

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

**Aplicación web para visualizar artículos científicos
de computación por su temática**

Proyecto tecnológico

Reporte final

Eliut Miguel Cruz Teoba

2112002965

undeuxtrois.alu@gmail.com

Asesor: Dr. José Alejandro Reyes Ortiz

Departamento de Sistemas

jaro@correo.azc.uam.mx

Fecha de entrega

21 de julio de 2017

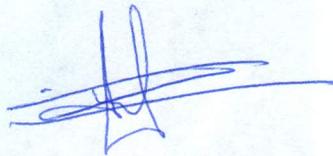
Declaratoria

Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



José Alejandro Reyes Ortiz

Yo, Eliut Miguel Cruz Teoba, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM-Azcapotzalco.



Eliut Miguel Cruz Teoba

Resumen

En este proyecto se ha realizado una aplicación Web que clasifica artículos científicos relacionados al área de computación. La finalidad de este proyecto es ofrecer un mecanismo de búsqueda que permita automatizar y mejorar el proceso de localización de un conjunto de artículos científicos.

Los artículos científicos con los que trabaja este proyecto son extraídos dentro de 5 ontologías que contienen la información de cada artículo en de forma estructurada.

El proyecto está conformado por 3 módulos con funciones específicas. El primer módulo es el encargado de leer y extraer toda la información importante de las ontologías, el segundo módulo se encarga de clasificar esta información y el tercer módulo organiza y muestra la información que ya ha sido clasificada.

Finalmente, una vez que los artículos han sido leídos, clasificados y organizados, se muestra un listado de los artículos relacionados a la temática que un usuario ha solicitado. Cada artículo es visualizado con la siguiente información: autores que lo han realizado; título del artículo, y por último, la información relevante a la revista que lo ha publicado.

Tabla de contenidos

1) Introducción.....	1
2) Antecedentes.....	2
3) Justificación.....	4
4) Objetivos.....	4
5) Marco Teórico.....	5
6) Desarrollo del Proyecto.....	8
6.1) Extracción de elementos de la ontología.....	8
6.1.1) Extracción de la información.....	8
6.1.2) Desarrollo de la extracción de información.....	9
6.1.3) Implementación del la extracción.....	11
6.2) Algoritmo clasificador.....	12
6.2.1) Descripción del algoritmo clasificador.....	12
6.2.2) Funcionamiento del algoritmo clasificador.....	13
6.2.3) Implementación del algoritmo clasificador	15
6.3) Consultar artículos por tema	17
6.3.1) Descripción del modulo consultar artículos.....	17
6.3.2) Funcionamiento del modulo consultar artículos.....	18
6.3.3) Implementación del organizador	19
7) Resultados.....	21
7.1 Pruebas y resultados del modulo de extracción de información.....	22
7.2 Resultados de la clasificación	24
7.3 Pruebas y resultados del modulo de consulta de artículos.....	26
8) Discusión de resultados.....	31
9) Conclusiones.....	33

Bibliografía.....	34
Apéndices.....	34
Anexo A Ontología DBLP.xml de ejemplo	34
Anexo B Clase xml.java	35
Anexo C articuloCientifico.java	36
Anexo D cosenoComparador.java	37
Anexo E Servlet ConsultarArticulos.java	39
Anexo F ArticuloHTML.java.....	43
Anexo G JSP vista principal del proyecto	44
Anexo H Javascript funcionesPI.js	49

1. Introducción

Inicialmente, un artículo científico era el resultado de una investigación publicada en alguna revista especializada. Hoy en día se pueden encontrar artículos científicos publicados en internet, lo cual se ha vuelto algo indispensable para su difusión. Este medio ha ofrecido que se pueda tener acceso a una mayor cantidad de contenido científico del que puede ofrecer una revista en formato impreso. Sin embargo, la clasificación que ofrecen los sitios dedicados al almacenamiento o búsqueda de contenido científico no siempre es eficiente y en ocasiones tiende a ser de forma muy generalizada.

No obstante, es posible modelar y clasificar el contenido de un artículo científico, gracias al uso de herramientas computacionales como las ontologías las cuales permiten una descripción y representación formal de los conceptos en un dominio o aplicación en específico.

Es necesario incorporar estas tecnologías en sistema de información como búsquedas, organizadas por temas para automatizar el proceso de localización de artículos científicos. Por ello, en este proyecto se planea implementar una aplicación web haciendo uso de una ontología académica, para modelar la temática del contenido de un artículo científico y, después, ofrecer estos resultados de forma adecuada a un usuario interesado en un conjunto de artículos pertenecientes a una temática específica.

2. Antecedentes

Proyectos de integración o terminales:

- Sistema de procesamiento de textos de investigación científicos [1].

En este proyecto se procesan artículos de investigación modelados con el formato de la IEEE. Lo que hace parecido este trabajo a esta propuesta de integración, es que en esta propuesta de integración se procesa información científica, para después extraer la información de interés para el usuario. La diferencia es que este trabajo extrae la información de archivos de texto y en esta propuesta se hace mediante el uso de ontologías.

- Arquitectura orientada a servicios web para establecer grupos de colaboración entre investigadores [2].

Este proyecto hace uso de la información extraída de las ontologías para modelar los grupos de colaboración de los investigadores y crear las correlaciones que existen entre cada uno de ellos. Este trabajo es parecido a esta propuesta ya que hace extrae información de las ontologías para crear un modelo de relación entre el contenido de la ontología. Lo que diferencia este trabajo de esta propuesta, es el contenido temático que se extrae de las ontologías y el hecho de que en esta propuesta de integración se propone una arquitectura basa en una aplicación web, mientras que en el trabajo descrito en este punto, se propone una arquitectura basada en servicios web.

- Sistema web para el poblado automático de ontologías a partir de textos [3].

En este proyecto se diseñó un sistema web para obtener información que permita ingresar el poblado automático de ontologías a partir de documentos de textos. Este trabajo se parece a esta propuesta ya que el uso principal de esta aplicación está basado en la utilización e implementación ontologías. Esta propuesta es diferente a la anterior ya que en este proyecto se extrae información de las ontologías y en el otro proyecto se añade información a las ontologías.

Tesis:

- Modelo basado en técnicas de procesamiento de lenguaje natural para extraer y anotar información de publicaciones científicas [4].

En esta tesis doctoral se define un modelo genérico que permite la modelización de una publicación científica, para que de esta forma esta información pueda ser identificada, localizada, clasificada y acceda por los recursos informáticos disponibles.

Lo que hace parecido este trabajo a esta propuesta es que en ambos se clasifican y modelan contenido de carácter científico en un área especializada. La diferencia es que este trabajo permite la modificación de este contenido...

Artículos:

- Categorización de resúmenes de publicaciones científicas basada en similitud semántica [5].

En este artículo se propone un modelo de medición de similitud de textos científicos basados en similitud de conocimiento semántico, el cual obtiene el grado de relación que existe entre los textos. Este trabajo es similar a esta propuesta ya que se trabaja con un modelado de textos científicos. Se diferencia de esta propuesta, ya que este proyecto se hace una similitud de textos para su categorización, y en esta propuesta el modelo de textos científicos es proporcionado por la ontología.

- Descubrimiento y representación de conocimiento sobre datos científicos y académicos [6].

En este artículo se presenta una metodología que permite descubrir relaciones semánticas entre los datos académicos y publicaciones científicas de un investigador. Esta información es representada mediante un modelo ontológico. Este trabajo es similar a esta propuesta porque modela y representa los datos científicos mediante el uso de ontologías. Lo que diferencia este trabajo de esta propuesta es que este trabajo usa las ontologías para describir una relación entre las publicaciones científicas y los datos académicos. En esta propuesta se usa la ontología para modelar el artículo científico del área de computación.

3. Justificación

Hoy en día, la búsqueda de artículos científicos por contenido temático tiende a ser una tarea en donde se implica una larga cantidad de tiempo. Esto es debido a que este tipo de publicaciones es almacenado en repositorios que se encuentran distribuidos por toda la web y en ocasiones sólo es visible un breve resumen sobre la temática que abordan. Por consiguiente, la clasificación del contenido sobre el que trata un artículo, no es algo que incluyan ciertas herramientas dedicadas a la búsqueda de material científico o académico, tal como lo es Google académico.

Por estas razones, en este proyecto se planea diseñar e implementar una aplicación web que ofrezca la posibilidad mostrar el contenido de una búsqueda por tema limitada al área de computación. Este proyecto se intentará reducir los tiempos de búsqueda de los artículos científicos que se encuentran modelados en una ontología académica. El contenido de dicha ontología será clasificado haciendo uso de un algoritmo de programación, así de esta forma el usuario podrá tener acceso a contenido científico de su interés en una menor cantidad de tiempo de que se gasta habitualmente.

4. Objetivos

Objetivo general:

- Diseñar e implementar una aplicación web que permita clasificar y visualizar publicaciones científicas basada en el contenido temático en el área de la computación.

Objetivos específicos:

- Diseñar e implementar un módulo que permita consultar elementos de una ontología.
- Aplicar un algoritmo que clasifique el contenido de un artículo mediante el uso de la taxonomía ACM.

- Diseñar e implementar un módulo que organice por temas el contenido de un artículo científico.
- Implementar una aplicación web que permita integrar los módulos del sistema para clasificar y visualizar publicaciones científicas.

5. Marco Teórico

Ontología

Una ontología es un conjunto estructurado de términos y conceptos que representa el sentido de un campo de información, esto se hace ya sea por metadatos de un espacio de nombres o por los elementos de un dominio de conocimientos. La ontología constituye en sí, un modelo de datos que representa un conjunto de conceptos en un dominio, así como las relaciones entre estos conceptos. Es utilizada principalmente para razonar a cerca de los objetos de un dominio concreto.

XML

XML (eXtensible Markup Language) es un lenguaje informático de etiquetado genérico que permite la descripción de datos gracias a la ayuda de etiquetas y reglas personalizables. XML es descendiente del lenguaje SGML (creado en los años 70) para facilitar la comunicación entre las aplicaciones. A partir de 1998 XML se volvió una recomendación del W3C (World Wide Web Consortium).

Taxonomía ACM

La taxonomía curricular de la *Association for Computing Machinery* (ACM) es un sistema de clasificación para el campo de la informática, publicado por primera vez en 1964. Después en 1982, la ACM publicó un sistema completamente nuevo. Las nuevas versiones están basadas en la versión de 1982. Actualmente el esquema de 2012 utiliza una nueva estructura jerárquica y un enfoque más profundo de la versión de 1998. En la tabla siguiente se muestran los conceptos que abarcan cada temática de la taxonomía.

Tema	Descripción de los conceptos
Hardware (HR)	En la rama de hardware se encuentran conceptos que tienen relación con circuitos informáticos, como lo son: circuitos integrados, microprocesadores, estructuras de memoria y estructuras lógicas. Actualmente se han agregado dos nuevos conceptos: el hardware de red y las máquinas pequeñas. Dentro de las máquinas pequeñas se

	encuentran términos relacionados con la nanotecnología y la informática cuántica, mientras que en hardware de red se organizan términos como adaptadores de red, puentes de red y routers.
Sistemas computacionales (CO)	En esta rama se encuentran las arquitecturas de procesador y se reorganizan algunos conceptos dentro ella. Dichos conceptos son: los sistemas basados en aplicaciones, los sistemas embebidos, los sistemas multinúcleo y los sistemas en tiempo real.
Redes (NW)	Redes es una mezcla de términos nuevos y conceptos existentes que han sido tomados de las redes de comunicación informática. Estos conceptos incluyen términos como: protocolos de red, Internet, comunicaciones entre datos, arquitecturas distribuidas y topologías de red. Actualmente se han introducido conceptos de hardware de red; rendimiento y confiabilidad de red y escala de red y redes inalámbricas.
Software (SW)	Esta rama abarca conceptos de lenguajes de programación y compiladores. En cuanto a los lenguajes de programación se agrupan todos los tipos de lenguajes y se mantiene una relación con las técnicas de programación. En cuanto a los compiladores, estos se utilizan en un sentido amplio e incluyen conceptos tales como: intérpretes y entornos en tiempo de ejecución, los cuales no se consideran estrictamente compiladores.
Teoría de la computación (TC)	En esta rama se encuentran términos relacionados al análisis y diseño de algoritmos. El análisis de algoritmos contiene métodos de análisis asintótico, numéricos y no numéricos. Actualmente se han agregado términos de complejidad computacional, lenguajes formales y lógica matemática.
Matemáticas computacionales (MC)	En esta rama se encuentran los conceptos de análisis numérico, análisis de algoritmos (en la teoría de la rama de cómputo), algoritmos y problemas numéricos. Se consideró que la teoría de la información encaja mejor en esta rama y se racionalizaron los aspectos de la codificación y la teoría de la información.
Sistemas de información (IS)	En esta rama se encuentra conceptos relacionados con el almacenamiento de datos, los sistemas de soporte de decisiones, los sistemas de información de empresas, los sistemas de información geográfica, dispositivos multimedia y las bases de datos. Actualmente, en esta rama también podemos hallar términos relacionados con la web como lo son: aplicaciones Web, idiomas, búsqueda y servicios Web.
Seguridad y privacidad (SP)	Esta nueva rama incorpora conceptos como: criptografía, políticas de privacidad y de seguridad, seguridad de hardware, sistemas operativos, sistemas distribuidos y seguridad de red, y seguridad de bases de datos.
Interacción humano computadora (HC)	La informática centrada en el ser humano consta de nuevos conceptos y términos existentes relacionados con las interfaces, la usabilidad, la informática personal, la visualización de información y la informática personal.

