UNIVERSIDAD POLITÉCNICA DE PUEBLA

PROGRAMA ACADÉMICO DE
INGENIERÍA EN INFORMÁTICA

Reconocimiento de Similitud entre Reportes Técnicos mediado por Ontologías

ALUMNA:
Silvia Titla Cosme

Reporte Técnico PII-21-08-09

COMITÉ EVALUADOR

Dra. María Auxilio Medina Nieto (*Asesor*)
M.C. Rebeca Rodríguez Huesca
Dra. Rita Marina Aceves Pérez

PROFESORA DE PROYECTO DE INVESTIGACIÓN II

Dra. María Auxilio Medina Nieto

Agosto 2009

Índice

Índice de figuras

Índice de tablas

Resumen

Los reportes técnicos son recursos de información relevantes, representan el trabajo realizado por estudiantes e investigadores en periodos de tiempo cortos. Este documento discute cómo la estructura y los términos de una colección pueden ser usados para comparar reportes técnicos. La colección está representada como una ontología ligera, la cual es una estructura que agrupa documentos similares de forma que los documento de un grupo en el k-ésimo nivel comparten los k-términos de su etiqueta. El documento describe la implementación de un sistema prototipo desarrollado en lenguaje PHP.

Capítulo 1. Planteamiento del Problema

1.1 Introducción

"Las Bibliotecas Digitales son espacios virtuales que facilitan el acceso, el uso, la diseminación y la generación del conocimiento [1]". Se necesita crear o innovar herramientas que procesen información de forma rápida considerando el almacenamiento. El análisis de textos requiere aplicaciones que tengan las características anteriores, las cuales facilitan la creación de nuevos prototipos o servicios de alta tecnología.

El proyecto intenta manifestar la protección de los derechos de autor a los alumnos de la Universidad Politécnica de Puebla (UPP). El derecho autoral se define como "un conjunto de normas de derecho social, que protegen el privilegio que el Estado otorga por determinado tiempo a la actividad creadora de autores y artistas, ampliando sus efectos en beneficio de intérpretes y ejecutantes [2]". Se deben globalizar los riesgos a los que se exponen sus creadores, siendo algunos de éstos los siguientes: "publicación, reproducción, edición, ejecución, representación, exhibición, adaptación, uso y explotación comercial, así mismo, oponerse a toda deformación, mutilación o modificación de la obra y el derecho de que citen su obra textualmente (art. 2, 3, 4, y 5 LFDA), también les otorga el reconocimiento a su calidad de autores. Estos derechos son perpetuos, inalienables, imprescriptibles e irrenunciables [3]".

Actualmente, en la UPP los reportes se revisan de forma manual, esto consume recursos humanos, costo en tiempo e indirectamente capital. Una herramienta de apoyo a la detección de la similitud entre reportes técnicos facilitaría su revisión y fomentaría el respeto a la propiedad intelectual.

Este proyecto revisará la similitud que hubiera entre reportes, lo cual se evaluará mediante ontologías, ya que algunas de sus funciones son: 1) describir sistemas de información, 2) facilitar el entendimiento del lenguaje natural, 3) apoyar la comprensión de los sistemas basados en el conocimiento. "Las ontologías son principalmente teorías de contenido, ya que su principal contribución es identificar determinadas clases de objetos y relaciones que existen en

algunos dominios [4].” En otras palabras, las ontologías son una representación de vocabularios, estructuras especializadas en la colección de datos comunes que existe en una materia o un dominio, las cuales proporcionarán un conjunto de términos para describir la similitud de los reportes.

1.2 Objetivo general

Reconocer la similitud entre reportes técnicos de la Universidad Politécnica de Puebla mediante ontologías.

1.3 Objetivos específicos

- Desarrollar un sistema de detección de plagio basado en ontologías
- Definir una representación de los documentos en XML
- Utilizar el método OntOAIr para construir ontologías
- Evaluar el grado de similitud entre reportes técnicos

1.4 Justificación

En la actualidad, las universidades aprovechan las tecnologías de información y comunicación (TIC). Un ejemplo de éstas son las bibliotecas digitales, las cuales reducen el tiempo de búsqueda de información validada. Contar con una biblioteca digital en la UPP sería de gran apoyo para los alumnos, en particular, un proyecto simultáneo está enfocado en el desarrollo de una colección de reportes técnicos. Esta colección permitiría a alumnos y profesores consultar los reportes técnicos que se han desarrollado, lo cual facilitaría la elaboración de nuevos reportes. Además, los alumnos podrían revisar los libros, las revistas, o artículos que se han referenciado, incluso la tecnología.

Sin embargo, contar con una colección de este tipo podría ocasionar que se viole la propiedad intelectual. Al crear una herramienta que detecte la similitud de reportes, se pretende evitar posibles plagios, así como difundir las sanciones que establece la Ley Federal de Derechos de Autor [5]. El proyecto fomentaría la ética en los estudiantes y haría énfasis en la propiedad

intelectual, dado que es un derecho que les corresponde. Por otro lado, los alumnos podrían citar a trabajos ya revisados. Esto favorecería al autor u otros titulares.

1.5 Metodología

Para llevar a cabo el proyecto, se proponen las actividades de la Tabla 1 y 2.

Tabla 1. Cronograma de actividades de Proyecto de Investigación 1

No.	**Actividad**	**ENERO**				**FEBRERO**				**MARZO**				**ABRIL**			
1.-	Revisión de la literatura																
2.-	Elaboración de la propuesta de investigación																
3.-	Revisión la sintaxis y semántica del metalenguaje XML																
4.-	Lectura de tutoriales de RDF																
5.-	Elaboración de protocolo y marco teórico																
6.-	Revisión el tutorial de la herramienta Protégé																
7.-	Diseño de mecanismo de similitud entre ontologías																
8.-	Proponer un mecanismo de similitud entre documentos de RDF																
9.-	Presentación del protocolo de investigación																
10.-	Elaboración del capítulo de metodología.																

Tabla 2. Cronograma de Actividades 2

No.	ACTIVIDAD	MAYO				JUNIO				JULIO				AGOSTO			
1.-	Revisión de metodología																
2.-	Implementación																
3.-	Pruebas																
4.-	Presentación de proyecto																
5.-	Conclusiones																
6.-	Elaboración de reporte de investigación																

1.6 Alcances y limitaciones

Alcances:

- El sistema facilitará la búsqueda de similitudes entre reportes técnicos.
- El sistema podrá ser soportado por diferentes plataformas como Windows y Linux.
- Las ontologías utilizadas se generarán de acuerdo al vocabulario de los reportes técnicos.

Limitaciones:

- La construcción de la ontología que representa una colección no forma parte del proyecto.
- El tiempo de respuesta del sistema se verá afectado por el tamaño de la ontología.
- El plagio se detectará a través de la implementación de un algoritmo que devuelve un porcentaje de similitud.

1.7. Recursos de hardware y software

Los recursos de software son los siguientes:

- Sistema operativo Windows XP
- Metalenguaje “Extensible Markup Language XML” Versión 1.0
- XAMPP Windows Versión 1.7.1

- Hypertext Pre-processor (PHP) Versión 5.2.9
- xml_parser_create Versión 3.0.6

Los recursos de hardware son los siguientes:

- Disco duro de 10 GB como mínimo.
- Procesador superior a 1Ghz
- Memoria RAM: 512 MB

Capítulo 2. Marco teórico

Este capítulo aborda temas como los derechos autor y las posibles sanciones que establece la Ley Federal de Derecho del Autor, se explica qué es un reporte técnico para la Universidad Politécnica de Puebla y se presentan brevemente las ontologías. También se describen los métodos OntOAIr y GrowBag que sirven para construir ontologías, así como una herramienta de edición llamada Protégé. Finalmente se citan trabajos relacionados.

2.1 Derechos de autor

Este proyecto intenta difundir información de "reportes técnicos" que se han realizado anteriormente. Se pretende dar crédito al autor a través de referencias a su trabajo, de tal forma que se pueda hacer manifiesto el uso total o parcial de éste de acuerdo con lo establecido por la Ley Federal de Derechos de Autor, al mismo tiempo, se pretende dar a conocer a los estudiantes las sanciones que protegen la propiedad intelectual.

La Ley Federal de Derechos de Autor intenta proteger y fomentar estos derechos. A manera de ejemplo, los siguientes artículos muestran algunas sanciones que pueden suceder al intentar plagiar un programa de computación o base de datos [5].

- Artículo 231 (fracciones II y VII): Contemplan dentro de las infracciones del comercio el "producir, usar, reproducir o explotar una reserva de derechos protegida o un programa de cómputo sin el consentimiento del titular"

- Artículo 215: Sanciona con multa de 300 a 3 mil días o pena de prisión de 6 meses a 6 años al que incurra en este tipo de delitos

En caso de que alguna persona realice copia total o parcial de los reportes técnicos, legalmente estaría sujeta a recibir las sanciones de los artículos 231 y 215.

2.2 Reporte técnico

La UPP es una institución pública que mediante un modelo de educación basado en competencias, prepara profesionales con una sólida formación científica, técnica y social, conscientes del contexto económico, político y sociocultural del Estado y del país; además, impulsa la investigación aplicada y la innovación tecnológica, asociadas a las necesidades de las empresas y de la sociedad [6]. En la UPP los reportes técnicos buscan impulsar la investigación científica y el desarrollo tecnológico en alumnos y profesores.

