinadas, evaluación de las ontologías, combinación de las ontologías, transformación del formato de la ontología resultante para ser adaptado a la aplicación donde será usada.

d) Herramientas de anotación basadas en ontologías: este tipo de herramientas han sido diseñadas para permitir a los usuarios insertar y mantener semi-automáticamente marcas en páginas Web basadas en ontologías. La mayoría de estas herramientas han aparecido recientemente, junto al surgimiento de la Web Semántica. La mayoría de ellas están siendo integradas en un entorno de desarrollo de ontologías (ver tabla 2.10).

Tabla 2.10: Herramientas de anotación basadas en ontologías.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
KIM	Esta herramienta proporciona infraestructura y servicios de anotación semántica, indexado y recuperación basándose en una ontología y en una masiva base de conocimiento. Las características mas destacadas de KIM son: Indexado y recuperación, consulta y explotación de conocimiento formal, seguimiento de acontecimientos simultáneos y clasificación, análisis de la evolución de población en entidades, instanciación automática de una ontología y anotación dinámica en dominios abiertos de contenidos no estructurados y semí-estructurados para la Web Semántica.
Ontomat	Es una herramienta de anotación de páginas Web. Ayuda al usuario en la tarea de crear y mantener ontologías basadas en OWL para crear instancias, atributos y relaciones. Incluye un navegador de ontologías para explorar la ontología y las instancias y un navegador HTML que visualiza las partes anotadas del texto.

e) Herramientas de almacenamiento y consulta de ontologías: este tipo de herramientas han sido creadas para permitir un fácil uso y consulta de ontologías. Debido a la extensa aceptación de la Web como plataforma de comunicación de conocimiento, han aparecido en este contexto nuevos lenguajes para la consulta de ontologías tales como RDQL, SeRQL o SPARQL (ver tabla 2.11).

Tabla 2.11: Herramientas de almacenamiento y consulta de ontologías.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
Kowari	Es una base de datos de código abierto, escalable, transaccional construida con el objetivo de almacenar, recuperar y analizar metadatos. Las características más relevantes de la herramienta Kowari son: Soporte nativo de RDF, soporta múltiples modelos de BD, lenguaje de consultas SQL y lenguaje de consultas interactivo.
Sesame	Es una arquitectura que permite el almacenamiento persistente de datos RDF e información en RDF-Schema y la subsiguiente consulta de dicha información. Una característica importante de Sesame es la abstracción de los detalles particulares de los repositorios que usa para el almacenamiento real. Esto hace posible trasladar Sesame a una gran variedad de repositorios, incluyendo Bases de Datos Relaciónales, almacenes de tripletas RDF, e incluso servicios de almacenamiento remoto en la Web.
Owlim	Owlim es la abreviatura de OWLMemSchemaRepositorySAIL para Sesame (capa de almacenamiento e inferencia), el cual soporta razonamiento parcial sobre OWL DL. Es una implementación en memoria, lo que permite una consulta y recuperación eficiente. Las características de esta herramienta son: Razonamiento y soporte de lenguaje OWL, la expresividad del lenguaje soportado no puede ser extendido en la dirección de DL.

f) Herramientas de aprendizaje basadas en ontologías: son usadas para derivar ontologías a partir de textos en lenguaje natural de manera semi automática (ver tabla 2.12).

Tabla 2.12: Herramientas de aprendizaje basadas en ontologías.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
KEA	KEA extrae automáticamente frases clave a partir del texto completo de documentos. El conjunto de frases candidatas de un documento se identifica usando procesamiento léxico, se computan las características de cada candidata, y se usa aprendizaje para realizar una clasificación que determine qué candidatos deben ser designados como frases clave. El mecanismo del esquema de aprendizaje construye primero un modelo predictivo usando documentos de entrenamiento con frases clave conocidas, y después usa el modelo para encontrar frases clave en nuevos documentos.
Text-to-Onto	Integra un entorno para construir ontologías de dominio a partir de un núcleo inicial de ontología. También es capaz de descubrir estructuras conceptuales de diferentes fuentes usando técnicas de adquisición de conocimiento y aprendizaje. Text-To-Onto ha implementado vanas técnicas de aprendizaje basada en ontologías a partir de texto libre y de texto semí-estructurado, diccionarios, ontologías y bases de datos. El resultado del proceso es una ontología de dominio que sólo contiene conceptos de dominio adquiridos de las fuentes de entrada citadas anteriormente.

g) Herramientas de Gestión de ontologías: estas herramientas incluyen varios aspectos del proceso de desarrollo de ontologías como la edición, navegación, almacenamiento y recuperación, versionado de ontologías. Este tipo de herramientas también ofrecen interfaces de programación de aplicaciones, también conocidas como APIs, que proporcionan a los programadores una infraestructura técnica que les permita desarrollar aplicaciones reales con ontologías, aislándolos de las particularidades de la sintaxis concreta, y proporcionándoles una perspectiva de alto nivel de los objetos de una ontología: clases, propiedades, axiomas (ver tabla 2.13).

