

Universidad Autónoma Metropolitana. Unidad Azcapotzalco  
División de Ciencias Básicas e Ingeniería  
Licenciatura en Ingeniería en Computación

Sistema clasificador de ontologías mediante métodos de máquinas de  
soporte vectorial

Moisés García Sierra  
Matrícula: 208333213

Trimestre 2013 Otoño

11 de diciembre de 2013

Reporte final

Asesores:

Dra. Maricela Claudia Bravo Contreras  
Profesor Asociado  
Departamento de Sistemas

Dr. José Guadalupe Rodríguez García  
Departamento de Computación del CINVESTAV  
Unidad Zacatenco

# Índice general

<b>Índice de figuras</b>	<b>4</b>
<b>Resumen</b>	<b>5</b>
<b>Objetivos</b>	<b>6</b>
Objetivo general . . . . .	6
Objetivos específicos . . . . .	6
<b>Introducción</b>	<b>7</b>
<b>Desarrollo del proyecto</b>	<b>9</b>
Arquitectura del sistema . . . . .	9
Repositorio de Ontologías . . . . .	9
Analizador Sintáctico: . . . . .	10
Índices . . . . .	10
Clasificador . . . . .	10
Interfaz gráfica . . . . .	10
Implementación . . . . .	11
Tecnologías utilizadas . . . . .	11
Hibernate . . . . .	11
OWL API . . . . .	11
Apache Tomcat . . . . .	11
MySQL . . . . .	11
Java Server Pages . . . . .	12
Java Server Faces . . . . .	12
PrimeFaces . . . . .	12
Implementación de los Modulos . . . . .	13
Repositorio de Ontologías . . . . .	13
Analizador Sintáctico . . . . .	14
Indices . . . . .	14
Clasificador . . . . .	15

Interfaz Gráfica . . . . .	17
<b>Pruebas</b>	<b>19</b>
Recursos . . . . .	19
Demostración . . . . .	19
Buscador . . . . .	19
Clasificador . . . . .	20
Primer ejemplo . . . . .	20
Segundo ejemplo . . . . .	20
<b>Conclusiones</b>	<b>26</b>
<b>Trabajos Futuros</b>	<b>27</b>
<b>Apéndice</b>	<b>28</b>
<b>Bibliografía</b>	<b>35</b>

# Índice de figuras

1.	La información y el sistema clasificador de ontologías. . . . .	7
2.	Sistema clasificador de ontologías . . . . .	9
3.	Directorio de ontologías preclasificadas . . . . .	13
4.	Directorio de ontologías a clasificar . . . . .	13
5.	Diagrama Entidad - Relación . . . . .	15
6.	Interfaz Web - Buscador de ontologías . . . . .	17
7.	Interfaz Web - Clasificador de ontologías . . . . .	18
8.	Buscador de ontologías . . . . .	19
9.	Ejemplo 1 - Clasificador de ontologías . . . . .	20
10.	Ejemplo 1 - Ontologías preclasificadas . . . . .	21
11.	Ejemplo 1 - Representación de ontologías a clasificar . . . . .	22
12.	Ejemplo 1 - Resultados . . . . .	22
13.	Ejemplo 1 - Archivo de texto . . . . .	23
14.	Ejemplo 2 - Clasificador de ontologías . . . . .	23
15.	Ejemplo 2- Ontologías preclasificadas . . . . .	24
16.	Ejemplo 2 - Representación de ontologías a clasificar . . . . .	24
17.	Ejemplo 2 - Resultados . . . . .	24
18.	Ejemplo 2 - Archivo de texto . . . . .	25

# Resumen

La información digital día a día crece en cantidad y volumen, sin embargo dicha información se encuentra dispersa y ubicua<sup>1</sup>, por lo que cada vez es más difícil procesarla manualmente. Por lo que se requiere de ayuda no humana para procesar y agrupar dicha información automáticamente.

Este sistema clasificador de ontologías, es un trabajo por clasificar ontologías de manera automática en base a un conjunto de ontologías ya preclasificadas.

Este proyecto Terminal también da una noción de la forma de hacerlo utilizando similitudes y Maquinas de Soporte Vectorial.

**Palabras claves:** *SVM, Maquinas de Soporte Vectorial, Clasificación de Ontologías, Similitud de Ontologías*

---

<sup>1</sup>Están en todas partes

# Objetivos

## Objetivo general

Diseñar e implementar una aplicación web que lea un conjunto de ontologías y las clasifique utilizando métodos de máquinas de soporte vectorial.

## Objetivos específicos

1. Seleccionar e implementar un analizador sintáctico para leer e identificar los componentes principales<sup>2</sup> de una ontología.
2. Seleccionar y adaptar un algoritmo basado en máquinas de soporte vectorial para la clasificación de las ontologías.
3. Diseñar e implementar una aplicación web para la manipulación, clasificación y visualización de las ontologías.
4. Diseñar e implementar un buscador básico de ontologías para comparar los resultados de búsqueda en un repositorio local simple y en un repositorio clasificado.

---

<sup>2</sup>Clases, propiedades, datos, instancias

# Introducción

Actualmente, la web es uno de los sistemas de acceso a información distribuida más usado en el mundo. Sin embargo, la información disponible se encuentra dispersa y ubicua (Ver figura 1).

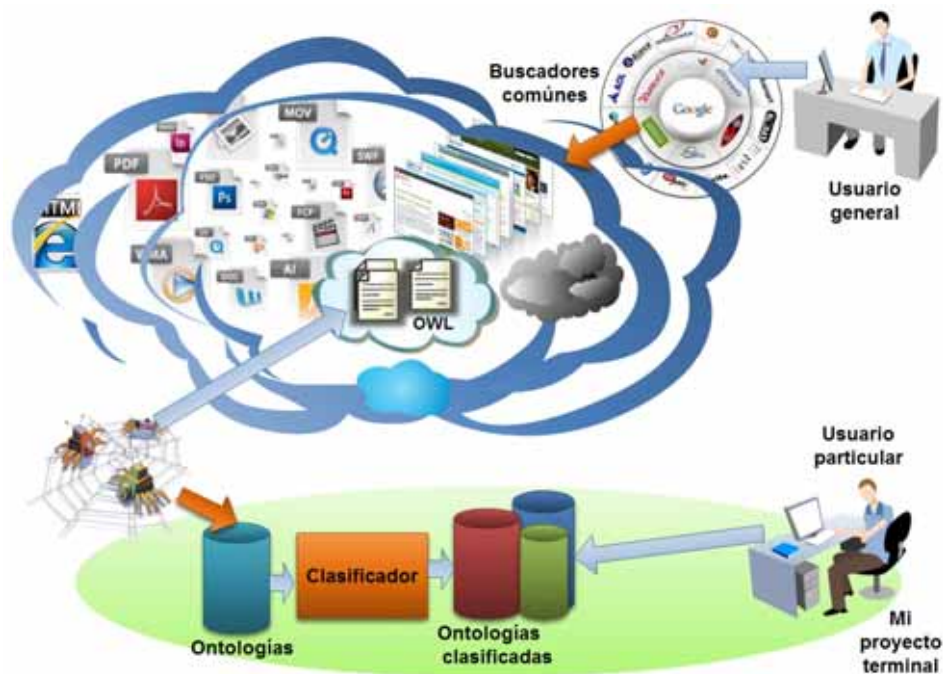


Figura 1: La información y el sistema clasificador de ontologías.

Por lo que es necesario integrarlos, ordenarlos y agruparlos de alguna manera ya sea en base a un dominio específico, tema, idioma, etc.

Para que la información pueda ser interpretada de manera automática, debe existir una representación común y formal de ésta. Las ontologías son un conjunto de vocabularios de conceptos y relaciones suficientemente enriquecidos que permiten representar el conocimiento y la información relativa a cualquier dominio de manera formal, además son legibles y reutilizables por programas informáticos.