Tabla 1.- Conceptos abarcados por cada temática de la taxonomía ACM

Servlet

Un Servlet es una clase de Java que permite crear dinámicamente datos dentro de un servidor HTTP. Estos datos son generalmente presentados en formato HTML, pero también pueden ser presentados en formato XML o cualquier otro formato

destinado a los navegadores Web. Un Servlet se ejecuta dentro de un servidor web y permite la extensión de funcionalidades de un servidor, como por ejemplo el acceso a la base datos.

JSP

El JSP (Java Server Page) es una técnica basada en Java que permite a los desarrolladores Web crear dinámicamente código HTML, XML o cualquier otro tipo de pagina web. Esta técnica permite que el código Java y ciertas acciones predefinidas, puedan ser agregados dentro de un contenido estático. Desde la versión 2.0, la sintaxis del JSP está dada conforme al estándar XML.

AJAX

Ajax (Asynchronous JavaScript and XML) es una arquitectura informática que permite construir aplicaciones Web y sitios interactivos del lado del cliente. Ajax hace uso de diferentes tecnologías como lo es: CSS, JSON, XML, el DOM, etc. Gracias Ajax se pueden realizar actualizaciones dentro del contenido de una página Web, sin necesidad de volver a recargar la pagina Web. La transferencia de datos es administrada exclusivamente por Javascript y se es actualizado únicamente el bloque que se quiere ver afectado en nuestro sitio Web.

DOM

El DOM (document Object Model) es una interfaz de programación normalizada por el W3C, que permite a los scripts examinar y modificar el contenido de un navegador Web. En el DOM la composición de un documento HTML o XML es representado bajo la forma de objetos unidos en una estructura de árbol. Gracias al DOM un script puede modificar un documento Web en el navegador agregando o suprimiendo nodos dentro del árbol.

W3C

El W3C (World Wide Web Consortium), es un organismo de estandarización sin ánimo de lucro, fundado en octubre de 1994 encargado de promover la compatibilidad de las tecnologías del World Wide Web.

6. Desarrollo del proyecto

Este proyecto consta de 3 módulos principales (lectura y extracción de información, clasificación de la información y consulta de la información). Cada módulo será descrito en los siguientes puntos.

6.1 Extracción de elementos de la ontología

Esta parte del proyecto es la encargada de leer y extraer toda la información necesaria de las ontologías. La lectura se lleva a cabo parseando el documento XML para obtener el contenido completo de la ontología de forma jerárquica. Una vez que se haya realizado la lectura de la ontología, la información que será ocupada será almacenada en memoria para que posteriormente sea clasificada.

6.1.1 Extracción de la información

En este proyecto se utilizan cinco ontologías en formato XML, las cuales han sido proporcionadas por el asesor de este proyecto. Estas ontologías contienen los artículos científicos que serán clasificados en esta aplicación. Los artículos contenidos en la ontología cuentan con la siguiente información: Autor, título, páginas, año, volumen, revista, número de la revista y URL. Esta información está organizada de forma arborescente, como se puede ver en la siguiente imagen:

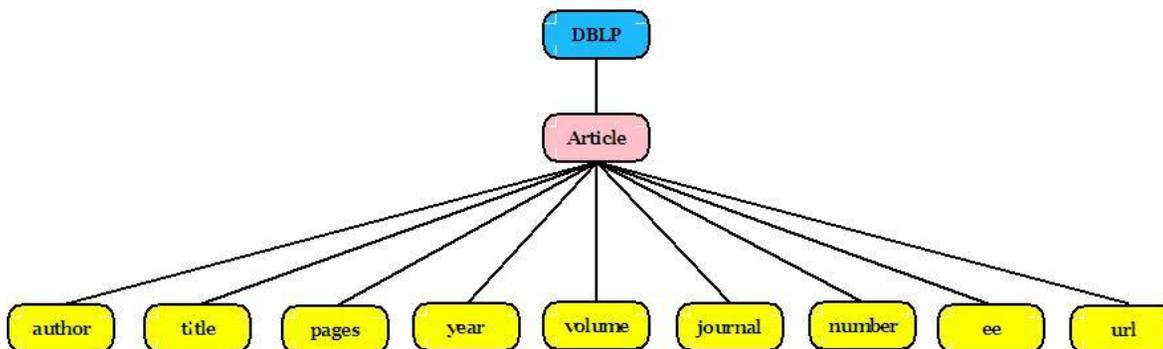


Figura 1.- Estructura jerárquica de la ontología DBLP
Se puede ver un ejemplo en formato XML de las ontologías en el Anexo A

Para la extracción de información de la ontología se hace uso de JDom (librería de Java). Con esta librería se parsea el documento XML y se extrae la información necesaria para almacenarla en memoria.

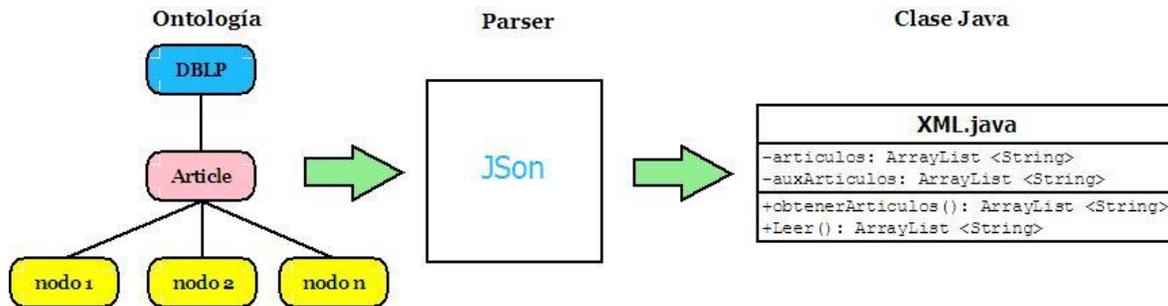


Figura 2.- Parseo de la ontología en formato XML

6.1.2 Desarrollo de la extracción

Las cinco ontologías con las que se trabajan derivan de dos ontologías cuyo peso aproximado es de 300 y 500 MB. El peso que tienen estas ontologías hace que el procesamiento de la aplicación se demore mucho, razón por la cual se ha decidido crear cinco ontologías, a partir de las originales, cuyo valor se encuentra entre 100 y 500 KB. Una vez hecho esto, cada una de estas ontologías es cargada de forma individual, según lo vaya solicitando el usuario.

Los datos que se extraen de la ontología son: el autor o los autores, el título, el año, el volumen, la revista donde se publicó el artículo y la URL. Posteriormente esta información es guardada en memoria en dos Arrays de Strings. El primer Array contiene únicamente los títulos de toda la ontología sin ningún signo de puntuación. Este Array se le será entregado a la clase “consultarArticulos” para que continúe con la clasificación.

El segundo Array contiene todos los datos extraídos de la ontología que ya se mencionaron anteriormente. En este Array se guarda todo el contenido de un Artículo en un String. Para hacer la diferenciación entre el diferente tipo de información contenida en dicha cadena de texto, se etiqueta previamente cada

nodo de un artículo mediante el uso de arrobas. La notación para este etiquetado se ve en la tabla siguiente:

Tipo de etiqueta	Contenido	Tipo de etiqueta	Contenido
@A@Contenido@A@	Autor/autores	@Y@Contenido@Y@	Año
@T@Contenido@T@	Título	@V@Contenido@V@	Volumen
@J@Contenido@J@	Revista	@U@Contenido@U@	URL

Tabla 2.- Etiquetado de los elementos de los artículos científicos

A continuación se muestran 3 artículos etiquetados con esta notación:

1) @A@Christian Noon, Ruqin Zhang, Eliot Winer, James Oliver, Brian Gilmore, Jerry Duncan@A@@T@A system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality.@T@@Y@2012@Y@@V@63@V@@J@Computers in Industry@J@@U@http://dx.doi.org/10.1016/j.compind.2012.02.003@U@

2) @A@Hans-Jörg Bullinger, Wilhelm Bauer, Günter Wenzel, Roland Blach@A@@T@Towards user centred design (UCD) in architecture based on immersive virtual environments.@T@@Y@2010@Y@@V@61@V@@J@Computers in Industry@J@@U@http://dx.doi.org/10.1016/j.compind.2009.12.003@U@

3) @A@C. N. Verdouw, Adrie J. M. Beulens, Jacques H. Trienekens, Tim Verwaart@A@@T@Towards dynamic reference information models: Readiness for ICT mass customisation.@T@@Y@2010@Y@@V@61@V@@J@Computers in Industry@J@@U@http://dx.doi.org/10.1016/j.compind.2010.07.008@U@

6.1.3 Implementación del la extracción

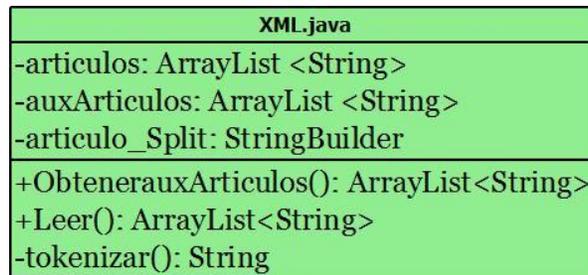


Figura 3.- Diagrama UML de la clase XML

En la figura 3 se muestra el diagrama UML de la clase XML, los elementos de esta clase serán descritos a continuación:

Clase XML

Es una clase que permite leer y extraer información dentro de las ontologías ocupadas en este proyecto y su implementación se puede consultar en el anexo B.

Atributos:

Artículos.- Arraylist que guarda los títulos de los artículos científicos sin signos de puntuación y sin letras en mayúsculas.

auxArtículos.- Arraylist que guarda todos datos de un articulo. Este Array almacena los artículos que *articuloSplit* recibe.

articuloSplit.- Almacena los autores, el titulo, el año, el volumen, la revista y la URL, de un artículo contenido en una ontología. Estos elementos están etiquetados con el formato de arrobas (tabla2).

Métodos:

Leer ().- Este método lee los elementos de la ontología, almacena los títulos de las ontologías en el Array *Articulos*, etiqueta y almacena el resto de los elementos de cada artículo en el String *articuloSplit* y por último, retorna el Array *Artículos*.

Obtener auxArticulos()

Retorna el Array *auxArticulos* que contiene todos los elementos un artículo.

Tokenizar()

Este método es el encargado de quitar todos los signos de puntuación a cada título leído de la ontología.

