Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

Reporte final del proyecto de integración:

Sistema Web para obtener la similitud entre publicaciones científicas mediante técnicas semánticas

Luis Enrique García García

Matrícula: 210332582

Trimestre: 2014-Otoño

Asesor

Dra. Maricela Claudia Bravo Contreras

Profesor Asociado “D”, Departamento de Sistemas

Co-asesor

Dr. José Alejandro Reyes Ortiz

Profesor Titular “A”, Departamento de Sistemas

Yo, Maricela Claudia Bravo Contreras, 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.

Firma del asesor

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.

Firma del asesor

Yo, Luis Enrique García García, 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.

Firma del alumno

Tabla de contenido

Resumen

El presente proyecto es un sistema web que determina la similitud entre dos texto como publicaciones científicas atraves de utilizar tres diferentes métricas (Tf-idf, frases y tripletas). El sistema consta de cuatro módulos descritos a continuación.

- Módulo extractor. Este módulo se encarga de conectarse, realizar consultas Insert, Update o Select y extraer textos desde la base de datos que serán utilizados para calcular la similitud con las tres métricas (tf-idf, frases, tripletas).
- Módulo métricas. Este módulo contiene las métricas Tf-idf, Frases y Tripletas para realizar la similitud. Para realizar el cálculo de Tf-idf se obtiene la frecuencia de cada palabra en los dos textos, después se realiza la división de la min. Frec entre la máx. Frec, al final la similitud se obtiene realizando el promedio de todas las divisiones. Para obtener la similitud con las métricas Frases y Tripletas, primero se realiza el etiquetado de los dos textos con lo cual se sabe sin las palabras son verbos, sustantivos, preposiciones etc. Ya obtenido este etiquetado se forman las Frases Nominales (FN), Frases Verbales (FV) y Frases Preposicionales (FP).
- En la métrica Frases se calcula la similitud de la Frase i del texto uno con su igual del texto dos y después se obtiene su promedio de las tres frases para obtener el valor de la similitud. Para realizar el cálculo de la métrica Tripletas se realiza un agrupación después de obtener las Frases, esta agrupación sigue la siguiente estructura FN FV (FN || FP) por tanto ahora el cálculo se realiza sobre las tripletas que genera cada texto.
- Módulo sistema web. Este módulo es el encargado de realizar la interfaz web entre el usuario y el sistema, realizando las llamadas a las clases necesarias para realizar el cálculo de similitud entre los textos seleccionados con cualquiera de las métricas, utilizando para ello páginas jsp´s, js y html.
- Módulo de visualización y enriquecimiento. Este módulo se encarga de mostrar de manera grafica los textos almacenados en la base de datos para seleccionar aquellos a los cuales se calculará la similitud, además presenta la similitud calculada por el sistema y también se realiza la inserción en la base de datos de los textos y la similitud encontrada con alguna de las métricas, y se muestra una gráfica de barras y una tabla de la similitud entre cada par de textos.

1.- Introducción

Algunas de las cosas que suele hacer una persona que va a publicar algún trabajo es tener referencias a otros temas relacionados con el suyo, esto implica, el tener que buscar dentro de la gran cantidad de textos científicos existentes, los apropiados para tal fin. También, debido a la extensión de Internet y su fácil acceso se pueden dar casos de plagios de documentos lo cual conlleva a pérdidas de diferente índole. Realizar esta búsqueda y detección de plagio de manera manual consume mucho tiempo.

La obtención automática de la similitud semántica en textos es un área del Procesamiento del Lenguaje Natural (PLN)que ha sido objeto de estudio por mucho tiempo, consiste en determinar un grado de relación o medida probabilística entre dos segmentos de texto, los cuales pueden ir desde una oración hasta un documento completo. Esta tarea se basa en métodos estadísticos, sintácticos o semánticos para encontrar el grado de similitud.

El proyecto a desarrollar será un sistema web el cual dados dos textos científicos descritos en lenguaje natural como entrada, aplicará algoritmos con técnicas de similitud semántica para determinar el grado de relación existe entre ellos con respecto a su contenido, de tal manera que se logre enriquecer una base de datos semántica con los resultados que superen cierto umbral el cual puede ser un porcentaje de similitud entre los textos.

2.- Justificación

El proceso de encontrar la similitud semántica entre dos textos es una tarea tediosa y costosa al realizarse manualmente, ya que existe una gran cantidad de documentos disponibles con los cuales se puede trabajar. Con el presente proyecto se pretende automatizar esta tarea con el cual se apoyará para consumir menos tiempo y se podrán revisar grandes cantidades de documentos.

La utilidad de este proyecto conlleva a muchos beneficios debido a que, al determinar el grado de similitud, se pueden clasificar textos científicos, detectar

plagio entre ellos, facilitar la localización de referencias a trabajos similares y generar redes temáticas.

Al tratarse de métodos de similitud semántica que involucrarán algoritmos con técnicas complejas de Procesamiento de Lenguaje Natural y que el sistema aportará una gran ayuda al realizar la similitud entre textos científicos resulta ser una herramienta novedosa en el área de la Ing. En Computación. Este proyecto se integrará con un proyecto de investigación que está siendo desarrollado en el Grupo de Investigación de Sistemas de Información Inteligentes.