La clasificación de ontologías es la tarea de identificar las características principales de un conjunto dado de ontologías y aplicar un algoritmo que permita discriminar entre ese conjunto de características para determinar la pertenencia de cada ontología a una clase en particular. Las clases resultantes de la ejecución de la clasificación pueden definirse como dominios de aplicación a las que pueden pertenecer o no las ontologías del conjunto evaluado.

Las máquinas de soporte vectorial son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T. Estos métodos están propiamente relacionados con problemas de clasificación y regresión<sup>3</sup>.

La utilización de un algoritmo basados en SVM permitirá una mejor interpretación de la información representada en una ontología, descubrir relaciones y agrupar las ontologías en clases de acuerdo a un criterio establecido. La ventaja de usar SVM respecto a redes neuronales es que el entrenamiento es menos costoso, tiene un alto rendimiento y funciona muy bien para problemas específicos mencionados: los problemas de clasificación y regresión. A pesar de que las SVM solo pueden tomar valores de uno y cero en su forma básica, en su extensión a multiclase soporta la asignación de etiquetas[3], donde las etiquetas son extraídas de un conjunto finito de elementos, permitiendo clasificar las ontologías en mas de dos categorías.

---

<sup>3</sup>Tipo de pruebas de software que intentan descubrir las causa de nuevos errores, carencias de funcionalidad o divergencias funcionales con respecto al comportamiento esperado del software



# Desarrollo del proyecto

## Arquitectura del sistema

Se desarrolló un sistema capaz de leer un conjunto de ontologías a partir de un repositorio y clasificarlos en base a un algoritmo basado en máquinas de soporte vectorial.

Este proyecto está conformado por los siguientes módulos (Ver figura 2).

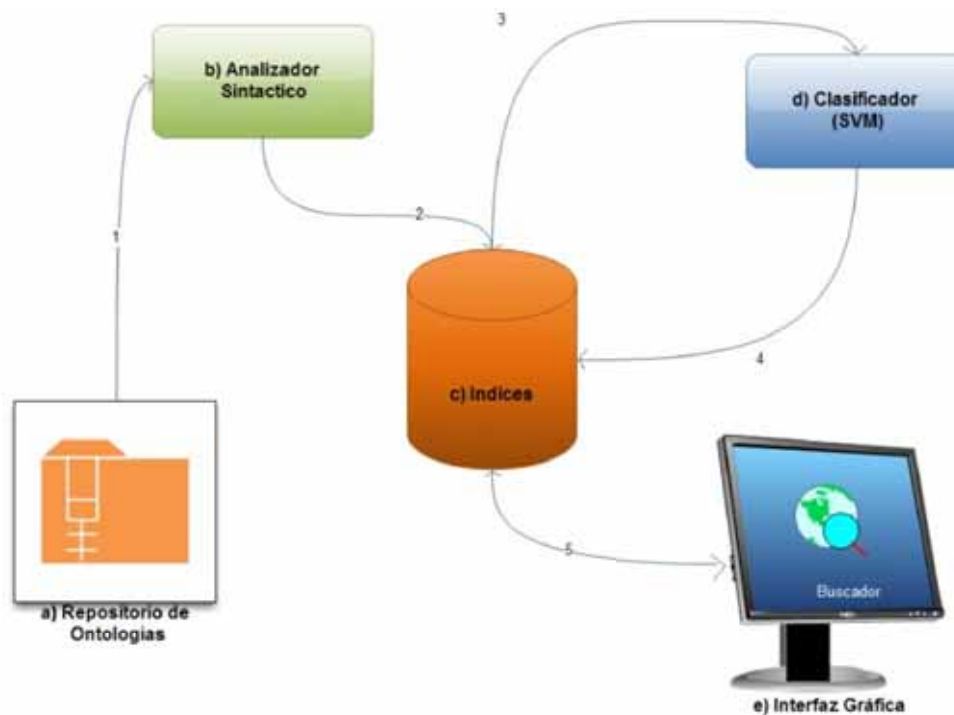


Figura 2: Sistema clasificador de ontologías

### a) Repositorio de Ontologías:

Este directorio contiene las ontologías preclasificadas y las ontologías a clasificar.

### **b) Analizador Sintáctico::**

Este módulo se encarga de analizar e identificar los componentes principales de cada ontología y almanenarlos en el Índice.

### **c) Índices:**

Para facilitar el procesamiento y la optimización de la memoria, este módulo se encarga de almacenar todos los datos de cada ontología y su contenido en una Base de Datos.

### **d) Clasificador:**

Este módulo se encarga de procesar cada característica de las ontologías, se creo este modulo que se encargara de hacer las comparaciones y posteriormente usando SVM clasificar las ontologías para finalmente actualizar el Índice.

### **e) Interfaz gráfica:**

Este módulo se encarga de proporcionar una interfaz por la cual se puedan controlar el sistema, incluyendo el buscador.

La interacción entre los módulos:

- 1- El analizador sintáctico extrae y procesa la colección de ontologías.
- 2 - El analizador sintáctico almacena los datos extraídos en las ontologías y los almacena en el Índice.
- 3 - Las SVM procesa y compara las ontologías para obtener características que permitan hacer su clasificación.
- 4 - Una vez hecha la clasificación por las SVM actualiza la Base de datos, asignando a cada ontología su nueva respectiva clase.
- 5 - En base a un conjunto de eventos se solicita realizar una operación ya sea de clasificación o de búsqueda de manera que el sistema pueda generar una consulta y recuperar ontologías relevantes.

# Implementación

## Tecnologías utilizadas

Se trabajó en un entorno desarrollado de Windows 8 posteriormente se adaptó e implementó en Ubuntu 13.10 para finalmente montarlo en un servidor con CentOS 5.

El sistema se desarrolló en el IDE<sup>4</sup>: Netbeans 7.4 y con el lenguaje de programación: Java. Adicionalmente se utilizaron las siguientes herramientas adicionales:

### **Hibernate:**

Hibernate <sup>5</sup> es una herramienta de Mapeo objeto-relacional que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos XML o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

### **OWL API**

OWL API <sup>6</sup> es una biblioteca para la manipulación de ontologías e incluye un analizador sintáctico para RDFS Y OWL .

### **Apache Tomcat**

Apache Tomcat es un servicio web que funciona como un contenedor de servlets y de Java Server desarrollado por Apache. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

### **MySQL**

MySQL<sup>7</sup> es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. desarrolla MySQL como software libre en un esquema de licenciamiento dual.

---

<sup>4</sup>IDE: Entorno de Desarrollado Integrado en español

<sup>5</sup>Hibernate: <http://hibernate.org/>

<sup>6</sup>OWL API: <http://owlapi.sourceforge.net/>

<sup>7</sup>MySQL: <http://www.mysql.com/>

### **Java Server Pages (JSP):**

JSP <sup>8</sup> es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML entre otros tipos de documentos. JSP es similar a PHP pero usa el lenguaje de programación Java.

### **Java Server Faces (JSF):**

JSF <sup>9</sup> es una tecnología y Framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario).

### **PrimeFaces 4:**

PrimeFaces <sup>10</sup> es un componente para JSF de código abierto que cuenta con un conjunto de componentes enriquecidos como paneles, gráficos, paneles entre otros, que facilitan la creación de las aplicaciones web. Tiene soporte para la tecnología Ajax que permite un despliegue parcial el cual permite controlar cuales componentes de la página actual se actualizarán.

---

<sup>8</sup>JSP: <http://www.oracle.com/technetwork/java/jsp-138432.html>

<sup>9</sup>JSF: <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

<sup>10</sup>Primefaces: <http://primefaces.org/>

## Implementación de los Módulos

### a) Repositorio de Ontologías

Se definieron manualmente dos directorios en el directorio local del usuario, una para las ontologías ya clasificadas, dentro del idioma español en este caso (ver Figura 3).