En la siguiente imagen se muestra un ejemplo de lo que almacena *auxArticulos*.

```
Array auxArticulos
@A@Jck-Hyun Kwon, Chang Ouk Kim, Kyung Pil Kim, Choonjong Kwak@A@@T@Recommendation of e-commerce sites by matching category-based buyer query and
product e-catalogs.@T@@Y@2008@Y@@V@59@V@@J@Computers in Industry@J@@U@http://dx.doi.org/10.1016/j.compind.2007.10.002@U@
@@A@Junhao Geng, Xitian Tian, Mingxing Bai, Xiaoliang Jia, Xiangwei Liu@A@@T@A design method for three-dimensional maintenance, repair and overhaul job card of
complex products.@T@@Y@2014@Y@@V@65@V@@J@Computers in Industry@J@@U@http://dx.doi.org/10.1016/j.compind.2013.08.008@U@
@@A@Marco Cantamessa, Francesca Montagna, Paolo Neirotti@A@@T@An empirical analysis of the PLM implementation effects in the aerospace
industry.@T@@Y@2012@Y@@V@63@V@@J@Computers in Industry@J@@U@http://dx.doi.org/10.1016/j.compind.2012.01.004@U@
@@A@Mohsen Shakeri@A@@T@Implementation of an automated operation planning and optimum operation sequencing and tool selection
algorithms.@T@@Y@2004@Y@@V@54@V@@J@Computers in Industry@J@@U@http://dx.doi.org/10.1016/j.compind.2003.12.002@U@
```

Figura 4.- Datos almacenada en el Array *auxArticulos*

6.2 Algoritmo clasificador

En esta parte del proyecto, se clasifican los artículos de cada ontología con base en la taxonomía curricular ACM. Esta clasificación se hace analizando el título de cada artículo científico. Una vez que el título de un artículo ha sido analizado, se determina la temática a la que éste pertenece.

6.2.1 Descripción del algoritmo clasificador

Para lograr la clasificación, es necesario tener un título que representa, lo mejor posible, cada temática de la taxonomía. Estos títulos han sido escogidos, analizando los artículos de las primeras 5 ontologías.

Para la clasificación de los artículos se implementa un algoritmo de similitud de textos. Este algoritmo compara dos cadenas de textos y determina, en términos de porcentaje, la similitud que existe entre ambos contenidos.

El proyecto utiliza dicho algoritmo para determinar la similitud que existe entre cada uno de los títulos contenidos en las ontologías y los títulos clave que ya se han seleccionado. Enseguida, se compara cada uno de los títulos de cada ontología con cada clave y se determina qué relación tiene el porcentaje de similitud más elevado. Con base en esta relación, se determina la pertenencia de un título a un determinado tema de la taxonomía.

La comparación del título: database support for evolving data in product design	con la temática: HR es de 0.0
La comparación del título: database support for evolving data in product design	con la temática: CO es de 0.0
La comparación del título: database support for evolving data in product design	con la temática: NW es de 0.0944911182523068
La comparación del título: database support for evolving data in product design	con la temática: SW es de 0.125
La comparación del título: database support for evolving data in product design	con la temática: TC es de 0.0
La comparación del título: database support for evolving data in product design	con la temática: MC es de 0.0
La comparación del título: database support for evolving data in product design	con la temática: IS es de 0.5
La comparación del título: database support for evolving data in product design	con la temática: SP es de 0.11180339887498948
La comparación del título: database support for evolving data in product design	con la temática: HC es de 0.11180339887498948

Figura 4.- Porcentaje de similitud que tiene el título: "database support for evolving data in product design" con cada una de las temáticas de la taxonomía.

En el ejemplo de la figura 4 se puede observar que el porcentaje de similitud más elevado ha sido el que se ha comparado con la temática de sistemas de información. Por lo tanto, el tema que se le será asignado al título "Data base support for evolving data in product design" será sistemas de información.

6.2.2 Funcionamiento del algoritmo clasificador

Para calcular la semejanza de los títulos, se utiliza el algoritmo de similitud de cosenos. Este algoritmo convierte cada texto en un vector y calcula la diferencia entre sus ángulos. Si las frases evaluadas son iguales, su ángulo valdrá 0 y el coseno 1, pero si son completamente diferentes, es de decir son dos vectores ortogonales el cálculo del coseno será 0.

Ahora, se utilizará este algoritmo para calcular la semejanza entre todos los títulos de la ontología y todas las semillas de cada temática. Los títulos propuestos para ser representantes de cada temática se describen en la tabla siguiente:

Tema	Semilla
Hardware	Efficient execution of speculative threads and transactions with hardware transactional memory.
Sistemas computacionales	Algebraic models of microprocessors architecture and organization.
Redes	An energy saving routing with a uniform clustering algorithm for wireless body sensor networks
Software	Automatic correction of syntax errors in programming languages.
Teoría de la computación	Project scheduling with irregular costs complexity approximability and algorithms.
Matemáticas computacionales	Project scheduling with irregular costs complexity approximability and algorithms.
Sistemas de información	Data management support for database management.
Seguridad y privacidad	On the security of auditing mechanisms for secure cloud storage.
Interacción humano computadora	Interactive visualization of complex dynamic virtual environments for industrial assemblies.

Tabla 4.- Semillas elegidas para la clasificación de artículos

Finalmente, gracias a la ayuda de dos variables auxiliares se determina qué porcentaje de similitud ha sido el más elevado. La primera variable comienza valiendo cero y se compara con la similitud de textos entre un tema y una semilla. Si se detecta que el valor de esta comparación es mayor al valor de la variable auxiliar, se le asigna el valor de la comparación a la variable auxiliar, si esto sucede, también se guarda la abreviatura del tema que dio esa similitud en la segunda variable auxiliar. Al final, gracias a estas dos variables se sabe la temática que dio el resultado más alto y se le asigna una tema a cada título de la ontología.

En la siguiente imagen se muestra las etapas por las que pasa el algoritmo de clasificación.

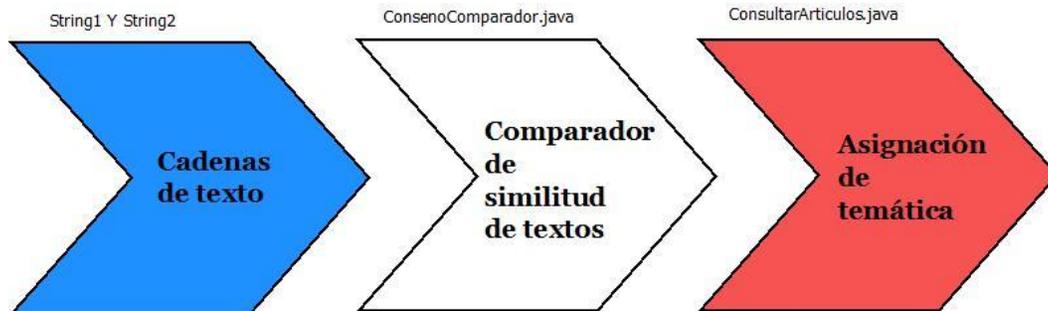


Figura 5.- Funcionamiento del algoritmo clasificador.

6.2.3 Implementación del algoritmo clasificador

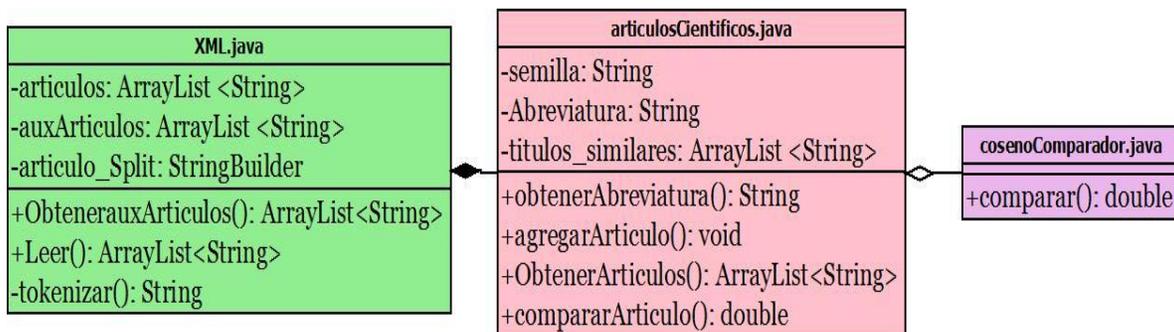


Figura 6.- Diagrama UML de la clases que conforman el clasificador XML

A continuación se describe las clases conforman el clasificador, la clase XML.java no será explicada debido a que ya se hizo en apartado 6.1.3

Clase articuloCientífico

Esta clase se encarga de administrar todos los artículos científicos relacionados a un área de la taxonomía ACM. Se crean 9 objetos en este proyecto para las 9 temáticas que se trabajan. La implementación de esta clase se puede ver en el anexo C.

Atributos:

Semilla.- String que guarda el título escogido para ser representante de la temática de la taxonomía.

Abreviatura.- String que guarda el nombre de cada temática en forma abreviada, ejemplo: Software es SW.

Articulos_similares.- Array que almacena los artículos, que han sido comparados con la semilla de esta clase y han tenido el resultado más alto.

Métodos:

ObtenerAbreviatura()

Devuelve el String *Abreviatura* que contiene la abreviatura de la temática con la que ha sido definida la clase.

ObtenerArticulos()

Retorna el Array *Articulos* que contiene todos los artículos que ha administrado esta clase. Estos elementos ya están clasificados y pertenecen a la temática de la clase.

agregarArticulo()

Sirve para agregar un artículo científico que ya ha sido clasificado. Este artículo se guarda en el Array *Articulos*.

compararArticulo()

Compara un título de un artículo científico con el título semilla con el que se ha definido la clase.

Clase comparadorCoseno

Es una clase estática que calcula la similitud de textos. Utiliza el algoritmo de similitud de cosenos. Su implementación se encuentra en el anexo D.

Métodos:

comparar()

Este método compara la similitud que existe entre dos textos y devuelve la semejanza existente en un número tipo double.

En la figura siguiente se muestran los títulos de los artículos científicos de la primera ontología que tienen relación con software:

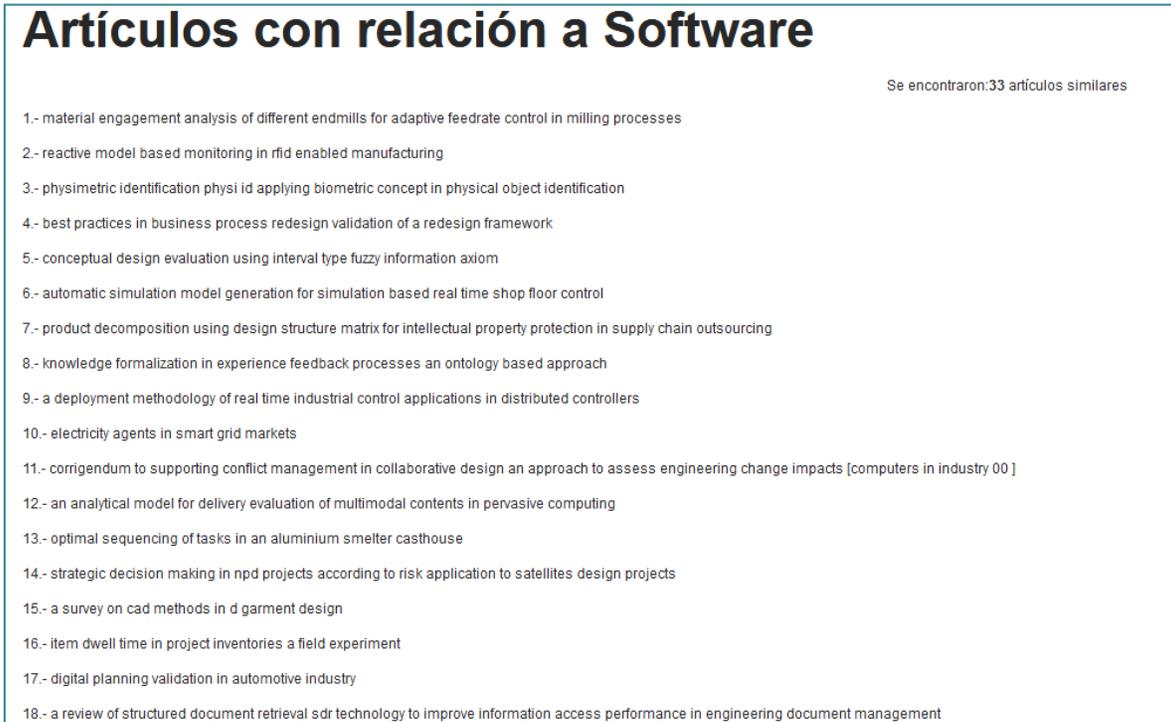


Figura 7.- Artículos clasificados de la primera ontología

6.3 Consultar artículos por tema

Este módulo se encarga de organizar los artículos clasificados de la ontología, y mostrar únicamente los artículos relacionados a la temática que el usuario ha seleccionado. En esta parte del proyecto ya muestra el resto de la información de los artículos y se les asigna un formato entendible a los títulos ya clasificados.