Un reporte técnico documenta a un proyecto de investigación. "Un proyecto de investigación consiste en describir los antecedentes teóricos, el trabajo de investigación, los estudios o ensayos que se han realizado, que permite fomentar, justificar y buscar la investigación propuesta [7]". La investigación debe ser verificable y replicable. Como resultado de la investigación aplicada, en la UPP se obtiene un reporte generado por alumnos y profesores, el cual se planea se lleve a cabo al cursar dos materias seriadas que llevan el mismo nombre, las cuales se diferencian por la fase de investigación que se realiza en cada una, (Proyecto de Investigación 1 corresponde a la fase de planeación, en tanto, Proyecto de Investigación 2 a la fase de ejecución).

Un reporte técnico aporta evidencias de conocimiento para profesores y estudiantes, integra un resumen de la extracción y recopilación de la información. También puede ser un prototipo o un servicio que requiere la UPP. Según [8], un proyecto de investigación o reporte técnico tiene los siguientes componentes:

- **Introducción**. Contiene una descripción clara de la estructura general del proyecto.

- **Objetivo general.** Constituye el enunciado global sobre el resultado final que se pretende alcanzar (qué, dónde y para qué de la investigación). Precisa la finalidad de la investigación en cuanto a sus expectativas más amplias y la orienta.

- **Objetivos específicos.** Facilitan el cumplimiento del objetivo general mediante la determinación de etapas; se derivan del objetivo general e inciden directamente en los

logros a obtenerlo. Los objetivos específicos deben ser concretos, claros, realistas, medibles y formulados en términos operativos.

- **Justificación.** Contiene los argumentos fundamentales que sustentan la investigación a realizar, enfatizando aquellos de carácter técnico y social.

- **Marco teórico.** Contiene una revisión de la literatura en la que se desarrolla la investigación, así como las investigaciones antecedentes y las teorías a manejar.

- **Metodología.** Contiene la descripción y argumentación de las principales decisiones metodológicas adoptadas según el tema de investigación y las posibilidades del investigador. La claridad en el enfoque y estructura metodológica es condición obligada para asegurar la validez de la investigación [9].

- **Recursos.** Determinan la infraestructura y los recursos humanos de acuerdo a las necesidades de la investigación.

- **Referencias**. Se utilizan a lo largo de la investigación para elaborar el marco teórico u otros propósitos.

Además de los componentes anteriores, en este trabajo se propone el empleo de ontologías para estimar la similitud entre reportes técnicos. Las ontologías se definen en la siguiente sección.

2.3 Definición de ontología

Una ontología define los términos básicos y las relaciones que comprende el vocabulario de un área temática, así como las reglas para combinar los términos y las relaciones para definir extensiones al vocabulario, es decir, determina los términos a utilizar como son tipos de estructuras, categorías de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad [4]. Cuando se pretende estandarizar de forma gráfica el conocimiento, la ontología se percibe como un árbol.

Las ontologías son fundamentales para muchas aplicaciones tales como portales de conocimiento científico, gestión de información y sistemas de integración, web semántica, gestión del conocimiento, sistemas de recomendación de consultas, hipertexto, teleeducación, aprendizaje a distancia (e-learning) y comercio electrónico [10]. Debido a que las ontologías en general representan información de un dominio, en el proyecto se consideran adecuadas para determinar la similitud entre reportes técnicos.

2.3.1 Representación de ontologías utilizando XML

Por sus siglas en inglés, extensible Markup Language (XML) o Lenguaje de Marcado Extensible, fue presentado por el World Wide Web Consorcio (W3C) en 1998 [11]. Es un lenguaje descriptivo de documentos creado para estructurar, almacenar y transportar información [12], permite definir etiquetas propias. El autor es quien domina la estructura del documento a realizar.

A diferencia de HTML, en XML las etiquetas no son limitadas, no mezcla la estructura con el diseño. XML permite mostrar contenido dinámico aunque recurre a hojas de estilo o cascada. Otras características de XML son [11]:

- Es posible definir etiquetas propias
- Se asignan atributos a las etiquetas
- Las etiquetas y atributos se definen de forma exacta mediante un esquema o un “Data Type Definition (DTD)”
- Está basado en texto, (no utiliza el formato binario), por lo que es relativamente fácil de convertir a otro formato

XML es un lenguaje que permite representar ontologías de acuerdo a agrupaciones de documentos, se emplea en operaciones de búsqueda y consultas al acceder a las etiquetas de los grupos [13]. XML es la base para lenguajes de la web semántica como Resource Description Framework (RDF) y Web Ontology Language (OWL). La Figura 1 muestra la estructura de la ontología que representa la colección de reportes técnicos. Los elementos en gris corresponden

a datos relacionados con la creación de la ontología, éstos no se utilizan para estimar similitud entre reportes.

```
<!ELEMENT ontologyofrecords (algorithm, cluster+)>
<!ATTLIST ontologyofrecords date  CDATA #REQUIRED>
<!ELEMENT algorithm EMPTY>
<!ATTLIST algorithm  name CDATA #FIXED "FICH"
globalsupport CDATA #REQUIRED
clustersupport CDATA #REQUIRED>

<!ELEMENT cluster (label, level, record*, cluster*)>
<!ELEMENT label (#PCDATA)>
<!ELEMENT level (#PCDATA)>
<!ELEMENT cluster (#PCDATA)>

<!ELEMENT record (title, subject?, description, identifier, url, dataprovider,
metadataformat, datestamp, author)>
<!ELEMENT  title (#PCDATA)>
<!ELEMENT  subject (#PCDATA)>
<!ELEMENT  description (#PCDATA)>
<!ELEMENT  identifier (#PCDATA)>
<!ELEMENT  url (#PCDATA)>
<!ELEMENT  dataprovider (#PCDATA)>
<!ELEMENT  metadataformat (#PCDATA)>
<!ELEMENT  datestamp (#PCDATA)>
<!ELEMENT  author (#PCDATA)>
<!ENTITY generateBy "OntoSIR 2.1">
```

Figura 1. Estructura de la colección de reportes técnicos

2.4. Métodos semi-automáticos para construir ontologías

Un método semi–automático crea ontologías por medio de agrupación jerárquica y palabras claves. Las secciones siguientes se describirán dos de estos métodos: GrowBag y OntOAIr.

2.4.1. Método OntOAIr

El método OntOAIr permite crear ontologías a partir de un conjunto de documentos. Consta de cuatro fases [Medina y Sánchez 2008]:

- **Cosecha:** utiliza la solicitud de recolección del protocolo OAI-PMH (*Open Archives Initiative Protocol for Metadata Harvesting)*. Este protocolo se usa con frecuencia en la representación de colecciones y recursos de bibliotecas digitales. La cosecha es un proceso repetitivo, ya que se coleccionan cientos o miles de documentos, depende de factores externos tales como la sobrecarga de la red y disponibilidad de las colecciones.

- **Representación:** utiliza un formato de metadatos para OAI-PMH o un proceso de extracción de información del documento, el cual obtiene palabras claves o términos que simplifica su representación. Por ejemplo, se podría extraer un identificador para discriminar entre los documentos y con una URL la ubicación de éste en una colección.

- **Agrupación:** usa el algoritmo FIHC para obtener un árbol con grupos de documentos similares. El algoritmo se basa en la hipótesis siguiente: si un grupo de documentos se refiere al mismo tema, los documentos comparten un conjunto de términos. El algoritmo crea un vocabulario a partir de los términos frecuentes.

- **Formalización:** es una representación jerárquica del árbol de grupos de documentos en un lenguaje accesible a la máquina como XML, RDF y OWL**.**

Este proyecto considera que la colección de reportes técnicos ya está representada como una ontología construida por el método OntOAIr. Independientemente del método de construcción, existen herramientas que facilitan la edición de ontologías. La siguiente sección describe una de dominio público.

2.4.1 Método GrowBag

GrowBag es un método que hace uso de palabras claves para encontrar objetos en bibliotecas digitales como documentos o imágenes. Se basa en los metadatos de objetos y

facetas de búsqueda, las cuales permiten a los usuarios filtrar los resultados. A menudo, los usuarios realizan diversas búsquedas de palabras claves de forma manual hasta perfeccionar los resultados y generar nuevas palabras claves. La semántica de GrowBag utiliza el algoritmo PageRank para calcular el orden superior de ocurrencias de similitudes transitivas. La idea básica de las facetas de búsqueda se basa en la dispersión, agrupación y recolección de enfoques [14].

Las facetas de búsqueda hacen uso del sistema de clasificación decimal "Dewey". Algunas formas de representar la faceta son [14]:

a) Filtrar los resultados de acuerdo a un conjunto o subárbol: por ejemplo, sólo mostraría los temas relacionados a la búsqueda.

b) Encontrar comunidades: realiza la consulta de acuerdo a su clasificación

c) Caracterizar a los autores o grupos de autores: requiere principalmente de los temas desarrollados.

d) Caracterizar el objeto de resultado: deberá ser conciso en proporcionar el resultado de la consulta de acuerdo a su categorización del tema.