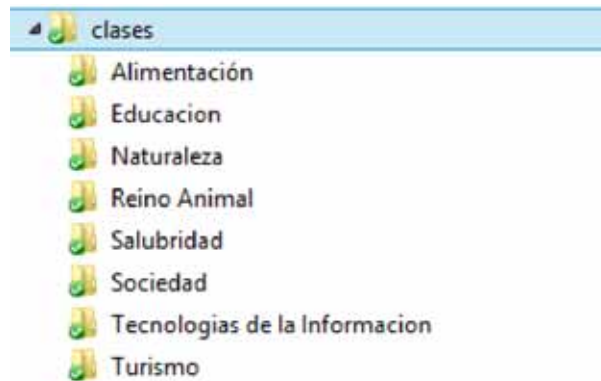


Figura 3: Directorio de ontologías preclasificadas

Y otro directorio (ver Figura 4) para las ontologías a clasificar también dentro del idioma definido manualmente:

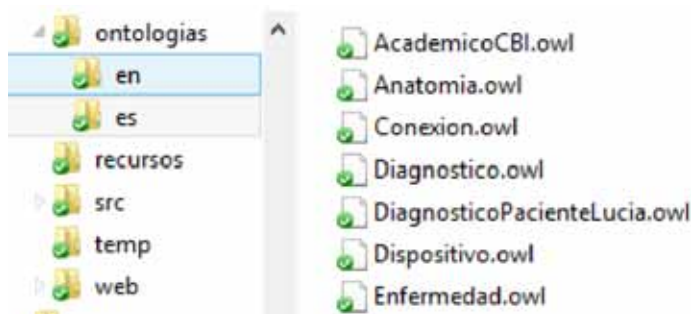


Figura 4: Directorio de ontologías a clasificar

## b) Analizador Sintáctico:

Utilizando la librería OWL API se procesaron e identificaron los componentes de las ontologías. Los componentes<sup>11</sup> a extraídos son:

1. *Class (Class)*
2. *ObjectProperty (OP)*
3. *Data Property (DP)*
4. *Individuals (Ind)*

Cada palabra fue etiquetada individualmente en base al componente al que corresponde para poder ser almacenados en el Índice.

## c) Índice:

Este modulo es una Base de Datos que incluye las siguientes tablas(ver Figura 5):

Donde la tabla Categorías contiene de manera única cada nombre de categoría, su identificador y una descripción. La tabla Ontologías contiene todos los datos sobre una ontología como su ubicación absoluta, el tipo<sup>12</sup>, su clase y otros campos necesarios para su procesamiento. La tabla Palabras contiene cada palabra existente en una ontología de manera única y campos adicionales como el identificador único y el idioma.

La tabla Índices contiene la relación entre una palabra y su relación con las ontologías el tipo de componente a la que pertenece, su frecuencia y otros datos necesarios.

---

<sup>11</sup>OWL: <http://www.w3.org/TR/owl-ref/>

<sup>12</sup>El tipo se refiere a si es una ontología ya preclasificada o es una nueva ontología por clasificar

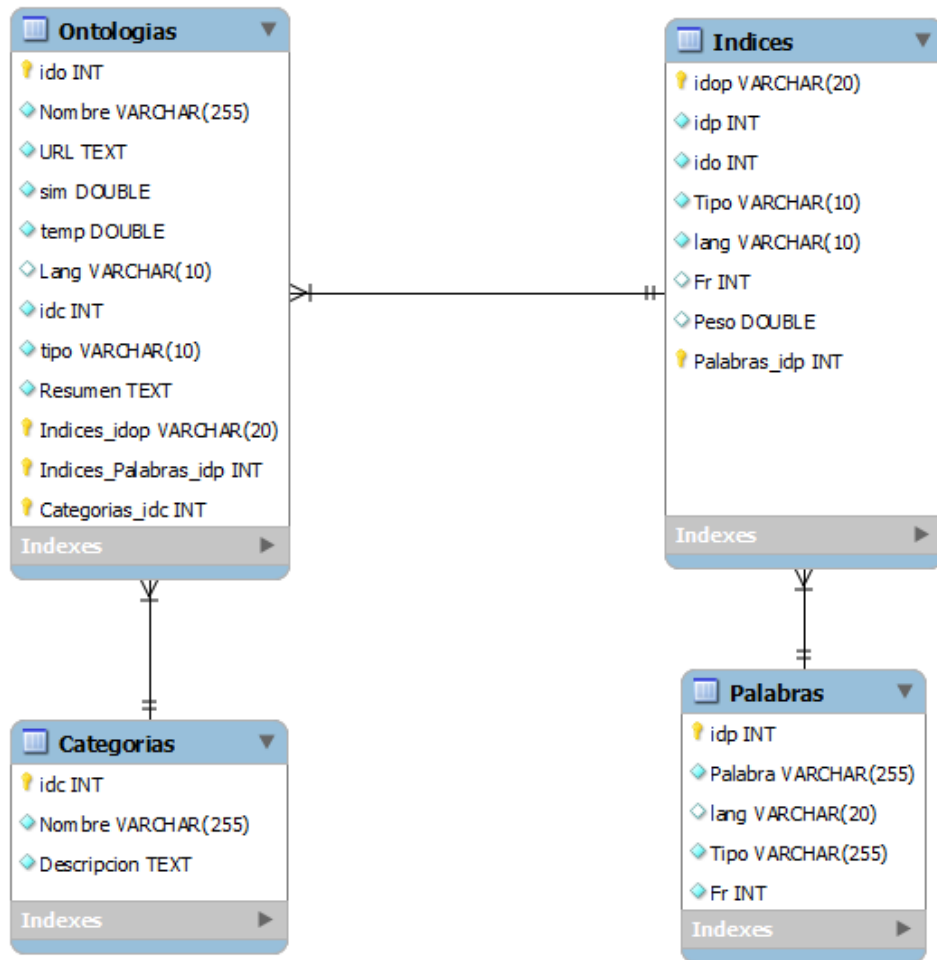


Figura 5: Diagrama Entidad - Relación

#### d) Clasificador

El clasificador se compone de dos partes: El comparador de ontologías que es básicamente implementar procedimientos capaces de comparar una ontología con otro y así devolver una similitud que se utilizará para representar las ontologías como vector. Y el procesamiento de las Maquinas de Soporte Vectorial que se encarga de ejecutar la clasificación con las ontologías representada en vectores.

#### Comparador de Ontologías:

Suponiendo que se tiene dos ontologías  $a$  y  $b$ , donde  $a$  se tiene que comparar con

b. Se toma a una palabra como el *token*<sup>13</sup>, ya que si toma las letras de las palabras como *tokens* estas ya no tendrían algún significado. Por lo que es necesario implementar una fórmula de comparaciones entre dos palabras.

$$SM(p_1, p_2) = \max(0, \frac{\min(|p_1|, |p_2|) - ed(p_1, p_2)}{\min(|p_1|, |p_2|)}) \in [0, 1]$$

Donde  $p_1$  y  $p_2$  son las palabras a comparar y  $ed$  es la distancia Euclidiana entre las dos palabras.