6.3.1 Descripción del módulo consultar artículos

Este módulo filtra los artículos científicos que el usuario desea ver. Primero obtiene todos los artículos que existen dentro de una ontología y luego los manda a clasificar. Una vez que todos los elementos han sido clasificados, este módulo se encarga de darle un formato DBLP a los artículos científicos que se desean ver. Este módulo es integrado directamente a la vista principal del proyecto, en donde

el usuario puede interactuar con la interfaz gráfica del programa y ver los artículos que ya se han clasificado.

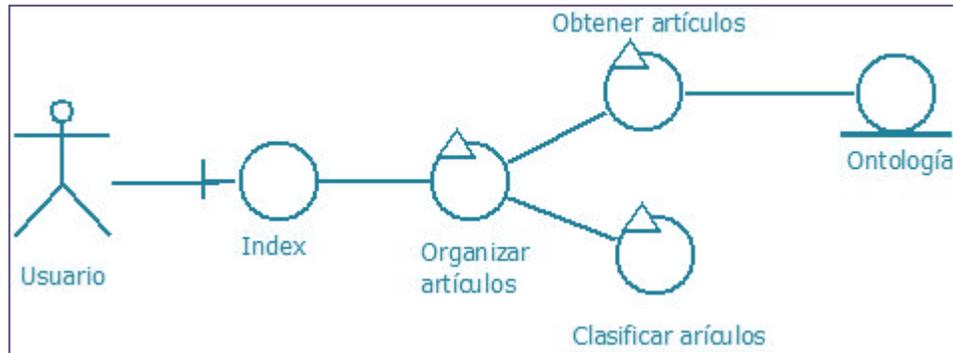


Figura 8.- Diagrama de robustez de los principales elementos que interactúan con la clase organizar artículos

6.3.2 Funcionamiento del modulo consultar artículos

La clase encargada de consultar los artículos es un Servlet que recibe una petición GET a esta dirección:

../PI/ConsultarArticulos?TA=Abreviatura_Ontol&no_Ontologia=Numero_ontologia

En esta petición se envía la información sobre la temática en la que se desea que sean visualizados los artículos y la ontología de donde serán extraídos estos artículos. En este proyecto esta petición se hace en segundo plano, mediante AJAX. Y se actualiza solo porción dentro la página principal para que sean visualizados estos artículos.

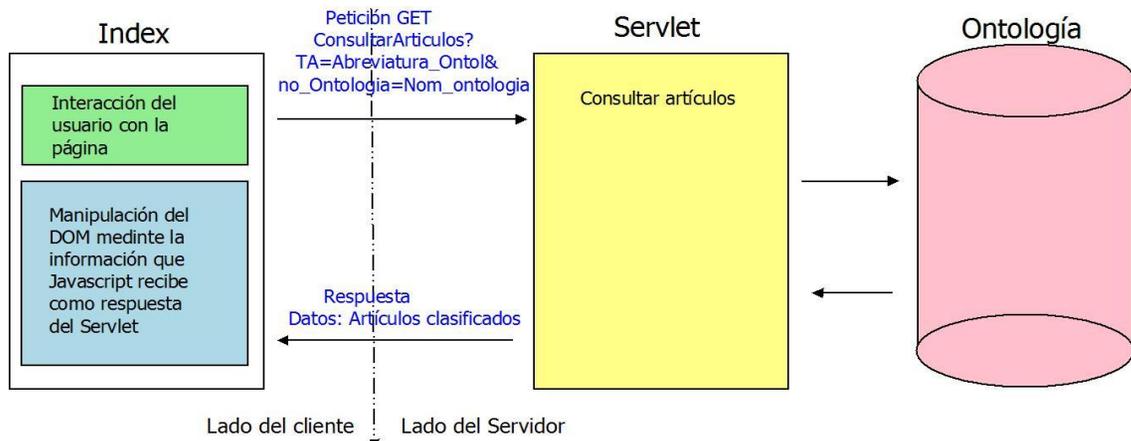


Figura 9.- Petición Ajax desde la pagina principal al Servlet

Como ya se menciono, este modulo también es el encargado de darle formato a los artículos científicos. Es decir, ahora corresponde quitar las etiquetas al contenido que se ha almacenado en el Array *articuloSimililares* de la clase *articulosCientificios*. Y darle un formato que respete la normativa de la DBLP en HTML. El formato de las etiquetas se puede ver en la tabla 2 del apartado 6.2.2.

Para llevar a cabo esto, hacemos uso de expresiones regulares, que detecten los elementos contenidos dentro de cada etiqueta, y lo reemplacen por código en HTML. El formato que esperamos es el siguiente:

Autor o autores, titulo (al hacer clic en éste se redirecciona a la pagina donde está dicho artículo), revista donde se ha publicado el artículo, el volumen de la revista y el año en el que fue publicado.

6.3.3 Implementación del organizador

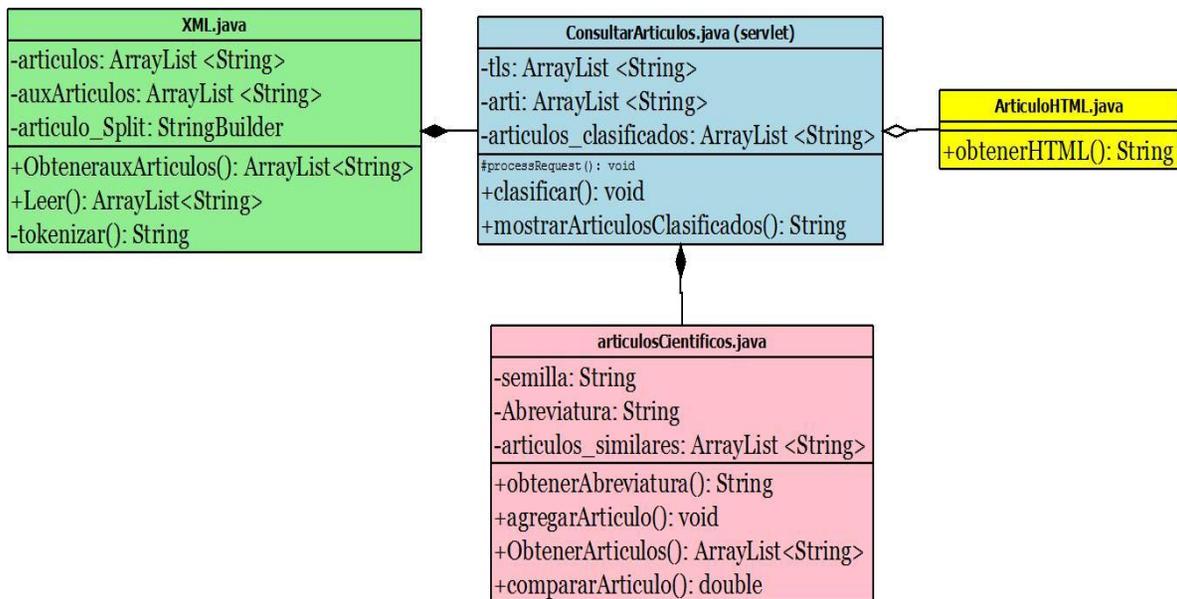


Figura 10.- Diagrama UML las clases que interactúan con el módulo para consultar artículos

A continuación se describe las clases que conforman el módulo de consulta de los artículos científicos, la clases XML.java y articulosCientificos.java no serán explicadas debido a que ya se hizo en apartado 6.1.3 y 6.2.3.

Clase ConsutarArituclo

Es una de las clases principales que se encarga de comunicarse con la mayoría de las otras clases de este proyecto, su función principal es la de filtrar los artículos relacionados a la temática que un usuario desee ver. El código de esta clase se puede ver en el anexo E.

Atributos:

TLS.- Array que almacena los títulos sin signos de puntuación, este Array es el que devuelve el método leer() de la claseXML.

Arti.- Array que alacena el resto del contenido de cada artículo, este Array es el que devuelve el método ObtenerauxArticulos() de la claseXML.

articulos_clasificados.- Array que almacena parcialmente todos los artículos clasificados de una ontología.

Métodos:

Clasificar()

Método que compara la similitud de cada texto con cada ontología. A la similitud más elevada se le será asignada la temática de la semilla con la que se comparo.

mostrarArticulos()

Método que muestra sólo los artículos clasificados que el usuario quiere ver. Recibe como parámetro la temática de la ontología que se recibió a través del método GET.

Clase AritucloHTML

Es una clase estática cuya función es extraer y dar formato a la información del contenido etiquetado de los artículos científicos. El código de esta clase se encuentra en el anexo F.

Métodos:

obtenerHTML ()

Este método recibe como parámetro el contenido completo de cada artículo con su respectivo formato de etiquetas. Este contenido se analiza y posteriormente se

extrae la información contenida en estas etiquetas, para después reemplazarla por código html. Al final, este método retorna el código que será impreso por la clase consultarArticulos. El formato en el que retorna los artículos se puede ver en la siguiente figura:



Figura 11.- Artículos científicos ya han sido formateados

El código fuente de la vista final del proyecto y la función Ajax que se implementan para mostrar los datos se pueden encontrar en el Anexo G y Anexo H.

7. Resultados

En esta sección se mostrará las pruebas y resultados que se tuvieron en la realización de este proyecto. En este apartado, se mostrara los resultados del modulo de extracción de información, el modulo de clasificación y el modulo de consulta (la vista final del proyecto).

7.1 Pruebas y resultados del modulo de extracción de información

```
PI (run) X Apache Tomcat 8.0.27.0 X Apache Tomcat 8.0.27.0 Log X
DBLP1
Articulos leidos: 384
modeling the permeability of carbonate reservoir using type fuzzy logic systems
business semantics management a case study for competency centric hrm
monitoring and predictive maintenance modeling and analyse of fault latency
computer aided activity planning caap in large scale projects with an application in the yachting industry
erp 0 what for and how
enabling collaborative product design through distributed engineering knowledge management
a novel fuzzy multi criteria decision framework for sustainable supplier selection with incomplete information
using svm based method for equipment fault detection in a thermal power plant
a demanufacturing projector vision system for combined manual and automated processing of used electronics
risk management dependability tools and case based reasoning integration using the object formalism
a system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality
determinants of choice of semantic web based software as a service an integrative framework in the context of e procurement and erp
material engagement analysis of different endmills for adaptive feedrate control in milling processes
sis stamp an integrated inspection system for sheet prints in stamp printing application
privacy preserving data infrastructure for smart home appliances based on the octopus dht
accurate d vision based obstacle detection for an autonomous train
architectures for enterprise integration and interoperability past present and future
ontooqualitas a framework for ontology quality assessment in information interchanges between heterogeneous systems
reactive model based monitoring in rfid enabled manufacturing
special issue on enterprise resource planning erp systems
a survey on industrial applications of fuzzy control
maintaining consistency between cad elements in collaborative design using association management and propagation
envisioning e logistics developments making spare parts in situ and on demand state of the art and guidelines for future developments
similarity based method for manufacturing process performance prediction and diagnosis
database support for evolving data in product design
modeling and specifications of dynamic agents in fractal manufacturing systems
towards user centred design ucd in architecture based on immersive virtual environments
a simulation study on maintainer resource utilization of a fast jet aircraft maintenance line under availability contract
```

