UNIVERSIDAD POLITÉCNICA DE PUEBLA

Maestría en Ingeniería

"Mantenimiento de colecciones de documentos basado en ontologías."

TESIS DE MAESTRÍA

EVERARDO CARLOS GUEVARA HERNÁNDEZ.

Juan C. Bonilla, Puebla. Agosto 2012

UNIVERSIDAD POLITÉCNICA DE PUEBLA

Maestría en Ingeniería

"Mantenimiento de colecciones de documentos basado en ontologías."

TESIS DE MAESTRÍA

EVERARDO CARLOS GUEVARA HERNÁNDEZ.

COMITÉ EVALUADOR

DRA. MARÍA AUXILIO MEDINA NIETO

ASESOR

DR. ANTONIO BENÍTEZ RUIZ.

SINODAL

DR. JORGE DE LA CALLEJA MORA.

SINODAL

Juan C. Bonilla, Puebla. Agosto 2012

Agradecimientos:

A YHVH por su amor infinito que sobre pasa todo entendimiento, por guiarme a concluir esta experiencia.

A Conacyt por el apoyo económico prestado para concluir mis estudios.

A mi asesora que con sus consejos, guía y paciencia puedo concluir esta tesis.

A mi hijo Carlos a su esposa y nieto por el amor que siempre me han tenido.

A mi madre por el ejemplo de superación que siempre tuve.

A mi padre y hermano por el apoyo moral que me brindo.

A mis compañeros y profesores por haberme acompañado en esta aventura.

Contenido

Capítulo 1 Planteamiento del problema de investigación

Introducción

Hoy en día estamos viviendo una revolución en las comunicaciones como nunca antes había existido; muchos la llaman la "era de la información", responder preguntas como ¿cuál información será útil para llevar a cabo una investigación documental? o ¿cuál proviene de una fuente confiable, para ser agregar como referencia bibliográfica?, es difícil.

Una manera de asegurar que la documentación es confiable es consultando una biblioteca, ésta puede ser tradicional (con libros en papel, estantes y mesas) o digital, donde sus colecciones se encuentren almacenadas en forma electrónica. La cantidad de información de la segunda posiblemente sea mayor a la primera.

Una de las tareas de las bibliotecas digitales es clasificar u organizar las colecciones para que los usuarios puedan encontrar los documentos de su interés. En esta tesis se considera que las ontologías podrían apoyar la realización de esta tarea.

Las ontologías son objeto de estudio en Inteligencia Artificial (IA), se utilizan para representar conocimientos. Algunos de los propósitos de las ontologías son (Noy & Mcguinness, 2001):

- Aumentar la efectividad en búsquedas
- Interoperar con otros sistemas o colecciones
- Analizar el conocimiento de un dominio
- Separar el conocimiento base de un dominio del conocimiento operacional
- Hacer suposiciones explícitas en el dominio
- Posibilitar la reutilización del conocimiento

Algunos trabajos relevantes que han utilizado ontologías para clasificar colecciones de documentos son:

- El proyecto GALEN, Financiado por la Unión Europea, cuyo objetivo es desarrollar herramientas y métodos para la elaboración y mantenimiento de clasificaciones de procedimientos quirúrgicos (Sandoval, 2007)
- La iniciativa mundial de clasificación (IMT), promovida por la ONU, que tiene como meta desarrollar un estrategia para unificar clasificadores diversos y apoyar así la implementación del Convenido sobre la Diversidad Biológica (Naciones Unidas, 2012).

En lo referente a las ciencias computacionales, en la asociación ACM (*Association for Computing Machinery*) donde participan expertos, académicos y empresas privadas, se ha hecho uso del sistema de clasificación CCS (de las siglas de *Computing Classification System*). La versión de 1998 expira en 2012, año en la cual se liberó la última versión caracterizada por extender la clasificación a una ontología poli-jerárquica que puede utilizarse en aplicaciones de tipo web semántica (ACM , 2012).

Algunos de los trabajos relevantes que utilizan la versión de 1998 son los siguientes:

- El proyecto OntologyNavigator realizado por la Universidad París 8 Vincennes-Saint Denis, se traduce la taxonomía de la ACM CCS de 1998 al francés (Kembellec, Saleh, & Catalina, 2009) para extender mecanismos de búsqueda.

- El sistema de análisis temático del conocimiento científico (Godínez, 2009) resultado de una tesis de maestría del Instituto Politécnico Nacional, una herramienta de software que permite identificar tendencias que describen la evolución en una disciplina cuyos recursos de información se encuentran ya clasificados, analiza la producción de los recursos de información de ciencia y tecnología y visualiza gráficas con respecto al tiempo.

- La organización *Advanced Knowledge Technologies* realizó una ontología en OWL 2.0 basada en el clasificador ACM CCS de 1998 (ATK, 2012)

El trabajo propuesto, consiste en diseñar una metodología basada en ontología para mantener de forma automática las colecciones de documentos de un dominio restringido. Esta metodología pretende hacer frente a los retos que implica organizar y clasificar un documento nuevo.

En una biblioteca tradicional, la clasificación de un documento nuevo se realiza en dos etapas:

1. *Registró en el inventario:* Se registra el autor o autores, título, lugar de edición, editorial y fecha de publicación. En ocasiones se utiliza también el código ISBN[1]; este es un estándar internacional desde 1970 a través de la norma ISO 2108 que permite la identificación de libros (ISBN, España Libros, 2010)

2. *Clasificación del documento nuevo:* se realiza mediante diversos sistemas predeterminados. Uno de los más utilizados es la CDU (Clasificación Decimal Universal).Los sistemas de clasificación son listas que permiten ordenar los documentos para que éstos estén juntos de acuerdo a una división principal, subordinando los temas relacionados. La clasificación permite designar una serie de números al libro, la cual se asocia con el tema al que pertenece

En este trabajo la clasificación utilizará una ontología que representa la versión del 2012 de ACM CSS. Se propone implementar un método semi automático de asignación a clases con base en el procesamiento de metadatos. Se diseñará una interfaz gráfica de usuario accesible vía web que permita seleccionar un documento nuevo y que despliegue la clase o clases de la ontología a la que pueda pertenecer. Esta interfaz permitirá también a los usuarios realizar tareas de búsqueda y navegación.

[1] Número Estándar Internacional de Libros o Número Internacional Normalizado del Libro.

Para probar la metodología propuesta, se presentará un caso de estudio en el cual se ejemplifiquen características semánticas.

Objetivo general

Diseñar una metodología basada en ontologías para mantener de forma automática colecciones de documentos de un dominio restringido.

Objetivos específicos

- Analizar la semántica de una ontología que extienda la versión del 2012 de ACM CCS.
- Presentar un caso de estudio en el cual el proceso de construcción de la ontología ejemplifique características semánticas.
- Implementar un método automático de asignación a clases basado en el procesamiento de metadatos.
- Implementar una interfaz para el método automático de asignación que apoye la búsqueda y recuperación en una colección organizada con la ontología.

Justificación

Los trabajos previos de clasificación que se han revisado en ciencias computacionales se basan en la versión de 1998 de ACM. Este trabajo propone una manera automática para realizar la clasificación de un documento nuevo con la versión del 2012, misma que ya se considera una ontología porque está representada en lenguajes como XML y RDF. Estas representaciones permitirán explorar algunas características semánticas y diferentes lenguajes de consulta. Se trabajará con la versión en inglés de la ontología y se experimentará con una versión para el idioma español, con el propósito de la metodología propuesta pueda apoyar algunas de las actividades que actualmente realizan bibliotecarios, estudiantes y profesores.

Contribuciones.

- Metodología para el mantenimiento de colecciones de documentos basado en ontologías.
- Interfaz gráfica para mantener una colección organizada con la versión en ontología de ACM CCS con tareas de búsqueda y recuperación

Capítulo II. Marco teórico

Este capítulo presenta una reseña de la representación del conocimiento desde la perspectiva de la IA[2] y la web semántica. La primer parte explica groso modo en qué consiste la lógica descriptiva, la segunda especifica a las ontologías como elementos descriptivos de un dominio, se identifican sus componentes principales, se citan algunas clasificaciones y las características de lenguajes utilizados comúnmente para codificar ontologías. La última sección forma el contexto de la investigación al presentar trabajos relacionados.

2.1 Representación del conocimiento con lógica descriptiva

La *representación del conocimiento* es un campo de la IA que trata de la concepción de formalismos que permiten el desarrollo de sistemas basados en conocimiento y específicamente en el estudio de las distintas maneras de definir y crear bases de conocimiento. Según Russel y Norving (2004), este campo utiliza la lógica de primer orden para modelar aspectos importantes del mundo real como acciones, espacio, tiempo y eventos.

Una vez que el conocimiento ha sido representado adecuadamente, puede utilizarse en un sistema inteligente junto con herramientas de análisis, tratamiento y manipulación automática para ofrecer la capacidad de inducir o deducir conocimientos nuevos.

Las redes semánticas y la lógica descriptiva (LD) se han utilizado para generar esquemas que faciliten la extracción del conocimiento, (según el Diccionario de la Real Academia de la Lengua, un *esquema* es la representación gráfica o simbólica

[2] Inteligencia Artificial.

de cosas materiales o inmateriales). Estos esquemas se sustentan en el razonamiento y la inferencia. Las *redes semánticas* ayudan a visualizar las bases de conocimiento, aportan algoritmos eficientes para inferir propiedades de un objeto con base a la pertenencia de una categoría. La LD proporciona lenguajes formales para construir y combinar definiciones de un grado de jerarquía, así como algoritmos eficientes para decidir las relaciones de subconjuntos y súper-conjuntos entre categorías.

En la literatura existen controversias entre los autores en relación de las redes semánticas y la lógica descriptiva. En esta tesis, se considera a la LD como una evolución de las redes semánticas, usada tradicionalmente para representar conocimiento taxonómico en aplicaciones de IA e implementada en los razonadores. Según (Baader, McGuinness, & Nardi, 2003), la LD es un conjunto de lenguajes que pueden ser usados para representar el conocimiento terminológico de un dominio de aplicación de una forma estructurada y con una semántica formal bien definida.

De acuerdo a Nardi & Brachman (2002), la LD se fundamenta en dos conceptos: la *lógica basada en representaciones*, que se explica como "*la descripción de los conceptos que se usan para definir el dominio*" y la "*lógica basada en formalismos"*, que incluye la traducción de predicados de lógica de primer orden.

Las tareas principales de inferencia para la LD son (Kroetzsch, Simancik, & Horrocks, 2012):

1. **Subsunción.-** Del inglés *subsumption***,** que consiste en comprobar si una categoría es un subconjunto de otra a través de la comparación de sus definiciones.
2. **Clasificación.-** Su función es comprobar si un objeto pertenece a una categoría.
3. **Consistencia.-** Una categoría se dice que es consistente si el criterio de pertenencia puede ser satisfecho lógicamente.

> Russell (2004) afirma que lo más importante de la LD es el énfasis que se pone en la maleabilidad de la inferencia. Esta lógica carece de negación y disyunción, lo cual obliga a los sistemas de primer orden a producir análisis de casos de tipo exponencial. En caso de descripciones disyuntivas, las definiciones anidadas dan lugar a una cantidad exponencial de rutas alternativas.

Formalmente, la DL se compone de axiomas tipo TBox y ABox. Estos axiomas se describen brevemente como sigue (Russel & Norving, 2004):

- **TBox:** Se le conoce como *caja terminológica*, contiene conocimiento intencional en forma de una terminología y está construida por declaraciones que describen propiedades generales de los conceptos. Otra forma de definirla es como un conjunto de axiomas que describen la estructura de dominio, contiene sentencias formadas por conceptos jerárquicos.
- **ABox:** Llamada *caja de aserciones*, contiene conocimiento extensional, esto es, conocimiento específico de los individuos del dominio, se representa por medio de un conjunto de axiomas que describen una situación concreta.

La LD es parte de la lógica de predicados y constituye la base de los lenguajes de la web semántica RDF y de OWL. Estos lenguajes se han diseñado buscando un consenso entre la capacidad de cómputo y la expresividad.

La sintaxis de la LD consiste en:

- **Conceptos atómicos:** equivalentes a los predicados unarios de la lógica de predicados.
- **Roles atómicos:** equivalentes a los predicados binarios de la lógica de predicados.
- **Operadores:** establecen relaciones entre conceptos y restricciones sobre los mismos.

Las herramientas que hacen uso de la sintaxis de la LD para modelar el mundo real son clave para el razonamiento y la inferencia. Muchos métodos de representación del conocimiento fueron probados a lo largo de la década de los 70's hasta principios de los 80's como las redes neuronales, las demostraciones de teoremas y los sistemas expertos. En los años 80 surgieron lenguajes formales de programación y sistemas de representación del conocimiento enfocados al manejo de información. Desde entonces, han evolucionado lenguajes de programación orientados a la representación del conocimiento como Prolog[3], desarrollado en 1972, que representa proposiciones y lógica básica.

En esta tesis se utilizan ontologías para representar conocimiento, éstas ayudan a compartir un entendimiento común de una estructura de información entre personas o agentes software, posibilitando la reutilización del conocimiento de un dominio y explicitando suposiciones (Noy & Mcguinness, 2001). Las ontologías estandarizan el significado de vocabularios y relaciones para un dominio en específico.

2.2 Ontologías, esquemas de representación del conocimiento para la web semántica

La Real Academia de la Lengua (2010) define una ontología como parte de la metafísica que trata del ser en general y de sus propiedades trascendentales. Especificaciones de lo que existe o de lo que se puede decir sobre el mundo se han utilizado desde la filosofía aristotélica[4]. En Filosofía, una ontología es una teoría sobre la naturaleza de la existencia, de qué tipo de cosas existen. Se confunde a menudo con la epistemología, que trata del conocimiento y el conocer. En el siglo XVII, ontología se usó como sinónimo de metafísica o como la rama de la metafísica que trata con la naturaleza del ser (Swartout & Tate, 1999).

[3] Lenguaje de programación desarrollado en la Universidad de Aix-Marseille I (Marsella, Francia).
[4] El término ontología fue introducido originalmente por Aristóteles en su intento de clasificar todo lo existente en el universo.

El término ontología es adoptado por la IA a finales de la década de los 80's para compartir y reutilizar conocimiento, mientras que en la segunda mitad de los 90's se incorpora a la ingeniería web para la inclusión de descripciones semánticas explícitas de recursos. En ambas disciplinas, una ontología se materializa en un documento o un archivo que define formalmente las relaciones entre términos (Berners-Lee, Hendler, & Lassila, 2001). En la web semántica, las ontologías se consideran como esquemas de representación de conocimiento (Brewste, y otros, 2004). En aplicaciones de interoperabilidad, una ontología hace referencia a la formulación de un esquema conceptual con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades. Algunas de las definiciones más citadas sobre ontologías son las siguientes:

- Una ontología es un vocabulario acerca de un dominio: términos + relaciones + reglas de combinación para extender el vocabulario (Neches, Fikes, Finin, & Gruber, 1991).
- Una ontología es la especificación de una conceptualización[5](Gruber, 1993).
- Una ontología es una especificación formal de una conceptualización compartida[6](Borst, 1997).
- Una ontología es una base de datos que describe los conceptos generales sobre un dominio, algunas de sus propiedades y cómo los conceptos se relacionan unos con otros.(Weigand, 1997).