Posteriormente es necesario hacer la comparación de palabras que en este caso sería un componente de la ontología  $a$  con el componente de la ontología  $b$  y así obtener una similitud entre la ontología  $a$  y la ontología  $b$ .

$$SMC(c_1, c_2) = \frac{1}{|c_1|} \sum_{c_i \in c_1} \max_{c_j \in c_2} SM(c_i, c_j)$$

Donde  $c_1$  y  $c_2$  son los componentes o grupo de palabras a comparar. La información extraída es representada de manera vectorial para que las SVM puedan procesarlas:

### Maquinas de Soporte Vectorial (SVM):

Para representar las ontologías preclasificada en vectores para el entrenamiento de las SVM, fue necesario hacer la comparación de cada ontología con todas las de una misma clase de tal manera que el vector de cada ontología es representada como:

$$\begin{aligned} \vec{D}_a &= \{ \vec{C}_{1a}, \vec{C}_{2a}, \vec{C}_{3a}, \vec{C}_{4a}, \dots, \vec{C}_{na} \} \\ \vec{D}_b &= \{ \vec{C}_{1b}, \vec{C}_{2b}, \vec{C}_{3b}, \vec{C}_{4b}, \dots, \vec{C}_{nb} \} \\ \vec{D}_c &= \{ \vec{C}_{1c}, \vec{C}_{2c}, \vec{C}_{3c}, \vec{C}_{4c}, \dots, \vec{C}_{nc} \} \\ \vec{D}_d &= \{ \vec{C}_{1d}, \vec{C}_{2d}, \vec{C}_{3d}, \vec{C}_{4d}, \dots, \vec{C}_{nd} \} \\ &\vdots \\ \vec{D}_z &= \{ \vec{C}_{1z}, \vec{C}_{2z}, \vec{C}_{3z}, \vec{C}_{4z}, \dots, \vec{C}_{nz} \} \end{aligned}$$

Para facilitar el procesamiento dinámico estos vectores se representaron en archivos de texto planos de la siguiente manera:

$$C_i \ x : c1 \ y : c2 \ \dots \ z : cn$$

Donde  $x, y, z$  son las coordenadas,  $C_i$  la clase a la que pertenece y  $n$  el numero de clases.

---

<sup>13</sup>Un token es un elemento que no se puede dividir



Algo similar se hizo con las ontologías a clasificar:

$$Cp_i \ x : c1 \ y : c2 \ \dots \ z : cn$$

Donde  $Cp_i$  es cualquier clase preliminar que asignemos a las nuevas ontologías.

### e) Interfaz Gráfica

Utilizando PrimeFaces junto con CSS<sup>14</sup> se crearon las interfaces web.

#### Buscador:

El buscador de ontologías( ver Figura 6) recupera todas las clases existentes en la Base de datos y los muestra para filtrar los resultados por su categoría y el tipo de componente a la que pertenece dicha palabra clave.

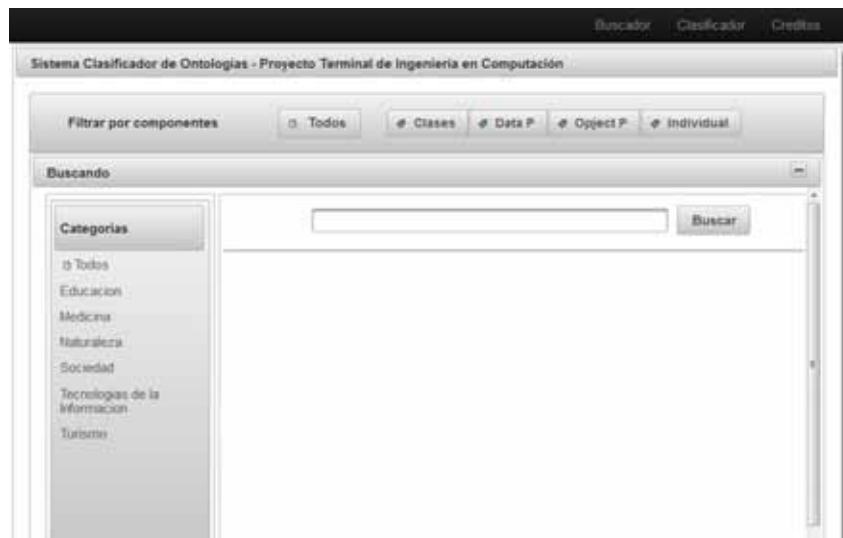


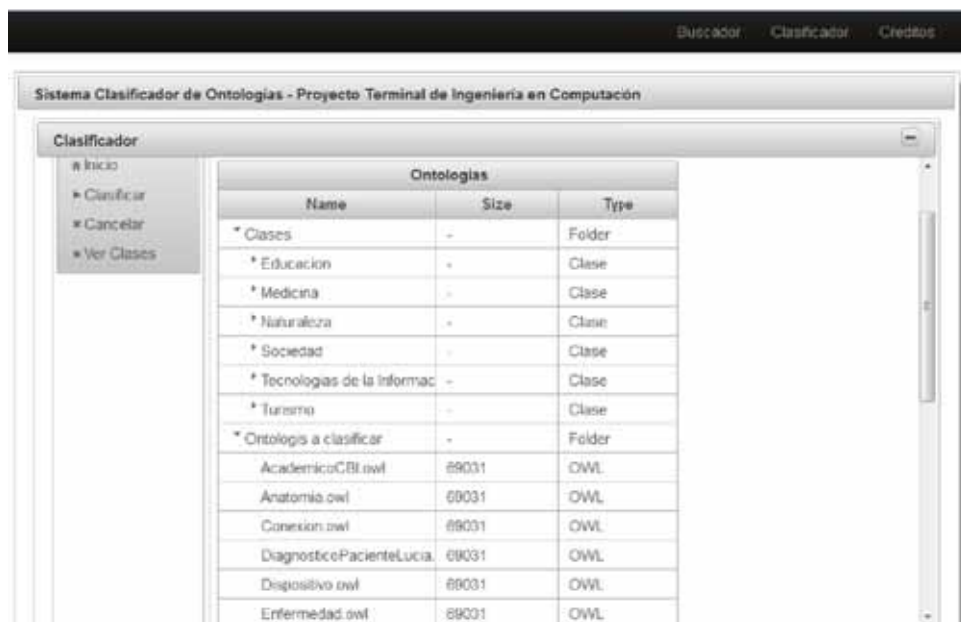
Figura 6: Interfaz Web - Buscador de ontologías

---

<sup>14</sup>CSS: Hojas de estilo en cascada con una amplia aplicación en dar estilo a páginas web

## Clasificador:

El clasificador (ver Figura 7) recupera y muestra las clases existentes, las ontologías que contiene cada clase y las nuevas ontologías a clasificar.



Ontologías		
Name	Size	Type
* Clases	-	Folder
* Educación	-	Clase
* Medicina	-	Clase
* Naturaleza	-	Clase
* Sociedad	-	Clase
* Tecnologías de la Informac	-	Clase
* Turismo	-	Clase
* Ontologías a clasificar	-	Folder
AcademicoCBI.owl	69031	OWL
Anatomia.owl	69031	OWL
Conexion.owl	69031	OWL
DiagnosticoPacienteLucia	69031	OWL
Dispositivo.owl	69031	OWL
Enfermedad.owl	69001	OWL

Figura 7: Interfaz Web - Clasificador de ontologías

El botón clasificar inicia la clasificación con las ontologías que muestra en la tabla. Primero realiza la indexación de las ontologías, es decir almacenarlo en la Base de datos para su rápido procesamiento. Posteriormente inicia los procesos de comparación entre ontologías generando el arreglo de textos plano listo para que las SVM las procese.

El botón ver clases muestra la clasificación ya existente en el Índice.

# Pruebas

## Recursos

El hardware que se utilizó es una computadora portátil con las siguientes características:

- 640 GB de disco duro.
- 4 GB de RAM.
- Procesador Intel Core i5 @ 2.67 GHz.
- Sistema operativo Windows 8.