Figura 12.- Extracción de todos los artículos de la ontología 1

```
Output X
PI (run) X Apache Tomcat 8.0.27.0 X Apache Tomcat 8.0.27.0 Log X
DBLP12
Articulos leidos: 581
recommendation of e commerce sites by matching category based buyer query and product e catalogs
a design method for three dimensional maintenance repair and overhaul job card of complex products
an empirical analysis of the plm implementation effects in the aerospace industry
implementation of an automated operation planning and optimum operation sequencing and tool selection algorithms
karmarkar s and interaction/prediction algorithms for hierarchical production planning for the highest business benefit
advances in photogrammetric measurement solutions
modelling of the life cycle of products with data acquisition features
editorial
dynamic approach to solve the daily drayage problem with transit time uncertainty
interoperability in collaborative networks independent and industry specific initiatives the case of the footwear industry
utilizing enterprise systems for managing enterprise risks
high fidelity rapid prototyping of 00 mm fabs through discrete event system modeling
an innovative freight traffic assignment model for multimodal networks
an ontological approach for reliable data integration in the industrial domain
a customer oriented methodology for reverse engineering software selection in the computer aided inspection scenario
sustainability trends in the process industries a text mining based analysis
an integrated design flow for the joint generation of control and interfaces from a business model
corrigendum to critical factors for successful erp implementation exploratory findings from four case studies [computers in industry 00 ]
information system modelling for engineering design co ordination
web based support for collaborative product design review
flexible ieee 0 deadline aware scheduling for dpccs using priority based csma ca
step based quality diagnosis of shape data of product models for collaborative e engineering
dixel based force torque rendering and volume updating for dof haptic product prototyping and virtual sculpting
a semi automatic approach for workflow staff assignment
vmes virtualization aware manufacturing execution system
an ontology change management approach for facility management
design for diagnosability of multistation manufacturing systems based on sensor allocation optimization
```

Figura 13.- Extracción de todos los artículos de la ontología 2

```
Output x
PI (run) x Apache Tomcat 8.0.27.0 x Apache Tomcat 8.0.27.0 Log x
DBLP5
Articulos leidos: 602
a formal definition of the phenomenon of collective intelligence and its iq measure
a new remote user authentication scheme for multi server architecture
netsec metrology based application for network security
manifold
methods of sensitivity theory and inverse modeling for estimation of source parameters
user transparent data and task parallel multimedia computing with pyxis dt
mars a framework for minimizing the job execution time in a metacomputing environment
flow logic for dolev yao secrecy in cryptographic processes
parallel computing in quantum chemistry message passing and beyond for a general ab initio program system
parallel processing the power and the portability experiments with reusable toolkits
using hybrid scheduling for the semi autonomous formation of expert teams
a st century web based telemedical information society
parallelization of local blast service on workstation clusters
grid infrastructures practice and perspectives
cooperative stalking of transient nomadic resources on overlay networks
a multi start local search heuristic for an energy efficient vms assignment on top of the opennebula cloud manager
an xml architecture for high performance web based analysis of remote sensing archives
metafol program tactics and logic tactics plus reflection
cloud mapreduce for monte carlo bootstrap applied to metabolic flux analysis
auction based adaptive sensor activation algorithm for target tracking in wireless sensor networks
poisonedwater an improved approach for accurate reputation ranking in p p networks
batch scheduling of consolidated virtual machines based on their workload interference model
message based cellular peer to peer grids foundations for secure federation and autonomic services
a performance comparative study on the implementation methods for ofdma cross layer optimization
vectorization and parallelization of a multi block navier stokes flow solver on different computer architectures
special section systems performance analysis and evaluation
locally constrained synthetic lods generation for natural terrain meshes
large scale computational science on federated international grids the role of switched optical networks
```

Figura 13.- Extracción de todos los artículos de la ontología 5

7.2 Resultados de la clasificación

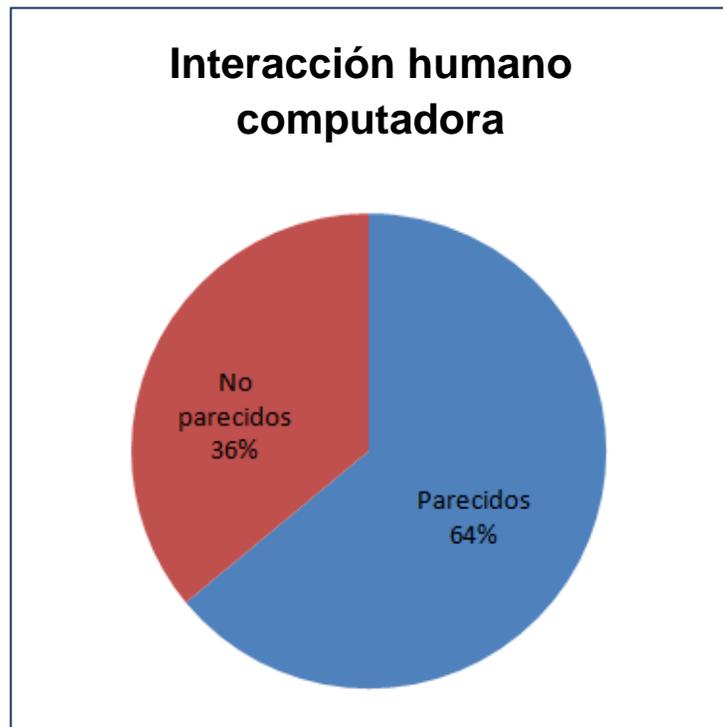


Figura 13.- Resultados de la clasificación del tema “Interacción humano computadora”



Figura 14.- Resultados de la clasificación del tema “Teoría de la computación”



Figura 15.- Resultados de la clasificación del tema “sistemas computacionales”

7.3 Pruebas y resultados del modulo de consulta y organización de artículos

En la siguiente imagen se muestra la vista principal de este proyecto.

Calificador de artículos científicos

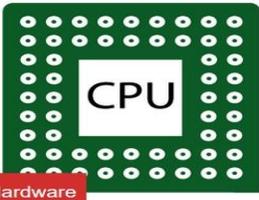
- Redes
- Hardware
- Software
- Sistemas de información
- Sistemas computacionales
- Teoría de la computación
- Matemáticas computacionales
- Seguridad
- Interacción H-C

Redes



Esta es una nueva rama de nivel superior que contiene una mezcla de términos nuevos y conceptos existentes que han sido extraídos de las redes de comu... +

Hardware



En la rama de hardware fueron consolidados recientemente de los conceptos de performance y confiabilidad en un único nodo. La clasificación 1998 tenía... +

Matemáticas computacionales



En esta rama se encuentran conceptos como: análisis numérico, análisis de algoritmos (en la teoría de la rama de cómputo), algoritmos y problemas numé... +

Sistemas computacionales



En esta rama se encuentran las arquitecturas de procesador y se reorganizan algunos conceptos dentro de ella. La clasificación 1998 tenía un concepto l... +