Tabla 2.13: Herramientas de Gestión de ontologías.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
Jena	Marco de trabajo Java para construir aplicaciones con ontologías. Proporciona un entorno de programación de RDF, RDFS y OWL, SPARQL e incluye un motor de inferencia basado en reglas. Las características de Jena son: API de RDF y OWL, lectura y escritura de RDF en RDF/XML, N3 y N-Tnples, almacenamiento en memoria y persistente, motor de consultas SPARQL.
Kaon2	Infraestructura para gestionar ontologías OWL-DL, SWRL y F-Logic. Proporciona las siguientes funcionalidades: Una API de programación para la gestión de ontologías OWL-DL, SWRL y F-Logic, un servidor propio que proporciona acceso a ontologías de forma distribuida usando RMI, un motor de inferencia para responder consultas expresadas en sintaxis SPARQL, interfaz DIG, proporcionada a herramientas tales como Protege y un módulo para extraer instancias de ontologías de Bases de Datos Relaciónales.
pOWL	Es un marco de trabajo en PHP para el análisis sintáctico, almacenamiento, consulta, manipulación, servicio y señalización de bases de conocimiento OWL en un entorno colaborativo Web. Las características de pOWL son: Navegación y edición de ontologías RDFS/OWL de tamaño arbitrario, edición de datos visuales, sistema de consulta RDQL y autenticación en el modelo.

h) Razonadores: dado que la base de la ontología es la lógica descriptiva y cuando se desarrolla una ontología se está realizando una representación formal de un dominio de aplicación en forma de sentencias lógicas (hechos y reglas), se hace necesario el uso de herramientas conocidas como motores de inferencia o razonadores, que no son más que programas que realizan deducciones siguiendo las reglas lógicas especificadas con el objetivo de inferir nuevo conocimiento (ver tabla 2.14).

Tabla 2.14: Razonadores.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
FaCT	Permite reglas transitivas, inversas, restricciones cualificadas y jerarquías. Es lo suficientemente expresivo para soportar y razonar sobre cualquier base de conocimiento. Esta escrito en Lisp[44] y ejecutable por cualquier programa Lisp de forma local, tiene escrita un versión servidor FaCT para ser usada vía interfaz CORBA[45] sobre cualquier sistema con acceso a la red. Actualmente es el razonador empleado por defecto por OilEd para clasificar los conceptos en una jerarquía según las descripciones que tenga.
RACER	Es un razonador diseñado para la Web Semántica. Permite la inferencia tanto en conceptos como en instancias, soporta ontologías escritas en RDF/RDFS/DAML/OWL y posee un lenguaje de consulta sencillo para la inferencia de instancias. Es utilizado en herramientas como OilEd y Protégé tanto para comprobar la consistencia de la ontología, como para hacer consultas sobre el conocimiento.
Pellet	Es un razonador de OWL-DL basado en Java. Es utilizado conjuntamente con bibliotecas del API[46] de Jena[47]. Mediante su uso es posible validar, comprobar la consistencia de ontologías, clasificar la taxonomía y contestar a un subconjunto de consultas RDQL[48] (conocido como consultas a ABox[49] en terminología de DL). Se trata de una razonador DL basado en los algoritmos tableaux desarrollados para DL expresiva.

[44] Del ingles LISt Processing. Lenguaje específico utilizado en el desarrollo de la inteligencia artificial.
[45] Acrónimo del ingles Common Object Request Broker Architecture.
[46] Acrónimo del ingles Application Programming Interface, Interfaz de Programación de Aplicaciones
[47] Framework desarrollado en Java aplicaciones Web Semantica
[48] Del ingles RDF Data Query Language.
[49] *TBox* = base de conocimientos, *ABox* = base de hechos, Las declaraciones de Tbox describen un sistema en términos.

i) Lenguajes de consulta ontologicos: los razonadores basados en lógica descriptiva permiten el razonamiento en las ontologías mediante un lenguaje de consulta para ontologías. Estos razonadores reciben una consulta basadas en ontologías y las ejecutan contra una base de conocimiento (ver tabla 2.15).

Tabla 2.15: Lenguajes de consulta ontologicos.
Fuente: [Navarro y Sarrios, 2007]

Herramienta	Descripción
RQL	Es un lenguaje de consulta para RDF y RDF-Schema. Permite navegar por los grafos que hay en el modelo RDF y proporciona un mecanismo para preguntar y seleccionar los nodos del modelo que se quiera recuperar. La característica más destacable de este lenguaje es que posee construcciones propias específicas para las relaciones semánticas dentro del RDF-Schema, como son las relaciones de clase/instancia, clase/propiedad o el dominio y rango de una propiedad, por lo que resulta más fácil recuperar información de los nodos del modelo.
RDQL	Fue desarrollado para que fuese el lenguaje de consulta para RDF en los modelos de Jena. No permite realizar ninguna inferencia, la utilización de filtros para obtener resultados es muy limitada. La ventaja es la sencillez de manejo.
SPARQL	Es un lenguaje de consultas para grafos RDF propuesto por W3C. Ofrece a los desarrolladores y usuarios finales un camino para presentar y utilizar los resultados de búsquedas. SPARQL también proporciona un camino de integración sobre recursos diferentes. Permite: Extraer información en diversas formas, incluyendo URIs, extraer subgrafos RDF. construir nuevos grafos RDF basados en la información de los grafos consultados.