## Demostración

### Buscador:

El buscador recibe como parámetros las palabras claves, formula las búsquedas y recupera las ontologías relevantes(ver Figura 8):



Figura 8: Buscador de ontologías

## Clasificador:

### Primer ejemplo:

Una vez establecido los directorios de clases y ontologías a clasificar al ejecutar la pestaña clasificar vemos la Figura 9:

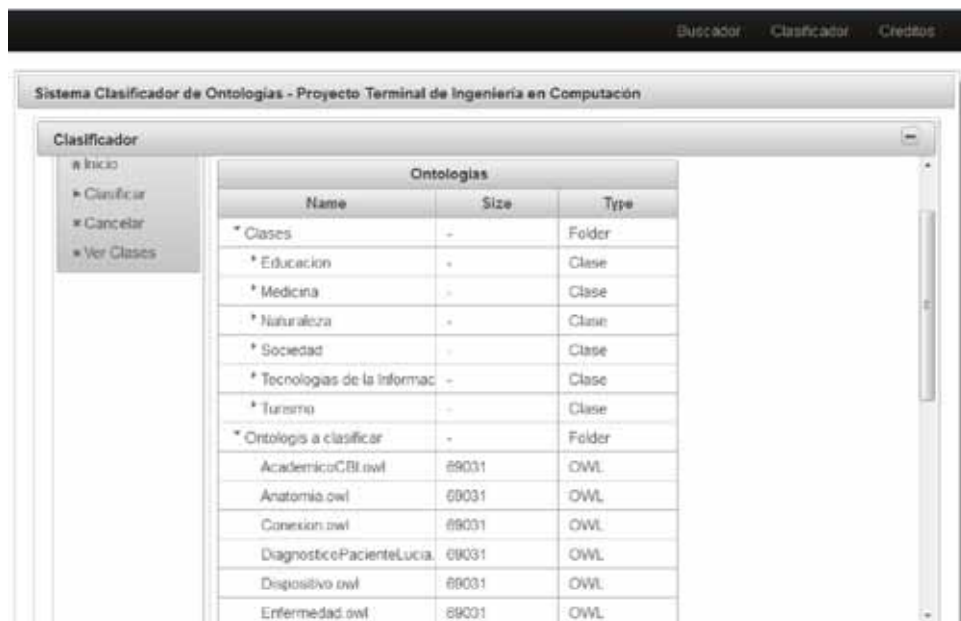


Figura 9: Ejemplo 1 - Clasificador de ontologías

Al presionar el botón de Clasificar se genera un evento que hace la petición para ejecutar la clasificación de las ontologías.

El archivo de vectores para las ontologías ya preclasificadas (Figura 10):

La representación de las nuevas ontologías a clasificar(Ver Figura 11)

Una vez terminado todo el proceso se refresca la pantalla mostrando las nuevas ontologías con su nueva clase (ver Figura 12).

Detalladamente podemos ver en el archivo de texto plano (ver Figura 13 ) Clasificación.out generado en la carpeta raíz de la cuenta del usuario todas las ontologías con su nueva respectiva clase.

Podemos ver dos detalles: AcademicoCBI.owl tiene datos de una persona normal si un enfoque hacia la educación por eso lo asigna a sociedad y Conexión.owl realmente por que no existe una ontología que contenga esos datos.

```

1 2 1:1.0 2:0.604821440080816 3:0.5734259362022082 4:0.6588690559069316 5:0.5904828106363614 6:0.5886824429702759
2 2 1:1.0 2:0.621856120389661 3:0.61472349354087 4:0.5710800181369524 5:0.5800622141844517 6:0.5692245903852824
3 2 1:1.0 2:0.6036375770966212 3:0.6010680205988903 4:0.6476747500269037 5:0.5845969154123675 6:0.5503620235543502
4 2 1:1.0 2:0.6002976295657648 3:0.5658870241962947 4:0.6152249552882453 5:0.5681006596520095 6:0.607928156432433
5 3 1:0.5537957702817723 2:1.0 3:0.6023350720905366 4:0.6030369394855981 5:0.5224958615643638 6:0.563176232333066
6 3 1:0.5371742182307779 2:1.0 3:0.5604341548990018 4:0.5614295576059694 5:0.5313216006314313 6:0.533572532062787
7 3 1:0.5317338640578748 2:1.0 3:0.588919703155681 4:0.5567437899653197 5:0.545889960785729 6:0.5353957985150123
8 3 1:0.5372778015567902 2:1.0 3:0.6000178559346602 4:0.5741493684497235 5:0.5288589013056055 6:0.5599964391043846
9 3 1:0.5374700612616989 2:1.0 3:0.591867132850413 4:0.550822247592908 5:0.539603933619783 6:0.5354784445942573
10 4 1:0.6214835226538797 2:0.58747253446381847 3:1.0 4:0.5794444519739884 5:0.5382229617338914 6:0.5924053797355064
11 4 1:0.6228355044668837 2:0.5732774280488979 3:1.0 4:0.5973322596520452 5:0.5240800985078818 6:0.5843560724366795
12 4 1:0.6152364472568167 2:0.6002728262295326 3:1.0 4:0.567013897622625 5:0.5203042393550277 6:0.6232721619007995
13 4 1:0.6306369637831663 2:0.590458342316576 3:1.0 4:0.5646011491616567 5:0.5271571118569714 6:0.5673484030442361
14 4 1:0.5089173468778718 2:0.5843665169730823 3:1.0 4:0.5714660174461788 5:0.4792902132280228 6:0.6620320534931039
15 4 1:0.6530990571847983 2:0.6272542398004812 3:1.0 4:0.5886854960437798 5:0.5854389466537913 6:0.5586653463941747
16 4 1:0.60827426353385733 2:0.615612098819589 3:1.0 4:0.62373342922229905 5:0.5559765509374906 6:0.653114816097364
17 5 1:0.5673440261118448 2:0.6132567782900227 3:0.5951573403913884 4:1.0 5:0.5279985423853149 6:0.594112778775068
18 5 1:0.5718065092352783 2:0.6373060169286627 3:0.6452192092428402 4:1.0 5:0.59470262231466996 6:0.6416936711389193
19 5 1:0.49772829939582588 2:0.5491124554475804 3:0.5544368785582168 4:1.0 5:0.51708394990866 6:0.5577915215721497
20 5 1:0.5267651972885711 2:0.5796319200047131 3:0.6165982576279805 4:1.0 5:0.512985782659859 6:0.6421250775970262
21 5 1:0.652952389717102 2:0.6787619125843048 3:0.6098412781953813 4:1.0 5:0.5762857222557067 6:0.5940952491760254
22 5 1:0.638403803736828 2:0.6354308554104396 3:0.6336734827075686 4:1.0 5:0.5224489910261971 6:0.6685374272721154
23 5 1:0.6090659348544079 2:0.6517399366085346 3:0.5942300779422173 4:1.0 5:0.525641028657674 6:0.614372568879113
24 5 1:0.5473545044660568 2:0.6272486278765924 3:0.6750000102652444 4:1.0 5:0.578835982103072 6:0.6564814903665887
25 5 1:0.6644444525241852 2:0.6998811204326153 3:0.5691765964031219 4:1.0 5:0.5722718313336372 6:0.6212600099092721
26 5 1:0.7334559948452971 2:0.5754527406259017 3:0.5967127397927371 4:1.0 5:0.668705238033627 6:0.596582718220624
27 5 1:0.5468750130389516 2:0.6927827516870687 3:0.576140882447362 4:1.0 5:0.553075407631893 6:0.5512648923322558
28 5 1:0.6080808147851522 2:0.6070541095165975 3:0.6021424432595571 4:1.0 5:0.5912887424497101 6:0.549220101227236
29 5 1:0.4988991341763927 2:0.5895459293934607 3:0.5353395792265092 4:1.0 5:0.5239695435570132 6:0.601574594936063
30 5 1:0.52738098199501 2:0.5959656139214834 3:0.5972222263614336 4:1.0 5:0.5092502634744181 6:0.5644841318129176
31 6 1:0.6004985834543521 2:0.613888888888300676 3:0.617109290491312 4:0.6408730242878963 5:1.0 6:0.6092796188134414
32 6 1:0.99990006739443 2:0.6256607923071479 3:0.655331240130525 4:0.602453237160467 5:1.0 6:0.59300093979652927
33 7 1:0.530400496693625 2:0.6405445490277332 3:0.6392438966726911 4:0.592655564228373 5:0.5979013810123222 6:1.0
34 7 1:0.6259945808332178 2:0.5821451485719862 3:0.6072131893302821 4:0.6519037654882744 5:0.5569688821140724 6:1.0
35 7 1:0.5115079447627068 2:0.6367889530165991 3:0.6131349265575409 4:0.6407836617732048 5:0.5327579450554585 6:1.0
36 7 1:0.6222989901206387 2:0.6217592706282934 3:0.5966882787611749 4:0.6496693193912506 5:0.6285493867133069 6:1.0
37 7 1:0.490746409182965 2:0.5361263823623041 3:0.5381257723157222 4:0.5382860275701835 5:0.49545559301399267 6:1.0
38 7 1:0.576517865806818 2:0.60521830559624 3:0.5906944543123245 4:0.6189801047606468 5:0.50659722946507822 6:1.0
39 7 1:0.513191903621082 2:0.5467847051528794 3:0.6048382287606214 4:0.6228785186241834 5:0.570563706076932 6:1.0
40 7 1:0.5923355735262524 2:0.615494237027385 3:0.601348314779527 4:0.5998061065646735 5:0.597902024299448 6:1.0