La semántica del algoritmo GrowBag es crear y clasificar los objetos digítales (como documentos e imágenes) de acuerdo a palabras clave como se muestra en la Figura 2. Las palabras X y Z están asociadas con el objeto C, el cual fue el primer objeto arrojado como resultado de la búsqueda de orden de ocurrencia. La palabra clave de Y y Z no están asociadas por el mismo objeto, pero existe otra palabra X que las conecta. Se asume que una ocurrencia de segundo orden u orden superior sucede más frecuentemente que las de primer orden. La creación automática de una clasificación jerárquica de los sistemas como las facetas de temas, al igual que las palabras claves, es útil para representar ontologías.

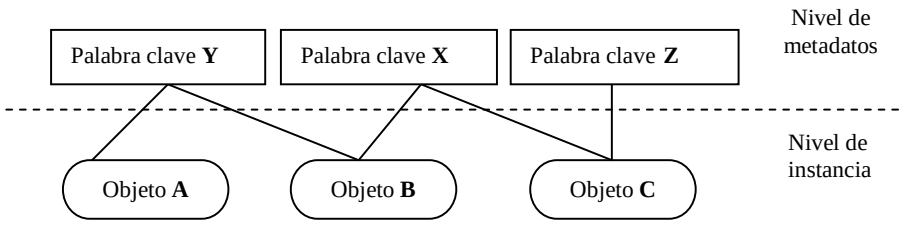

Figura 2. Ejemplo del primer y segundo orden de ocurrencia

2.5 Editor de ontologías Protégé

Protégé es un editor basado en ontologías, construye un modelo de acuerdo al dominio y conocimiento de las aplicaciones como la web semántica. Es un servicio de código abierto, apoya la creación, visualización y manipulación de ontologías en distintos formatos de representación como XML, RDF y OWL [15]. Protégé soporta dos formas de modelado de ontologías: el editor de marcos y el editor de Protégé – OWL. El primero permite a los usuarios construir y poblar las ontologías, consta de un conjunto de clases organizadas jerárquicamente que representan el dominio de conceptos principales, donde las clases describen las propiedades y sus relaciones. El segundo implementa mecanismos definidos para el lenguaje OWL como la validación de la consistencia y la formalización de algunas propiedades.

2.6 Trabajos relacionados

Hoy en día, información importante se almacena en grandes bases de datos y documentos. Su extracción necesita pasar por procesos de transformación, formato, limpieza, deducción y extracción. Una operación común es la agrupación o "clustering", conocida también como aprendizaje no supervisado, cuyo propósito es agrupar la información de acuerdo a criterios de similitud. La agrupación se ha aplicado a una amplia gama de temas y áreas como física, matemáticas, programación, estadística, computación científica, entre otros. En el proyecto, el método OntOAIR emplea el algoritmo de agrupación FIHC para formar grupos de documentos similares [16].

Algunas técnicas de medición de similitud comúnmente utilizadas son coseno e índice de Jaccard, las cuales consideran superposición de términos. En ocasiones también se emplea similitud semántica, la cual considera las relaciones entre conceptos utilizando ontologías. Este tipo de similitud propone el empleo de métricas que utilizan la representación de grafo de las ontologías. Al recorrer un grafo, se implementa un proceso de propagación que considera la relación intrínseca que puedan existir entres dos conceptos [17].

Algunos sistemas implementan medidas de similitud semántica en documentos con dominios particulares. Por ejemplo, el sistema Eureka accede a una colección que tiene alrededor de 40,000 documentos con consejos técnico–autor generados en una organización encargada de la reparación de fotocopiadoras. Este sistema realiza las tareas siguientes: 1) identifica documentos similares, 2) detecta las secciones de dos documentos que se superponen y 3) establece si hay relación o contradicciones entre dos documentos. Estas tareas requieren de técnicas de procesamiento de lenguaje natural y el uso de la ontología WordNet (WordNet es una ontología de propósito general que se considera una base de datos léxica, contiene la definición de cerca de 110 000 palabras), [18].

Una ontología que apoya la representación del contenido de los textos comprende eventos, entidades y las relaciones que hay entre los diferentes dominios, expresa conceptos específicos con un nivel medio de abstracción. La abstracción en la representación y la normalización de las dimensiones comparativas son dos criterios de diseño de ontologías que apoyan la búsqueda de similitudes de documentos [19].

Las imágenes al igual que texto requieren de herramientas de búsqueda, recuperación, clasificación, visualización y procesamiento. Estas tareas se realizan en una gran variedad de dominios como medicina, patrimonio cultural, medio ambiente e ingeniería. La búsqueda de imágenes se realiza en ocasiones a través de descripciones textuales y especificando palabras claves. Se emplean ontologías para simplificar y ayudar a la recuperación e indexación de regiones de imágenes basándose en las características de la forma del objeto contenido. El uso de ontologías facilita la búsqueda, incorpora recuperación semántica y apoya la visualización de las imágenes [20].

Los trabajos descritos en este capítulo tienen como objetivo mostrar la incorporación de las ontologías para implementar tareas que procesan texto en lenguaje natural. En el siguiente capítulo se describe la metodología a seguir para utilizarlas al estimar similitud entre documentos.

Capítulo 3. Metodología

Este capítulo describe la metodología propuesta para llevar a cabo el proyecto. Incluye la descripción de requerimientos, los casos de uso principales, la especificación de actores y el diagrama conceptual del sistema.

3.1 Requerimientos

Las Tablas 3-5 describen los requerimientos funcionales para el sistema.

Tabla 3. Requerimiento funcional "ver grupo por niveles"

Descripción corta:	Muestra los grupos de los reportes técnicos por niveles (1, 2, 3 o todos).
Descripción detallada:	1.- El usuario elige un nivel 2.- El sistema abre el archivo con la colección de reportes técnicos y muestra las etiquetas de los grupos del nivel seleccionado 3.- El usuario selecciona una etiqueta y consulta los datos del grupo o de los reportes

Tabla 4. Requerimiento funcional "consulta de referencias"

Descripción corta:	Consultar las referencias de un reporte técnico
Descripción detallada:	1.- El usuario elige un reporte técnico 2.- El sistema abre un archivo con las referencias del reporte seleccionado

Tabla 5. Requerimiento funcional "calcular la similitud"

Descripción corta:	Se calcula la similitud entre un nuevo reporte técnico y la

	colección de acuerdo al título, tema y descripción.
Descripción detallada:	1.- El usuario tiene los datos descriptivos de un nuevo reporte técnico 2.- El usuario elige si la comparación se hará por sección o en todas las secciones 3.- El sistema despliega los resultados de la comparación en texto

Los requerimientos no funcionales son los siguientes: la plataforma y compatibilidad del sistema considera el sistema operativo Windows XP. Se accede a él a través de un navegador de Internet como Mozilla Firefox. Se asignan identificadores de usuario y administrador para controlar el acceso al sistema. El usuario puede realizar consultas, el administrador además acceso a los datos y da mantenimiento continuo al sistema. El sistema no dependerá de otras partes a menos que sea el servidor web no esté en funcionamiento. La eficiencia y el desempeño dependen del tamaño de la colección. Será un sistema hecho a la medida, permitirá extensibilidad para agregar algún módulo en el futuro y se desarrollará con software libre con fines educativos.

3.2. Casos de uso

La Figura 3 muestra las tareas que podrán ser realizadas por el usuario y el administrador. Cada elipse representa un caso de uso, los cuales se describen a continuación:

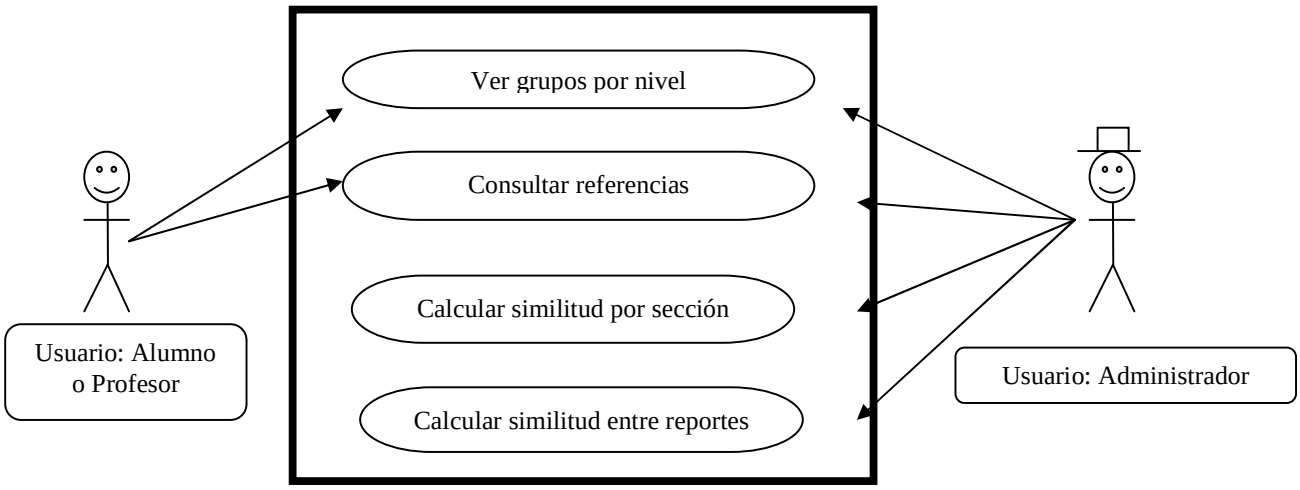

Figura 3. Acceso a módulos

1) Caso de uso "ver_grupos_por_nivel"