En esta tesis, una ontología se refiere a la especificación de un vocabulario relativo a cierto dominio, formado por entidades, clases, propiedades, predicados, funciones y las relaciones entre estos componentes.

Los programas informáticos pueden utilizar ontologías para propósitos como razonamiento inductivo, en técnicas de resolución de problemas y problemas de clasificación. Por ejemplo, las ontologías se usan para solucionar problemas

[5]Aquí el término conceptualización se refiere a un modelo conceptual.
[6] El término formal se refiere a que es procesable por una computadora.

semánticos de dominio y de nombre, los conflictos de dominio aparecen cuando conceptos similares en relación al significado, pero no idénticos, se representan en distintos dominios. Los conflictos de nombre son de dos tipos: sinónimos y homónimos. Los sinónimos ocurren cuando los sistemas usan distintos nombres para referirse al mismo concepto. Por ejemplo, trabajador y empleado. Los homónimos surgen cuando los sistemas usan el mismo nombre para representar cosas distintas. Por ejemplo, asiento “contable” o asiento “mueble” (Moreno & Sánchez, 2012).

Las ontologías en esta tesis se utilizan como una alternativa para atender un problema de clasificación particular que es el mantenimiento de colecciones, el cual de forma breve, puede describirse como sigue: dado un esquema de clasificación y un conjunto de documentos que tienen asociada una o más clases, utilizar la información de un documento nuevo para asignar de forma automática la clase apropiada.

2.2.1 Componentes de las ontologías

Al revisar la literatura, se distinguen componentes distintos de una ontología de acuerdo al dominio de interés y a las necesidades de los desarrolladores. Según Gruber (Gruber, 1993), las ontologías se componen de:

- **Conceptos:** Son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias o procesos de razonamiento.
- **Relaciones:** Representan la interacción y enlace entre los conceptos de un dominio, suelen formar una taxonomía. Algunas relaciones son: *subclase-de*, *parte-de*, *parte-exhaustiva-de* y *conectado-a*.
- **Funciones:** Son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, *asignar-fecha* o *categorizar-clase*.

- **Instancias:** Se utilizan para representar objetos determinados de un concepto.
- **Reglas de restricción o axiomas**: Son teoremas que se declaran sobre las relaciones que deben cumplir los elementos de la ontología. Los axiomas, junto con la herencia de conceptos, permiten inferir conocimiento no explícito en la taxonomía.

De acuerdo a (Sowa, 2000) y (Noy & Mcguinness, 2001), los componentes de las ontologías son:

- **Axiomas:** Elementos que permiten el modelado de verdades que se cumplen siempre en la realidad. Los axiomas pueden ser *estructurales* si establecen condiciones relacionadas con la taxonomía, los conceptos o atributos y *no estructurales* si se refieren a relaciones entre atributos de un concepto pero son específicos del dominio.
- **Clases o tipos**: Una clase es un conjunto de objetos físicos, tareas o funciones. Cada objeto es una instancia de esa clase. Desde el punto de vista de la lógica, los objetos se describen conforme a una relación de pertenencia a la clase. Las clases son la base de la representación del conocimiento en las ontologías, ya que se modelan los conceptos del dominio. Una clase puede dividirse en subclases que se asocian a conceptos más específicos; una clase cuyos componentes son clases se denomina superclase o meta-clase.
- **Conceptualización:** Conjunto de conceptos, relaciones, objetos y restricciones que caracterizan un domino.
- **Marco** (en inglés *frame*)**:** Es un objeto que incluye clases, instancias y relaciones.
- **Instancias o individuos:** Son objetos, miembros de una clase, que no pueden ser divididos sin perder su estructura y características funcionales.
- **Relaciones:** Se establecen entre conceptos de una ontología para representar las interacciones entre éstos. Algunas relaciones comunes son:
    - *Instancia de*: Asocian objetos a clases.

    - *Relaciones temporales*: Implican precedencia en el tiempo.
    - *Relaciones topológicas*: Establecen conexiones espaciales entre conceptos.
- **Propiedades**: Los objetos se describen por medio de un conjunto de atributos almacenados en *slots*. Las especificaciones, rangos y restricciones sobre los valores se denominan *propiedades, características o facetas*. Dada una clase, los slots y las restricciones se heredan por las subclases e instancias.
- **Taxonomía:** Conjunto de conceptos organizados jerárquicamente; define las relaciones entre los conceptos, pero no sus atributos.

2.2.2 Tipos de ontologías

Existen varias maneras de clasificar las ontologías. Michael Uschold (Uschold & Gruninger, 1996) identifica los tres tipos siguientes:

- **Formales.-** Trata el grado de formalismo del lenguaje usado para expresar la conceptualización.
- **Propósito.-**Conforme a la intención de uso de la ontología.
- **Materia.-**Si expresa la naturaleza de los objetos que la ontología caracteriza.

yb

Guarino (1998) clasifica a las ontologías con base en su dependencia y relación con una tarea específica en los siguientes tipos:

- **Ontologías de alto nivel o genéricas.-** Describen conceptos generales como espacio, tiempo o acción, es decir, conceptos independientes de un problema o dominio particular. En sistemas de información, describirían conceptos básicos.
- **Ontologías de dominio.-** Forman un vocabulario relacionado con un dominio que especializa los conceptos introducidos en una ontología de nivel superior.

- **Ontologías de tareas o de técnicas básicas.-**Representan una tarea, actividad o artefacto.
- **Ontologías de aplicación.-** Son las ontologías más específicas, describen conceptos que dependen de las ontologías de dominio y de tarea; frecuentemente, son especializaciones de ambas. Los conceptos corresponden a los roles propios de las entidades del dominio mientras se realiza una actividad.

Guarino (1998) identifica dos tipos de ontologías más de acuerdo a la forma de uso en un sistema:

- **No refinadas.-** Conocidas también como *on-line*, tienen un número mínimo de axiomas y se comparten por usuarios que las utilizan de manera concurrente
- **Las refinadas.-** Emplean un lenguaje con expresividad alta y un número grande de axiomas. Se utilizan como referencia y se conocen también como *off-line*.

La clasificación de las ontologías propuesta por Heijst (1997) se basa en la cantidad y tipo de estructura de la conceptualización. Los tipos son:

- **Ontologías terminológicas.-** Definen los términos usados para representar el conocimiento, unifican vocabularios en un campo determinado.
- **Ontologías de información.-**Caracterizan la estructura de almacenamiento de bases de datos, ofrecen un marco para el almacenamiento estandarizado.
- **Ontologías de modelado de conocimiento.-**Especifican conceptualizaciones del conocimiento, contienen una estructura interna rica y suelen ajustarse al uso particular del conocimiento que describen.

En sistemas de información, las ontologías tienen un papel clave en la resolución de problemas de interoperabilidad semántica. Su codificación puede realizarse en diferentes lenguajes legibles por las computadoras como los que se describen en la sección siguiente.

2.2.3 Lenguajes de la web semántica para ontologías

Los lenguajes para codificar ontologías como KIF (*Knowledge Interchange Format*), OCML (*Operational Conceptual Modelling Language*) y F-Logic (*Frame Logic*) surgieron en los 80's, sin embargo, en la actualidad tienen mayor aceptación los lenguajes que se desarrollaron después de la aparición de la web semántica.

En 1998, el consorcio W3C (*World Wide Web Consortium*) emplea XML como una solución las carencias de HTML en lo que respecta al tratamiento de la información. En 1999, W3C liberó RDF y RDFS, dirigidos a la representación de metadatos y al modelado de datos, respectivamente. En el mismo año apareció SHOE (*Simple HTML Ontology Extensions*) y en el 2000 se presentó DAML+OIL. OWL apareció en 2004. En las secciones siguientes se describen algunas de las características relevantes de estos lenguajes con respecto al contexto de esta tesis.

2.2.3.1 XML

De SGML[7] surgen lenguajes como HTML[8] y XML, ambos utilizados ampliamente en web. Los documentos escritos en estos lenguajes se emplean como fuentes de datos en tareas de recuperación de información, minería de datos y de manera más reciente, en representación del conocimiento.

Reutilizando la sintaxis y semántica de XML, se han derivado lenguajes de representación del conocimiento como RDF, RDF Schema y Web Ontology Lenguaje (OWL).

XML son las siglas provenientes de *e**X**tensible **M**arkup **L**anguage* o lenguaje de etiquetado extensible. Es un metalenguaje que permite la definición de etiquetas propias y lenguajes nuevos, estructura la información de los documentos que pueden ser usados en la web de forma jerárquica facilita la reutilización de documentos

[7] Standard Generalized Markup Lenguaje o Estándar de Lenguaje de Marcado Generalizado.
[8] HyperText Markup Lenguaje lenguaje de marcado de hipertexto.

completos o de algunas secciones del contenido (W3C, 2012). La sintaxis de XML consiste de una serie de reglas, pautas o convenciones para planificar formatos de texto, de manera que los archivos se generan e interpretan por una computadora fácilmente, esta característica apoya la interoperabilidad entre diferentes plataformas (Bray, Paoli, McQueen, & Eve, 2008). Sin embargo, como los conjuntos de etiquetas de XML no están predefinidos para todos los usuarios, en XML no existe una semántica preconcebida (Cowan & Tobin, 2004).

2.2.3.2 RDF

RDF corresponde a las siglas de ***R****esource* ***D****escription* ***F****ramework* o marco de descripción de recursos. Más que un lenguaje de propósito general para representar información en la web, es un modelo de datos utilizado para representar recursos y sus relaciones con una semántica básica que puede escribirse en XML. El conocimiento se representa como un conjunto de *sentencias* (o *ternas*) de recursos usando propiedades y valores. Las sentencias se forman por los componentes que se describen a continuación:

- **Sujeto:** Identifica a un recurso, puede ser cualquier objeto de un modelo como una página, un documento, un usuario o un enlace. A cada recurso se le asigna un identificador único en forma de URI[9].
- **Predicado:** Representa una propiedad como nombre, ciudad, título, color, forma o característica que describen al sujeto. Los predicados se identifican también con URI's.
- **Objeto:** especifica el valor para la propiedad del sujeto.

2.2.3.3 RDFS

El esquema de RDF (RDF Schema o RDFS) describe el vocabulario de RDF para clases de recursos, propiedades y relaciones; facilita mejoras a los procesos de búsqueda y permite hacer inferencias (Brickley, 2004). Las propiedades se describen

[9] *Uniform Resource Identifier*, una cadena corta de caracteres que identifica un recurso de la web.

en términos de las clases a las que éstas se pueden aplicar, aparecen los roles de ***dominio*** y ***rango.*** Así, por ejemplo, se podría definir la propiedad *"autor"* con un dominio *"documento"* y un rango *"persona"*.

RDFS es un lenguaje para escribir ontologías, contiene primitivas para el modelado de clases, relaciones de subclases, propiedades, relaciones de subpropiedades y restricciones de dominio y rango con un significado determinado (Antoniou & Van Harmelen, 2004). Tanto RDF como RDFS utilizan la inferencia para encontrar relaciones nuevas en una base de conocimiento.

2.2.3.4 OWL

OWL (*Ontology Web Language* o lenguaje web para ontologías) es un lenguaje que se basa en lógica descriptiva para definir diccionarios con características semánticas formales, apoya el procesamiento de aplicaciones que infieren el contenido de documentos con mayor precisión que RDF y RDFS al admitir expresiones lógicas.

En OWL, las relaciones tienen propiedades como cardinalidad, simetría, transitividad o relaciones inversas. Su semántica formal especifica cómo derivar sus consecuencias lógicas. La versión 1.0 de OWL proporciona tres sub-lenguajes(W3C, 2004):

- **OWL Lite.-** Se utiliza para migrar clasificaciones jerárquicas y restricciones simples.
- **OWL DL.-** Las siglas provienen de lógica descriptiva, la expresividad es máxima porque incluye a todos los constructores bajo ciertas restricciones y mantiene completitud computacional, es decir, todos los cálculos del sistema de razonamiento terminan en un tiempo finito.
- **OWL Full.-** Mantiene expresividad máxima y libertad sintáctica de RDF pero sin garantías computacionales dado que se aumenta el significado del vocabulario preestablecido.

La versión OWL 2 es una actualización de OWL añadiendo varias características nuevas (Golbreich de & Wallace, 2008):

- Las construcciones aumentan la expresividad de las propiedades, por ejemplo, restricciones de cardinalidad calificados o inclusión cadena de propiedad, las claves de estilo de base de datos; esto es extra para hacer algunas declaraciones comunes más fáciles de decir.
- Soporte extendido para los tipos de datos, por ejemplo, restricciones de tipos de datos y facetas para restringir un tipo de datos a un subconjunto de sus valores.
- Las capacidades de anotaciones extensas para anotar entidades, ontologías y también axiomas.
- Otras innovaciones importantes son: las declaraciones, los perfiles de los idiomas nuevos.

Los lenguajes para representar ontologías responden a los requerimientos de la arquitectura de capas de la web semántica que se muestra en la Figura 1.

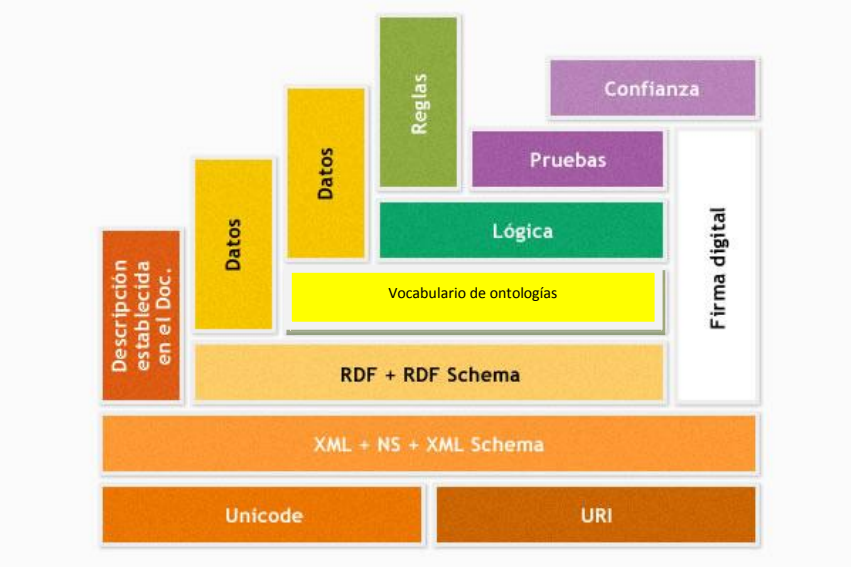


Figura. 1 Arquitectura de capas de la web semántica**(Berners-Lee, Hendler, & Lassila, 2001)**

2.2.4 Razonadores

Uno de los aspectos de la inteligencia es la capacidad de razonar, es decir, la capacidad de obtener información nueva de la ya disponible. En la IA existen diversas técnicas para hacer razonamiento aplicando en diferentes estrategias de inferencia sobre el conocimiento con esto permiten la obtención de conocimiento nuevo, éstas responden en gran parte a la naturaleza del mismo, pero también dependerá del tipo de manipulación que se vaya a realizar, según los objetivos que se persigan.