Software



```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Un cambio importante que ha sufrido esta rama es la eliminación de las anteriores técnicas de programación. Se consideró la posibilidad de ubicar este... +

Teoría de la computación



La teoría de la computación se dividió en dos conceptos: Análisis de algoritmos, y diseño de algoritmos. El análisis de algoritmos contiene métodos d... +

Sistemas de información



Los sistemas de información ahora reflejan una mayor variedad de aplicaciones, incluyendo almacenamiento de datos, sistemas de soporte de decisiones, ... +

Seguridad y privacidad



Esta nueva rama incorpora conceptos dispersos de la antigua clasificación así como otros nuevos términos. El cifrado de datos es ahora criptografía y está más... +

Interacción humano - computadora



La informática centrada en el ser humano (HCC) se ha convertido en una disciplina por derecho propio desde la última iteración de la clasificación. De... +

Proyecto de integración

Aplicación web para visualizar artículos científicos de computación por su temática.

Alumno: Eliut Miguel Cruz Teoba
Asesor: Dr. José Alejandro Reyes Ortiz

© 2017 Eliut M. Cruz Teoba UAM-A

Links

Universidad Autónoma Metropolitana
UAM - Azcapotzalco
AISSI - UAM
Taxonomía ACM

[Acerca de este proyecto](#)

Figura 16.- Pagina principal del proyecto (index.jsp)

En la siguiente imagen se muestra la representación de las temáticas de la taxonomía ACM en este proyecto.

- 1) Temática de la taxonomía ACM
- 2) Breve información sobre la tematica
- 3) Menú fijo inamovible que contiene todas Las temáticas de la aplicación

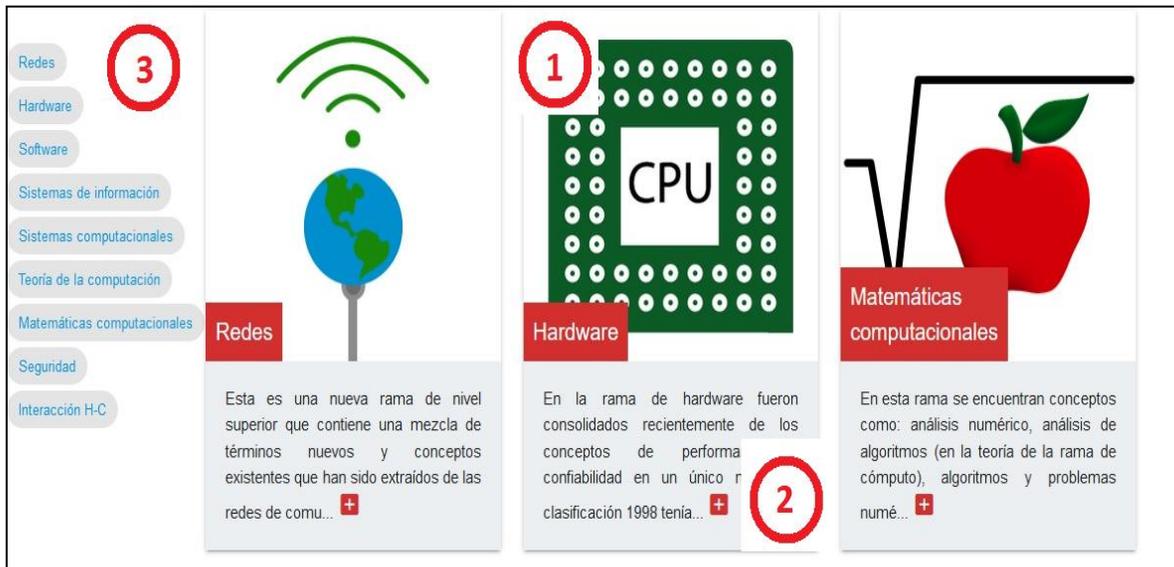


Figura 17.- Temáticas de la taxonomía ACM (index.jsp)

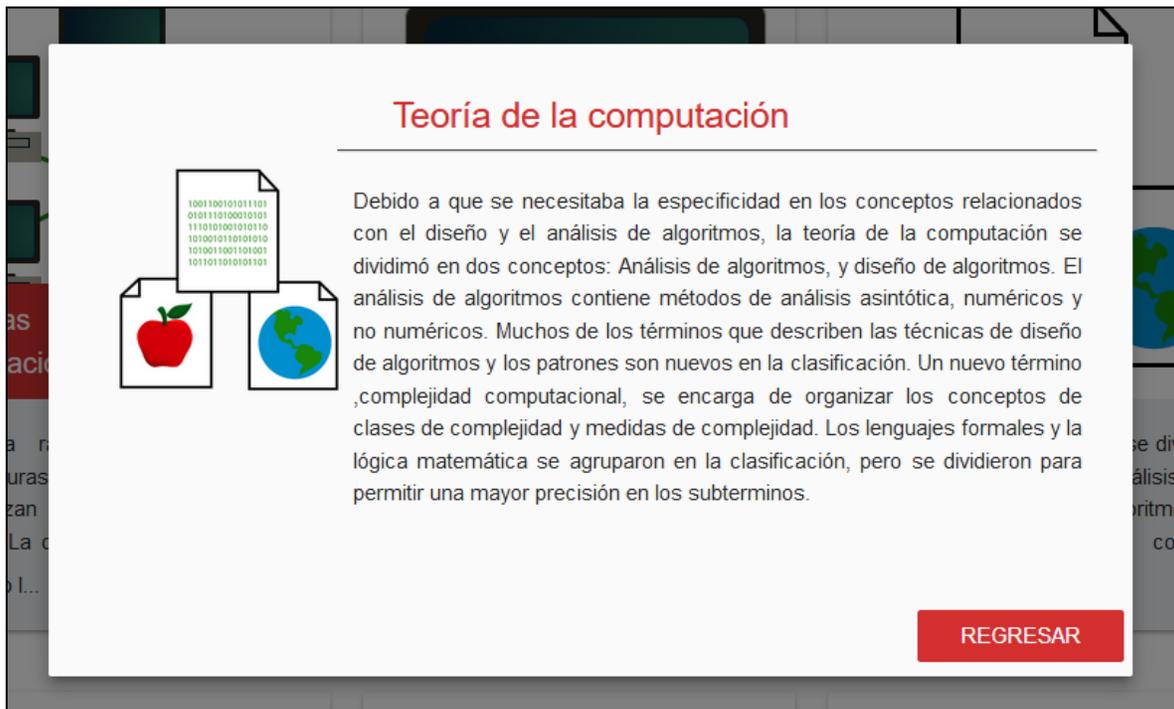


Figura 18.- Despliegue de la información de la temática “Teoría de la computación”.

Artículos con relación a sistemas de información

 Pieter De Leenheer, Stijn Christiaens, Robert Meersman

 [Business semantics management: A case study for competency-centric HRM.](#)

 Computers in Industry - 61 (2010)

 Yuh-Jen Chen, Yuh-Min Chen, Hui-Chuan Chu

 [Enabling collaborative product design through distributed engineering knowledge management.](#)

 Computers in Industry - 59 (2008)

 Rafael Gouriveau, Daniel Noyes

 [Risk management - dependability tools and case-based reasoning integration using the object formalism.](#)

 Computers in Industry - 55 (2004)

 Borhen Louhichi, Louis Rivest

 [Maintaining consistency between CAD elements in collaborative design using association management and propagation.](#)

 Computers in Industry - 65 (2014)

 Dietmar Posselt, Gerd Hillebrand

 [Database support for evolving data in product design.](#)

 Computers in Industry - 48 (2002)

 Robert W. Brennan, Douglas H. Norrie

 [Metrics for evaluating distributed manufacturing control systems.](#)

 Computers in Industry - 51 (2003)

 John Ryan, Cathal Heavey

 [Process modeling for simulation.](#)

 Computers in Industry - 57 (2006)

Imagen 19.- Artículos clasificados con relación a: “Sistemas de información”.

Artículos con relación a interacción humano computadora

-  [Christian Noon, Ruqin Zhang, Elliot Winer, James Oliver, Brian Gilmore, Jerry Duncan](#)
-  [A system for rapid creation and assessment of conceptual large vehicle designs using immersive virtual reality.](#)
-  Computers in Industry - 63 (2012)
-  [Hans-Jörg Bullinger, Wilhelm Bauer, Günter Wenzel, Roland Blach](#)
-  [Towards user centred design \(UCD\) in architecture based on immersive virtual environments.](#)
-  Computers in Industry - 61 (2010)
-  [C. N. Verdouw, Adrie J. M. Beulens, Jacques H. Trienekens, Tim Verwaart](#)
-  [Towards dynamic reference information models: Readiness for ICT mass customisation.](#)
-  Computers in Industry - 61 (2010)
-  [Amar Kumar Behera, Bert Lauwers, Joost R. Dufflou](#)
-  [Tool path generation framework for accurate manufacture of complex 3D sheet metal parts using single point incremental forming.](#)
-  Computers in Industry - 65 (2014)
-  [Wei Xiang, Sai Cheong Fok, Georg Thimm](#)
-  [Agent-based composable simulation for virtual prototyping of fluid power system.](#)
-  Computers in Industry - 54 (2004)
-  [Robin Chavanne, Bernard Anselmetti](#)
-  [Functional tolerancing: Virtual material condition on complex junctions.](#)
-  Computers in Industry - 63 (2012)

Figura 20.- Artículos clasificados con relación a: “Interacción humano computadora”.

Artículos con relación a software

 Jirawan Kloypayan, Yuan-Shin Lee

 [Material engagement analysis of different endmills for adaptive feedrate control in milling processes.](#)

 Computers in Industry - 47 (2002)

 Yi Huang, Brian C. Williams, Li Zheng

 [Reactive, model-based monitoring in RFID-enabled manufacturing.](#)

 Computers in Industry - 62 (2011)

 S. K. Kwok, Ocean P. H. Ng, Albert H. C. Tsang, H. M. Liem

 [Physimetric identification \(Physi-ID\) - Applying biometric concept in physical object identification.](#)

 Computers in Industry - 62 (2011)

 Selma Limam Mansar, Hajo A. Reijers

 [Best practices in business process redesign: validation of a redesign framework.](#)

 Computers in Industry - 56 (2005)

 Diyar Akay, Osman Kulak, Brian Henson

 [Conceptual design evaluation using interval type-2 fuzzy information axiom.](#)

 Computers in Industry - 62 (2011)

 Young-Jun Son, Richard A. Wysk

 [Automatic simulation model generation for simulation-based, real-time shop floor control.](#)

 Computers in Industry - 45 (2001)

 Xiaoguang Deng, Gregory Huet, Suo Tan, Clement Fortin

 [Product decomposition using design structure matrix for intellectual property protection in supply chain outsourcing.](#)

 Computers in Industry - 63 (2012)

Figura21.- Artículos clasificados con relación a: “Software”.

8. Discusión de los resultados

La funcionalidad principal de este proyecto se logra gracias al algoritmo clasificador. Sin embargo, este algoritmo ha dado en ocasiones resultados incoherentes en relación con algunas temáticas. Esto es debido a la elección de títulos clave que se han elegido como representantes de las temáticas. Si bien, se realizaron pruebas para elegir las semillas más adecuadas a cada temática, un cambio en una sola de las semillas afecta de manera considerable los resultados de la clasificación proporcionando, en algunos casos, resultados desproporcionados para algunas temáticas. Como se puede observar en la clasificación de: Seguridad y privacidad. Se analizó el 20% de los resultados obtenidos de esta clasificación, de los cuales solo el 12% de los resultados tenía relación con dicha temática.



Estos resultados se deben a las siguientes razones:

1. No se escogió bien el título representante para esta temática.
2. Los temas relacionados a esta temática, son escasos en las ontologías que se están leyendo.
3. Seguridad es un término que se puede encontrar con facilidad en otras temáticas, lo que provoca que los temas relacionados a seguridad tengan otra categorización.

Como se puede observar, un factor importante es la elección de estos títulos semilla, ya que no nada más afecta el resultado de su propia clasificación, sino que afecta también el resultado de otras temáticas.

Para la elección de un título clave o semilla. Es necesario tomar algunas consideraciones:

- El título tiene que tener palabras clave muy representativas (o incluso únicas) a la temática que está representando.
- Se debe evitar que la frase contenga palabras repetidas.
- Se debe evitar la utilización de palabras que se pueden encontrar en otros temas. Como por ejemplo: sistemas, jerarquía, datos, etc.
- Todos los títulos deberían tener un número similar de palabras. Es decir, no debe haber títulos que contengan 5 palabras y otros que contengan 20.

A pesar de que las consideraciones sean tomadas correctamente, el algoritmo puede presentar resultados erróneos. Un claro ejemplo de esto, es que para este algoritmo una palabra en singular y una palabra en plural son totalmente diferentes. Así que si en nuestro título semilla se ha escogido una oración con palabras en plural, es muy probable que al comparar otros textos que contengan las mismas palabras, pero en singular, dé un resultado de similitud bajo, y por consiguiente este texto no sea clasificado correctamente.

La solución sería convertir todas las palabras de todos los títulos a singular. Con esto los porcentajes de similitud serían más altos. Para ello, es necesario conocer las reglas gramaticales del idioma en el que se está trabajando y saber detectar cuando una palabra está en plural y cuál es su equivalente en singular.

Otra solución, sería aplicar el mismo algoritmo de comparación, pero esta vez enfocado a calcular la similitud de cada palabra de un título con cada palabra un segundo título. Así se detectaría que “sistema” y “sistemas” son dos palabras muy similares y no diferentes, haciendo que la similitud total entre los títulos sea más alta. Sin embargo, esto haría que el tiempo de ejecución sea más lento, al tener que comparar título por título, palabra por palabra.

9. Conclusiones

A lo largo de este proyecto se fueron cumpliendo los objetivos establecidos en la propuesta. Se logró construir un modulo que leyera el contenido de las ontologías, así como también se logró optimizar el tiempo de lectura de éstas mismas.

Se consiguió crear un algoritmo clasificador y estudiar su comportamiento. Asimismo, se observaron las ventajas y desventajas que tiene este algoritmo. Como el hecho de que las semillas que recibe este algoritmo tienen que ser muy explícitas en relación a la temática que quieren representar o el hecho de que es un algoritmo que cuyo tiempo de ejecución es un poco lento, ya que compara todas las semillas contra todos los títulos.

Con esto también se aprendió un poco el principio de lo que es el procesamiento lenguaje natural y una de las tantas aplicaciones que este tiene en aplicaciones reales.

Se observó las diversas aplicaciones que puede tener esta aplicación en otros proyectos, como por ejemplo: en un sistema de búsqueda de artículos podemos implementar un modulo de sugerencias donde se muestre una lista de otros artículos que tengan relación a la temática del artículo que se está viendo.

Otra aplicación podría ser en la UAM para la toma de decisiones sobre el plan de estudios de ingeniería y maestría en computación. En lugar de clasificar artículos científicos clasificamos el plan de estudios de ingeniería en computación. Esta información podría servir para observar hacia qué temas tiene más tendencia dichos planes. Permitiendo decidir si hace falta la creación de más UEAs o perfiles en ingeniería o maestría en computación. Se podría ser más específico y analizar programa de estudios de cada UEA y así determinar si existe alguna posibilidad de mejorar dicha UEA.

En un futuro este proyecto podría expandirse y dejar de limitarse a clasificar artículos relacionados al área de computación. De igual manera se podría dejar de clasificar solamente artículos científicos y comenzar a clasificar cualquier tipo de documento. En una biblioteca esto podría ser muy útil, ya que se clasificaría automáticamente cada libro existente. Con esto sería más fácil el ordenamiento de los libros y dejaría a un lado la labor humana de tener que asignarle manualmente un tema (o varios) a cada uno de los libros existentes.

Por último, este proyecto consiguió implementar una arquitectura de 3 capas MVC. Separando el jsp para la visualización de datos, el Servlet para el procesamiento de datos y dejando a las ontologías en la capa del modelo.

Referencia bibliográfica

- [1] F. Alejandro Acosta, "Sistemas de procesamiento de textos de investigación". México 2013: Universidad Autónoma Metropolitana Azcapotzalco. 2013.
- [2] P. G. Malagón Torres, "Arquitectura orientada a servicios web para establecer grupos de colaboración entre investigadores". México, 2016: Universidad Autónoma Metropolitana. Azcapotzalco, 2016.
- [3] C. Pilapanta Herrera, "Sistema web para el poblado automático de ontologías a partir de textos". México, 2015: Universidad Autónoma Metropolitana. Azcapotzalco, 2015
- [4] G. de la Calle Velazco, "Modelo basado en técnicas de procesamiento de lenguaje natural para extraer y anotar información de publicaciones científicas.", Doctorado, Universidad Politécnica de Madrid, 2014.
- [5] J. A. Reyes Ortiz, M. C. Bravo Contreras "Categorización de resúmenes de publicaciones científicas basada en similitud semántica" Información tecnológica, no. 1, 2015.
- [6] J. A. Reyes Ortiz, "Descubrimiento y representación de conocimiento sobre datos científicos y académicos" Información tecnológica, no. 1, 2015.

Apéndices

Anexo A Ontología DBLP.xml de ejemplo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dblp SYSTEM "dblp.dtd">
<dblp>
<article mdate="2011-04-18" key="journals/cii/OlatunjiSR11">
<author>Sunday Olusanya Olatunji</author>
<author>Ali Selamat</author>
<author>Abdul Azeez Abdul Raheem</author>
<title>Modeling the permeability of carbonate reservoir using type-2 fuzzy logic systems.</title>
<pages>147-163</pages>
<year>2011</year>
<volume>62</volume>
<journal>Computers in Industry</journal>
<number>2</number>
<ee>http://dx.doi.org/10.1016/j.compind.2010.10.008</ee>
<url>db/journals/cii/cii62.html#OlatunjiSR11</url>
</article>
</dblp>
```