Datos de salida:

- ❖ Mensaje de error si no hay información sobre el tema
- ❖ Mensaje de error si el servidor web no está disponible
- ❖ Lista de etiquetas de cada grupo

2) Caso de uso "consultar_referencias"

Datos de entrada:

- Título de un reporte técnico

Datos de salida:

- ❖ Mensaje de error si "no hay información sobre el reporte técnico" o
- ❖ Mensaje de error si el servidor web no está disponible o
- ❖ Lista de referencias del reporte técnico seleccionado

3) Caso de uso "calcular_similitud_por_sección"

Datos de entrada:

- Archivo con un reporte técnico nuevo descrito por título, tema y descripción (o resumen)
- Selección de secciones a comparar

Datos de salida:

- ❖ Mensaje de error si el servidor web no está disponible
- ❖ Ventana de resultado que muestre la similitud en las secciones seleccionadas (título, tema o descripción) con su respectivo porcentaje de similitud en caso de que éste sea mayor a cero

4) Caso de uso calcular_similitud_entre_reportes

Datos de entrada:

- Archivo con un reporte técnico nuevo completo
- Comparar toda la estructura de los reportes

Datos de salida:

- ❖ Mensaje de error si el servidor web no está disponible
- ❖ Ventana de resultado que muestra la similitud en las secciones (título, tema y descripción) con su respectivo porcentaje.
- ❖ Enlace para visualizar los documentos similares

3.3 Especificación de actores

Las Tablas 6-7 contienen la especificación de los actores del sistema.

Tabla 6. Especificación del actor administrador

Descripción:	Persona que administra el sistema de similitud de reportes técnicos
Características:	El administrador se encarga de dar mantenimiento al sistema
Relaciones:	Interactúa con el sistema para consultar las referencias, calcular la similitud y ver los grupos por niveles de los reportes técnicos. Se comunica con los asesores y sinodales para dar a conocer los resultados.

Tabla 7. Especificación del actor usuario

Descripción:	Persona que ingresa y consulta el sistema de detección de similitud
Características:	Alumno regular inscrito en un semestre superior al séptimo o profesor
Relaciones:	Interactúa con el sistema para consultar las referencias de un reporte técnico o datos de la colección.

3.4 Diagrama conceptual

La Figura 4 muestra el diagrama navegacional del *sistema de detección de similitud entre reportes técnicos* (en adelante se denotará como *DSRT*), el cual indica que desde la página principal se muestra un menú de ayuda o se inicia una sesión según el tipo de usuario.

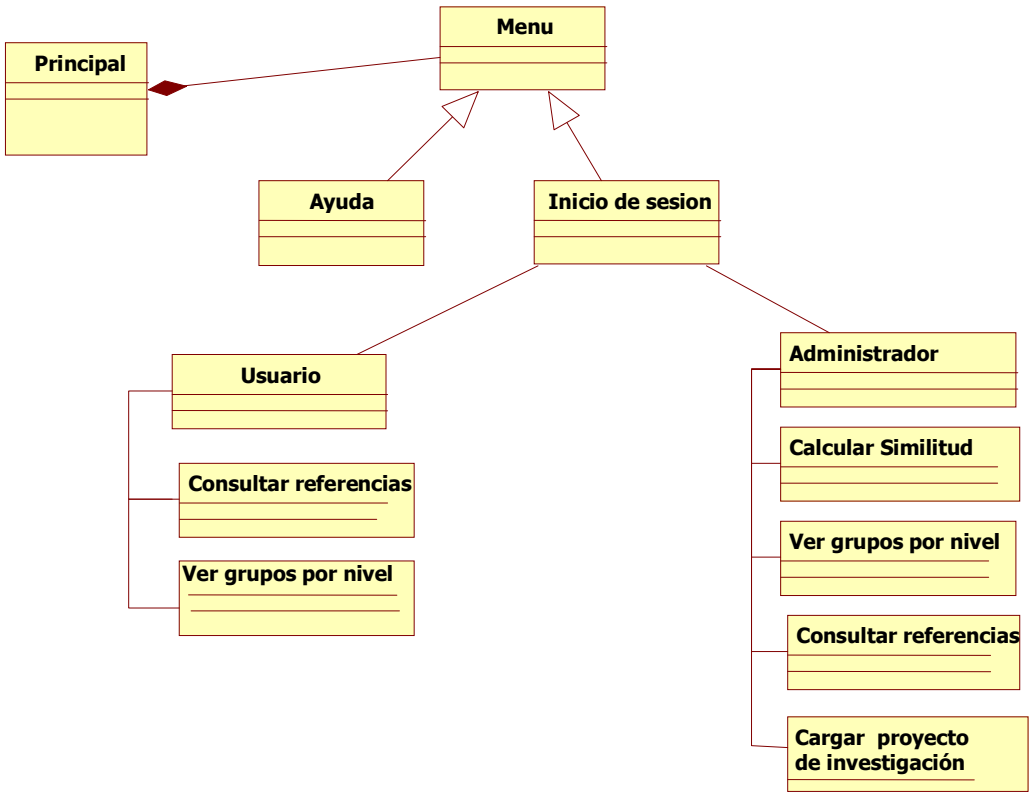

Figura 4. Diagrama navegacional del sistema

Los datos de entrada, datos de salida y la descripción de algunos módulos de la Figura 4 se muestran en las tablas 8-12.

Tabla 8. Descripción del módulo "ayuda".

Descripción	Contiene la descripción de cada una de las tareas con ejemplos.
Datos de entrada	Tarea a consultar
Datos de salida	Descripción y ejemplo de la tarea

Tabla 9. Descripción del módulo "ver por nivel"

Descripción	Muestra por nivel las palabras que corresponden a las etiquetas de los grupos en los niveles seleccionados.
Datos de entrada	Nivel
Datos de salida	Lista de los título de los reportes técnicos del nivel seleccionado.

Tabla 10. Descripción de módulo "calcular similitud" por sección o entre reportes.

Descripción	Cuenta con las siguientes opciones al cargar un nuevo reporte técnico: 1) comparar el documento entre reportes o 2) comparar el documento por sección. Muestra los datos descriptivos de los reportes en caso de que encuentre similitud entre reportes o en secciones.
Datos de entrada	Datos descriptivos de un reporte técnico nuevo
Datos de salida	Lista con grado de similitud entre reportes técnicos por sección o entre documentos.

Tabla 11. Descripción de módulo "consultar referencias".

Descripción	Muestra la lista de referencias de un reporte técnico seleccionado por título
Datos de entrada	Título del reporte técnico
Datos de salida	Lista de referencias del reporte técnico

Tabla 12. Descripción de módulo “carga de reporte técnico”.

Descripción	Se realiza la carga del archivo que representa la colección de reportes técnicos. Este archivo tiene la estructura descrita en la Figura 1.
Datos de entrada	Archivo que representa la colección de reportes técnicos
Datos de salida	Mensaje de éxito si se cargó el archivo correctamente o mensaje de error en caso contrario.

La Figura 5 muestra la pantalla de acceso al sistema, ésta contiene el nombre del sistema, la universidad en donde se desarrolló y una breve descripción.

Figura 5. Pantalla de acceso

Las Figuras 6-8 muestran el menú y algunas tareas que podrán realizarse.

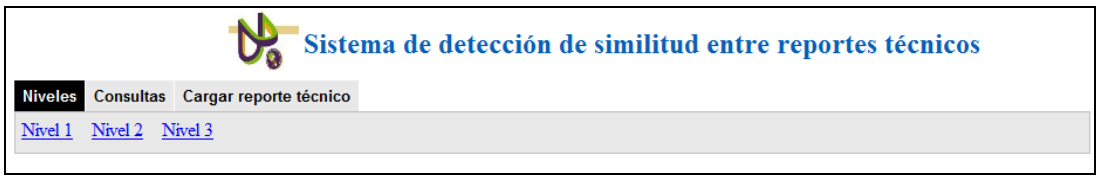

Figura 6. Pantalla para seleccionar los grupos de un nivel

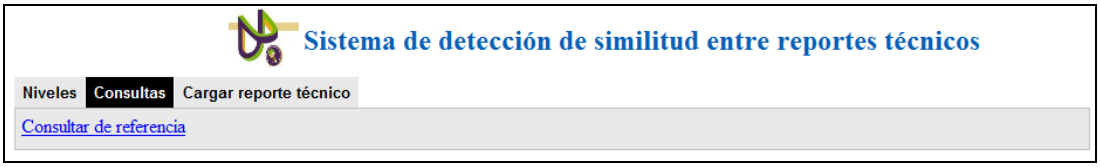

Figura 7. Pantalla para consultar referencias

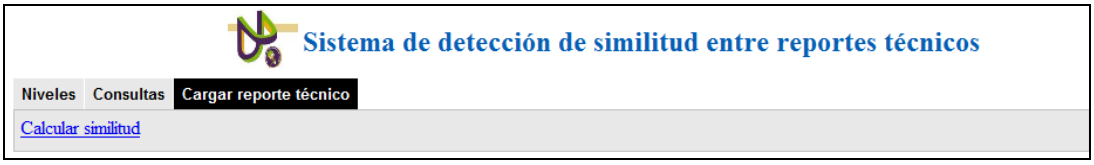

Figura 8. Pantalla para seleccionar la sección a comparar

Capítulo 4. Implementación

Este capítulo está organizado de la siguiente manera: primero se explica un ejemplo de una ontología y se indican cuáles de sus elementos se emplean en el sistema. Después se describe la función que calcula la similitud entre cadenas, ésta se utiliza para establecer el porcentaje de coincidencia entre secciones y reportes. Finalmente se presenta la arquitectura del sistema X y se describe la implementación de sus módulos.