Los razonadores pueden ser agrupados en dos categorías: razonadores de lógica descriptiva y razonadores de programación lógica.

Los elementos principales de los que habla una lógica descriptiva son:

– Objetos.

– Conceptos (clases), que son conjuntos de objetos.

– Roles (relaciones), que son usados para especificar propiedades o atributos de objetos.

– Tipos de datos (opcional) a los cuales pertenecen los objetos.

Razonadores como Pellet, HermiT, FacT++, Racer Pro proporcionan los siguientes servicios (Polo, Berrueta, Rubiera, & Fernández, 2007):

- **Validación de la consistencia de una ontología:** el razonador puede comprobar si una ontología contiene hechos contradictorios.
- **Validación del cumplimiento de los conceptos de la ontología:** el razonador determina si es posible que una clase tenga instancias. En el caso de que un concepto no sea satisfecho, la ontología es inconsistente.
- **Clasificación de la ontología:** el razonador calcula a partir de los axiomas declarados en el TBox, las relaciones de subclase entre todos los conceptos declarados explícitamente, a fin de construir la jerarquía de clases.

- **Posibilita la resolución de consultas**: a partir de la jerarquía de clases, se pueden formular consultas como conocer todas las subclases de un concepto, inferir subclases nuevas, recuperar las superclases directas, etc.
- **Precisiones sobre los conceptos de la jerarquía:** el razonador puede inferir cuáles son las clases a las que pertenece directamente un concepto o individuo y mediante la jerarquía inferida obtener todas las clases a las que pertenece indirectamente.

En esta tesis el razonador nos permitirá verificar la consistencia dentro de la ontología de una manera automática además nos permitirá realizar búsquedas donde podremos inferir información.

2.3 Clasificación de documentos

En bibliotecas digitales, los documentos comúnmente se organizan con base en una taxonomía o jerarquía de temas que representa el consenso de expertos en un dominio. En Ciencias de la Computación, la ACM[10]desde 1964 emplea el sistema de clasificación computacional (*Computing Classification System*, CCS) para clasificar documentos, indexar, buscar y encontrar semejanzas, crear perfiles de autor e identificar áreas de investigación en perfiles institucionales, en aplicaciones y proyectos propios de investigadores e instituciones. La Tabla 1 y 2 muestran las clases principales y las diferencias de las últimas dos versiones de CCS, respectivamente, ambas están disponibles en formato XML desde la página de ACM.

[10]Siglas de *Association for Computing Machinery*, la sociedad internacional de cómputo reconocida a nivel mundial

Tabla 1. Clases principales en la versión de 1998 y 2012 de CCS

CCS versión 1998	CCS versión 2012
A. Literatura general B. Hardware C. Organización de sistemas informáticos D. Software E. Datos F. La teoría de la computación G. Matemáticas de la computación H. Sistemas de información I. Metodologías de computación J. Aplicaciones informáticas K. Ambientes de cómputo suave	• General y de referencia • Hardware • Organización de sistemas informáticos • Redes • Software e ingeniería • Teoría de la computación • Las matemáticas de la computación • Los sistemas de información • Seguridad y privacidad • Computación centrada en el humano. • Metodologías de computación • Computación aplicada • Temas sociales y profesionales • Nombres propios: personas, tecnologías y empresas

Tabla 2. Diferencias entre la versión de 1998 y 2012 de CCS

CCS	Versión 1998	Versión 2012
Estructura poli-jerárquica	No	Si
Letra y números de codificación	Si	No
Disponible en SKOS[11]	No	Si
Número de clases principales	11	14
Disponible en OWL	Si	No

La versión de 2012 se representa como una ontología disponible en formato SKOS, proporciona soporte para datos vinculados a las aplicaciones web. De acuerdo a (Rous, 2012), se considera un mapa cognitivo moderno del campo de la informática.

[11]*Simple Knowledge Organization* System es una iniciativa del W3C en forma de aplicación de RDF que proporciona un modelo para representar la estructura básica y el contenido de esquemas conceptuales.

En esta tesis trata de cómo se puede dar manteniendo a una colección de documentos existente, es decir, cuando un documento nuevo se agrega a la colección y es clasificado correctamente; Para clasificar un documento de las ciencias computacionales anteriormente se utilizaba el sistemas clasificador computacional de ACM versión 1998, pero desde enero de 2013 se utiliza la versión 2012; Un ejemplo de los datos descriptivos de un documento clasificado en esta versión se muestra en la tabla 3.

Tabla 3. Documento perteneciente a DL de ACM

Nombre del libro: Una taxonomía de las redes de datos para el intercambio de datos distribuida, gestión y procesamiento.
Autores: Srikumar Venugopal, Rajkumar Buyya y Kotagiri Ramamohanara.
Publicado en: Diario ACM Computing Surveys Volume 38 Issue 1, 2006.
Resumen: Redes de datos se han adoptado como la plataforma de próxima generación de muchas comunidades científicas que necesitan compartir, acceso, transporte, proceso y gestionar grandes colecciones de datos distribuidos en todo el mundo. Se combinan las tecnologías de computación de gama alta con la creación de redes de alto rendimiento y técnicas de gestión de almacenamiento de área amplia. En este artículo se analizan los conceptos fundamentales detrás de cuadrículas de datos y los comparamos con otros el intercambio de datos y los paradigmas de distribución, tales como las redes de entrega de contenido, redes peer-to-peer, y bases de datos distribuidas. Luego se dedica a taxonomías integrales que abarcan diversos aspectos de la arquitectura, el transporte de datos, replicación de datos y la asignación de recursos y la programación. Por último, hacemos un mapa de la taxonomía propuesta a varios sistemas de cuadrícula de datos no sólo para validar la taxonomía, sino también para identificar áreas para la exploración futura.

La aplicación ACM Digital Library (DL) es la colección más completa de artículos de texto y registros bibliográficos que existen hoy en día, cubren los campos de la tecnología informática y de información. La base de datos incluye revistas, actas de congresos y boletines. La ACM Digital Library se indexa con el sistema de clasificación computacional (CCS) versión 2012 de ACM, cuenta con servicios de letras, tiene formato de exportación como BibTex[12] así como el cumplimiento de OpenURL[13] (ACM, 2012)

[12]Facilita la realización de citas bibliográficas de un modo consistente mediante la separación de la información bibliográfica de la presentación de esta información.
[13] Es un tipo de URL que contiene metadatos para su uso fundamentalmente en bibliotecas.

Es con esta información se puede clasificar el documento en varias manera; según la librería digital de ACM (ACM, 2012) [14] como sigue: En la versión 1998 se clasifican por medio de los temas identificados dentro del documento como se muestra en el siguiente ejemplo:

Lista de clases jerarquías por versión	
1998	2012
Clasificación Principal: H. Sistemas de Información H.3 INFORMACIÓN almacenamiento y recuperación H.3.4 Sistemas y Software	Los sistemas de información Sistemas de almacenamiento de información. Arquitecturas de almacenamiento Almacenamiento distribuido

Tabla 4: lista de clases jerarquías por versión.

El Sistema de Clasificación de 1998 es una revisión actualizada de la versión 1991 del sistema de clasificación que se organiza en forma de árbol, esto hace que el formato sea más fácil de representar en una estructura jerárquica que en un formato de publicación lineal.

El árbol de clasificación se limita a tres niveles, a fin de reflejar con precisión la estructura esencial de la disciplina durante un período prolongado.

Está formado por 11 nodos de primer nivel y uno o dos niveles en cada una de ellas. El conjunto de los hijos de todos los de primer y segundo nivel de nodos comienza con un nodo General y termina con un nodo Varios. Los nodos de primer nivel tienen designaciones de letras de la A hasta K. Los segundo y tercer niveles de combinación tienen designaciones de letras y números.

En el uso real de la clasificación, los nodos de primer nivel como *B. Hardware* nunca se usan para clasificar el material. Para el material en un nivel general, se utiliza el nodo General en este caso *B.0* en su lugar. El nodo General en el primer o segundo

[14] http://dl.acm.org/citation.cfm?id=1132952.1132955&coll=DL&dl=GUIDE&CFID=238187228&CFTOKEN=14066677

nivel puede servir a dos propósitos: se utiliza para los documentos que incluyen tratamientos amplios del tema comprendido en su nodo principal el nodo inmediatamente anterior en el árbol, o puede cubrir varios temas relacionados con algunos pero no necesariamente todos sus nodos hermanos. Por ejemplo, bajo *K.7 La Profesión Informática* el nodo *K.7.0 general* se utiliza para clasificar un artículo general sobre la profesión informática, sino que también podría ser utilizado para un artículo que trata específicamente de *K.7.1 Ocupaciones de computación K.7.2 Organizaciones* y *K.7.3 ensayos, certificación y concesión de licencias*.

En el sistema clasificador 2012 según la librería digital asigna las clases al documento de la siguiente manera ver la figura 2.

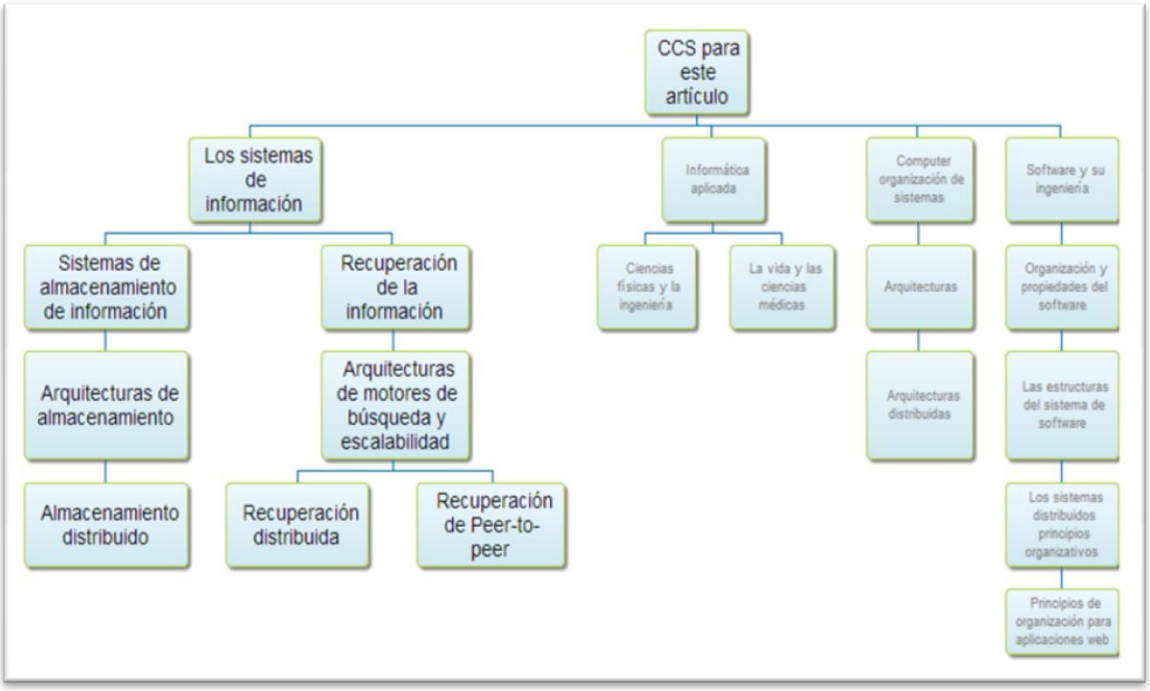


Figura. 2 Asignación de clases en la versión 2012.

El sistema de Clasificación ver 2012 mantiene una jerarquía de nivel n y no utiliza números de clasificación, por lo que no hay distinción entre los niveles esto hace que la clasificación sea poli-jerárquica.

2.4 Trabajos relacionados

Las ontologías se han utilizado para clasificar y mantener colecciones de documentos en diferentes dominios, por ejemplo, en el área de la salud, el proyecto GALEN[15] financiado por la Unión Europea, las emplea para clasificar procedimientos quirúrgicos (Sandoval, 2007), o en ciencias naturales, la iniciativa mundial de clasificación (IMT) adoptada en mayo de 2000 las utiliza para unificar los diversos clasificadores y apoyar la implementación del Convenido sobre la Diversidad Biológica de las Naciones Unidas (Naciones Unidas, 2012). En Ciencias de la Computación, el uso de la versión de 1998 de CCS se ha reportado en trabajos como los siguientes:

El consorcio AKT[16] agrupa a 5 universidades del Reino Unido y fue fundado por el *Engineering and Physical Sciences Research Council* (EPSRC). Su fin es ayudar a desarrollar tecnologías del conocimiento para dar soporte a las organizaciones. Uno de los proyectos de AKT consistió en codificar la versión de 1998 de CCS a OWL en el idioma inglés. Una traducción para el francés se realizó en el contexto del proyecto OntologyNavigator, realizado por la Universidad París 8 Vincennes-Saint Denis; la ontología producida se emplea en un software de apoyo a la investigación documental que ayuda a los estudiantes de posgrado a encontrar documentos científicos. Los usuarios interactúan con la traducción al proporcionar traducciones alternativa y enriquecer las etiquetas de los nodos de la ontología al favorecer la retroalimentación (Kembellec, Saleh, & Catalina, 2009).

En la tesis (Godínez, 2009), se presenta una herramienta de software que permite identificar tendencias de uso del acervo bibliográfico y describir la evolución en una disciplina del conocimiento científico cuyos recursos de información se encuentran clasificados haciendo uso de ontologías y el CCS versión 1998, analiza la producción

[15]General Architecture for Languages Encyclopedias and Nomenclatures in Medicine.
[16] Advanced Knowledge Technologies

de los recursos de información de ciencia y tecnología y su visualización en gráficas con respecto al tiempo.

En la tesis de doctoral de (Gábor, 2005), trata el estudio de cómo mejorar la eficiencia de recuperación de información de un dominio de la ciencia en este caso es la Historia, en este trabajo examina cómo las ontologías se puede explotar durante el proceso de recuperación de información y sostiene la hipótesis de que Las ontologías permiten almacenar el conocimiento del dominio mucho más sofisticado que un tesauros[17]. Por lo tanto, supone que mediante el uso de ontologías en recuperación de información en un sistema mejora la efectividad.