```

Figura 10: Ejemplo 1 - Ontologías preclasificadas

## Segundo ejemplo:

La interfaz web de clases y ontologías (ver Figura 14):

Después de ejecutar la clasificación podemos ver el archivo de vectores para las ontologías ya preclasificadas (ver Figura 15):

La representación de las nuevas ontologías a clasificar (ver Figura 16)

Las nuevas ontologías con su nueva clase (Ver Figura 17).

El archivo de texto plano (ver Figura 18 ) Clasificación.out generado en la carpeta raíz de la cuenta del usuario, contiene los nombres de las ontologías y la clase asignada

```

1 1:0.4095065442958102 2:0.5758261499999153 3:0.588008978833229 4:0.467902159888016 5:0.5739941784818998 6:0.48681379919466766
2 1:0.5181235710584201 2:0.793277559172095 3:0.4050746392502418 4:0.41157893245488949 5:0.5340096297779277 6:0.47289483029805494
3 1:0.31644664828095913 2:0.48074404887855053 3:0.3875060011175871 4:0.4906235032154064 5:0.34270833618938923 6:0.39027777973297964
4 1:0.5339740332941314 2:0.5428311800954726 3:0.5294805288314819 4:0.6019350755214691 5:0.6255454609394673 6:0.5435238218307495
5 1:0.5005036708417148 2:0.5924270946073923 3:0.5314284474684297 4:0.494049387793643 5:0.4751445231049479 6:0.45842332814471035
6 1:0.7111111182981233 2:0.4789441348033859 3:0.7023809891929119 4:0.4992062372009405 5:0.4237936684475067 6:0.4184444793810064
7 1:0.5259000059404445 2:0.6041446766007742 3:0.4288133422740301 4:0.5730156885320828 5:0.57023810346802129 6:0.5644444676600774
8 1:0.41111111640930176 2:0.588889034588132 3:0.41111111640930176 4:0.588889034588132 5:0.488888994852702 6:0.522222230611782
9 1:0.48247067695078644 2:0.426328512378361 3:0.4347481116009432 4:0.7425987538206636 5:0.572877893091823 6:0.6128344490426103
10 1:0.4879948151442051 2:0.4887400855494826 3:0.543482294872139 4:0.5227704896105744 5:0.4902173601090908 6:0.47489194172141
11 1:0.414761919158029556 2:0.6044444561004838 3:0.53777778444647436 4:0.431111114025116 5:0.5783323380951136 6:0.585277888311615
12 1:0.525827721299184 2:0.458420066090328 3:0.5448478617388348 4:0.554207895533376 5:0.5037862942821307 6:0.5179400348901344
13 1:0.694444414811039 2:0.55704363470681 3:0.570436314914038 4:0.505209399421503 5:0.4200396940112114 6:0.958333338168602

```

Figura 11: Ejemplo 1 - Representación de ontologías a clasificar

Sistema Clasificador de Ontologías - Proyecto Terminal de Ingeniería en Computación

Clasificador

- Clasificar
- Cancelar
- Ver Clases

Name	Size	Type
* Ontologías	-	Folder
* Educación	-	Clase
* Medicina	-	Clase
Anatomia.owl	es	new
Conexion.owl	es	new
Enfermedad.owl	es	new
Humano.owl	es	new
Medicamento.owl	es	new
Procedimientos_Medicina.owl	es	new
* Naturaleza	-	Clase
Estado_Clima.owl	es	new
* Sociedad	-	Clase
AcademicoCBI.owl	es	new
La_Musica.owl	es	new
Persona.owl	es	new

Figura 12: Ejemplo 1 - Resultados

```

1 AcademicoCBI.owl:Sociedad
2 Anatomia.owl:Medicina
3 Conexion.owl:Medicina
4 Dispositivo.owl:Tecnologias de la Informacion
5 Enfermedad.owl:Medicina
6 Espacio_Fisico.owl:Educacion
7 Estado_Clima.owl:Naturaleza
8 Humano.owl:Medicina
9 La_Musica.owl:Sociedad
10 Medicamento.owl:Medicina
11 Persona.owl:Sociedad
12 Procedimientos_Medicos.owl:Medicina
13 Transporte.owl:Turismo
14

```

Figura 13: Ejemplo 1 - Archivo de texto

Sistema Clasificador de Ontologías - Proyecto Terminal de Ingeniería en Computación

Clasificador

Medicina	-	Clase
Naturaleza	-	Clase
Sociedad	-	Clase
Tecnologías de la Informac	-	Clase
Turismo	-	Clase
Ontologías a clasificar	-	Folder
Anatomia.owl	69031	OWL
Enfermedad.owl	69031	OWL
Estado_Clima.owl	89031	OWL
Humano.owl	69031	OWL
La_Musica.owl	69031	OWL
Medicamento.owl	69031	OWL
Persona.owl	69031	OWL
Procedimientos_Medicos	69031	OWL
Transporte.owl	69031	OWL
Tratamiento.owl	69031	OWL

Figura 14: Ejemplo 2 - Clasificador de ontologías

```

1 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
2 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
3 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
4 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
5 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
6 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
7 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
8 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
9 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
10 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
11 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
12 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
13 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
14 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
15 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
16 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
17 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
18 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
19 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888
20 1:0.410021440000018 2:0.574493122222222 3:0.888888888888888 4:0.308821188888888 5:0.888888888888888

```

Figura 15: Ejemplo 2- Ontologías preclasificadas

```

1 1:0.5153239710584201 2:0.793277559372095 3:0.6050366392502418 4:0.611575324548969 5:0.5360098297779377 6:0.47295483
2 1:0.5005036708817148 2:0.5924270946073923 3:0.5316264676654297 4:0.4940698837793643 5:0.4751465231049479 6:0.4584235
3 1:0.5250000055604645 2:0.6041666766007742 3:0.6208333422740301 4:0.5730158885320028 5:0.5702381034692129 6:0.56666666
4 1:0.41111111440930176 2:0.58888889034589132 3:0.41111111440930174 4:0.58888889034589132 5:0.48888889482702 6:0.52222
5 1:0.48247667695078644 2:0.626328512378361 3:0.6347481114604612 4:0.742598353206836 5:0.572877859291823 6:0.61283644
6 1:0.4578964151442051 2:0.6587400855496526 3:0.536382296872139 4:0.5227704998105764 5:0.490217361090908 6:0.47488104
7 1:0.4147619195029556 2:0.6044444461004638 3:0.5377777844467435 4:0.651111114025116 5:0.5783333390951156 6:0.58527779
8 1:0.5255827721299194 2:0.6568620046090328 3:0.5449679817368345 4:0.554307895535376 5:0.5037892942821107 6:0.51794001
9 1:0.4944444414811039 2:0.557043656704681 3:0.5704365149140358 4:0.5052083395421505 5:0.4200396940112114 6:0.95833333

```

Figura 16: Ejemplo 2 - Representación de ontologías a clasificar

Sistema Clasificador de Ontologías - Proyecto Terminal de Ingeniería en Computación