4.1. Manejo de la ontología

La Figura 9 muestra la estructura del archivo *ontologyofrecords.xml* que describe una ontología. El número entre paréntesis corresponde al nivel de cada grupo, el texto forma parte de la etiqueta. El nivel 0 es la raíz, el nivel 1 contiene la etiqueta de los grupos procesamiento, robótica y simulación. Las etiquetas del nivel 2 contienen 2 palabras, la del nivel y la de su ancestro. Las etiquetas del nivel 3 contienen 3 palabras, la del nivel 3 más 2 de sus ancestros. La imagen de un libro representa un reporte técnico. Considere el ejemplo siguiente. La etiqueta del grupo del nivel más profundo es natural, se podría interpretar como sigue: el grupo padre de natural es lenguajes y el abuelo es procesamiento, entonces quiere decir que los reportes del grupo del nivel más profundo corresponden a Procesamiento (de) →Lenguaje → Natural. El título y el resumen de todos los reportes en este grupo contienen las 3 palabras de la etiqueta.

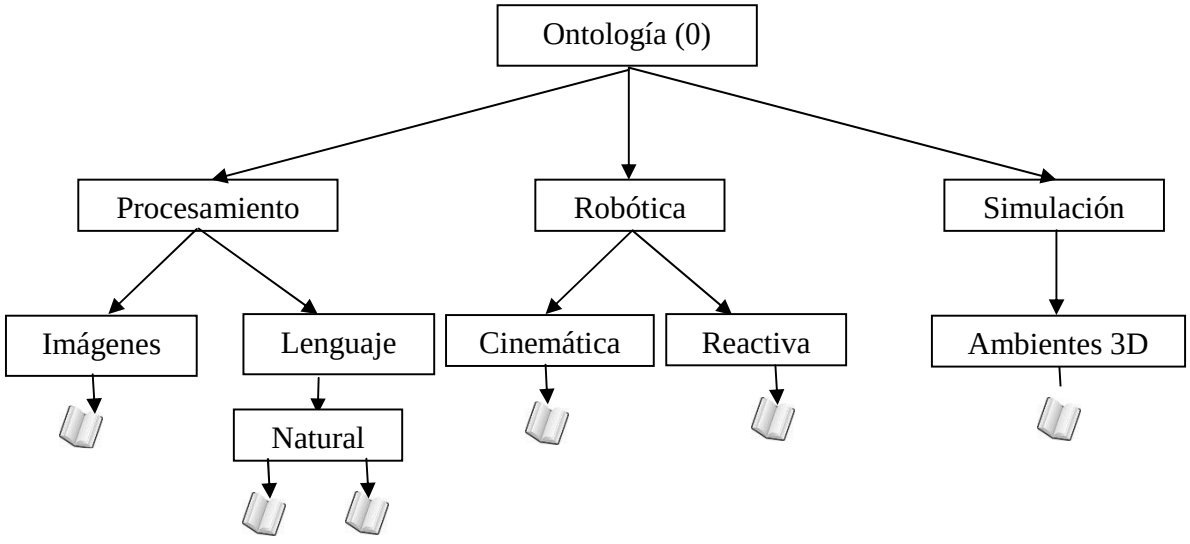

Figura 9. Ejemplo de la ontología del archivo *ontologyofrecord.xml*

La Tabla X detalla los elementos que contiene el archivo de ontologyofrecords.xml necesarios para extraer los datos. Estos elementos forman parte del estándar de metadatos Dublin Core [21], el cual se emplea en bibliotecas digitales. Dublín Core es el estándar por omisión del protocolo OAI-PMH.

Tabla 13. Elementos del DTD utilizado por DSRT

Elemento	Descripción
CLUSTER	Representa un grupo, tiene una etiqueta y un nivel, así como otros grupos.
LABEL	Es la etiqueta del grupo, muestra 1, 2 o 3 palabras dependiendo del nivel.
LEVEL	Indica la profundidad del grupo en la ontología.
RECORD	Representa un reporte, contiene a los elementos título, tema, descripción e identificador.
TITLE	Es el título de cada uno de los reportes técnicos.
SUBJECT	Almacena el tema, puede o no tenerlo.
DESCRIPTION	Contiene el resumen del reporte realizado.
IDENTIFIER	Es un identificador que contiene cada reporte, está asociado a un archivo .txt el cual muestra la bibliografía.

4.2. Eliminación de palabras vacías

Para el desarrollo del sistema DSRT fue necesario hacer uso de un arreglo de palabras vacías (stopwords), éste contiene palabras como artículos, pronombres, preposiciones y algunos adverbios. En la búsqueda de información y en el procesamiento de los campos de entrada, fue necesario eliminar las palabras vacías para reducir el tiempo de procesamiento de datos en lenguaje natural (texto).

4.3. Calculo de similitud entre cadenas

La fórmula de similitud entre 2 cadenas considera la primer cadena como base (proviene del archivo *ontologyofrecords.xml*) y la segunda como nueva (la teclea el usuario). La

comparación se realiza calculando la longitud de la cadena base y la longitud de la cadena que representa la intersección entre la cadena base y la cadena nueva. Previamente se han eliminado de estas cadenas las palabras vacías y las palabras restantes se convierten a mayúsculas. Esta fórmula de similitud es llamada por la función *cmpString*. En el cuadro siguiente se describe.

Longitud_cadena_intersección: Es el número de palabras que coinciden con la cadena base y la cadena nueva, se aplica en la comparación de título, tema o descripción.

Longitud_cadena_base: Es el número de palabras que se encuentran en el archivo *ontologyofrecords.xml* entre etiquetas que representa algún elemento (título, tema o descripción).

El ejemplo que se muestra a continuación los arreglos ya están sin palabras vacías y en mayúsculas:

Longitud_cadena_interseccion=

IMAGENES	3D

=2

Longitud_cadena_base=

COMPRESION	*IMAGENES*	*RED*	*NEURONAL*	*ARTIFICIAL*

=5

$$\text{Similiud}=\frac{2*100}{5}=\frac{200}{5}=40\ \%$$

4.4 Algoritmo de detección de plagio

Para detectar similitud entre reportes técnicos, se emplea el algoritmo que usa la notación siguiente:

- Un informe técnico *tr* de *m* palabras se representa como una tupla tr (tr_1,tr_2,…tr_m) donde *m* es el número de palabras clave de *tr*.

- Una ontología de los registros se representa como una tupla *O (t, d, c, r)* donde *t* es el número de nodos en el primer nivel (que corresponde con el número de grupos con

etiquetas de una sola palabra), *d* es el número de niveles, *c* es el número de grupos y *r* es el total número de documentos.

Entrada: *tr:* nuevo reporte técnico, *O (t,d,c,r):* ontología de reportes técnicos

Salida: Lista de reportes técnicos similares

```
Inicio
     For each etiqueta  l ∈ O do (5)
          Aplicar S(tr,I) (6)
          Formar un arreglo Lc en un orden descendente de
     acuerdo a S(tr, l) (7)
          if (S(tr,I)) > un umbral)
               Mostrar el porcentaje de similitud
Fin
```

Nota: S*(tr,I)* denota la similitud entre la etiqueta *l* y un reporte técnico *tr* $(tr_1, tr_2, \ldots, tr_m)$, es una notación simplificada de la función de similitud de la sección 4.3.

4.5 Implementación de módulos y funciones

La Figura 10 describe la arquitectura (interacción entre módulos, funciones y clases) que se implementó en el desarrollo del sistema DSRT. Cada uno de los módulos hace uso de las función y clases de acuerdo a la líneas, llaves y formato de letra marcados en la figura. A continuación se describen los módulos.