Además trata de la relación que existe entre la complejidad del algoritmo y el rendimiento cuando son empleadas ontologías. Gábor dice que la ontología no tiene procedimientos de razonamiento manejables computacionalmente. Sin embargo, demuestra que es que mediante la combinación de ontologías con los métodos tradicionales de recuperación de información es posible proporcionar resultados con un rendimiento aceptable, y así satisfacer las necesidades para los repositorios de gran tamaño que son lo que existen en el mundo real.

Basado en lo expuesto por Gábor, es conveniente utilizar algoritmos tradicionales para la recuperación de información que reducirían la carga computacional, en este trabajo de tesis se utiliza métodos tradicionales para la recuperación de información como el método de frecuencia directa y frecuencia inversa apoyado con el uso de una ontología codifica sobre OWL.

En el proyecto Conversión de Clasificaciones en ontologías OWL realizado por (Giunchiglia, Zaihrayeu, & Farazi, 2009) de la Universidad de Trento, Italia, describe la forma en que un esquema de clasificación se pueden convertir en ontologías y también trata algunos puntos que se deben tomar en cuenta a la hora de estas codificando la ontología en OWL; Al tomar en cuenta estos puntos se logra que la

[17] Es una lista que contiene los términos empleados para representar los conceptos, temas o contenidos de los documentos.

ontología se consistente, Giunchiglia sugiere la utilización los razonadores lógicos Pellet y Fact ++ que se encuentran adicionados en Protégé para detectar inconsistencias en la clasificación, esta sugerencia es muy importante ya que en este tema de tesis es la herramienta que ha seleccionado para verificar la consistencia.

Con el proyecto Conversión de Clasificaciones en ontologías OWL nos dice que podemos partir de un esquema conceptual como lo que ya tenemos el CCS de ACM versión 2012, y luego la estructura del sistema de clasificación esto se logra al organizar las clases son clases padre que clases son hijos y que clases son excluyentes o que clases son equivalente. En el siguiente capítulo se trata a detalle como en esta tesis se codifico la estructura del CCS de ACM, siempre tomando en cuenta que se debe mantener la constencia de la ontología.

Capítulo III Codificación y búsquedas en la ontología.

En este capítulo se trata el cómo se ha codificado el Sistema Clasificador Computacional de ACM versión 2012 en lenguaje OWL. Hay que recordar que en su versión de liberación se codificó en SKOS. En teoría este lenguaje puede ser exportado fácilmente a OWL, pero aquí se explica la serie de complicaciones que se enfrentaron para este fin. Además, se describe el proceso de cuando se añade un documento nuevo en una colección de documentos y esto tiene que ser evaluado para identificar la clase apropiada se hace uso taxonomía y en concreto el sistema de clasificación de computación ACM versión 2012. Para comprobar la validez de la ontología se elaboraron algunas consultas en el del servidor http Fuseki y lenguaje de consulta SPARQL[18]; Para garantizar la consistencia de la información se utilizó el razonador de Jena.

3.1 Codificación de la ontología.

Uno de los propósitos del CCS es clasificar documentos; para lograr este fin, se propone organizarlos en clases. Se asume que cada documento tiene un identificador único.

La jerarquización y la organización de las clases y sub clases se basan en el CCS de ACM la versión 2012, fue necesario codificar esta ontología ya que ACM sólo proporciona el diseño estructural, pero no proporciona la ontología en ningún lenguaje con el que se pueda realizar inferencia. Para la codificación de la ontología se empleó Protégé un editor desarrollado por la Universidad de Stanford, tiene interfaz para la codificación de RFD y OWL, puede ser empleado por razonadores lógicos para realizar tareas de inferencia.

La ontología basada en CCS ver 2012, se codificó en inglés y español, a través de clases de equivalencia por ejemplo las, <Sistema de Clasificación computacional

[18] Protocol and RDF Query Language. Se trata de un lenguaje estandarizado para la consulta de grafos RDF

2012 de ACM> es equivalente a <The 2012 ACM Computing Classification System> con esto se logra usar los dos idiomas.

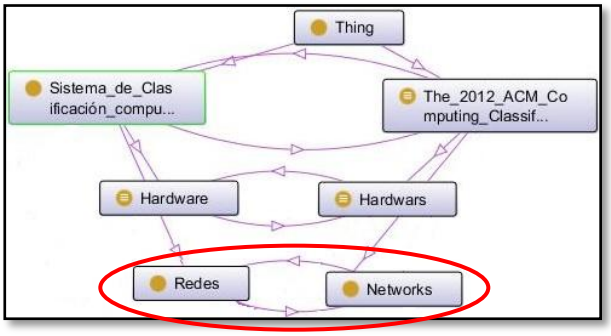


Figura 3. Miembros de una clase.

Se definieron clases disjuntas, por ejemplo, Hardware es distinta a la clase Redes con la siguiente ternas se cumple este objetivo: <Hardware><*disjointWith*><Redes>.

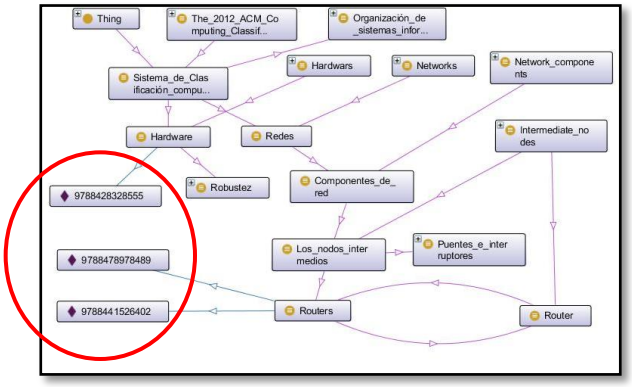


Figura 4. Equivalencias de clases

Para colocar una instancia o elemento nuevo en una clase, se siguen los siguiente pasos, primero se le asigna como nombre de este miembro como un código único como se ve en la línea 1, en este caso es el código ISBN[19] con lo que aseguramos que es un código único e irrepetible, además se le incorporan los metadatos que se moldean como anotaciones heredadas como se ve en la líneas 3 a la 7, además se

[19] Número Estándar Internacional de Libros o Número Internacional Normalizado del Libro abreviado ISBN, es un identificador único para libros, previsto para uso comercial.

indica a qué clases de la ontología pertenece, en este caso pertenece a Hardware como se muestra en la línea 2 de la tabla 4; esta línea es muy importante, y no es necesario que se le agreguen las clases padre como en la anterior versión, ya que con una sola clase se puede inferir todas las clases ancestro.

El código que se encuentra en la figura 4, es una de las aportaciones de esta tesis, ya que con estas líneas se puede se integra un nuevo documento a la ontología sin afectar la consistencia. Para poder cumplir con los requerimientos de Dublin Core fue necesario crear propiedades de datos jerárquicas, en la line 3 podemos ver <*onto:DC.Title*> que es un ejemplo de esto.

```
1.<owl:NamedIndividual rdf:about="&ont;9788428328555">
2.      <rdf:type rdf:resource="&ont;Hardware"/>
3.      <ont:DC.Title>MONTAJE, CONFIGURACIÓN Y REPARACIÓN DEL PC</ont:DC.Title>
4.      <ont:DC.Publisher>PARANINFO</ont:DC.Publisher>
5.      <ace_lexicon:PN_sg>9788428328555</ace_lexicon:PN_sg>
6.      <ont:DC.Language>Español</ont:DC.Language>
7.      <ont:DC.Creator>David Zurdo Saiz</ont:DC.Creator>
8.</owl:NamedIndividual>
```

Tabla 5 Código para la inserción de una elemento en una clase

3.2 Herramientas.

Existen herramientas que ayudan a representar lo que existe en el mundo real, cimentándose en la sintaxis de la lógica descriptiva; mediante estas herramientas se establecen las reglas y los predicados que son la base para el razonamiento y la inferencia para la representación del conocimiento.

Las herramientas que se utilizaron es este proyecto son:

- **Protégé**: editor para RDF y OWL. OWL es un lenguaje de definición de diccionarios semánticos creado por el W3C, OWL o “Lenguaje de Ontologías Web” fue desarrollado para facilitar el procesamiento en aplicaciones que necesitan interpretar e inferir el contenido de la información de algún documento. OWL interpreta el contenido web con mayor precisión XML, RDF, y esquema RDF (RDF-S), proporciona vocabulario adicional junto con una semántica formal. OWL se puede formular en RDF incluye toda la capacidad

expresiva de RDF y RDF-S y la extiende con la posibilidad de utilizar expresiones lógicas.

- **Jena**: soporta los formatos RDF/XML, N3, N-Triples y OWL. En Jena se almacena la información como tripletas o ternas de RDF en grafos dirigidos, esto permite que su código pueda agregar, eliminar, manipular, almacenar y publicar esa información (Prud & Seaborne, 2013). Jena proporciona los medios para que estas ternas que contienen información sean inferidas de manera transparente al usuario.
- **Fuseki:** es una plataforma servicio de HTTP con esto se proporciona la facilidad de realizar alta, consulta y actualizaciones, por medio del lenguaje de consulta SPARQL.

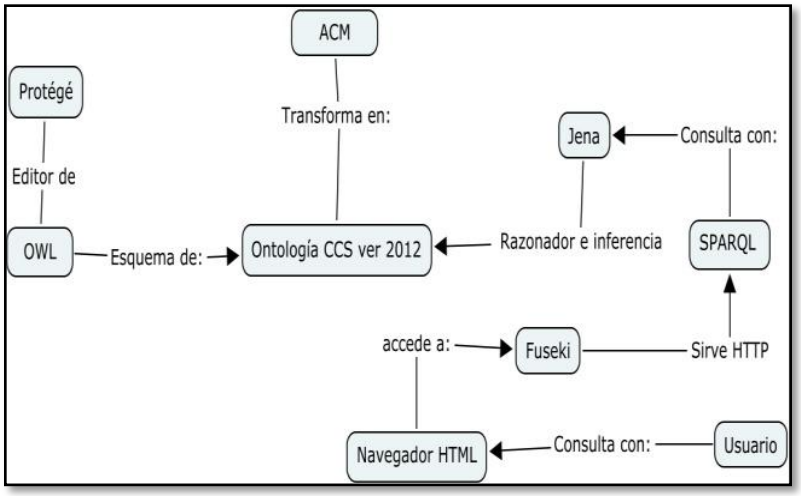


Figura 5. Mapa de transferencia de información

En la figura 5 se ve cómo interactúan todas las herramientas que son empleadas para realizar una consulta dentro de la ontología, cabe aclarar que estas herramientas son empleadas solo para el proceso de consulta ya que para otro proceso que están integrados en esta tesis, algunas de estas herramientas no serán necesaria.

3.3 Consultas en la ontología.

SPARQL es un acrónimo del inglés “Protocol and RDF Query Language”. Es un lenguaje estandarizado para la consulta de grafo RDF, normalizado por el RDF Data Access Working Group (DAWG) del World Wide Web Consortium (W3C) (Prud & Seaborne, 2013).

Al igual que sucede con SQL, es necesario distinguir entre el lenguaje de consulta y el motor para el almacenamiento y recuperación de los datos. Existen múltiples implementaciones de SPARQL, generalmente ligados a entornos de desarrollo y plataforma tecnológica, en un principio SPARQL únicamente incorpora funciones para la recuperación sentencias RDF, sin embargo, algunas propuestas también incluyen operaciones como: creación, modificación y borrado de datos.

En esta tesis se describen consultas de ejemplo para mostrar la utilidad de la ontología construida:

El grupo de prefijos, encabezado, cabeceras o URIs de tabla 5 son importante porque señalan la ubicación donde se encuentran la entidad a consultar, La palabra PREFIX permite abreviar el lugar donde se encuentra la entidad, en esta tesis se usa está cabeceras para todas las consultas.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prop: <http://www.co-ode.org/ontologies/ont.owl#>
```

Tabla 6: Prefijos de consultas.

- **Consulta 1:** información de un libro que tiene como código de ISBN 9788428328555.

```
***Cabecera de prefijos ***
SELECT  ?Propiedades
WHERE {prop:9788428328555 ?p ?Propiedades.
            FILTER (?p != rdf:type)}
```

Tabla 7: Consulta uso de filtro.

En esta consulta muestra los datos de libro por medio de un identificador en este caso se emplea un filtro para excluir la información que no es útil, observe en la tabla 7 los resultados, solo muestra los datos del documento.

- **Resultados:**

Propiedades
"MONTAJE, CONFIGURACIÓN Y REPARACIÓN DEL PC"
"PARANINFO"
"9788428328555"
"Español"
"David Zurdo Saiz"

Tabla 8: Resultados de la consulta uso de filtro.

- **Consulta 2**: Recupera todos los libros que pertenecen a la clase Routers.

```
*****Cabecera de prefijos *****
SELECT ?ISBN ?Propiedades
WHERE {{?ISBN rdf:type prop:Routers.}
optional {?ISBN ?p ?Propiedades.
FILTER (?p != rdf:type && ?p != lexico:PN_sg ) }
}
```

Tabla 9: Consulta de una clase específica.

En la tabla 10 muestra los resultados de la consulta 2, para ser más facilitar la visualización de los datos de esta consulta se le agregó el campo metadatos, con esto se quiere decir que el servidor no entregará esa columna y el filtro que incluye esta búsqueda quita las ternas que contiene datos que no interesan al usuario, al ser presentados en la pantalla.

- **Resultados:**

ISBN	Propiedades	Metadatos
9788441526402	"ROUTERS CISCO: EDICION REVISADA Y ACTUALIZADA 2010 (GUIA PRACTICA)"	Título
9788441526402	"ANTONIO GALLEGO DE TORRES"	Autor

9788441526402	"ANAYA MULTIMEDIA"	Editorial
9788441526402	"Español"	Idioma
9788478978489	"TÉCNICAS DE CONFIGURACIÓN DE ROUTERS CISCO"	Título
9788478978489	" ERNESTO ARIGANELLO"	Autor
9788478978489	"AG Canarias"	Editorial
9788478978489	"Español"	Idioma

Tabla 10: Resultado de la consulta de una clase específica.

Consulta 3: Muestra los nombres de las clases a las que pertenece un libro que tiene como identificador el ISBN 9788441526402 en otras palabra muestra todos las clases ancestros que anteceden a la clase del libro con el identificador el ISBN 9788441526402. Cabe señalar que la consulta de la tabla 11 se realiza por medio de ternas es por esta razón que es necesario que la clase hijo exista y así poder obtener la clase ancestro y así sucesivamente para poder inferir los antecesores.