Anexo B: Clase xml

```
public class xml {
    public ArrayList<String> leer (String url){
        ArrayList <String> titulos = new ArrayList<String>();
        String etiqueta = new String();
        int auxAutor=0;
        try{
            SAXBuilder builder = new SAXBuilder();
            File xmlFile = new File(url);
            Document document = (Document) builder.build( xmlFile );
            Element rootNode = document.getRootElement();
            List list = rootNode.getChildren( "article" );
            int i, j, cont = 0;
            for (i = 0; i <list.size(); i++ ) {
                Element tabla = (Element) list.get(i);
                List lista_campos = tabla.getChildren();
                for (j = 0; j < lista_campos.size(); j++ ){
                    Element campo = (Element)lista_campos.get( j );
                    etiqueta = campo.getName();
                    switch (etiqueta) {
                        case "title":
                            System.out.println("titulo:" +campo.getText() +"\n");
                            titulos.add(tokenizar(campo.getText().toLowerCase()));
                            Articulo_split.append("@A@");
                            Articulo_split.append("@T@");
                            Articulo_split.append(campo.getText());
                            Articulo_split.append("@T@");
                            break;
                        case "ee":
                            Articulo_split.append("@U@");
                            Articulo_split.append(campo.getText());
                            Articulo_split.append("@U@");
                            break;
                        case "author":
                            if (auxAutor==0) {
                                Articulo_split.append("@A@").append(campo.getText());
                                auxAutor++;
                            }
                            else {
                                Articulo_split.append(", ").append(campo.getText());
                            }
                            break;
                        case "journal":
                            Articulo_split.append("@J@");
                            Articulo_split.append(campo.getText());
                            Articulo_split.append("@J@");
                            break;
                    }
                }
            }
        }
    }
}
```

```

        case "volume":
            Articulo_split.append("@V@");
            Articulo_split.append(campo.getText());
            Articulo_split.append("@V@");
        break;
        case "year":
            Articulo_split.append("@Y@");
            Articulo_split.append(campo.getText());
            Articulo_split.append("@Y@");
        break;
    }
}
if(Articulo_split.toString() != null || Articulo_split.toString()!=""){
    auxArticulos.add(Articulo_split.toString());
}
auxAutor=0;
Articulo_split.delete(0, (Articulo_split.length()-1));
Articulo.clear();
}
return titulos;
}
catch ( IOException io ){
    return null;
}
catch ( JDOMException jdomex ){
    return null;
}
}
}

private String tokenizar (String tituloOnto)
{
    String cadTokenizada="";
    StringTokenizer tokens= new StringTokenizer(tituloOnto, " ,.:='\\"?!( )123456789+ -_");
    while(tokens.hasMoreTokens()){
        cadTokenizada+=tokens.nextToken()+" ";
    }
    return cadTokenizada;
}
}

```

Anexo C: Clase articuloCientifico

```

public class articuloCientifico {
    private String semilla;
    private String abreviatura;
    private ArrayList <String> articulos_similares = new ArrayList<String>();
}

```

```

public articuloCientifico(String semilla, String abreviatura){
    this.semilla=semilla;
    this.abreviatura=abreviatura;
}

public String obtenerAbreviatura (){
return abreviatura;
}

public void agregarArticulo(String Articulo){
    articulos_similares.add(Articulo);
}

public ArrayList<String> ObtenerArticulos()
{
    if(articulos_similares.isEmpty()) {
        articulos_similares.add("No se encontraron resultados");
        return articulos_similares;
    }
    else{
        return articulos_similares;
    }
}

public double compararArticulo(String semillaAC) {
    double similitud;
    similitud = cosenoComparador.comparar(semilla, semillaAC);
    return similitud;
}
}

```

Anexo D: cosenoComparador

```

public class cosenoComparador {
    static public double comparar(String cadena1, String cadena2)
    {
        ArrayList <String> todosloselem = new ArrayList();
        String[] cadena1aux;
        String[] cadena2aux;
        cadena1aux = cadena1.split(" ");
        cadena2aux = cadena2.split(" ");
        int i, j;
        for(i=0;i<cadena1aux.length; i++){
            if(!todosloselem.contains(cadena1aux[i]))
                todosloselem.add(cadena1aux[i]);
        }
        for(i=0;i<cadena2aux.length; i++){
            if(!todosloselem.contains(cadena2aux[i]))

```

```

        todosloselem.add(cadena2aux[i]);
    }
    int [] v1 = new int [todosloselem.size()];
    int [] v2 = new int [todosloselem.size()];
    for(i=0; i<todosloselem.size();i++){
        v1[i]=0;
        v2[i]=0;
    }
    int ax=0;
    for(i=0; i<todosloselem.size(); i++){
        for(j=0;j<cadena1aux.length;j++){
            if(cadena1aux[j].equals(todosloselem.get(i))){
                ax++;    v1[i]=ax;
            }
        }
        ax=0;
    }
    for(i=0; i<todosloselem.size(); i++){
        for(j=0;j<cadena2aux.length;j++){
            if(cadena2aux[j].equals(todosloselem.get(i))){
                ax++;    v2[i]=ax;
            }
        }
        ax=0;
    }
    int cont1=0, cont2=0,contc=0,cont1c=0,cont2c=0;
    String V1="", V2="";
    for(i=0; i<todosloselem.size();i++){
        cont1=cont1 + v1[i];
        contc= contc +(v1[i]*v2[i]);
        System.out.print(v1[i] + " , ");
        V1= V1.concat(String.valueOf(v1[i]));
        if(i<(todosloselem.size()-1))
            V1 += " , ";
    }
    for(i=0; i<todosloselem.size();i++){
        cont2=cont2 + v2[i];
        cont1c= cont1c + (v1[i]*v1[i]);
        cont2c= cont2c + (v2[i]*v2[i]);
        V2= V2.concat(String.valueOf(v2[i]));
        if(i<(todosloselem.size()-1))
            V2 += " , ";
    }
    double cos=0.f;
    cos= (contc)/(Math.sqrt(cont1c*cont2c));
    return cos;
}
}

```

Anexo E: Servlet ConsultarArticulos.java

```
public class ConsultarArticulos extends HttpServlet {
    ArrayList <String> tls = new ArrayList<String>();
    ArrayList <String> arti = new ArrayList<String>();
    ArrayList <String> articulos_clasificados = new ArrayList<String>();

    private final articuloCientifico HR = new articuloCientifico(hr,"HR");
    private final articuloCientifico CO = new articuloCientifico(co,"CO");
    private final articuloCientifico NW = new articuloCientifico(nw,"NW");

    private final articuloCientifico SW = new articuloCientifico(sw,"SW");
    private final articuloCientifico TC = new articuloCientifico(tc,"TC");
    private final articuloCientifico MC = new articuloCientifico(mc,"MC");

    private final articuloCientifico IS = new articuloCientifico(is,"IS");
    private final articuloCientifico SP = new articuloCientifico(sp,"SP");
    private final articuloCientifico HC = new articuloCientifico(hc,"HC");

    private StringBuilder HTMLArticulo = new StringBuilder();

    public void clasificar(){
        double indice = 0, aux=0;
        String Abreviatura = "";
        for(i=0; i<tls.size(); i++){
            aux = HR.compararArticulo(tls.get(i));
            if(aux>indice){
                indice = aux;
                Abreviatura=HR.obtenerAbreviatura();
            }
            aux = CO.compararArticulo(tls.get(i));
            if(aux>indice){
                indice = aux;
                Abreviatura= CO.obtenerAbreviatura();
            }
            aux = NW.compararArticulo(tls.get(i));
            if(aux>indice){
                indice = aux;
                Abreviatura=NW.obtenerAbreviatura();
            }
            aux = SW.compararArticulo(tls.get(i));
            if(aux>indice){
                indice = aux;
                Abreviatura=SW.obtenerAbreviatura();
            }
            aux = TC.compararArticulo(tls.get(i));
            if(aux>indice){
```

```

indice = aux;
Abreviatura=TC.obtenerAbreviatura();
}
aux = MC.compararArticulo(tls.get(i));
if(aux>indice){
indice = aux;
Abreviatura=MC.obtenerAbreviatura();
}
aux = IS.compararArticulo(tls.get(i));
if(aux>indice){
indice = aux;
Abreviatura=IS.obtenerAbreviatura();
}
aux = SP.compararArticulo(tls.get(i));
if(aux>indice){
indice = aux;
Abreviatura=SP.obtenerAbreviatura();
}
aux = HC.compararArticulo(tls.get(i));
if(aux>indice){
indice = aux;
Abreviatura= HC.obtenerAbreviatura();
}
indice=0;
switch (Abreviatura){
    case "HR":
        HR.agregarArticulo(arti.get(i));
        break;
    case "CO":
        CO.agregarArticulo(arti.get(i));
        break;
    case "NW":
        NW.agregarArticulo(arti.get(i));
        break;
    case "SW":
        SW.agregarArticulo(arti.get(i));
        break;
    case "TC":
        TC.agregarArticulo(arti.get(i));
        break;
    case "MC":
        MC.agregarArticulo(arti.get(i));
        break;
    case "IS":
        IS.agregarArticulo(arti.get(i));
        break;
    case "SP":
        SP.agregarArticulo(arti.get(i));

```

```

        break;
        case "HC":
            HC.agregarArticulo(arti.get(i));
            break;
    }
}

public String MostrarArticulosClasificados(String respuesta)
{
    articulos_clasificados = HR.ObtenerArticulos();
    switch (respuesta){
        case "HR":
            for(i=0;i<articulos_clasificados.size();i++){
                HTMLArticulo.append("<p>")
                .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
                .append("<p>");
            }
            break;
        case "CO":
            articulos_clasificados = null;
            articulos_clasificados = CO.ObtenerArticulos();
            for(i=0;i<articulos_clasificados.size();i++){
                HTMLArticulo.append("<p>")
                .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
                .append("<p>");
            }
            break;
        case "NW":
            articulos_clasificados = null;
            articulos_clasificados = NW.ObtenerArticulos();
            for(i=0;i<articulos_clasificados.size();i++){
                HTMLArticulo.append("<p>")
                .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
                .append("<p>");
            }
            break;
        case "SW":
            articulos_clasificados = null;
            articulos_clasificados = SW.ObtenerArticulos();
            for(i=0;i<articulos_clasificados.size();i++){
                HTMLArticulo.append("<p>")
                .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
                .append("<p>");
            }
            break;
        case "TC":
            articulos_clasificados = null;

```

```

articulos_clasificados = TC.ObtenerArticulos();
for(i=0;i<articulos_clasificados.size();i++){
    HTMLArticulo.append("<p>")
    .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
    .append("<p>");
}
break;
case "MC":
articulos_clasificados = null;
articulos_clasificados = MC.ObtenerArticulos();
for(i=0;i<articulos_clasificados.size();i++){
    HTMLArticulo.append("<p>")
    .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
    .append("<p>");
}
break;
case "IS":
articulos_clasificados = null;
articulos_clasificados = IS.ObtenerArticulos();
for(i=0;i<articulos_clasificados.size();i++){
    HTMLArticulo.append("<p>")
    .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
    .append("<p>");
}
break;
case "SP":
articulos_clasificados = null;
articulos_clasificados = SP.ObtenerArticulos();
for(i=0;i<articulos_clasificados.size();i++){
    HTMLArticulo.append("<p>")
    .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
    .append("<p>");
}
break;
case "HC":
articulos_clasificados = null;
articulos_clasificados = HC.ObtenerArticulos();
for(i=0;i<articulos_clasificados.size();i++){
    HTMLArticulo.append("<p>")
    .append(ArticuloHtml.obtenerHTML(articulos_clasificados.get(i)))
    .append("<p>");
}
break;
default:
HTMLArticulo.append("");
}
return HTMLArticulo.toString();
}

```

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {

        String respuesta = request.getParameter("TA");
        String num_ontologia = request.getParameter("no_Ontologia");
        String ruta="C:\\03\\dblp"+num_ontologia+".xml";
        double inicio,fin, resultadoTiempo;
        xml onto = new xml();
        t1s = onto.leer(ruta);
        arti = onto.ObtenerauxArticulos();

        clasificar();
        out.print(MostrarArticulosClasificados(respuesta));
    }
}
}
}

```

Anexo F: ArtículoHTML.java