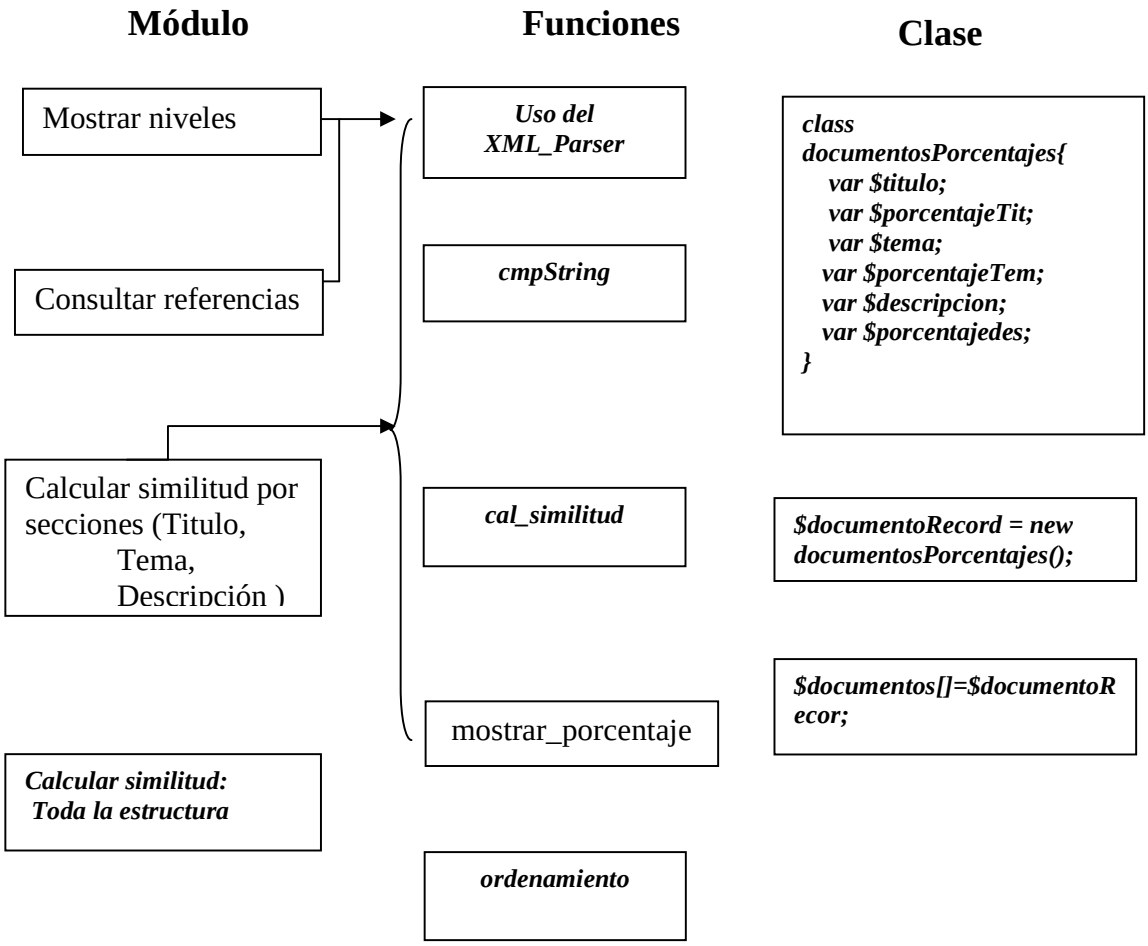

Figura 10. Arquitectura de DSRT

A) Módulo mostrar niveles

El módulo "mostrar niveles" hace uso de la función "uso del XML_Parser", realiza una comparación de la etiqueta nivel <LEVEL> y muestra la palabra del grupo. La función "uso del XML_Parser" llama a otras funciones que son necesarias para abrir, extraer los elementos de cada etiqueta y la información que está dentro de las etiquetas.

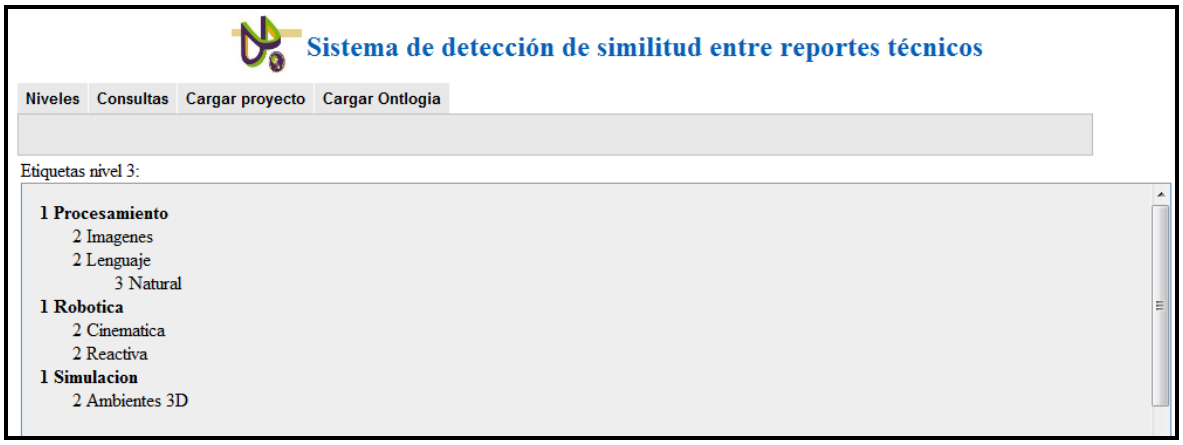

Figura 11. Nivel de etiquetas

B) *Módulo consultar referencias*

El módulo "consulta de referencia" al igual que el módulo anterior hace uso del la función "uso del XML_Parser", la cual extrae la información de las etiquetas <TITLE> e <IDENTIFIER> del reporte. Los valores de estos elementos se muestran en la Figura 12.

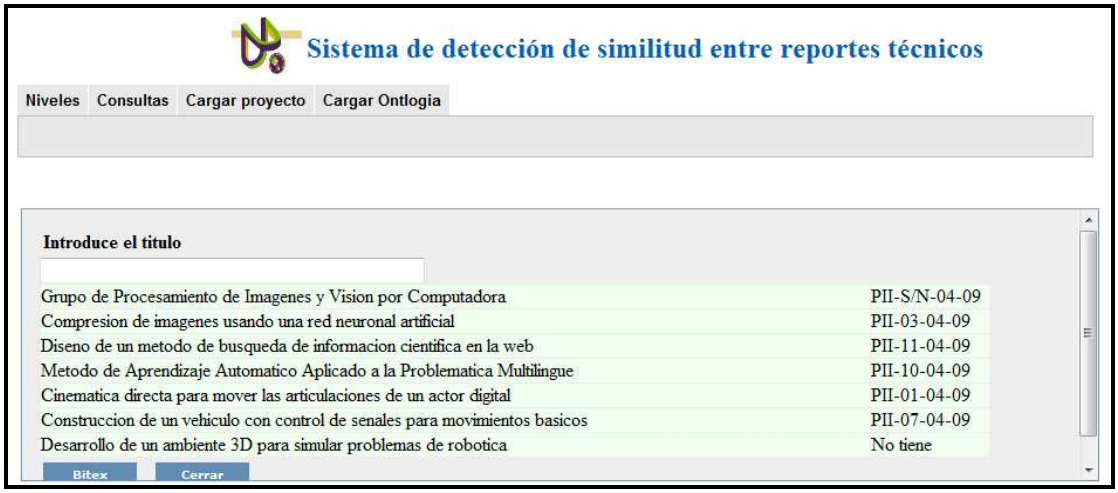

Figura 12. Consulta de referencias

C) *Módulo calcular similitud por secciones*

El módulo "calcula similitud por secciones" necesita de las primeras cuatro funciones que se muestran en la Figura 10. La primera realiza la agrupación de información por etiquetas de título, tema y descripción, que pertenecen a un grupo de reportes de la ontología. La segunda

prepara las cadenas base y nueva para calcular su intersección. La tercera, si encuentra intersección, calcula la similitud de las cadenas, para ello hace uso de la fórmula descrita en la sección 4.2. La cuarta función muestra el porcentaje de similitud por título, tema o descripción; emplea 3 arreglos, uno por sección. En cada arreglo se almacena el contenido de la sección y el porcentaje de similitud. Los arreglos se ordenan de forma descendente según el porcentaje.

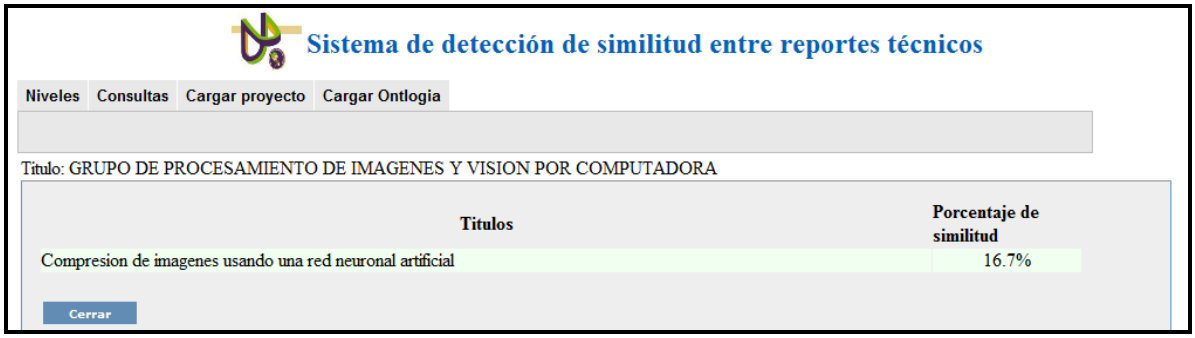

Figura 13. Cálculo de similitud por título

D) Módulo calcular similitud por estructura

El último módulo "cálculo de similitud entre reportes" hace uso de la clase *documentosPorcentajes* , el objeto $*documentoRecord* , el arreglo de objetos $*documentos*, la función "uso del XML_Parser", *cmpString*, ***cal_similitud*** y el método de ordenamiento. El módulo revisa el título, tema y descripción de todos los reportes. En caso de tener un valor de similitud mayor a cero en las 3 secciones, se almacena el contenido de las secciones y sus porcentajes respectivos en un objeto. Los objetos se almacenan en un arreglo, el cual se ordena descendentemente según el porcentaje del título. La Figura 9 muestra las funciones internas y la secuencia de llamadas que componen a la función "uso del XML_Parser".