```
Cabecera de prefijos
SELECT *
WHERE {?ISBN rdf:type ?Clase.
FILTER (?ISBN = prop:9788441526402)
      ?Clase  rdf:type owl:Class.
       optional{?Clase  rdfs:subClassOf  ?Ancestro1.}
       optional{?Ancestro1  rdfs:subClassOf  ?Ancestro2.}
       optional{?Ancestro2  rdfs:subClassOf  ?Ancestro3.}
       optional{?Ancestro3  rdfs:subClassOf  ?Ancestro4.}
       optional{?Ancestro4  rdfs:subClassOf  ?Ancestro5.}
}
```

Tabla 11: Consulta muestra los nombre de las clases.

- **Resultados:**

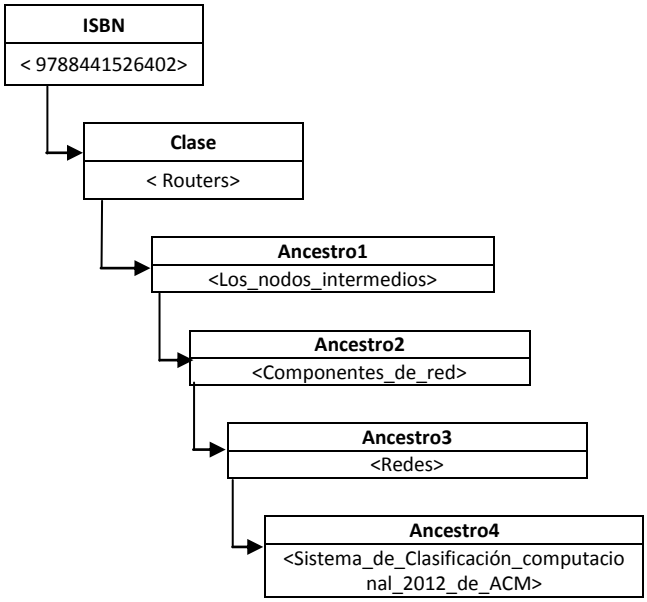


Tabla 12: Consulta muestra los nombre de las clases.

Resultados parciales

Con esta serie de consultas ya se han podido visualizar algunos resultados parciales que tiene el trabajar con ontologías, los primeros se en lista aquí:

- La ontología cuenta con una estructura y codificación consistentes
- La sintaxis de codificación es correcta.
- La ontología puede ser explotada desde un servidor HTTP.
- Los miembros de una clase puede contener uno o más meta datos.
- Es necesario contar con un identificar único para no caer en inconsistencias dentro de la ontología.

En el siguiente capítulo se trata como con la ontología se integra con método tradicional de recuperación de información sin nunca perder la consistencia lograda en este capitulo

Capítulo IV Recuperación de información.

En este capítulo se trata de la implementación de una interfaz para el método automático de asignación que apoye la búsqueda y recuperación en una colección organizada por una ontología. Hay que recordar que la utilidad de esta aplicación se encuentra en las bibliotecas digitales y en particular podría ser de interés para los participantes en el proyecto REMERI[20] el que agrupa varias universidades del país; Sin embargo, la aplicación será de utilidad para mantener colecciones que implementen el protocolo OAI-PMH.

El esta parte se explica él como se le asignar una clase utilizando la ontología CCS de ACM 2012 cuando nuevo documento es ingresado a la colección ya existente, este documento nuevo se compara con todos los documento que ya están clasificados en la colección, y utilizando métodos de extracción de información tradicionales se le asigna un clase.

Para la realizar esta parte fueron utilizadas varias herramientas de programación pero la característica en común de estas herramientas es que fueron desarrollado bajo plataforma libre y gratuita.

Los servicios de web son proporcionados por un servidor XAMPP versión 3.1.0.1.3.1.0 el que cuenta con apache server, PHP Server y Mysql server; Además para programar la interfaz se empleo PHP 5.4.7, Ajax y JavaScript; Con el fin de aprovechar la potencialidad que ofrece el manejador de bases de datos se empleo leguaje de consultas SQL; También se hace uso de OWL ya que con este, brinda las facilidades de verificar la consistencia de la ontología.

4.1 metadatos

Literalmente, un metadato es “algo” que está por sobre los datos. Una definición comúnmente utilizada es "datos sobre datos", es decir que son objetos de información que describen o dicen algo sobre otro objeto de información.

[20] Red Mexicana de Repositorios Institucionales.

Los metadatos son utilizados en ambientes de bibliotecas y sistemas documentales. Un caso común es asociar autor, título y fecha a una publicación particular con el objetivo de organizarlo bajo una estructura definida. Esto sirve para minimizar esfuerzos de organización y facilitar su mantenimiento.

Existen normas que definen diferentes estructuras de metadatos, las cuales persiguen objetivos diferentes, por ejemplo:

- Dublin Core (DC) (DCMI, 2013) .Referentes descripciones de los recursos.
- Consortium for the Interchange of Museum Information[21] (CIMI) Información sobre museos.
- Federal Data Geographic Committee[22] (FGDC). Descripción de datos geoespaciales.

Estas normas definen los estándares que especifican la sintaxis para generar estructuras de datos y proveen especificaciones semánticas necesarias que explicar el significado de las expresiones sintácticas.

Actualmente, a los efectos de ayudar a solucionar el problema de sobrecarga de información derivada de la constante expansión del espacio web se promueve el uso de la denominada web semántica. La cual tiene entre sus objetivos modificar la forma en que se presenta la información en el espacio web en pos de facilitar el procesamiento automático de la misma, y de esta forma establecer facilidades para lograr un factible procesamiento, integración y reutilización de la información contenida en tal espacio. Los metadatos juegan un rol importante en el área mencionada, debido a que proveen una categorización semántica de su contenido, permitiendo razonar de forma automática sobre la información.

[21] http://old.cni.org/pub/CIMI/framework.html.
[22] http://www.fgdc.gov/

4.1.2 Iniciativa de metadatos Dublin Core

La iniciativa mundial de Dublin Core propone la idea de estandarización de metadatos básicos para las descripciones de los recursos. Dublin Core (DC) es un sistemas de 15 definiciones semánticas descriptiva, con estas definiciones DC logro amplia difusión al unirse con la “Iniciativa de Archivos Protocolo abiertos para la recolección de metadatos” (OAI-PMH) y ha sido ratificado como IETF RFC 5013, ANSI / NISO Z39.85-2007 estándar, y la norma ISO 15836 : 2009 (DCMI, 2013).

A partir de 2000, la comunidad de DC se centró en la idea de que los registros de metadatos DC se usarían junto con otros vocabularios especializados para satisfacer las necesidades de aplicación específicas. Durante ese tiempo, la Wide Web Consortium genero un modelo para los metadatos, el Resource Description Framework (RDF). DC se convirtió en uno de los vocabularios más populares para el uso con RDF.

Los 15 elementos que conforman a DC están organizados en tres grupos que indican la clase o el ámbito de la información que se guarda en ellos:

Contenido:

1. **DC.Title (***Título)* el nombre dado a un recurso, habitualmente por el autor.
2. **DC.Subject:** (Palabras *claves*) trata los temas del recurso. Típicamente, Subject expresará las claves o frases que describen el título o el contenido del recurso. Se fomenta el uso de vocabularios controlados y de sistemas de clasificación formales, como es el CCS de ACM Versión 2012.
3. **DC.Description:** *(Descripción)* una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual.
4. **DC.Source:** *(Fuente)* secuencia de caracteres usados para identificar un trabajo a partir del cual proviene el recurso actual.

5. **DC.Type:** *(Tipo del Recurso)* la categoría del recurso. Por ejemplo, página personal, romance, poema, diccionario, etc.
6. **DC.Relation:** *(Relación)* es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos.
7. **DC.Coverage:** *(Cobertura)* es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso.
    - La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas.
    - La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado (que ya lo encontramos en el elemento Date).

Propiedad Intelectual:

8. **DC.Creator:** *(Autor o Creador)* la persona o organización responsable de la creación del contenido intelectual del recurso. Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de recursos visuales.
9. **DC.Publisher:** *(Editor)* la entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual.
10. **DC.Contributor:** *(Otros Colaboradores)* una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento Creator. (por ejemplo: editor, ilustrador y traductor).
11. **DC.Rights** *(Derechos)* son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso.

Instanciación:

12. **DC.Date** *(Fecha)* una fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la

medida que el contenido intelectual está de alguna manera relacionado con aquella fecha.

13. **DC.Format:** *(Formato)* es el formato de datos de un recurso, usado para identificar el software y posiblemente el hardware que se necesitaría para mostrar el recurso.
14. **DC.Identifier** *(Identificador del Recurso)* secuencia de caracteres únicos utilizados para identificar un recurso. Ejemplos para recursos en línea pueden ser URLs y URNs. Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo ISBN ("International Standard Book Number").
15. **DC.Language** *(Lengua)* lenguaje en el que se encuentra el contenido del recurso.

4.2 Repositorio OntOAIr

El propósito de este repositorio institucional es apoyar la gestión, el almacenamiento, la preservación, la organización y la diseminación de recursos académicos de la UPPuebla (Universidad Politécnica de Puebla, 2012). Este repositorio se encuentra construido bajo el estándar de OAI-PMH[23], este protocolo permite compartir los recursos académicos que se estén elaborando dentro de la Universidad Politécnica de Puebla.

El protocolo de la iniciativa de archivos abiertos para la recolección de metadatos (OAI-PMH) es una propuesta para auxiliar interoperabilidad repositorio. Proveedores de datos son depósitos que exponen metadatos estructurados mediante el protocolo OAI-PMH.

OAI-PMH es un conjunto de seis verbos o servicios que se invocan en HTTP (Lagoze & Van de Sompel, 2008):

- **Identify (?verb=Identify).** Accede a los datos de identificación del repositorio

[23] Siglas en ingles Open Archives Initiative Protocol for Metadata Harvesting.

- **GetRecord (?verb=GetRecord)**. Recupera los metadatos de un registro
- **ListIdentifiers (?verb=ListIdentifiers).** Devuelve una lista de los identificadores de los registros.
- **ListMetadataFormats (?verb=ListMetadataFormats).** Recupera los tipos de formatos de metadatos de los registros.
- **ListRecords (?verb=ListRecords).** Accede a la lista de registros.
- **ListSets (?verb=ListSets).** Recupera los nombres de los grupos utilizados para organizar los registros por tema.

4.2.1 Estructura de repositorio.

Esta es la estructura para los repositorios que se empleó y será utilizó para agregar un nuevo documento en RDF; Esta estructura cumple con el protocolo OAI y con lo establecido con DC incluida en proyecto REMERI.

La primera parte son la cabeceras las que indican los requisitos y protocolos utilizados.

```
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2012-06-13T19:20:47Z</responseDate>
<request identifier="http://172.16.30.170/~oai_uppuebla/index.html" metadataPrefix="oai_dc" verb="GetRecord">
</request>
```

En esta parte se tienen identificado cada unos de los elementos de un documento bajo los lineamientos que propone iniciativa DC, para este proyecto de tesis los metadatos más importante son:

- DC.Identifier es el identificador único para el documento que puede ser ISBN o algún otro que se propongan.
- DC.Title el titulo del documento regularmente asignado por el autor.

- DC.Description es un breve resumen del contenido del documento que regularmente lo realiza el autor del documento.
- DC:subject este es el motivo de esta tesis ya que es aquí donde se coloca la clasificación de sé a estado explicando en capítulos anteriores en esta caso al solo tratar con documentos de la ciencias computaciones utilizamos el CCS de ACM versión 2012.

```
<GetRecord>
<record>
<header>
<identifier>172.16.30.170/~oai_uppuebla/rt/xique061100650.pdf</identifier>
<datestamp>2012-06-13T11:13:36Z</datestamp>
</header>
<metadata>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dc="http://purl.org/dc/elements/1.1/"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:coverage>Licenciatura</dc:coverage>
<dc:title> Web Semanticas</dc:title>
<dc:publisher>Universidad Politécnica de Puebla</dc:publisher>
<dc:creator>Everardo Carlos Guevara Hernandez </dc:creator>
<dc:contributor>Dr. Antonio Benitez</dc:contributor>
<dc:contributor>Dr. Jesus Antonio </dc:contributor>
<dc:contributor>Dra. María Auxilio Medina Nieto</dc:contributor>
<dc:contributor>M.C. Argelia Berenice Urbina Nájeta</dc:contributor>
<dc:language>es</dc:language>
<dc:type>Technical report</dc:type>
<dc:type>Reporte técnico</dc:type>

<dc:subject>H. Sistemas de información </dc:subject>
<dc:subject>H.4 Aplicaciones de sistemas de información </dc:subject>
<dc:subject>H.4.0 General </dc:subject>

<dc:subject>H. Information Systems </dc:subject>
<dc:subject>H.4 Information Systems Applications </dc:subject>
<dc:subject>H.4.0 General </dc:subject>

<dc:date>2003-05-09</dc:date>
<dc:identifier>
172.16.30.170/~oai_uppuebla/rt/xique061100650.pdf
</dc:identifier>
<dc:description> El presente proyecto de investigación expone una alternativa de apoyo a la gestión de los datos de los tutorados de Ingeniería en Informática de la UPPuebla, mediante la propuesta
de construcción dinámica de reportes gráficos sobre el desempeño académico de tutorados, desarrollada con el lenguaje de programación PHP en conjunto con la herramienta FushionCharts, con el objetivo de obtener de manera dinámica reportes en forma tabular y gráfica sobre el
desempeño de los tutorados en la UPPuebla. </dc:description>
```

```
<dc:format>application/pdf</dc:format>
```

En esta última parte del archivo RDF es donde se cierra todos los requerimientos del mismo archivo.