2.4. PROCESO DE DESARROLLO DE ONTOLOGÍAS

El proceso de construir una ontología no difiere mucho, en líneas generales del usado para construir software. Las ontologías dentro el ámbito de la Inteligencia Artificial son un producto software y por lo tanto su desarrollo deberá seguir los estándares establecidos para el desarrollo de software, pero adaptados a las características de las ontologías [Ramos y Nuñez, 2007].

2.4.1. Actividades del proceso de desarrollo de ontologías

En el trabajo de Fernández [Fernández et al, 1997] se presenta una propuesta del proceso de desarrollo de ontologías, tal propuesta esta basada en el proceso de desarrollo de software propuesta por la IEEE[50]. El proceso de desarrollo de ontologías se refiere a que actividades son realizadas al construir ontologías. En la figura 2.8 se muestra las tres categorías de actividades orientadas al desarrollo de ontologías predesarrollo, desarrollo y post desarrollo.

[50] Acrónimo del ingles Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Electricistas y Electrónicos,

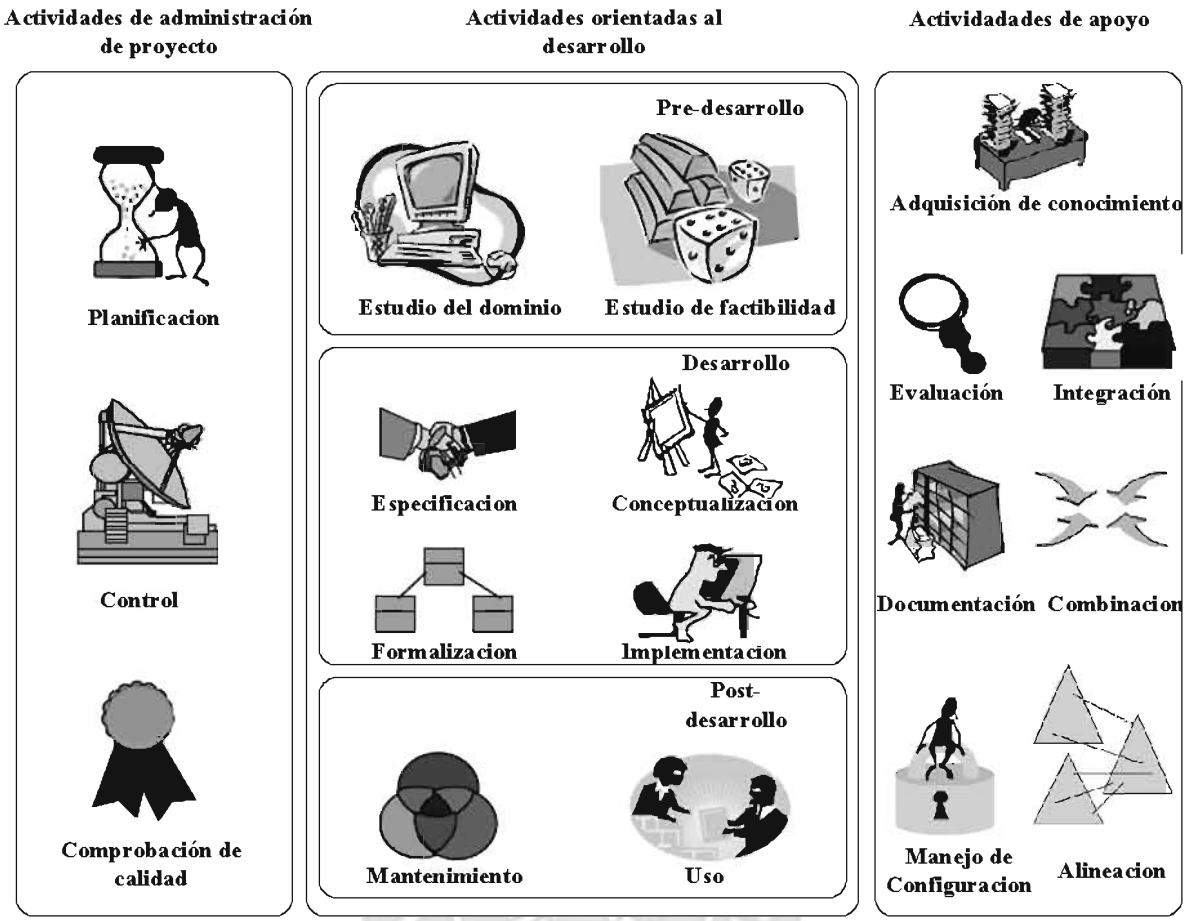

Figura 2.8: Proceso de desarrollo de ontologías.
Fuente: [Fernández et al, 1997].

A continuación se describe las actividades y los objetivos del proceso de desarrollo de la figura 2.8 descritas por Gómez y Fernández [Gómez et. al., 2004] [Fernández et al, 1997].

a) Actividades de administración de proyectos: dentro de las actividades de administración de proyectos se encuentran actividades de planificación, control y comprobación de calidad [Ramos y Núñez, 2007], a continuación se describe cada una de estas:

1. Actividad de planificación: Esta actividad es esencial para el desarrollo del proyecto, su objetivo es identificar las tareas a realizar y los recursos disponibles para el proyecto, tales como software, hardware, recursos humanos.
2. Actividad de control: Esta actividad garantiza que las tareas previstas son completadas de acuerdo a lo planificado, su objetivo es especificar los mecanismos para garantizar que las tareas se realicen según lo planificado.