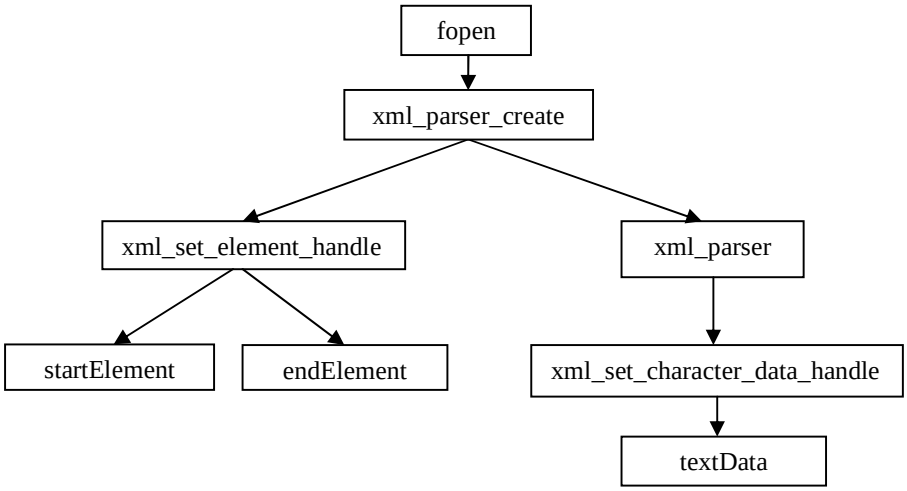

Figura 14. Arquitectura de la función "uso del XML_Parser"

Las tablas 14-17 describen las funciones de la Figura 11, consideran los datos de entrada, los datos de salida y la descripción de la función. Estos datos provienen del manual de Php Xml Parser [22].

Tabla 14. Función *fopen*, lectura del archivo XML

Datos de entrada:	La función fopen recibe 2 parámetros de tipo cadena: Nombre del archivo: 'ontologyofrecords.xml' y operación a realizar con él, ejemplo: *'ontologyofrecords.xml'*, 'r', donde 'r' indica que la operación es de tipo lectura.
Datos de salida:	No contiene parámetros de salida
Descripción:	Realiza la lectura del archivo *'ontologyofrecords.xml'*

Tabla 15. Función *xml_parser_create*, parser de XML

Datos de entrada:	No contiene parámetros de entrada
Datos de salida:	Devuelve un recurso para manejar el nuevo analizador XML.
Descripción:	La función xml_parser_create(), crea una variable de tipo parser XML, la cual crea un nuevo analizador XML y devuelve una referencia de recursos que manejan para ser utilizado por las otras funciones XML.

Tabla 16. Función x*ml_set_element_handler*, configuración de inicio y final elemento manipuladores

Datos de entrada:	La función xml_set_element_handler($parser, "startElement", "endElement"), tiene 3 parámetros de entrada. El primero es de tipo xml_parser() y los últimos 2 parámetros son de tipo cadena. $parser: analiza el documento, llama a la función: "startElement", (etiqueta de inicio del un elemento) y a la función: "endElement", (etiqueta de fin del elemento)
Datos de salida:	Devuelve un valor booleano que valida si existen las etiquetas de inicio y término de los elementos.
Descripción:	Esta función recorre los elementos del archivo '*ontologyofrecords.xml*'.

Tabla 17. Función *xml_set_character_data_handler*, crea datos de carácter manejador

Datos de entrada:	La función xml_set_character_data_handler ($parser, "textData"), tiene 2 parámetros de entrada, el primero es de tipo xml_parser(),el segundo es una función que extrae el texto de un elemento cuyo contenido es texto. $parser: Analiza el documento Llama a la función: " textData "
Datos de salida:	Devuelve un valor booleano que indica si el elemento contiene texto o no.
Descripción:	La información de caracteres es por definición todo el contenido no "marcado" de los documentos XML, incluidos los espacios en blanco entre etiquetas. Note que el intérprete XML no añade o elimina ningún espacio en blanco, depende de la aplicación decidir si el espacio en blanco es significativo o no.

La Tabla 18 describe a las funciones de la Figura 11 que se encuentran en la parte inferior, éstas se emplean para extraer el nombre y el contenido de los elementos. No forman parte de la documentación de [22]. Se extrajeron de un ejemplo disponible en [23]

Tabla 18. Funciones *textData, startElement y endElement*

Función	**Descripción**
textData	La función textData($parser, $text) es llamada por la función xml_set_character_data_handler ($parser, "textData"). El primer parámetro analiza el documento, el segundo extrae el texto.
startElement	Esta función busca la etiqueta de inicio de un elemento.
endElement	Esta función busca la etiqueta de término de un elemento.

Las Tablas 19-22 describen las funciones que se implementaron para realizar el cálculo de similitud propuesto.

Tabla 19. Función *cmpString,* comparación de cadenas

Datos de entrada:	Cadena base: es aquella que está en el archivo .XML. Cadena nueva: tecleada por el usuario, se procesa para eliminar las palabras vacías Arreglo de palabras vacías
Datos de salida:	La función regresará un 0 si no intersección entre las cadenas. En otro caso, se regresa el porcentaje de similitud

A continuación se muestra el pseudocódigo de la función *cmpString*:

```
1. Eliminar espacios en blanco de las cadenas
2. Convertir las cadenas a mayúsculas
3. Dividir las cadenas en elementos para formar un arreglo por
   cadena
4. Ordenar los arreglos alfabéticamente
5. Eliminar las palabras vacías de los arreglos
6. Obtener el tamaño de los arreglos
7. Calcular la intersección entre arreglos
8. Si la intersección == 0
9.      Regresar 0
```

10. Si no, llamar a la función cal_similitud para obtener el porcentaje de similitud

Tabla 20. Función *cal_similitud,* calcula similitud

Datos de entrada:	Longitud cadena base: contiene el tamaño del arreglo de la cadena base. Longitud cadena intersección: tiene el tamaño del arreglo de intersección
Datos de salida:	La función regresa el porcentaje de similitud.
Descripción:	La similitud entre 2 cadenas se realiza con la longitud de cadenas de intersección y base, ambas cadenas están sin palabras vacías. La función aplica la fórmula descrita en la sección 4.3 del capítulo 4.

Tabla 21. Función *ordenamiento,* método de ordenamiento de la burbuja

Datos de entrada:	Arreglo de objetos: es aquel que contiene los siguientes atributos titulo, porcentaje del título, tema, porcentaje del tema, descripción y porcentaje descripción.
Datos de salida:	La función regresa un arreglo de objeto ordenado descendentemente porcentaje de título.
Descripción:	Esta función obtiene el tamaño del arreglo de objetos y recorre cada uno de los elementos por el porcentaje de titulo y lo ordena descendentemente.

Tabla 22. Función *mostrar_porcentaje*, muestra el porcentaje

Datos de entrada:	Datos: Es un arreglo, tiene 2 atributos, el primero es el título y el segundo es el porcentaje de título, o tema y porcentaje de tema o descripción y porcentaje de descripción. num: Esta variable puede tener los siguiente valores 1 o 2 o 3, indica qué va a mostrar: 1) el arreglo de títulos y porcentajes, 2) el arreglo de temas y porcentajes o 3) el arreglo descripción y porcentaje.
Datos de salida:	Regresa un valor booleano si se realizó correctamente el despliegue del arreglo o en caso de algún fallo.
Descripción:	La función regresa un arreglo de objeto ordenado descendentemente por porcentaje de similitud del título, o porcentaje del tema o porcentaje de descripción. Se usa para ordenar varias matrices a la vez o una matriz multi-dimensional por una o más dimensiones. Emplea las banderas de orientación del ordenamiento siguientes: • *SORT_ASC* - Ordenar ascendentemente • *SORT_DESC* - Ordenar descendentemente

Capítulo 5. Pruebas

El tipo de pruebas que se proponen para el sistema DSRT son 2, ambas utilizan el archivo *'ontologyofrecords.xml'* que contiene 6 reportes técnicos. Esto porque el archivo se construyó como un caso de prueba. El sistema DSRT no está limitado por algún número fijo de reportes.