```
</oai_dc:dc>
</metadata>
</record>
</GetRecord>
</OAI-PMH>
```

El problema encontrado al realizar esta tesis es que si existen repositorios que ya cuenten con una clasificación pero en versión de ACM 1998, por lo que para llevar a cabo esta tesis fue necesario construir los repositorios de datos, se hiso necesario la captura de cada documento de manera manual y la fuente proviene de la Librería Digital de ACM que es la única que actualmente se encuentra clasificada bajo CCS versión 2012.

Para facilitar esta operación y para cumplir con los requerimientos de los estándares, se diseño e implemento una aplicación que facilitara esta tarea que más delante de describe.

4.3 Proyecto REMERI

La Red Abierta de Bibliotecas Digitales (RABID) forma parte de la comunidad de Bibliotecas Digitales de la Corporación de Universidades para el Desarrollo, siendo creada en abril de 2006 con la finalidad de contribuir al desarrollo de las bibliotecas digitales en México. A lo largo de cinco años, se han alcanzado importantes logros: el desarrollo conjunto de diversas aplicaciones y herramientas tecnológicas, la interoperabilidad de repositorios o colecciones digitales y el diseño de indicadores sobre cibermetría y estructuras de metadatos entre otros.

En la reunión de primavera 2011 de CUDI[24], se determina iniciar un nuevo proyecto denominado “Red Mexicana de Repositorios Institucionales (REMERI)”, para el cual se conforma un grupo de trabajo con miembros de nueve instituciones de educación superior públicas y privadas integrantes de RABID.

El objetivo general del proyecto REMERI es integrar una red federada de repositorios de acceso abierto de las instituciones de educación superior mexicanas, para dar visibilidad a su producción científica, académica y documental, y constituirse como el nodo nacional del Proyecto “Estrategia Regional y Marco de Interoperabilidad y Gestión para una Red Federada Latinoamericana de Repositorios Institucionales de Documentación Científica”.

Para ello se proponen acciones estratégicas y metas a mediano y largo plazo, así como los requerimientos para su implementación y el impacto esperado (Vázquez Tapia, 2012).

4.4 Diseño de interface.

Como ya se ha descrito en la parte anterior, los nuevos documentos que se van ingresar a la ontología codificada, a este proceso lo conocemos como “mantenimiento de colecciones”. Los como ya se menciono los reservorios existentes no fueron útiles por que se encontraban clasificado con la versión 1998, así que los reservorios tuvieron que capturarse manual mente, utilizamos la biblioteca digital de ACM, al generar un reservorio de 87 documentos lo que cumplían con las especificaciones descritas.

Para poder asignar una clase a un nuevo documento se selecciono el titulo del documento, la descripción que es un breve resumen además del asunto o palabra clave de esta manera fue tratada la información para poder ser ingresada.

[24] Corporación Universitaria para el Desarrollo de Internet.

La pantalla de construcción de repositorio, ver la figura 6 que tiene el nombre de “alta de documento” las áreas de esta pantalla de captura son:

1.- Metadatos cumplen con lo establecido por DC diseño de esta aplicación.

2.- Son áreas contiene mensajes dinámicos para ayuda a la captura de información.

3.- Esta zona llama a las clases de la ontología construida en OWL de forma dinámica realizando una búsqueda de cada clase para realizar una captura más amigable.

En importante mensionar que los campos que podemos estraer información relevante del documeto son los campos de titulo, descripcion y la clase los demas campos para el proceso de extraccion de infomacion fueron ignorados, ya que no son represtativos para asiganar una clase.

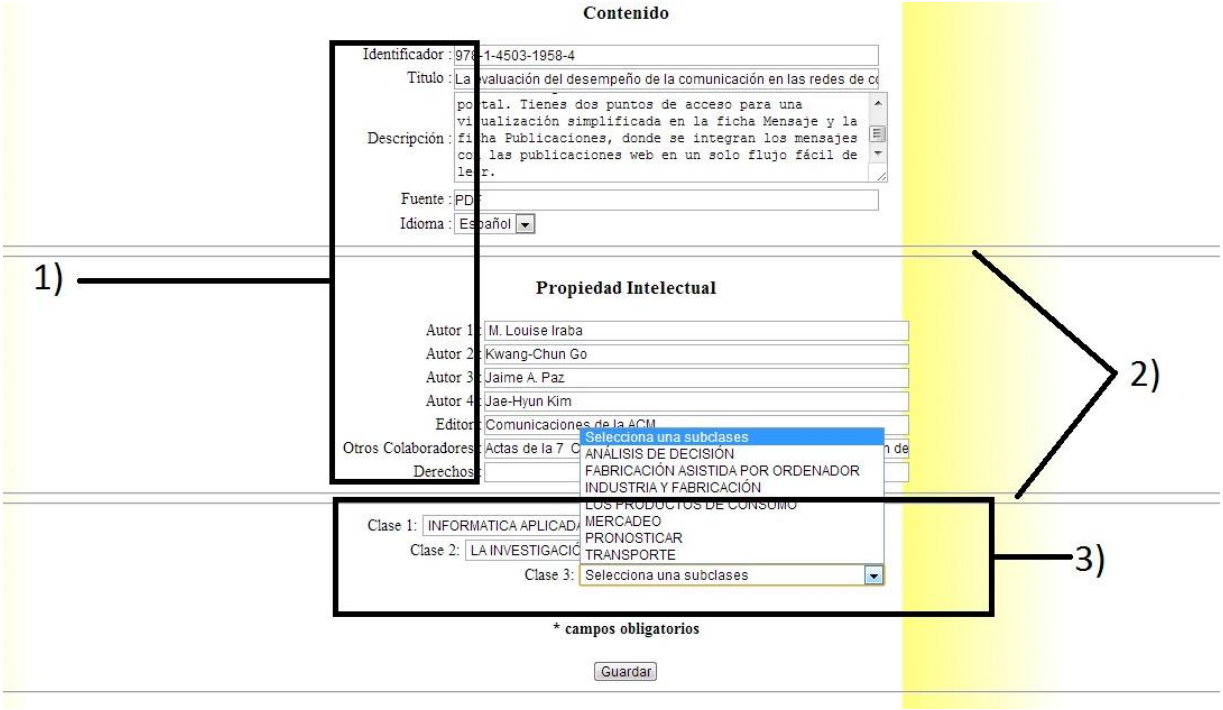


Figura 6. Pantalla de Captura

4.4.1 El proceso de recuperación de información.

La recuperación de información es un proceso de varias etapas que finaliza con la representación del contenido completo de la colección sobre las estructuras de datos definidas.

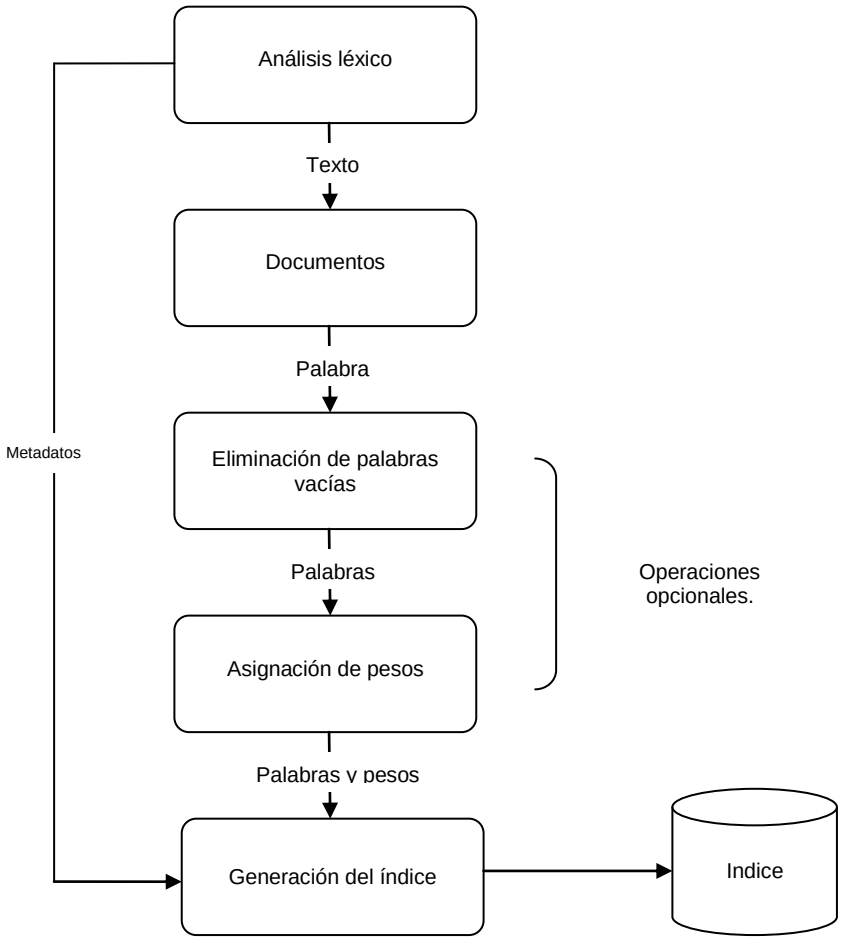


Figura 7. Proceso de recuperación de información

Las etapas que conforman ver figura 7 este proceso son las siguientes:

1. Diagrama de flujo del proceso de indexación
2. Análisis lexicográfico: Se extraen las palabras y se normalizan.
3. Eliminación de palabras vacías o de alta frecuencia en la colección.
4. Selección de los términos a indexar: Se extraen aquellas palabras simples o compuestas que mejor representan el contenido de los documentos.

5. Asignación de pesos o ponderación de los términos que componen los índices de cada documento. En algunos modelos de Recuperación de Información es fundamental asociar la importancia de un término en un documento a los efectos de mejorar las prestaciones.

Se debe tener en cuenta que esta es una descripción general y que en algunos casos se pueden omitir algunas etapas, mientras que otras varían de acuerdo a las necesidades particulares en la tarea posterior de recuperación.

4.4.2 Eliminación de palabras vacías.

Como primer paso se realizo la extracción de “stop words” que son aquellas palabras ignoradas en una consulta por los motores de búsqueda por ser muy comunes. El término stop word puede traducirse al español como “palabras vacías” y, a pesar de ser utilizadas naturalmente en la redacción de contenido, en las consultas de búsqueda tienen nula relevancia. Al eliminar palabras como “de”, “con”, “en”, etc., el motor de búsqueda puede encontrar los resultados más relevantes.

En los anexos 1 se presentan listas de palabras vacías extraídas del sistema SMART y la lengua española (Salton, 1971). Una ventaja adicional de la eliminación de éstos términos es la reducción del tamaño de almacenamiento necesario, que se llegan a reducir los archivos de índices en un 40% (Francis & Kucera, 1982).

El proceso de eliminación de palabras vacías fue auxiliado por el servidor mysql con el que utiliza la indexación que acelera las búsquedas, fue necesario crear base de datos llamada motor

Tabla Vacia	
Id	Int
palabra	String

La aplicación fue desarrollada en PHP el siguiente script permite realizar una extracción de las palabras vacías.

```
$pal=explode(" ",$linea);
  for($i=0;$i<count($pal);$i++)
    {
    $palabra='';
    $ls=0;
    $palaux=strtolower($pal[$i]);
    for($j=0;$j<strlen($palaux);$j++){
      for($r=1;$r<=32;$r++)
         if($puntos[$r]== $palaux[$j]) $ls=1;
      if($ls==0)
         $palabra=$palabra.$palaux[$j];
                                    }
          $palabra=trim($palabra, "\x00..\x1F");
    $result = mysql_query("SELECT * FROM vacia WHERE palabra ='".$palabra."'",$db);
        if (!mysql_fetch_array($result)){
         //$palabra=Spanish::Stemm($palabra);
           $result = mysql_query("SELECT * FROM peso WHERE palabra = '".$palabra."' and iddocumento='".$t."'",$db);
           if ($row = mysql_fetch_array($result)){
              $nf=$row["frecuencia"]+1;
                mysql_query("update peso set frecuencia='".$nf."',especial='".$v."' where palabra='".$palabra."' and
iddocumento='".$t."'  );
                                        }
           else
                                       {
                mysql_query("insert into peso(palabra,frecuencia,iddocumento,especial,calculo,clase)
values('".$palabra."',1,'".$t."','".$v."',0,'".$clas."')");
                                         }

           $linea2=$linea2.$palabra." ";
                  }
        }

    echo "--> ". $linea2."<br>";
```

También para poder extraer solo las palabras que sean útiles fue necesario limpiar el texto de signos de puntación como (; ,. :) en el anexo 2 se muestra la lista de esto caracteres.

En este fragmento de código entra un párrafo o un grupo de párrafos estos son desmenuzados para extraer las palabras vacías y caracteres que no son útiles; Si una palabra es relevante es detectada y contabilizada además asigna una ponderación según de donde proceda 1 para clase, 2 para la descripción y 3 para titulo.

4.4.3 Asignación de pesos.

Método TF*IDF

Métodos de ponderación utilizado en esta tesis es el denominado TF*IDF [25] por sus siglas en ingles el cual plantea establecer una relación entre la frecuencia de un término dentro de un documento y su frecuencia en los documentos de la colección.

Básicamente, se utiliza la técnica de la "frecuencia del término por la frecuencia inversa", es decir, se obtiene la frecuencia pura del término t_i en el documento d_j (TF) y se lo multiplica por la inversa de la cantidad de documentos de la colección donde aparece t_i (Fi) (Baeza-Yates & Ribeiro-Neto, 2011).

$$TF * IDF_{ij} = TF_{ij} \; x \; log_2 \frac{N}{n}$$

Donde:

TF_{ij} : Corresponde a la frecuencia del término t_i (opcionalmente, se normaliza con la longitud de d_j en el documento d_j ó la máxima frecuencia de un término en dicho documento).

N: Es el tamaño de la colección (cantidad de documentos).

n: Es la cantidad de documentos donde el término t_i aparece.

En el cálculo del IDF, los valores cercanos a cero indican que el término posee poco peso y por lo tanto bajo valor de discriminación. Por otro lado, los valores positivos lejanos a cero indican que el término es poco frecuente y por consiguiente resulta más adecuado para caracterizar a los documentos donde se encuentran.

A continuación, se plantean algunos ejemplos:

N = 100 documentos

[25] Term Frequency, Inverse Document Frequency

Palabra	Valor de n	IDF
“el”	100 documentos	$Log_2(100/100) = 0.00$
“programa”	50 documentos	$log_2 (100/50) = 1.00$
“redes”	15 documentos	$log_2 (100/15) = 2.73$
“clase”	10 documentos	$log_2 (100/10) = 3.32$
“Adn”	1 documento	$log_2 (100/1) = 6.64$

Aquí se nota claramente que para cualquier término t_i, su peso será mayor mientras aparezca en menos documentos de la colección. Por otro lado, el valor de IDF varía entre 0 y el log(N).

Una variante a la fórmula de TF*IDF considera utilizar las frecuencias normalizadas a los efectos de relativizar los pesos en los documentos largos y cortos.

$$TF * IDF_{ij} = \frac{TF_{ij}}{largo(d_j)} \; x \; IDF$$

Donde:

Largo (d_j): es la cantidad de términos del documento d_j.

A continuación, se presenta un desarrollo completo del cálculo de pesos utilizando un ejemplo:

Documento 1	Mouse, Mouse, Mouse
Documento 2	Mouse, Mouse, Mouse, PC
Documento 3	Mouse, PC, Teclado
Documento 4	Mouse, Mouse, Mouse, PC, PC, PC
Documento 5	Teclado.

Representación de matriz de términos

	Mouse	**PC**	**Teclado**	**Largo**
Documento 1	3	0	0	3
Documento 2	3	1	0	4
Documento 3	1	1	1	3
Documento 4	3	3	0	6
Documento 5	0	0	1	1

El sigue paso se normaliza la tabla esto es dividimos la columna entre el largo

	Mouse	**PC**	**Teclado**
Documento 1	1	0	0
Documento 2	0.75	0.25	0
Documento 3	0.33	0.33	0.33
Documento 4	0.50	0.50	0
Documento 5	0	0	1

El cálculo de IDF al vocabulario del corpus resulta:

N= 5 (número de documentos)

	n	IDF
Mouse	4	$Log_2(5/4) = 0.32$
PC	3	$Log_2(5/3) = 0.73$
Teclado	2	$Log_2(5/2) = 1.32$

Luego, el cálculo del TF*IDF normalizado:

	Mouse	**PC**	**Teclado**
Documento 1	1*0.32=0.32	0*0.73=0	0*1.32=0
Documento 2	0.75*0.32=0.24	0.25*0.73=0.18	0*1.32=0

Documento 3	0.33*0.32=0.10	0.33*0.73=0.24	0.33*1.32=
Documento 4	0.50*0.32=0.16	0.50*0.73=0.36	0*1.32=0
Documento 5	0*0.32=0	0*0.73=0	1*1.32=1.32

En el siguiente código de PHP muestra como fue codificado el proceso de asignación de pesos.