```

public class ArtículoHtml {
    static public String obtenerHTML(String texto)
    {
        String regex = "@@A@([a-zA-Z. -BáàâäëèèëíîïïóôðöóúüüÁÀÁÁÉÉÉÉÉÍÎÏÓÓÓÓÓÚÚÚÚÚýçñã?\\.\\.]+)@A@" //ñ
        + "@T@([a-zA-Z. -BáàâäëèèèèíîïïóôðöóúüüÁÀÁÁÉÉÉÉÉÍÎÏÓÓÓÓÓÚÚÚÚÚ\\.\\.]+)@T@"
        + "@Y@([0-9]{4})@Y@"
        + "@V@([0-9]{1,5})@V@" //añadir signo -
        + "@J@([a-zA-Z. -BáàâäëèèèèíîïïóôðöóúüüÁÀÁÁÉÉÉÉÉÍÎÏÓÓÓÓÓÚÚÚÚÚ\\.\\.]+)@J@"
        + "@U@([:a-zA-Z0-9.()//_-]+)@U@"; //|@U@([:a-zA-Z0-9.()//_-]+)@U@

        Pattern patron = Pattern.compile(regex);
        Matcher emparejador = patron.matcher(texto);
        String sustitucion = "<i class=\"material-icons\">perm_identity</i>"
        + "<span class=\"red-text text-darken-2\">$1</span><br>"
        + "<i class=\"material-icons\">language</i> <a href=\"$6\" target=\"_blank\">$2</a><br>"
        + "<i class=\"material-icons\">class</i> $5 - $4 ($3)<br>";
        if(emparejador.find()){
            String resultado = emparejador.replaceAll(sustitucion);
            return resultado;
        }else{
            regex = "@@A@([a-zA-Z. -BáàâäëèèèèíîïïóôðöóúüüÁÀÁÁÉÉÉÉÉÍÎÏÓÓÓÓÓÚÚÚÚÚýçñã?\\.\\.]+)@A@" //ñ
            + "@T@([a-zA-Z. -BáàâäëèèèèíîïïóôðöóúüüÁÀÁÁÉÉÉÉÉÍÎÏÓÓÓÓÓÚÚÚÚÚ\\.\\.]+)@T@"
            + "@J@([a-zA-Z. -BáàâäëèèèèíîïïóôðöóúüüÁÀÁÁÉÉÉÉÉÍÎÏÓÓÓÓÓÚÚÚÚÚ\\.\\.]+)@J@"
            + "@V@([0-9]{1,5})@V@"
            + "@Y@([0-9]{4})@Y@"
            + "@U@([:a-zA-Z0-9.()//_-]+)@U@";
        }
    }
}

```



```

        <a href="#articulos" onclick="AjaxArticulos('TC' , 1)">
        Teoría de la computación
        </a>
    </div><br>
    <div class="chip">
        <a href="#articulos" onclick="AjaxArticulos('MC' , 1)">
        Matemáticas computacionales
        </a>
    </div><br>
    <div class="chip">
        <a href="#articulos" onclick="AjaxArticulos('SP' , 1)">
        Seguridad
        </a>
    </div><br>
    <div class="chip">
        <a href="#articulos" onclick="AjaxArticulos('HC' , 1)">
        Interacción H-C
        </a>
    </div><br>
</div>
<div class="container" id="inicio">
    <div class="row">
        <div class="col s4">
            <div class="card">
                <div class="card-image">
                    <a href="#articulos" onclick="AjaxArticulos('NW' , 1)">
                        
                        <span class="card-title red darken-2">Redes</span>
                    </a>
                </div>
                <div class="card-content blue-grey lighten-5">
                    <p align="justify">Esta es una nueva rama de nivel superior que contiene una
mezcla de términos nuevos y
                    <a href="#modal_redes"><i class="material-icons red-text text-darken-
2">add_box</i></a>
                    </p>
                </div>
            </div>
        </div>
        <div class="col s4">
            <div class="card">
                <div class="card-image">
                    <a href="#articulos" onclick="AjaxArticulos('HR' , 1)">
                        
                        <span class="card-title red darken-2">Hardware</span>
                    </a>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="card-content blue-grey lighten-5">
            <p align="justify">En la rama de hardware fueron consolidados
recientemente de los conceptos de
                performance y confiabilidad en un único nodo. La clasificación 1998
tenía...
            <a href="#modal_hardware"><i class="material-icons red-text text-darken-
2">add_box</i></a>
        </p>
    </div>
</div>
</div>

```

```

<div class="col s4">
    <div class="card">
        <div class="card-image">
            <a href="#articulos" onclick="AjaxArticulos('MC' , 1)">
                
                <span class="card-title red darken-
2">Matemáticas<br>computacionales</span>
            </a>
        </div>
        <div class="card-content blue-grey lighten-5">
            <p align="justify">En esta rama se encuentran conceptos como: análisis
numérico,
                análisis de algoritmos (en la teoría de la rama de cómputo),
algoritmos y problemas numé...
            <a href="#modal_matematicas"><i class="material-icons red-text text-
darken-2">add_box</i></a>
        </p>
    </div>
</div>
</div>

```

```

<div class="row">
    <div class="col s4">
        <div class="card">
            <div class="card-image">
                <a href="#articulos" onclick="AjaxArticulos('CO' , 1)">
                    
                    <span class="card-title red darken-2">Sistemas <br>computacionales</span>
                </a>
            </div>
            <div class="card-content blue-grey lighten-5">
                <p align="justify">
                    En esta rama se encuntran las arquitecturas de procesador y se
reorganizan algunos conceptos
                        dentro de ella. La clasificación 1998 tenía un concepto l...
                </p>
            </div>
        </div>
    </div>
</div>

```

```

        <a href="#modal_scomputacionales"><i class="material-icons red-text
text-darken-2">add_box</i></a>
    </p>
</div>
</div>
</div>
</div>
<div class="col s4">
    <div class="card">
        <div class="card-image">
            <a href="#articulos" onclick="AjaxArticulos('SW' , 1)">
                
                <span class="card-title red darken-2">Software</span>
            </a>
        </div>
        <div class="card-content blue-grey lighten-5">
            <p align="justify">
                Un cambio importante que ha sufrido esta rama es la eliminación de las
anteriores
                técnicas de programación. Se consideró la posibilidad de ubicar
este...
            <a href="#modal_software"><i class="material-icons red-text text-
darken-2">add_box</i></a>
        </p>
    </div>
</div>
</div>
</div>
<div class="col s4">
    <div class="card">
        <div class="card-image">
            <a href="#articulos" onclick="AjaxArticulos('TC' , 1)">
                
                <span class="card-title red darken-2">Teoría de la<br>computación</span>
            </a>
        </div>
        <div class="card-content blue-grey lighten-5">
            <p align="justify">
                La teoría de la computación se dividimó en dos conceptos: Análisis de
algoritmos,
                y diseño de algoritmos. El análisis de algoritmos contiene métodos
d...
            <a href="#modal_teoría"><i class="material-icons red-text text-darken-
2">add_box</i></a>
        </p>
    </div>
</div>
</div>
</div>

```

```

</div>

<div class="row">
  <div class="col s4">
    <div class="card">
      <div class="card-image">
        <a href="#articulos" onclick="AjaxArticulos('IS' , 1)">
          
          <span class="card-title red darken-2">Sistemas de<br>información</span>
        </a>
      </div>
      <div class="card-content blue-grey lighten-5">
        <p align="justify">
          Los sistemas de información ahora reflejan una mayor variedad de
aplicaciones, incluyendo
          almacenamiento de datos, sistemas de soporte de decisiones, ...
          <a href="#modal_sininformacion"><i class="material-icons red-text text-
darken-2">add_box</i></a>
        </p>
      </div>
    </div>
  </div>

  <div class="col s4">
    <div class="card">
      <div class="card-image">
        <a href="#articulos" onclick="AjaxArticulos('SP' , 1)">
          
          <span class="card-title red darken-2">Seguridad y <br>privacidad</span>
        </a>
      </div>
      <div class="card-content blue-grey lighten-5">
        <p align="justify">
          Esta nueva rama incorpora conceptos dispersos de la antigua
clasificación así como
          otros nuevos términos. El cifrado de datos es ahora criptografía y está
más...
          <a href="#modal_seguridad"><i class="material-icons red-text text-
darken-2">add_box</i></a>
        </p>
      </div>
    </div>
  </div>

  <div class="col s4">
    <div class="card">
      <div class="card-image">
        <a href="#articulos" onclick="AjaxArticulos('HC' , 1)">

```

```

        
        <span class="card-title red darken-2">Interacción <br>humano -
computadora</span>
    </a>
</div>
<div class="card-content blue-grey lighten-5">
    <p align="justify">
        La informática centrada en el ser humano (HCC) se ha convertido en una
disciplina por
        derecho propio desde la última iteración de la clasificación. De...
    <a href="#modal_interaccion"><i class="material-icons red-text text-
darken-2">add_box</i></a>
    </p>
</div>
</div>
</div>
</div>
</div>
<div class="container">
    <div class="row" id="articulos">

    </div>
</div>
<div class="container">
    <div class="row" id="cargando">

    </div>
</div>
<div class="container">
    <div class="row" id="masr">

    </div>
</div>
</div>
<jsp:include page="HAF/Footer.jsp" />
</body>
</html>

```

Anexo H: Javascript funcionesPI.js

```

var cont=0;
var ont_max=5;
$(document).ready(function(){
    modales();
});
function modales(){
    $(''.modal').modal();
}

```

```

}
function titulos(TA) {
    var text;
    switch(TA) {
    case "HR":
        text = "<H3><b>Artículos con relación a hardware</b></H3>";
        return text;
        break;
    case "CO":
        text = "<H3><b>Artículos con relación a sistemas computacionales</b></H3>";
        return text;
        break;
    case "NW":
        text = "<H3><b>Artículos con relación a redes</b></H3>";
        return text;
        break;
    case "SW":
        text = "<H3><b>Artículos con relación a software</b></H3>";
        return text;
        break;
    case "TC":
        text = "<H3><b>Artículos con relación a teoría de la computación</b></H3>";
        return text;
        break;
    case "MC":
        text = "<H3><b>Artículos con relación a matemáticas de la computación</b></H3>";
        return text;
        break;
    case "IS":
        text = "<H3><b>Artículos con relación a sistemas de información</b></H3>";
        return text;
        break;
    case "SP":
        text = "<H3><b>Artículos con relación a seguridad y privacidad</b></H3>";
        return text;
        break;
    case "HC":
        text = "<H3><b>Artículos con relación a interacción humano computadora</b></H3>";
        return text;
        break;;
    }
}
function AjaxArticulos(TA , primeravez){
    if(primeravez == 1)
    {
        cont=0;
        $("#articulos" ).html("");
    }
}

```

```

no_Ontologia=(cont+1);
if(cont<ont_max)
{
$.ajax({
url:"ConsultarArticulos",
method:"GET",
data : { TA : TA, no_Ontologia:no_Ontologia},
beforeSend: function(){
$("#cargando").html('<center><IMG SRC="img/icon_loading.gif "><h2>Cargando</h2><center>');
$("#masr" ).hide();
},
success : function(data) {
$("#cargando" ).html('');
if(cont ==0){
var titulo = titulos(TA);
$("#articulos").html(titulo);
}
$("#articulos").append(data);
if(no_Ontologia!=ont_max){
$("#masr").html('<a href="#inicio" class="col s3 grey lighten-1 waves-effect waves-grey btn-flat white-text">Regresar a menu principal <i class="material-icons">expand_less</i></a>');
$("#masr").append('<a class="col offset-s6 s3 red darken-2 waves-effect waves-red btn-flat white-text" onclick="AjaxArticulos(\''+TA+\'\' , 0)">Mostrar más resultados <i class="material-icons">add</i></a>');$
$("#masr").show();
}else{
$("#masr").html('<a href="#inicio" class="col s3 grey lighten-1 waves-effect waves-grey btn-flat white-text">Regresar a menu principal <i class="material-icons">expand_less</i></a>');
$("#masr").show();
}
cont++;
}
});
}
}

```