El primer tipo de pruebas consiste en elegir un reporte técnico de los 6, cambiarlo de posición en una rama para alterar su nivel y calcular la similitud en cada caso. Se eligió el reporte técnico que estuviera en el nivel más profundo, es decir, al azar se eligió uno de los 2 reportes técnicos del grupo “Procesamiento de Lenguaje Natural”. A continuación se muestra el reporte que se utilizó para la prueba.

```
Agrupo: Natural
Nivel: 3
Titulo: Metodo de Aprendizaje Automatico Aplicado a la
Problematica Multilingue
Tema: Busqueda de respuestas
Descripción: Los sistemas BR poseen hoy en dia un amplio
interes, debido al aumento de la informacion en la web y a la
necesidad cada vez mas urgente de obtener informacion precisa.
La busqueda de respuestas en un entorno multilingue hace frente
a problemas como la fusion de respuestas de colecciones de
diferentes idiomas. Este problema no ha sido bordado de manera
amplia. El principal conflicto es manejar l orden que presentan
las respuestas a una pregunta. Este proyecto propone la fusion
de respuestas multilingues a traves del procesamiento de las
listas de respuestas y de la regunta. Dicho procesamiento
incluye etapas como la identificacion del tipo de pregunta, la
identificacion de una misma respuesta entre las distintas listas
```

usando la traduccion, la identificacion de respuestas correctas a una pregunta, entre otras.

Tabla 23. Resultados del tipo de prueba 1

Nivel	Etiqueta	Nivel	Etiqueta	Porcentaje de similitud
1	Procesamiento	1	Procesamiento	Titulo:100 %. Tema: 100%. Descripción:100%
		3	Lenguaje	Titulo:100 %. Tema: 100%. Descripción:100%
2	Lenguaje	2	Lenguaje	Titulo:100 %. Tema: 100%. Descripción:100%
		3	Natural	Titulo:100 %. Tema: 100%. Descripción:100%
3	Natural	3	Natural	

El segundo tipo de prueba consiste en suponer un séptimo reporte técnico y compararlo con las ramas de la ontología para calcular la similitud entre reportes técnicos en los diferentes niveles. La rama: Procesamiento (de)→Imágenes es la que se usó para la prueba.

Tabla 24. Resultados del tipo de prueba 2

Nivel	Etiqueta	Nivel	Etiqueta	Porcentaje de similitud
1	**Procesamiento**	**1**	**Procesamiento**	Titulo:100 %. Tema: 100%. Descripción:100%

Capítulo 6. Conclusiones

Para la construcción de la ontología de ejemplo se hizo uso del método OntOAIr y sus cuatro fases. En la primera fase (*cosecha*) se consiguió reunir 6 reportes al inicio de Mayo del 2009. En la segunda fase, que es la *representación* se utilizó el formato de metadatos Dublín Core y el protocolo OAI-PMH, la representación de los documentos consideró las palabras clave. En la tercera fase, la de *agrupación*, manualmente se aplicó el algoritmo FIHC para formar una jerarquía de los reportes preservando las características de este algoritmo, en particular, recordar que los términos de las etiquetas aparecen en cada uno de los documentos que forma el grupo. La cuarta fase, la *formalización*, se hizo uso del editor de ontologías Protégé para crear y validar el archivo en XML denominado ontologyofrecords.xml. Este archivo representa en una estructura jerárquica al conjunto de reportes técnicos.

En el proyecto se creó una nueva versión del DTD que representa a la ontología de los reportes técnicos. La construcción del documento hizo uso de elementos propuestos por el estándar de metadatos Dublin Core, el cual se emplea en el área de bibliotecas digitales. El DTD muestra la composición de cada uno de los elementos y sus relaciones, de forma que su interpretación es legible por los humanos y las computadoras.

Se realizó un sistema de detección de plagio basado en la ontología construida el cual automatiza el cálculo de la similitud entre reportes técnicos de acuerdo a un conjunto de palabras (vocabulario) común entre reportes. La similitud se aplica en secciones del reporte o entre reportes. La forma de evaluar el grado de similitud entre reportes técnicos propuesta, permite evaluar el número de palabras que coinciden entre los reportes técnicos que se encuentran en el archivo *ontologyofrecords.xml* y el nuevo reporte técnico que es tecleado por el administrador.

El sistema permite también que los usuarios puedan consultar la lista de referencias bibliográficas de los reportes, con el propósito de fomentar los derechos de autor y el reconocimiento a la propiedad intelectual de los documentos revisados durante un proyecto de investigación.

Se concluye que se logró el reconocimiento de similitud entre reportes técnico mediado por ontologías. El sistema permite acceder a la ontología por niveles, lo cual ofrece un panorama general o específico del contenido de la colección. Por otro lado, el empleo de niveles fungirá como un mecanismo de filtrado de documentos cuando el número de éstos se incremente.

Como trabajo a futuro se propone el empleo de otras fórmulas de similitud entre cadenas base y las nuevas, así como la extensión de la búsqueda mediante dos métodos: 1) incorporación del uso de sinónimos para palabras clave y 2) manejo de sufijos y prefijos de cada palabra clave para extender la similitud a familias de palabras. Otra extensión de este propone la búsqueda de similitud en la sección de referencias a través de un análisis sintáctico y semántico de los archivos Bibtex.

Referencias

[1] Corporación Universitaria para el Desarrollo de Internet A.C., Día Virtual sobre Bibliotecas Digitales, Romo F.Z. 2005, 11 de febrero 2009. Disponible en: http://www.cudi.edu.mx/aplicaciones/dias_cudi/05_03_09/dia_cudi_05_02_09.htm

[2] Hocsman Abogados, Deficiencias legislativas en la protección de derechos de autor en software: propuesta de creación de la Ley de Derechos de Autor en Software, Aldana C.Z., 1997, 11 de febrero 2009. Disponible en: http://www.justiniano.com/revista_doctrina/Derecho_de_autor_en_software.htm

[3] Instituto de investigaciones eléctricas, Propiedad Intelectual Ley Federal de Derechos de 2006, 11 de febrero 2009. Disponible en: http://axp16.iie.org.mx/promocio/patentes/paginas/derinf2.htm

[4] Gruber, T.R.,1999 , What are Ontologies and why do we need them?, *IEEE Intelligent Systems*,1233276 (Enero), 20-21

[5] Ley Federal de Derechos de Autor, Artículos de la ley federal del derecho de autor que protegen los programas de cómputo, 2005, 03 de marzo de 2009. Disponible en: http://www.cem.itesm.mx/di/seguridad/servicios/documentos/articulosProtegenProgramasComputo.html

[6] Universidad Politécnica de Puebla, Modelo Educativo, 2006, 05 de marzo de 2009, http://www.uppuebla.edu.mx/U_modelo.html

[7] Resenos, E., 1998, Guía para la elaboración de protocolos de investigación. Instituto Politécnico Nacional, México DF.

[8] Hernández R. y Fernández C. 2003. Metodología de la investigación. Mc Graw Hill.

[9] Saravia Gallardo M.A. 2006. Metodologías de Investigación. CONACYT, México, D.F.

[10] Institut Universitari de Linguisitca Aplicada Universitat Pompeu Fabra, Generación semiautomática de ontologías, 2007, 05 de marzo de 2009. Disponible en: http://www.iula.upf.edu/materials/070223pedraza.pdf

[11] Hanke, J.C y KnowWare EURL, 2002, Introducción a XML, *Cuadernos técnicos,* 8 (Septiembre), 7-9.

[12] W3schools. Introduction to XML, copyright 1999 - 2009, 02 de Marzo de 2009. Disponible en: http://www.w3schools.com/xml/xml_whatis.asp

[13] Medina, M. A., Sánchez, J. A. 2008. OntOAIr: A method to construct lightweight ontologies from document collections. Proceedings of the Ninth Mexican International Conference on Computer Science (ENC 2008, Mexicali, Mexico), 115-125.

[14] Diederich J. and Balke W. 2007. The Semantic GrowBag Algorithm: Automatically Deriving Categorization Systems. 11th European Conference on Research and Advanced Technology for Digital Libraries (ECDL).

[15] Stanford Center for Biomedical Informatics Research. Protégé. Copyright 2009, 06 de marzo 2009. Disponible en: http://protege.stanford.edu/overview

[16] Glenn F. 2001, A Comprehensive Overview of Basic Clustering Algorithms. 3-4

[17] Hewlett-Packard Labs, Bangalore, India,2008,Computing Semantic Similarity Using Ontologies, 87

[18] Christiane F., WordNet: An Electronic Lexical Database (Language, Speech, and Communication), Mayo 15 1998, 2009

[19] John O., Daniel G., Martin V., Xerox PARC, FXPAL, Making Ontologies Work for Resolving Redundancies Across Documents, 1-2

[20] Sánchez L., Tesis: modelo de indexación de formas en sistemas VIR basado en ontologías. Universidad de las Américas Puebla, 2007, 15 de marzo de 2009, Disponible en:
http://catarina.udlap.mx/u_dl_a/tales/documentos/mcc/sanchez_l_se/capitulo_4.html

[21] Lamarca M.J., Metadatos Dublin Core, September 2005, 02 de Mayo de 2009. Disponible en: http://www.hipertexto.info/documentos/dublin_core.htm

[22] PHP, XML Parser, 15 de julio 2009. Disponible en:
http://mx.php.net/xml

[23] hospedajeydominios.com, CXI_ Funciones de intérprete XML, 2007, 20 de Julio de 2009.
Disponible en:
http://www.hospedajeydominios.com/mambo/documentacion-manual_php-pagina-ref_xml.html

[24] DYNAMICDRIVER, Mouseover Tabs Menu, Copyright © 1998-2009, 02 Abril de 2009. Disponible en: http://www.dynamicdrive.com/dynamicindex1/mouseovertabs.htm