```
mysql_select_db("motor",$db);
$r=mysql_query("SELECT COUNT(DISTINCT iddocumento) FROM peso");
$n=mysql_result($r,0);
$unico=0;
$i=0;
//echo $n."<br>";
$r1=mysql_query("SELECT * FROM peso");
if (! $r1){
       echo "SQL contiene errores.".mysql_error();
       exit();
       }else {
     set_time_limit (0);
          while ($row = mysql_fetch_row($r1)){
          $aux=$row[0];
          $rpal=mysql_query("SELECT COUNT(palabra) FROM peso where palabra='".$aux."'");
                    $npal=mysql_result($rpal,0);
                    if($npal==1) {$unico=$unico+1;}
                    $i=$i+1;

          $rpaln=mysql_query("SELECT COUNT(iddocumento) FROM peso where iddocumento='".$row[2]."'");
                    $paln=mysql_result($rpaln,0);


                   $pes=(log($n/$npal)/log(2));
                   $pesn=$pes*($row[1]/$paln);

          mysql_query("update peso set calculo='".$pes."',calculon='".$pesn."' where palabra='".$row[0]."' and
iddocumento='".$row[2]."'");
                                   }
            }
```

En la línea *"$pes=(log($n/$npal)/log(2));"* se calcula el IFD, la esta parte del código *"$pesn=$pes*($row[1]/$paln);"* se calcula TF*IDF normalizado.

4.4.4 Generación de índices.

Para la generación de índices, se empleo el manejador de base de datos, para poder utilizarlo se crearon algunas tablas muy simples, con esto se quiere decir que no fue necesaria la utilizamos ninguna entidad relacional y todas las tablas son

independiente una a otra; Básicamente el manejador de base de datos fue utilizado para ahorrar la programación de los algoritmos de búsqueda. A continuación se muestra el código SQL las tablas más empleadas:

La tabla “peso” es en la que se agrega todo los pesos de las palabra que son representativas de los documento que ya se encuentran integrados a la ontología; En especial podemos identificar a el campo *“calculo”* este almacena el IDF, el campo *“calculon”* en este se almacena TF*IDF normalizado, también el importante mencionar el campo *“especial”* este permite cuantificar y diferenciar de que parte proviene la palabra a la que se le está asignado un peso, hay que recordar que el titulo tiene un peso mayor que el de resumen pero con esta simple modificación permite jugar con la asignación de pesos para realizar modificaciones en futuros trabajos.

```
CREATE TABLE `peso` (
  `palabra` varchar(45) DEFAULT NULL,
  `frecuencia` int(11) DEFAULT NULL,
  `iddocumento` varchar(90) DEFAULT NULL,
  `especial` int(11) DEFAULT NULL,
  `calculo` float DEFAULT NULL,
  `clase` varchar(100) DEFAULT NULL,
  `calculon` float DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1$$
```

Tabla de “auxpeso” en esta tabla se almacenan los pesos del documento que se va agregar y no conocemos su clase.

```
CREATE TABLE `auxpeso` (
  `palabra` varchar(45) DEFAULT NULL,
  `frecuencia` int(11) DEFAULT NULL,
  `iddocumento` varchar(90) DEFAULT NULL,
  `especial` int(11) DEFAULT NULL,
  `calculo` float DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1$$
```

La tabla “semejante” es ahí donde almacenamos los cálculos de palabras de nuevo documento con respecto a la nueva colección básicamente es en esta tabla donde se almacena los resultados y la diferencia más cerna a valor de 0 son los documentos mas semejantes y por lo tanto el nuevo documento puede ser parte de la clase de este documentó que ya pertenece a la colección.