3. Actividad comprobación de calidad: Finalmente esta actividad asegura que la calidad de toda y cada salida de productos cono ontologías, software y documentación sea satisfactorio, su objetivo es especificar los estándares de calidad que se deben cumplir en todas las tareas.

b) Actividades orientadas al desarrollo: las actividades orientadas al desarrollo son agrupados en predesarrollo, desarrollo y posdesarrollo que a continuación se describen [Gómez et. al., 2004] [Ramos y Núñez, 2007]:

1. Predesarrollo.- Dentro de esta actividad se encuentran las actividades estudio del dominio y estudio de factibilidad que a continuación se describe:
   Actividad estudio del dominio: Esta actividad es realizada para identificar las plataformas donde la ontología será utilizada, aplicaciones donde la ontología será integrada.
   Actividad estudio de factibilidad: Esta actividad contesta a preguntas, ¿es posible construir la ontología?, ¿es conveniente construir la ontología?

2. Desarrollo.- Esta actividad es la mas importante porque es aquí donde se construye la ontología, dentro este grupo se encuentra las actividades de especificación, conceptualización, formalización y de implementación que a continuación se describe:
   Actividad de especificación: En esta actividad se especifica porque se construye, de que manera se utilizara la ontología, quienes serán los usuarios finales de la ontología, el objetivo de esta actividad es realizar un documento que contenga información referente a usuarios finales, objetivo, metas y grado de formalidad de la ontología.
   Actividad de conceptualización: En esta actividad se estructura el conocimiento del dominio como modelos significativos en el nivel de conocimiento, el objetivo de esta actividad es construir un modelo conceptual que describa el problema y su posible solución.
   Actividad de formalización: El Objetivo de esta actividad es transformar el modelo conceptual en un modelo semi computable, utilizando representaciones lógicas, grafos conceptuales, esquemas.
   Actividad de implementación: El objetivo de esta actividad es codificar la ontología en un lenguaje antológico.

3. Posdesarrollo.- En este grupo se encuentra las actividades de mantenimiento y de uso:
   Actividad de mantenimiento: En esta actividad se actualiza y corrige la ontología cuando sea necesario.

Actividad de uso: La ontología en algunas ocasiones es utilizada por otras ontologías o aplicaciones

c) Actividades de apoyo: finalmente se encuentran las actividades de apoyo de ontología que incluyen actividades que son realizadas al mismo tiempo que las actividades orientadas al desarrollo sin las cuales no podría construirse, a continuación se describe cada una de estas actividades [Ramos y Núñez, 2007].

1. Actividad de adquisición de conocimiento: El objetivo de esta actividad es adquirir conocimiento de expertos de un dominio dado mediante la aplicación de técnicas apropiadas.
2. Actividad de evaluación: La evaluación de la ontología se realiza durante cada etapa y entre etapas del ciclo de vida de la ontología.
3. Actividad de integración: Esta actividad es necesaria cuando se necesita construir una nueva ontología y se cuenta con ontologías disponibles ya desarrolladas o integrar ontologías existentes para garantizar la utilización del conocimiento.
4. Actividad de documentación: Se documenta cada una de las etapas completadas y productos generados de la ontología para garantizar el éxito al se compartida y reutilizada.
5. Actividad de manejo de configuración: Esta actividad registra todas las versiones de la documentación y el código de la ontología para controlar los cambios.

El proceso de desarrollo de ontología indica *que actividades deben ser realizadas* para construir una ontología, sin embargo no identifica el orden en el cual las actividades deben se realizadas. El ciclo de vida de la ontología indica *cuando las actividades deben ser realizadas* es decir indica el conjunto de etapas a través de la cual la ontología tiene que pasar para ser construida, describe que actividades deben ser realizadas en cada una de las etapas y como se relacionan estas etapas.

En general las metodologías proporcionan un conjunto de directrices que indican *como deben realizarse las actividades* identificadas en el proceso de desarrollo, que técnicas son las mas apropiadas en cada actividad y que produce cada una de ellas [Ramos y Nuñez, 2007].

A continuación se describe la metodología Methontology para el desarrollo de ontologías.

2.4.2. Metodología Methontology

Esta metodología fue desarrollada dentro del grupo de ontología en el laboratorio de Inteligencia Artificial de la universidad Politécnica de Madrid por Corcho, Fernández y Gómez, tiene sus raíces en las actividades principales identificadas por el proceso de desarrollo de software y en metodologías de ingeniería del conocimiento. Esta metodología incluye: la identificación de del proceso de desarrollo de la ontología, un ciclo de vida basado en la evolución de prototipos y técnicas particulares para realizar cada actividad [Ramos y Nuñez, 2007].

a) *Ciclo de vida de la ontología:* el proceso de desarrollo de ontología descrito en las actividades de la figura 2.8: actividades de administración de proyectos, actividades orientadas al desarrollo y actividades de apoyo, fue propuesto en el framework de esta metodología y hace referencia a las actividades a ser realizadas durante la construcción de una ontología. Este proceso no indica el orden en el cual tales actividades deben ser realizadas, este es el papel del ciclo de vida de la ontología. Methontology propone la construcción de una ontología basado en la evolución de prototipos, porque esto permite añadir, cambiar y quitar términos en cada versión nueva (prototipo) [Gómez et. al., 2004].