**Clasificador**

- Clasificar
- Cancelar
- Ver Clases

Name	Size	Type
• Ontologías	-	Folder
• Educación	-	Clase
• Medicina	-	Clase
Anatomia.owl	es	new
Conexion.owl	es	new
Enfermedad.owl	es	new
Humano.owl	es	new
Medicamento.owl	es	new
Procedimientos_Medic	es	new
• Naturaleza	-	Clase
Estado_Clima.owl	es	new
• Sociedad	-	Clase
AcademicoCBI.owl	es	new
La_Musica.owl	es	new
Persona.owl	es	new

Figura 17: Ejemplo 2 - Resultados



```
1 Anatomia.owl:Medicina
2 Enfermedad.owl:Medicina
3 Estado_Clima.owl:Naturaleza
4 Humano.owl:Medicina
5 La_Musica.owl:Sociedad
6 Medicamento.owl:Medicina
7 Persona.owl:Sociedad
8 Procedimientos_Medicos.owl:Medicina
9 Transporte.owl:Turismo
10 Tratamiento.owl:Medicina
11
```

Figura 18: Ejemplo 2 - Archivo de texto

# Conclusiones

La clasificación de ontologías y cualquier tipo de documento es una labor complicada incluso para los humanos. Hay cosas que para un humano es algo trivial pero para una máquina no, por ejemplo si se está hablando de la UAM un estudiante asumiría que se trata de nuestra universidad, sin embargo la máquina podría confundirlo con cualquier otra inicial similar o incluso con las iniciales de la Universidad Autónoma de Madrid.

Este proyecto terminal da una noción de lo difícil que es procesar la información y también que la clasificación es una actividad necesaria día a día por que permite tener una mejor administración de la información. Pero a su vez puede ser complicado y variado, por lo que es de gran ayuda mejorar cada vez la automatización.

Este proyecto paso por varias versiones hasta llegar a al actual, mejorando cada vez la forma de hacer las comparaciones y también se implementó la comparación IDF<sup>15</sup>. Sin embargo, por el tamaño de procesamiento y el consumo de memoria, el equipo tardaba muchos minutos para procesar grandes matrices de palabras.

---

<sup>15</sup>Frecuencia inversa de documento, es decir la frecuencia de ocurrencia del termino en la colección de documento o clase.

# Trabajos Futuros

Existen muchas mejoras que se pueden hacer al proyecto como optimizar las comparaciones entre las ontologías, implementar el método de IDF <sup>16</sup> u algún otro para obtener la similitud entre las ontologías.

Otro aspecto importante a trabajar es el acceso a disco por lo que se puede mejorar el índice de la Base de Datos aplicando algún tipo ordenamiento e implementar algún algoritmo de búsqueda como la búsqueda binaria u algún otro similar.

También se podría trabajar en la detención automática de idiomas e implementar Wordnet en todos los idiomas legibles y aceptables por el sistema.

---

<sup>16</sup>IDF: <http://es.wikipedia.org/wiki/Tf-idf>

# Apéndice

Script SQL de la Base de Datos para el módulo Índice.

```
1 CREATE TABLE IF NOT EXISTS Palabras(  
2     idp INT NOT NULL AUTO_INCREMENT,  
3     Palabra VARCHAR(255) NOT NULL ,  
4     lang VARCHAR(20),  
5     Tipo VARCHAR(255) NOT NULL,  
6     Fr int NOT NULL,  
7     PRIMARY KEY (idp)  
8     ENGINE=InnoDB;  
9  
10 CREATE TABLE IF NOT EXISTS Ontologias(  
11     idc INT NOT NULL AUTO_INCREMENT,  
12     Nombre varchar(255) NOT NULL UNIQUE,  
13     URL TEXT NOT NULL,  
14     sim Double NOT NULL,  
15     temp Double NOT NULL,  
16     Lang varchar(10),  
17     idc INT NOT NULL,  
18     tipo VARCHAR(10) NOT NULL,  
19     Resumen TEXT NOT NULL,  
20     PRIMARY KEY (idc)  
21     ENGINE=InnoDB;  
22 CREATE TABLE IF NOT EXISTS Indices(  
23     idop VARCHAR(20) NOT NULL UNIQUE,  
24     idp INT NOT NULL,  
25     idc INT NOT NULL,  
26     Tipo VARCHAR(10) NOT NULL,  
27     lang varchar(10) NOT NULL,  
28     Fr int,  
29     Peso Double,  
30     PRIMARY KEY(idop)  
31     ENGINE=InnoDB;  
32 CREATE TABLE IF NOT EXISTS Categorias(  
33     idc INT NOT NULL AUTO_INCREMENT,  
34     Nombre VARCHAR(255) NOT NULL UNIQUE,  
35     Descripcion TEXT NOT NULL,  
36     PRIMARY KEY (idc)  
37     ENGINE=InnoDB;  
38 CREATE TABLE IF NOT EXISTS Opciones(  
39     Opcion VARCHAR(50) NOT NULL UNIQUE,  
40     Valor TEXT,  
41     PRIMARY KEY (Opcion)  
42     ENGINE=InnoDB;
```

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package com.gasiem.pt.clasificador;
6
7  import com.gasiem.pt.clases.cClase;
8  import com.gasiem.pt.clases.cOntologia;
9  import com.gasiem.pt.clasificador.comparar.CompOntologias;
10 import com.gasiem.pt.hibernate.clases.Categorias;
11 import com.gasiem.pt.hibernate.clases.Ontologias;
12 import com.gasiem.pt.hibernate.controlador.ctCategorias;
13 import com.gasiem.pt.hibernate.controlador.ctOntologias;
14 import com.gasiem.pt.svm.svm_predict;
15 import com.gasiem.pt.svm.svm_train;
16 import com.gasiem.pt.util.Auxiliar;
17 import java.io.BufferedOutputStream;
18 import java.io.DataOutputStream;
19 import java.io.FileOutputStream;
20 import java.io.IOException;
21 import java.io.Serializable;
22 import java.util.ArrayList;
23 import java.util.List;
24
25 /**
26  *
27  * @author GASIEM
28  */
29 public class Clasificador implements Serializable {
30
31     Auxiliar aux;
32     Ontologias ont = new Ontologias();
33     cOntologia co = new cOntologia();
34     ctOntologias cont = new ctOntologias();
35     ctCategorias ccat = new ctCategorias();
36     List<Categorias> lcat;
37     List<Ontologias> lontc = new ArrayList();
38     List<Ontologias> lonto = new ArrayList();
39     Integer idunclass = 1;
40     String mysql = "";
41     Double sim;
42     CompOntologias compo;
43     int nroclases = 0;
44     DataOutputStream fpc, fpt, fpo;
45     cClase cc;
46     cClase ccc;
47

```

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package com.gasiem.pt.clasificador;
6
7  import com.gasiem.pt.clases.cClase;
8  import com.gasiem.pt.clases.cOntologia;
9  import com.gasiem.pt.clasificador.comparar.CompOntologias;
10 import com.gasiem.pt.hibernate.clases.Categorias;
11 import com.gasiem.pt.hibernate.clases.Ontologias;
12 import com.gasiem.pt.hibernate.controlador.ctCategorias;
13 import com.gasiem.pt.hibernate.controlador.ctOntologias;
14 import com.gasiem.pt.svm.svm_predict;
15 import com.gasiem.pt.svm.svm_train;
16 import com.gasiem.pt.util.Auxiliar;
17 import java.io.BufferedOutputStream;
18 import java.io.DataOutputStream;
19 import java.io.FileOutputStream;
20 import java.io.IOException;
21 import java.io.Serializable;
22 import java.util.ArrayList;
23 import java.util.List;
24
25 /**
26  *
27  * @author GASIEM
28  */
29 public class Clasificador implements Serializable {
30
31     Auxiliar aux;
32     Ontologias ont = new Ontologias();
33     cOntologia co = new cOntologia();
34     ctOntologias cont = new ctOntologias();
35     ctCategorias ccat = new ctCategorias();
36     List<Categorias> lcat;
37     List<Ontologias> lontc = new ArrayList();
38     List<Ontologias> lonto = new ArrayList();
39     Integer idunclass = 1;
40     String mysql = "";
41     Double sim;
42     CompOntologias compo;
43     int nroclases = 0;
44     DataOutputStream fpc, fpt, fpo;
45     cClase cc;
46     cClase ccc;
47