```
CREATE TABLE `semejante` (
```

```
  `palabra` varchar(45) DEFAULT NULL,
  `frecuencia` int(11) DEFAULT NULL,
  `iddocumento` varchar(90) DEFAULT NULL,
  `especial` int(11) DEFAULT NULL,
  `frecutotal` int(11) DEFAULT NULL,
  `calculo` float DEFAULT NULL,
  `calculonue` float DEFAULT NULL,
  `diferencia` float DEFAULT NULL,
  `clase` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1$$
```

Capítulo V Resultados.

En este capítulo se trata los resultados obtenidos durante todo el trabajo de tesis desarrollado, en el se explica la tipo de ontología que se codifico y los aportes que se están hecho, además que se presentan las métricas que fueron empleadas para medir el rendimiento del clasificador y por último se sugieren los trabajos a futuro que puede tener esta tesis.

5.2 Tipo de ontología

La ontología codifica puede caer en varias clasificaciones pero la que es la más idónea es la propone Heijst en relación a la forma de uso, él la clasifica como una *ontología de información,* por que proporciona una estructura de almacenamiento; En esta tesis la ontologías además de proporcionar un esquema de organización, la misma ontologías funciona como una base de datos donde se guardan los metadatos de las colecciones de libros.

Según la clasificación de Guarino esta ontología caería en la clasificación de *ontología de dominio,* por que usa un vocabulario relacionado con un dominio especifico como es el de ciencias computacionales.

5.2. Codificación de ontología.

La codificación de la ontología se encuentra en OWL 2 y se realizo con la herramienta editora de lenguaje para ontologías Protégé, como ya se explico en el capítulo 1 existen tres sub-lenguajes de OWL, para esta tesis se empleo el OWL DL ya con él, se puede hacer uso de la lógica descriptiva, OWL también cuida el costo computacional, con esto sé garantizar que los cálculos de razonamiento terminan en un tiempo finito.

El código generado es un aporte para futuros trabajos, la estructura de código cuenta con 25,023 líneas codificadas en OWL, de las que 14 son clases principales y 2,465 subclases, además se genero una estructura para el almacenamiento datos dentro

de la ontología por ello fue necesario crear los 15 jerarquías de propiedades de datos todas cumpliendo con es estándar de DC.

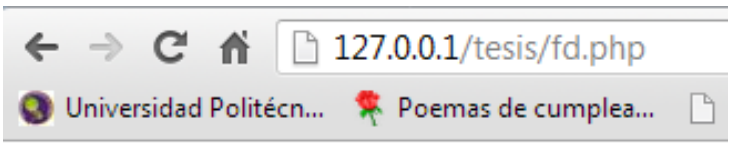
El aporte de la estructuración de los datos que se muestra en el capítulo 3 en la tabla 5, esas líneas de código son con las que se puede introducir un nuevo miembro a una clase. Para no afectar la ontologías o dejarla inconsistente, estas líneas de código se agregan en la sección *"Individuals*".

Una de las preguntas que saltan es ¿por qué usar una ontología para como una estructura de almacenamiento? Esto tiene que ver con el potencial que tiene las ontologías para poder realizar tareas de inferencia, pero la más importante es que por medio de los razonares se verifica la consistencia de la ontología y de los datos que se encuentran almacenados en ella.

Para detectar la inconsistencia se utiliza Pellet, fact ++ que son parte de Protégé y Jena que está unido con Fuseki en un servicio HTTP; Ninguno de esto razonadores informó que la ontología OWL tuviera inconsistencia.

5.3 Extracción de información.

Para poder extraer la información es necesario ejecutar el la aplicación en la siguiente dirección *http://127.0.0.1/tesis/fd.php*, esta aplicación calcula los pesos y prepara las bases de datos para cuando se desee agregar un nuevo documento a la colección de manera automática. Al finalizar la ejecución de esta aplicación presenta un resumen ver la figura 8, los datos que se presentan en este capítulo provienen de un repositorio de 100 articulo con un total de 5475 términos.



Concepto	total
Total de articulos en la colección	100
Total de terminos unicos	1714
Total de terminos repetidos	3761
Total de terminos	5475

Figura 8. Reporte finalización de cálculo de pesos

En esta tesis se empleo una pantalla de captura ver la figura 9, la que se denomina pantalla *mantenimiento de colecciones* en la que se capturan los datos de un documento, este es analizado y le es asignado los pesos y comparado con los documentos de la colección.

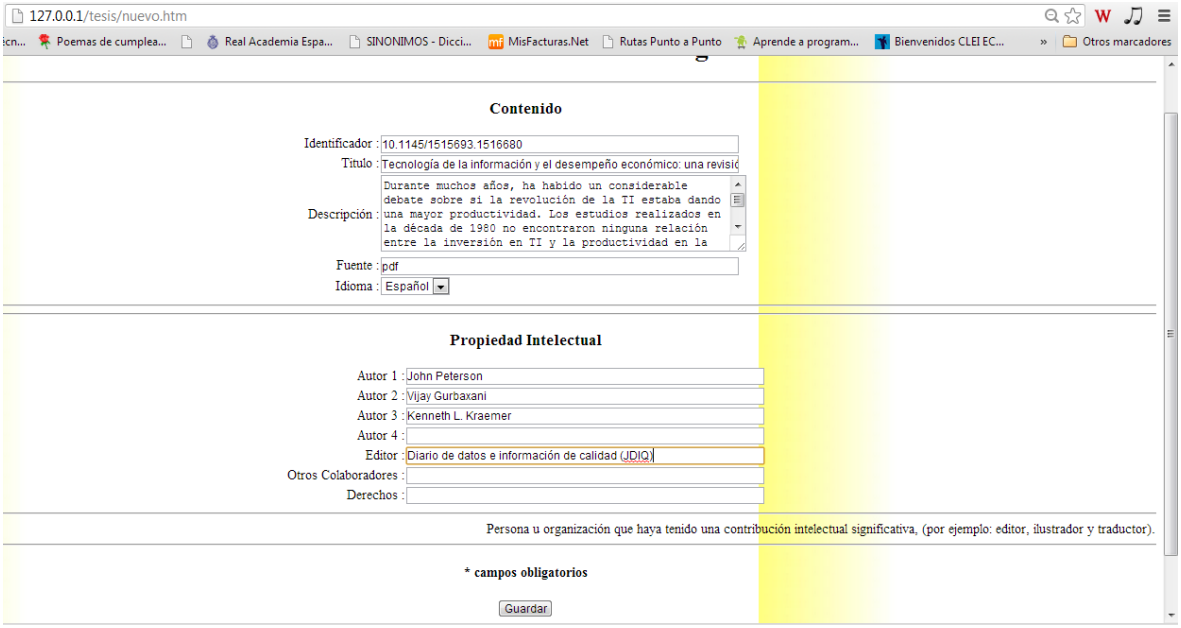


Figura 9. Captura de nuevo documento sin conocer la clase a la que pertenece.

En este caso la biblioteca digital de ACM proporcionó también los documento con los cuales se va a realizar el cálculo de resultados y como ejemplo se tomo el siguiente documento[26], solo se selecciono un fragmento de la información.

> **Titulo**: Tecnología de la información y el desempeño económico: una revisión crítica de la evidencia empírica.
> **Resumen**: Durante muchos años, ha habido un considerable debate sobre si la revolución de la TI estaba dando una mayor productividad. Los estudios realizados en la década de 1980 no encontraron ninguna relación entre la inversión en TI y la productividad en la economía de EE.UU., una situación conocida como la *paradoja de la productividad*.

[26] Dirección donde se tomos el ejemplo http://dl.acm.org/citation.cfm?id=641865.641866&coll=DL&dl=GUIDE&CFID=239906361&CFTOKEN=46261887

La biblioteca de ACM clasifica el documento de la siguiente manera como lo se muestra en la figura 10.

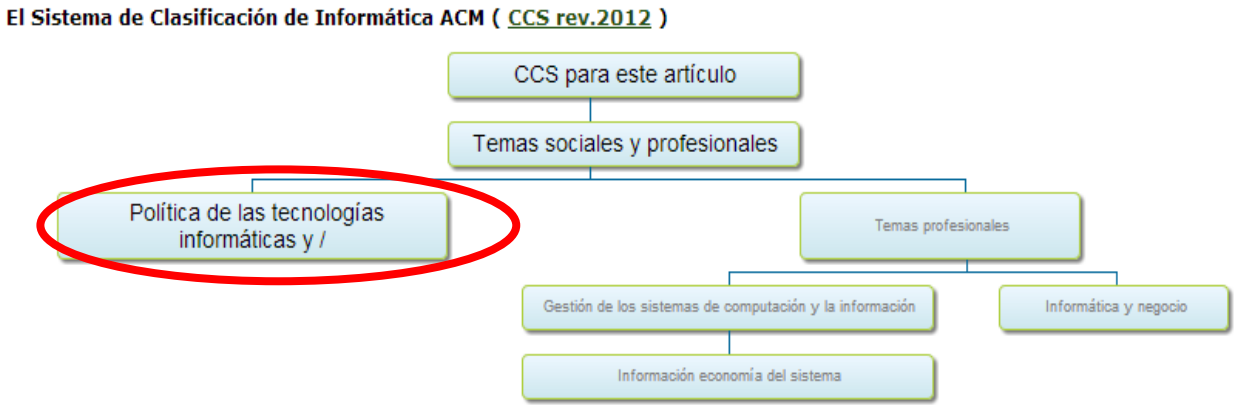


Figura 10. Esquema de clasificación del documento.

La clasificación de este documento puede caer en varias clases y todas estarían correctas esa es una característica de la CCS de ACM 2012, como se observa en la figura 10. Las clases con color más obscuro y letra más grande es la más adecuada para clasificarla en este caso puede caer la clasificación en las siguientes clases:

- Temas sociales y profesionales
- Política de las tecnologías informáticas y /
- Temas profesionales
- Gestión de los sistemas de computación y la información
- Informática y negocio
- Información economía del sistema

En a figura 11 se ve el menú de selección de clases sugeridas, en el se muestra las 5 primeras coincidencias una lista de todas las clases que podemos utilizar para clasificar el nuevo documento, en este ejemplo podemos ver que 3 de las 5 coincidencias son las que sugiere la biblioteca digita de ACM. Estas coincidencias son:

- Temas sociales y profesionales
- Política de las tecnologías informáticas y /
- Información economía del sistema

Ordena por la suma total

SELECCIÓN	CLASE	PROMEDIO	TERMINOS	SUMA TOTAL
○	NFORMÁTICA / POLÍTICA TECNOLÓGICA	4.11795	137	564.159
○	LOS_DATOS_DE_GESTIÓN_DE_SISTEMAS	43.4779	13	565.212
○	HUMAN_COMPUTER_INTERACTION_(HCI)	56.4503	10	564.503
○	ECONOMÍA_DE_LA_INFORMACIÓN_DEL_SISTEMA	56.5177	10	565.177
○	LA_MINERÍA_DE_DATOS	62.8105	9	565.294

Guardar

Figura 11. Menú de selección de clases sugerida.

En este menú se permite que el usuario seleccione la clase que más le parezca conveniente, se recomienda utilizar la primera.

5.4 Métrica de evaluación

Según Ricardo Baeza (Baeza-Yates & Ribeiro-Neto, 2011) sugiere que la métrica que funciona mejor para un clasificador de texto se define en relación a una clase determinada es Accurancy and Error. El acurancy es la parte de los documentos de prueba que están asignados a sus clases correctas por el clasificador. Y el error es la parte de los documentos de prueba que están incorrectamente asignados a sus clases por el clasificador. En esta tesis es una instancia binaria con esto se dice que o está bien clasificado o no se está bien clasificado.

Accuracy

$$Acc_{(Cp)} = \frac{tp + tn}{N}$$

Error

$$Err_{(Cp)} = \frac{fn + fp}{N}$$

Siempre se debe de cumplir

$$Acc_{(Cp)} + Err_{(Cp)} = 1$$

Donde

tp: clasifico bien en positivos donde el resultado es correcto.

tn: clasifico bien a los que no pertenecen a esta clase el resultado es correcto.

fp: falso positivos los coloco en la clase y no corresponden.

fn: lo clasifico como positivos y no corresponden

Como ya se dijo este problema de tesis es binaria así que solo existe dos opciones clasificar bien o mal clasificada, por lo tanto solo usaremos los valores de tp o fp.

En la parte anterior se describió como se da el mantenimiento a una colección de documento en base a esa explicación se genero los datos para aplicar la métricas que aquí se muestran.

- Número total de documentos en el repositorio 100.
- Numero de documentos que le es aplicado el proceso de mantenimiento de colecciones 20.
- Numero de documentos que entre las 5 primeras clases se encuentra bien clasificado 14.
- Numero de documentos no se encontró una relación entre las 5 primeras clases 6.

Resultados de las métricas

Accuracy = 70%
Error = 30%

5.5 Metodología diseñada

El objetivo principal de esta tesis es diseñar una metodología basada en ontologías para mantener de forma automática colecciones de documentos de un dominio restringido, para entregar esta metodología primero definiremos que es una metodología dice RAE que metodología es un conjunto de métodos o pasos que se

siguen en una investigación científica o en una exposición doctrinal, partiendo de esto la metodología es la siguiente:

1. Codificación de una ontología sobre OWL.
2. Verificar la consistencia de la ontología con los razonadores de LD disponible.
3. Generar la estructura de datos necesarios para poder soportar las colecciones de documentos.
4. Agregar un método como convencional para la extracción de información como por ejemplo IDF.
5. El nuevo documento pasa por el método extracción de información, para que posterior mente le sea comparado con todos los documentos de la colección y así el documento de la colección que más se asemejen al nuevo documento este último tomara la clase de documento de la colección.
6. El nuevo documento es agregado a la ontología verificando la consistencia.

5.6 Trabajos a futuro.

En esta parte se sugieren los trabajos que se pueden realizar a partir de esta tesis. El aporte de la codificación de la otología sobre OWL da pie a muchos trabajos a futuro, como por ejemplo agregar dentro de la ontología un conjunto de términos para que esta pueda ser utilizada para describir las clases y así transformar la ontología en de conceptos. Con esta modificación ya se podría solo utilizar solo la ontología para poder realizar la clasificación, algunos autores como Gabor aseguran que este tipo de prácticas tiene un costo computacional muy alto.

Otro trabajo que se puede realizar es el clasificar un nuevo libro desde cero en esta tesis se requiere tener una colección ya que los patrones que dan la extracción de información es lo que dicta hacia donde se va clasificar el documento sería interesante empezar desde ceros.

Bibliografía

ACM. 2012. “Sistema de clasificación computacional de 2012”. En línea http://delivery.acm.org/10.1145/2380000/2371137/ACMCCSTaxonomy.html?ip=187.186.146.173&acc=OPEN&CFID=198433416&CFTOKEN=82154632&__acm__=1352663893_7854295f3b5bb1d8e02051c84075ca0.12 septiembre 2012.

Antoniou, G. & Van Harmelen, F. 2004. *A Sematic Web Primer*. ,The Mit Press, Cambridge, Massachusetts, London England.

Baader, F., McGuinness, D. L. & Nardi, D. 2003.*The Description Logic Handbook*. University Press, CambirdgeReino Unido.

Berners-Lee, T., Hendler, J. & Lassila, O. 2001.“The Semantic Web”, En línea http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-:_;84A9809EC588EF21, 7 Noviembre 2012.

Borst, W. N.1997. *Construction of engineering ontologies for knowledge sharing and reuse.* 1 ed. Enschede, The Netherlands: Centre for Telematics and Information.

Bray, T., Paoli, J., McQueen, S. & Eve, M. 2008. “Extensible Markup Language (XML) 1.0 (Fifth Edition)”.http://www.w3.org/TR/REC-xml/,10 Septiembre 2012.

Brewste, C. y otros.2004. Knowledge Representation with Ontologies: The Present and Future*. IEEE Intelligent System*,19(1), pp. 72-81.

Brickley, D. 2004. “Descripción de Vocabularios RDF 1.0: Esquema RDF.”En línea http://www.w3.org/TR/rdf-schema/, 12 noviembre 2012.

Cowan, J. & Tobin, R. 2004. “Conjunto de información XML”, En línea http://www.w3.org/TR/xml-infoset ,2 ed.12 Septiembre 2012.

Gabor N. 2005. Improving Information Retrieval Effectiveness by Using Domain Know-ledge Stored in Ontologies. In proceeding son the Move to Meaningful Internet Systems 2005: OTM Workshops. Lecture Notes in Computer Science.pp. 780—789

Gruber T.R. 1993. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220.

Gruber, T. R. 1993. “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”. En línea http://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCQQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.91.6025%26rep%3Drep1%26type%3Dpdf&ei=qlVyUN6pLcSfqwGYq4HoDw&usg=AFQjCNH6E4XzTsjjmWIRRePKNA2vGSY0Dw&s, 7 Octubre 2012.

Guarino, N. 1998. Formal Ontology and Information Systems. *Proceedings of FOIS’98*, 6(8), pp. 3-15.

Giunchiglia F., Zaihrayeu I. Farazi F. 2009. Converting classifications into OWL ontologies. 2009.Work shop on matching and meaning (WMM 2009). The University of Edinburgh. School of informatics. Edinburgh, United Kingdom.

Giunchiglia F., Marchese M., Zaihrayeu I. 2006. Encoding Classifications into Light weight Ontologies. Proceedings of the Proceedings of the 3rd European Semantic

Web Conference, The Semantic Web: Research and Applications(ESWC 2006).pp. 80-94

Harmelen, F. v. & Horrocks, I. 2008. “Quenstions and answers on OIL: the Ontology Inference Layer”. En línea http://oil.semanticweb.org/, 12 junio 2012.

Heijst, V. G., Schreiber, T. A. & Wielinga, J. B., 1997. Using explicit ontologies in KBS development. *International Journal of Human and Computer Studies*, 46(2), pp. 183-292.

Hepp, M.; de Bruijn, J. 2007. GenTax: A Generic Methodology for Deriving OWL and RDF-S Ontologies from Hierarchical Classifications, Thesauri, and Inconsistent Taxono-mies. Proceedings of the 4th European Semantic Web Conference (ESWC 2007), June 3-7, Innsbruck, Austria, Springer LNCS Vol. 4519, Springer 2007, pp.129-144.

http://www.heppnetz.de/files/hepp-de-bruijn-ESWC2007-gentax-CRC.pdf

ISBN, España Libros. 2010. “Código de identificación para libros”. En línea http://www.isbn.com.es/ , 03 octubre 2012.

Klyne, G. & Carroll, J. J. 2004. “Resource Description Framework (RDF): Conceptos y Sintaxis Abstracta”. En líneahttp://www.w3.org/TR/rdf-concepts/, 12 Octubre 2012.

Kroetzsch, M., Simancik, F. & Horrocks, I. 2012. *A Description Logic Primer*. Oxford University Press, Oxford, Reino Unido.

Khan, L., D. McLeod, and E.H. Hovy. 2004. Retrieval Effectiveness of an Ontology-Based Model for Information Selection. Journal for Very Large Data Bases (VLDB). 13(1), pp. 71–85.

Medina M.A., Sánchez J.A., De la Calleja J., Benitez A. 2012. A practicalapproachtomo-delclassificationschemeswith OWL ontologies.Proceedings of theProceedings of theEigh-thLatin American WorkshoponLogic / Languages, Algorithms and New Methods of Rea-soning(LANMR 2012). Volume 911 of CEUR WorkshopProceedings, pp. 75-88. CEUR-WS.org, (2012).

Mahesh, K., 1996. *Ontología para el Desarrollo de la traducción automática: Ideología y metodología*. New Mexico State University,New Mexico.USA.

Mindswap. 2006. “Maryland information and network dynamics lab semantic web agents project”. En línea, http://www.mindswap.org/2003/pellet/index.shtml ,20 octubre 2012.

Moreno, C. A. & Sánchez, Y. 2012. Prototipo de buscador semántico aplicado a la búsqueda de libros de ingeniería de sistemas y computación en la biblioteca Jorge Roa Martínez de la universidad tecnológica de Pereira. Tesis de Ingeniería,Facultad de ingenierías: eléctrica, electrónica, física y ciencias de la computación Ingeniería de sistemas y computación, Universidad Tecnológica de Pereira, Pereira, Colombia, Marzo.

Nardi, D. & Brachman, R. J. 2002. *An Introduction to Description Logics*. Cambridge University Press, Cambridge,Reino Unido :

Neches, R., Fikes, R., Finin, T. & Gruber, T. 1991. Tecnología de Activación para Compartir Conocimiento. *Revista AI*, 3(12), pp. 36-56.

Noy, N. F. & Mcguinness, D. 2001. *Desarrollo de Ontologías 101: Guía para la Creación de la ontología en primera*. Stanford University Press, CA,USA.

Polo, L., Berrueta, D., Rubiera, E. & Fernández, S. 2007. “Experimento semántico para definir contextos y recursos:1.0”. En línea http://forge.morfeo-project.org/wiki/index.php/D_2.4_Experimento_sem%C3%A1ntico_para_definir_contextos_y_recursos:1.0#Referencias , 08 Noviembre 2012.

RALE .2010. “Diccionario de la lengua Española”. En línea http://lema.rae .es/drae/?al=ontolog%C3%ADa7, 7 Noviembre 2012.

Rodríguez C. 2010. Razonadores semánticos: en estado del arte. *Revista Ingenium de la Facultad de Ingeniería*, 11(21).

Rous, B., 2012. Actualización principal en el sistema de ACM Computing Clasificación. *Communication of the ACM*, 55(11), p. 12.

Russel, S. & Norving, P., 2004. *Inteligencia Artificial. En: Un enfoque moderno*. Madrid: Prentice Hall, p. 1240.

Sowa, J. F., 2000. “Representación del conocimiento :fundamentos lógicos y filosóficos, y computacionales. Pacific Grove, CA: Brooks Cole Publishing Co..

Stanford Center for Biomedical Research Informatics , 2012. Welcome to protégé”. En línea http://protege.stanford.edu/, 15 Noviembre 2012.

Stamou S. 2006. Retrieval Effectiveness of an Ontology-Based Model for Conceptual In-dexing. In Proceedings of the 5th International Conference on Formal Approachesto South Slavic and Balkan Languages (FASSBL-5), October 18-20, Sofia, Bulgaria.

Swartout, W. & Tate, A. 1999. Ontologies. *IEEE Intelligent Systems*, pp. 18-19..

UDC, Consortium, 2006. “AENOR Asociación Española de Normalización y Certificación”. En línea http://www.udcc.org/aenor.htm , 2012 Octubre 2012.

Uschold, M. & Gruninger, G., 1996. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2), pp. 93-155.

W3C, 2004. “OWL Ontology Web Language Guie”. En línea http://www.w3.org /TR/owl-guide/ , 10 noviembre 2012.

W3C, 2012. “Extensible Markup Language (XML)”. En líneahttp://www.w3. org/XML/, 20 Septiembre 2012.

Weigand, H., 1998. Multilingual Ontology-based Lexicon For News Filtering the TREVI Project. Reporte técnico pp.138-159, Infolab,Tilburg University,.Tilburg, Netherlands

Anexo

Anexo 1: Lista de palabras vacías de la lengua española

él ésta éstas éste éstos última últimas último últimos a añadió aún actualmente adelante además afirmó agregó ahí ahora al algún algo alguna algunas alguno algunos alrededor ambos ante anterior antes apenas aproximadamente aquí así aseguró aunque ayer bajo bien buen buena buenas bueno buenos cómo cada casi cerca cierto cinco comentó como con conocer consideró considera contra cosas creo cual cuales cualquier cuando cuanto cuatro cuenta da dado dan dar de debe deben debido decir dejó del demás dentro desde después dice dicen dicho dieron diferente diferentes dijeron dijo dio donde dos durante e ejemplo el ella ellas ello ellos embargo en encuentra entonces entre era eran es esa esas ese eso esos está están esta estaba estaban estamos estar estará estas este esto estos estoy estuvo ex existe existen explicó expresó fin fue fuera fueron gran grandes ha había habían haber habrá hace hacen hacer hacerlo hacia haciendo han hasta hay haya he hecho hemos hicieron hizo hoy hubo igual incluso indicó informó junto la lado las le les legó lleva llevar lo los luego lugar más manera manifestó mayor me mediante mejor mencionó menos mi mientras misma mismas mismo mismos momento mucha muchas mucho muchos muy nada nadie ni ningún ninguna ningunas ninguno ningunos no nos nosotras nosotros nuestra nuestras nuestro nuestros nueva nuevas nuevo nuevos nunca o ocho otra otras otro otros para parece parte partir pasada pasado pero pesar poca pocas poco pocos podemos podrá podrán podría podrían poner por porque posible próximo próximos primer primera primero primeros principalmente propia propias propio propios pudo pueda puede pueden pues qué que quedó queremos quién quien quienes quiere realizó realizado realizar respecto sí sólo se señaló sea sean según segunda segundo seis ser será serán sería si sido siempre siendo siete sigue siguiente sin sino sobre sola solamente solas solo solos son su sus tal también tampoco tan tanto tenía tendrá tendrán tenemos tener tenga tengo tenido tercera tiene tienen toda todas todavía todo todos total tras trata través tres tuvo un una unas uno unos usted va vamos van varias varios veces ver vez y ya yo.

Anexo 2: Lista de caracteres

1=> '.',	7=>'¿',	13=> '+',	19=> ')',	27=> "!",
2=> '-',	8=> '¡',	14=> '*',	20=> '=',	28=> '@',
3=> '_',	9=> '>',	15=> '/',	21=> '$',	29=> '#',
4=> ';',	10=> '<',	16=> '%',	22=> '¨',	30=>'\r',
5=> ',',	11=> '?',	17=> '&',	23=> '{',	31=>'\n',
6=> ':',	12=> '^',	18=> '(',	24=> '"',	32=> ' ',
	26=> "\\",		25=> "",	