Para cada prototipo Methontology propone comenzar con las actividades de planificación que identifica las tareas a ser realizadas para determinar el tiempo y recursos necesarios para su terminación. Después de esto la actividad de especificación de ontología es iniciada paralelamente con las actividades de administración, orientadas al desarrollo y de apoyo durante el ciclo de vida de la ontología.

Una vez que el primer prototipo ha sido especificado el modelo conceptual de la ontología es construido dentro de la actividad de conceptualización del ciclo de vida. Esta conceptualización es realizada con los recursos suministrados por la actividad de adquisición de conocimiento. Luego de la conceptualización las actividades de formalización e implementación son ejecutadas. Si alguna carencia es descubierta después de haber terminado algunas des estas actividades se regresa a una estas actividades anteriores para realizar la modificaciones o refinamientos. Cuando se utiliza una herramienta como el editor de ontologías WebODE, el modelo de conceptualización es automáticamente es implementado en varios lenguajes antológicos. Por lo tanto la formalización no es una actividad obligatoria en Methontolgy [Ramos y Nuñez, 2007].

En la figura 2.9 muestra el ciclo de vida para la construcción de ontologías propuesta en Methontology y resume la descripción anterior sobre cuando realizar las actividades. Note que las actividades de administración, desarrollo y de apoyo son realizadas simultáneamente.

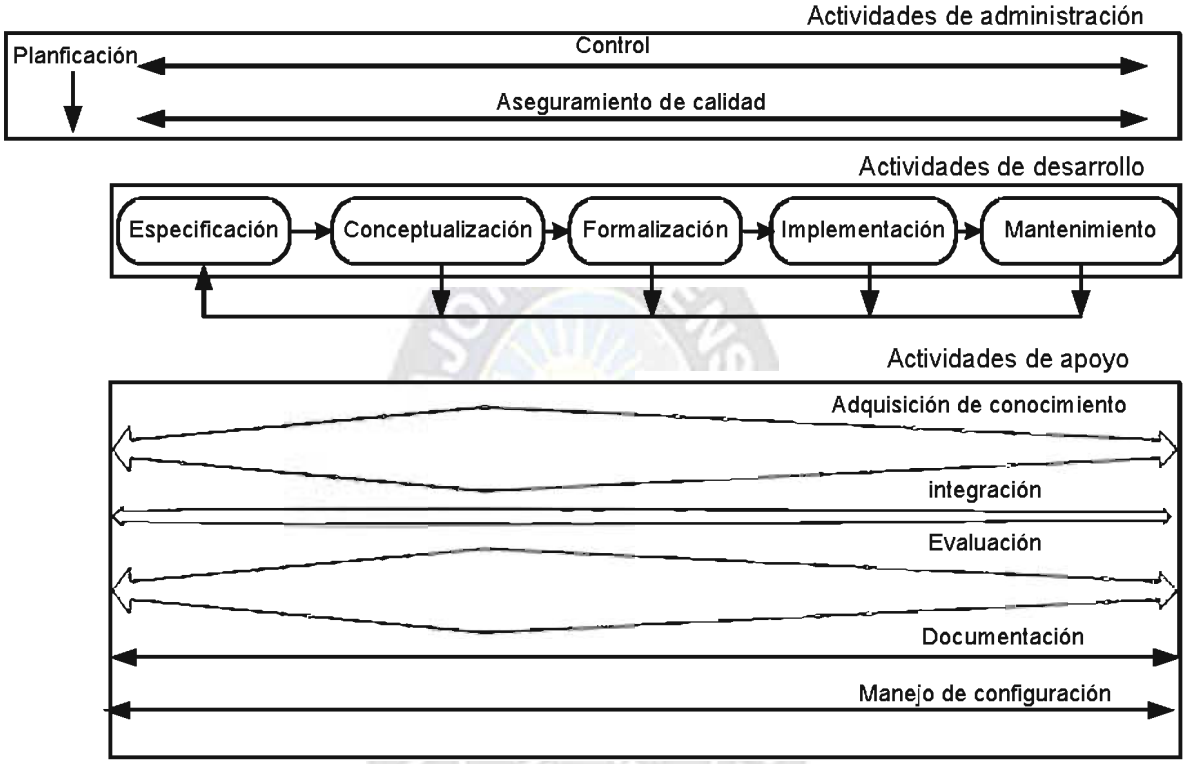

Figura 2.9. Ciclo de vida de Methontolgy.
Fuente. [Gómez et. al., 2004].