```

```

48 public Clasificador(Auxiliar aux) {
49     this.aux = aux;
50     compo = new CompOntologias(aux);
51     cc = new cClase(aux);
52     ccc = new cClase(aux);
53 }
54
55 public void Clasificar() throws IOException {
56     aux.addInfo("El proceso de clasificacion ha iniciado");
57     ObtenerCategorias();
58     aux.addInfo("Recuperando ontologias");
59     Clases();
60     Ontologias();
61     aux.addInfo("Procesando Clases ontologias");
62     ProcesarClases();
63     aux.addInfo("Procesando Nuevas Ontologias");
64     ProcesarOntologias();
65     aux.addInfo("Ejecutando las SVM");
66     EjecutarSVM();
67     aux.addInfo("Finalizando proceso...");
68     ActualizarOntologias();
69 }
70
71
72 public void ActualizarOntologias() {
73     List<Ontologias> listo = cont.ListaOntologias("FROM Ontologias where tipo='new'");
74     List<String> lista = aux.RecuperarClaseOntologias(aux.getDir()+"svm.out");
75     try {
76         fpo = new DataOutputStream(new BufferedOutputStream(
77             new FileOutputStream(aux.getDirout() + "Clasificacion.out")));
78         int i = 0;
79         for (Ontologias on : listo) {
80             String nro = i <- lista.size() ? lista.get(i) : "0";
81             nro = nro.replaceAll(".0", "");
82             on.setIdc(Integer.parseInt(nro));
83             i++;
84             cont.ActualizarOntologia(on);
85             fpo.writeBytes(cont.ObtenerOntologia(
86                 on.getIdc()).getNombre() + ":" + ccat.ObtenerCategoria(
87                     on.getIdc()).getNombre() + "\n");
88         }
89
90         fpo.close();
91     } catch (IOException e) {
92         System.err.println(e);
93     }
94 }
95

```

```

96 public void ProcesarOntologias() {
97     String cadena;
98     int i, idc = 0;
99     Double max = 0.0;
100     try {
101         fpt = new DataOutputStream(new BufferedOutputStream(
102             new FileOutputStream(aux.getDir()+"svm.t")));
103         for (Ontologias o : lonto) {
104             co = co.ObtenerDatos(o.getIdo());
105             cadena = "";
106             i = 1;
107             for (Categorias myc : lcat) {
108                 if (!myc.getNombre().equalsIgnoreCase("new")) {
109                     cc = cc.ObtenerDatos(myc.getIdc());
110                     sim = compo.dCompara(co, cc);
111                     cadena += i + ":" + sim + " ";
112                     i++;
113                     if (sim > max) {
114                         max = sim;
115                         idc = myc.getIdc();
116                     }
117                 }
118             }
119             fpt.writeBytes(idc + " " + cadena + "\n");
120         }
121         fpt.close();
122     } catch (IOException e) {
123         System.err.println(e);
124     }
125 }
126
127 public void ProcesarClases() {
128     try {
129         fpc = new DataOutputStream(new BufferedOutputStream(
130             new FileOutputStream(aux.getDir()+"svm.dat")));
131         ProcesarClasesO();
132         fpc.close();
133     } catch (IOException e) {
134         System.err.println(e);
135     }
136 }
137

```



```

138 void ProcesarClasesC() throws IOException {
139     for (Categorias mycata : lcat) {
140         if (!mycata.getNombre().equalsIgnoreCase("new")) {
141             cc = cc.ObtenerDatos(mycata.getIdc());
142             fpc.writeBytes(mycata.getIdc() + " ");
143             int i = 1;
144             for (Categorias mycatb : lcat) {
145                 if (!mycatb.getNombre().equalsIgnoreCase("new")) {
146                     ccc = ccc.ObtenerDatos(mycatb.getIdc());
147                     sim = compo.dCompara(cc, ccc);
148                     fpc.writeBytes(i + ":" + sim + " ");
149                     i++;
150                 }
151             }
152             fpc.writeBytes("\n");
153         }
154     }
155 }
156
157 void ProcesarClasesO() throws IOException {
158     for (Ontologias oc : lontc) {
159         co = co.ObtenerDatos(oc.getIdc());
160         fpc.writeBytes(oc.getIdc() + " ");
161         int i = 1;
162         for (Categorias mycatb : lcat) {
163             if (!mycatb.getNombre().equalsIgnoreCase("new")) {
164                 ccc = ccc.ObtenerDatos(mycatb.getIdc());
165                 sim = compo.dCompara(co, ccc);
166                 fpc.writeBytes(i + ":" + sim + " ");
167                 i++;
168             }
169         }
170         fpc.writeBytes("\n");
171     }
172 }
173
174 public void Ontologias() {
175     Categorias micat = lcat.get(0);
176     if (micat.getNombre().equalsIgnoreCase("new")) {
177         this.idunclass = micat.getIdc();
178         mysql = "FROM Ontologias where tipo='new'";
179         lonte = cont.ListaOntologias(mysql);
180     } else {
181         aux.Mensaje("Ocurrio un error con la estructura de los directorios");
182     }
183 }
184

```

```

184
185 public void Clases() {
186     mysql = "FROM Ontologias where idcl=" + this.idunclass;
187     lontc = cont.ListaOntologias(mysql);
188 }
189
190 public void ObtenerCategorias() {
191     lcat = ccat.ListaCategorias("FROM Categorias");
192     this.nroclases = lcat.size() - 1;
193 }
194
195 void EjecutarSVM() throws IOException {
196     String[] param = {aux.getDir() + "svm.dat"};
197     svm_train.main(param);
198     String[] paramp = {aux.getDir() + "svm.t", aux.getDir() + "svm.dat.model",
199                       aux.getDir() + "svm.out"};
200     svm_predict.main(paramp);
201 }
202 }
203

```

# Bibliografía

- [1] E. Jiménez Ruiz, *Ontologías en Informática*, <http://krono.act.uji.es/publications/techrep/Book-Chapter-Protege-report2007.pdf>, enero del 2013
- [2] G. Colmenares, *Máquina de vectores de soporte*, [http://www.webdelprofesor.ula.ve/economia/gcolmen/programa/economia/maquinas\\_vectores\\_soporte.pdf](http://www.webdelprofesor.ula.ve/economia/gcolmen/programa/economia/maquinas_vectores_soporte.pdf), enero del 2013
- [3] E. Vlachos, *Active Learning with Support Vector Machines*, <http://www.inf.ed.ac.uk/publications/thesis/online/IM040138.pdf>, febrero 2013
- [4] E.S. Estrada, *Clasificación de servicios web semánticos mediante ontologías*, propuesta de Proyecto Terminal, Universidad , Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2012.
- [5] J. P. Martínez, *Extracción automatizada y representación de servicios Web mediante ontologías*, Proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2011.
- [6] L. A. Toral, *Modelo Multidimensional para la representación de Perfiles de Aprendizaje y Estilos de Pensamiento mediante Ontologías y Reglas de Inferencia*, propuesta de Proyecto terminal, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2011.
- [7] A.Z. Mendialdua, *Aproximaciones a SVM semisupervisado multiclase para clasificación de páginas web*, Universidad Nacional de Educación a Distancia, España, 2008.
- [8] F.M. Rangel, *Clasificación de páginas web en dominio específico*, tesis Master , Universidad Nacional de Educación a Distancia, España, 2007.
- [9] J. E Rodríguez, *Software para el filtrado de páginas web pornográficas basado en el clasificador KNN - UDWEBPORN*, Universidad Distrital "Francisco José de Caldas", Colombia, 2011.