La figura muestra que la adquisición de conocimiento, integración y evaluación son mayores durante la conceptualización de la ontología, y que estas actividades disminuyen durante la formalización y la implementación. Las razones para este mayor esfuerzo en la conceptualización son [Gómez et. al, 2007]:

- La mayor parte del conocimiento es adquirido al principio de la construcción de la ontología.
- La conceptualización de la ontología debe ser evaluada para evitar propagar errores en etapas posteriores del ciclo de vida de Methontolgy.
- Las relaciones entre las actividades realizadas durante el desarrollo de la ontología son llamadas intradependencias o lo que es lo mismo que las relaciones definen el ciclo de vida de la ontología.

La planificación se realiza antes del desarrollo de la ontología, por lo tanto no forma parte del ciclo de vida.

b) Especificación de ontologías: la especificación de ontología es un conjunto de requisitos que la ontología debe cumplir. Esta tarea es realizada por un equipo de expertos en el dominio acompañado por expertos en el modelado de ontologías [Sure y Studer, 2002].

El objetivo de la especificación de la ontología es obtener un ORSD[51]. El describe los requisitos relevantes como: propósito y alcance de la ontología, recursos de conocimiento, usuarios, grupo de preguntas. El ORSD debe guiar al ingeniero de ontologías a tomar decisiones a cerca de la inclusión o exclusión de conceptos y relaciones y construir la estructura jerárquica de la ontología [Sure y Studer, 2002] [Gómez y otros, 2007].

Los requisitos son necesidades que la ontología debe ser capaz de solucionar, las preguntas relevantes CQs[52] son preguntas que la ontología debe ser capaz de contestar, por lo tanto las CQs es una forma para definir requisitos explícitos para la ontología. Típicamente las CQs son obtenidas como resultado de entrevistas con expertos en el dominio y ayuda a encontrar los conceptos y relaciones importantes entre ellos, estructurar conocimiento y evaluar la ontología en una fase posterior. Las CQs se escribe en lenguaje natural y son formalizadas en un lenguaje de consulta de ontología como SPARQL [Gómez y otros, 2007].
El ORSD debe tener los siguientes puntos:

- Propósito y Alcance (dominio, meta.)
- Usos pretendidos (escenarios)
- Usuarios
- Recursos de conocimiento usados durante esta actividad
- Grupos de preguntas relevantes con prioridades
- En lenguaje natural
- En lenguaje de consulta de ontología (optativo)
- El pre-glosario de términos con frecuencias

La figura 2.10 muestra la secuencia de tareas para la especificación de ontología y la tabla 2.16 indica el objetivo, salida, técnicas y herramientas recomendadas para realizar cada tarea propuesta por Gómez [Gómez y otros, 2007].

[51] Acrónimo del ingles, Ontology Requirements Specification Document, Documento de especificación de requisitos de la ontología.
[52] Acrónimo del ingles, Competency Questions, Preguntas de verificación.

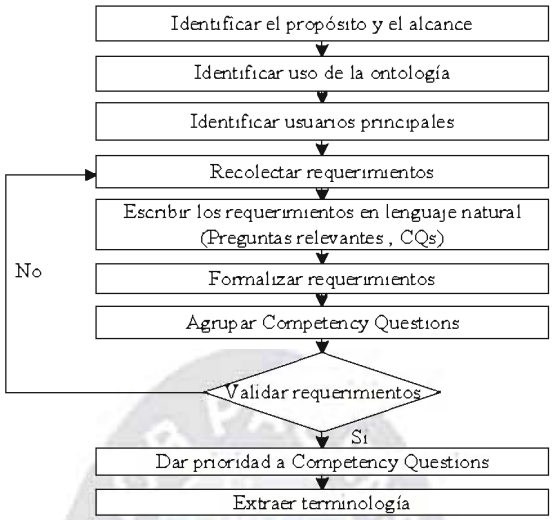

Figura 2.10: Secuencia de tareas para la especificación de la ontología.
Fuente: [Gómez y otros, 2007].

Tabla 2.16: Objetivo, salida técnicas y herramientas para la secuencia de tareas.
Fuente: [Gómez y otros, 2007]

	Especificación de la ontología: Propósito y alcance.
Objetivo	Obtener el dominio y objetivo principal para la ontología.
Salida	El propósito y alcance para la plantilla ORSD.
Técnicas recomendadas	Entrevistas con usuarios y expertos en el dominio.
Herramientas recomendadas	¿?
	Especificación de la ontología: Uso de la ontología.
Objetivo	Obtener cual es el uso principal para la ontología, es decir en que tipos de escenarios la ontología será usada.
Salida	Uso de la ontología para la plantilla ORSD.
Técnicas recomendadas	Entrevistas con usuarios y expertos en el dominio.
Herramientas recomendadas	¿?
	Especificación de la ontología: Usuarios principales.
Objetivo	Obtener quienes son los usuarios principales de la ontología.
Salida	Usuarios principales para la plantilla ORSD.
Técnicas recomendadas	Entrevistas con usuarios y expertos en el dominio.
Herramientas recomendadas	¿?
	Especificación de la ontología: Recolectar requerimientos.
Objetivo	Obtener el conjunto de requisitos (necesidades) que la ontología debe satisfacer.
Salida	Un conjunto de requisitos informales para ser usados en la siguiente tarea.
Técnicas recomendadas	Entrevistas con usuarios y expertos del dominio. Tormenta de ideas. Explorar escenarios y casos de uso usando plantillas.
Herramientas recomendadas	¿?

	Especificación de la ontología: Escribir Preguntas relevantes CQs).
Objetivo	Transformar el conjunto informal de requisitos obtenidos en la anterior tarea en CQs.
Salida	Conjunto de CQs escrito en lenguaje natural y conjunto de respuestas para las CQs.
Técnicas recomendadas	Top-Down: Las preguntas complicadas se descompone en preguntas simples. Bottom-Up: Las preguntas simples por composición se deriva en preguntas complejas. Middle out.
Herramientas recomendadas	Mapas mentales MindMap.
	Especificación de la ontología: Formalizar CQs.
Objetivo	Formalizar la CQs.
Salida	Un conjunto de preguntas formalizadas en un lenguaje de consulta de ontologías.
(Esta tarea es optativa pero se recomienda realizarlo).	Formalizado las CQs servirá para la evaluación semi automática de la ontología contra los requerimientos.
	Especificación de la ontología: Agrupar CQs
Objetivo	Clasificar las CQs en grupos o categorías diferentes.
Salida	Una clasificación del conjunto de CQs.
Técnicas recomendadas	Clasificación en tarjetas (manual). Agrupar oraciones en lenguaje natural.
Herramientas recomendadas	MinMap (mapas conceptuales.)
¿Por qué es útil la agrupación?	Desarrollo basado en módulos. Desarrollo basado en prototipos.
¿Criterios de agrupación?	Basado en términos principales que aparecen en CQs. (Criterios dependientes del dominio) Basado en dimensiones o términos generales. (Criterios independientes del dominio), Ejemplo tiempo y fecha, unidades de medidas, localización, lenguajes.
	Especificación de la ontología: Validar CQs.
Objetivo	Identificar posibles conflictos o contradicciones en CQs
Criterios	Exactitud, integridad, coherente, comprensible, verificable, sin ambigüedad, sin redundancia.
	Especificación de la ontología: Prioridad para CQs.
Objetivo	Dar niveles diferentes de prioridad a las CQs.
Salida	Conjunto de CQs con una prioridad concreta.
(Esta tarea es optativa, pero recomendada)	Las prioridades de CQs servirán para planificar el desarrollo de ontología.
	Especificación de la ontología: Pre glosario.
Objetivo	Extraer un pre glosario para ser usado en la actividad de la conceptualización.
Salida	Pre glosario con los términos principales usado en CQs.
Técnicas recomendadas	Extracción de terminología: los términos son nombres, adjetivos, verbos. o De las CQs se extrae el modelo conceptual. o De las respuestas para las CQs se extrae el universo de discurso (la base de conocimiento). Herramientas de extracción de terminología.
Herramientas recomendadas	Herramientas de frecuencia de terminología.

c) Modelo conceptual Methontology: en este inciso se presenta la propuesta de Methontology para la conceptualización de la ontología. Su objetivo es organizar y estructurar el conocimiento adquirido durante la actividad de adquisición de conocimiento, utilizando representaciones que son independientes de los paradigmas de representación de conocimiento y los lenguajes de implementación en los cuales la ontología es formalizada e implementada. Esta actividad de conceptualizacion tiene una relación fuerte con la actividad de adquisición de conocimiento [Ramos y Nuñez, 2007].

Esta actividad de conceptualización en Methontolgy organiza y convierte una percepción informal de un dominio en una especificación semi-formal usando representaciones intermedias basadas en notaciones graficas y que son entendibles por expertos del dominio y desarrolladores de ontologías. Methontology plantea conceptualizar la ontología usando un conjunto de representaciones intermedias en forma de gráficos y tablas. La figura 2.11 muestra los componentes de la ontología (conceptos, atributos, relaciones, constantes, axiomas formales, reglas e instancias) construidos en cada tarea, esta figura también ilustra el orden propuesto para crear tales componentes durante la actividad conceptualización. El orden para realizar las tareas no es secuencial como un modelo de ciclo de vida en cascada, sin embargo algún orden debe ser seguido para asegurar la consistencia y la integridad del conocimiento representado [Gómez et. al., 2004].

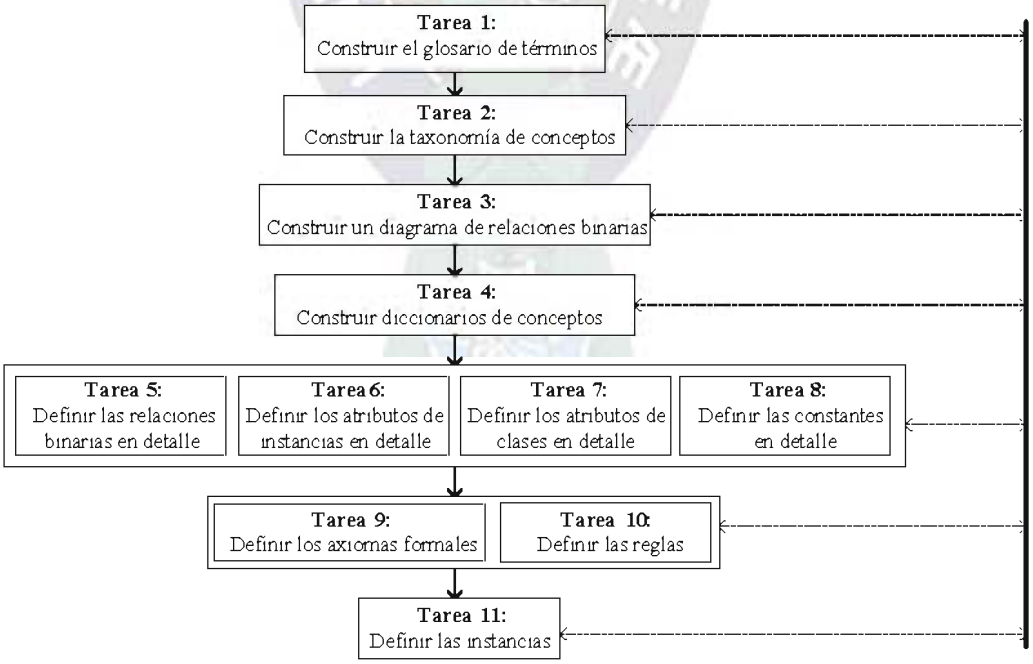

Figura 2.11. Tareas de la actividad de conceptualización según Methontology.
Fuente: [Ramos y Nuñez, 2007].

A continuación se describe cada tarea de la conceptualización [Ramos y Nuñez, 2007] [Gómez et. al., 2004].

Tarea 1: Construir el glosario de términos. El glosario de términos debe incluir todos los términos relevantes del dominio (conceptos, instancias, atributos, relaciones entre conceptos), sus descripciones en lenguaje natural, sus sinónimos y acrónimos.

Tarea 2: Construir la taxonomía de conceptos. Cuando el glosario de términos tenga una cantidad importante de elementos, se debe construir una taxonomía que defina la jerarquía entre los conceptos. Se debe evaluar que la taxonomía creada no contenga errores.

Tarea 3: Construir un diagrama de relaciones binarias. El objetivo de este diagrama es establecer las relaciones entre los conceptos de una o más taxonomías de conceptos. Se debe evaluar que el diagrama creado no contenga errores.

Tarea 4: Construir el diccionario de conceptos. El diccionario de conceptos contiene los conceptos del dominio, sus relaciones, instancias, atributos de clases y atributos de instancias. Las relaciones, atributos de instancias, y atributos de clases son locales al concepto, lo que significa que sus nombres se repeten en diferentes conceptos.

Tarea 5: Definir las relaciones binarias en detalle. Se crea la tabla de relaciones binarias en la que se describe detalladamente todas las relaciones binarias incluidas en el diccionario de conceptos. Para cada relación binaria se debe especificar: nombre, conceptos fuente y destino, cardinalidad y relación inversa.

Tarea 6: Definir los atributos de instancia en detalle. Se crea la tabla de atributos de instancias en la que se describe detalladamente todos los atributos de instancias incluidos en el diccionario de conceptos. Los atributos de instancias son aquellos atributos que describen las instancias de un concepto, y sus valores son diferentes para cada instancia del concepto. Para cada atributo de instancia, se debe especificar: nombre, concepto al que pertenece, tipo de valor, rango de valores (en el caso de valores numéricos) y cardinalidad.

Tarea 7: Definir los atributos de clases en detalle. Se crea la tabla de atributos de clases en la que se describe detalladamente todos los atributos de clases incluidos en el diccionario de conceptos. Para cada atributo de clase, se debe especificar: nombre, concepto donde es definido, tipo de valor, valor y cardinalidad.

Tarea 8: Definir las constantes en detalle. Se crea la tabla de constantes en la que se describe detalladamente cada una de las constantes definidas en el glosario de términos. Para cada constante, se debe especificar: nombre, tipo de valor, valor y unidad de medida (para constantes numéricas).

Tarea 9: Definir los axiomas formales. Se deben identificar los axiomas formales necesarios en la ontología y describirlos con precisión en una tabla. Para cada definición de axioma formal de debe especificar: nombre, descripción, expresión lógica que formalmente lo describe (preferiblemente utilizando lógica de primer grado), los conceptos, atributos y relaciones binarias a las cuales el axioma hace referencia y las variables utilizadas.

Tarea 10: Definir las reglas. Se deben identificar cuáles reglas son necesarias en la ontología y describirlas en una tabla de reglas. Para cada regla, se debe especificar: nombre, descripción, expresión que formalmente la describe, los conceptos, los atributos y las relaciones a los que hace referencia y las variables usadas en la expresión. Para la especificación de las reglas se sugiere la forma: *Si* <condiciones> *entonces* <consecuencias o acciones>.

Tarea 11: Definir las instancias. Una vez que el modelo conceptual de la ontología ha sido creado, se deben definir las instancias relevantes que aparecen en el diccionario de conceptos en una tabla de instancias. Para cada instancia se debe especificar: nombre, concepto al que pertenece y valores de los atributos.

El siguiente paso es la transformación del modelo conceptual en un modelo formal semi computable, esto es la etapa de formalización. En la etapa de implementación de la ontología se codifica la ontología utilizando un lenguaje formal XOL, DAML, RDF-Shema, OWL u otro. La Etapa de mantenimiento permite la actualización y corrección de la ontología.