3.- Objetivo general

Diseñar e implementar un sistema web para determinar el grado de similitud entre publicaciones científicas descritas en inglés con un formato libre (lenguaje natural), utilizando técnicas semánticas.

3.1.- Objetivos específicos

- Diseñar e implementar un módulo para extraer la información relevante de las publicaciones científicas a partir del modelo de datos.

- Diseñar e implementar módulo el cual utilice algoritmos de similitud semántica y considerar menos 3 distancias para obtener la similitud entre textos escritos en inglés.

- Diseñar e implementar un módulo de sistema web que integre las métricas para obtener la similitud semántica entre publicaciones científicas.

- Diseñar e Implementar un módulo para visualizar y enriquecer el modelo de datos mediante las similitudes encontradas.

4.- Antecedentes

4.1.- Proyectos de integración

Sistema de detección de plagio en archivos de texto. Este proyecto es similar a mi trabajo debido a que es un sistema que tiene como entrada archivos a los cuales se les aplica técnicas de procesamiento de lenguaje natural (PLN) y devuelve el grado de similitud, algunas diferencias son que recibe los documentos en texto plano, además mi sistema trabajará sobre textos científicos escritos en ingles. [1]

Sistema de recuperación de información semántico. Este proyecto es un sistema que también utiliza técnicas de procesamiento de lenguaje natural con diferencia de que las técnicas no las utiliza para encontrar similitud sino para obtener determinados datos del archivo de entrada. [2]

Obtención de una medida cuantitativa de similitud de códigos fuente escritos en lenguaje C. este proyecto es similar debido a que también realiza similitud entre textos y da como resultado una medida de esto, se diferencia en que este es especifico a códigos fuente de lenguaje C y mi trabajo es para textos científicos escritos en inglés. [3]

4.2.- Tesis

ANÁLISIS DE LA SIMILITUD ENTRE PROGRAMAS DE ALTO NIVEL. Esta tesis es similar a mi trabajo ya que se enfrenta al problema de similitud entre dos textos, aunque difiere en que está específicamente enfocado a programas de alto nivel y realiza la transformación del código fuente en dos secuencias numéricas (abstracta y en detalle) y entonces aplica el algoritmo Fast Dynamic Time Warping (FDTW) para calcular la similitud entre los codigos. [4]

4.3.- Artículos

*UsingGrammarPatterns to Evaluate Semantic Similarity for Short Texts.*Este artículo es similar a mi trabajo ya que trata sobre utilizar técnicas de procesamiento de lenguaje natural para dar una medida de similitud semántica

ente textos cortos y se diferencia en que utiliza la técnica de 'GrammarPatterns' y en mi trabajo utilizaré métodos estadísticos, sintácticos y semánticos. [5]

4.4.- Software

iThenticate[6],*TurnitinOriginalityCheck*[7]. Estas dos herramientas funcionan de manera parecida en las que, el usuario sube un documento al sistema y este lo compara contra documentos ya almacenados en una base de datos y al finalizar arroja varios reportes de similitud como lo son: porcentaje de similitud con distintos documentos, gráficas.

5.- Marco Teórico

5.1 Similitud en textos

La similitud entre textos es importante en distintas aplicaciones del lenguaje natural como extracción de la información, clasificación etc. Trabajo existente en esta área a intentado computar la similitud en textos analizando co-ocurrencia de las palabras y estadística de palabras con modelos probabilísticos.

Una de las primeras aplicaciones de la similitud en textos es quizá en modelos vectoriales para la recuperación de la información.

El acercamiento típico para encontrar la similitud entre dos segmentos de texto es usar un método léxico simple y producir un resultado basado en el número de unidades léxicas que ocurren en ambos documentos.

Las medidas de similitud son funciones que calculan el grado de relación entre un par de textos el cual se encuentra en el rango de [0,1] donde cero representa que no hay nada en común y uno indica que representan lo mismo.

Hay distintas medidas de similitud algunas de las cuales son:

1.- Dice

$$S_{A,B} = \frac{2\,|words_A \cap words_B|}{|words_A| + |words_B|}$$

2.- Jaccard

$$S_{A,B} = \frac{|words_A \cap words_B|}{|words_A \cup words_B|}$$

3.- Cosine

$$S_{A,B} = \frac{|words_A \cap words_B|}{\sqrt{|words_A|}\,|words_B|}$$

5.1.1 Similitud TF-IDF

TF-IDF es la unión de dos métricas de similitud TF (Term Frequency) y IDF (Inverse Document Frequency), este método genera listas de palabras clave con una calificación o peso que indica qué tan relevante es la palabra con respecto al documento seleccionado y al corpus en general, esta medida se puede implementar de distintas formas siendo una de ellas la siguiente.
El cálculo de TF se lleva a cabo calculando la frecuencia relativa de las palabras en cada uno de los textos, Los términos que más se repiten en un documento son, en principio, más relevantes que los que se emplean menos.Representandoesta métrica de la siguiente manera:

$$TF_{i,j} = f_i^j$$

Donde f es el número de ocurrencias del término i en el documento j.

IDF se basa en contar el número de documentos en los que aparece la palabra buscada. Los términos más frecuentes en la colección serán menos relevantes que los más raros.

$$IDF_i = \log (N/n_i)$$

Donde N es el número de documentos que existen en el corpus, n es el número de documentos donde el término i se encuentra.
Obteniendo así la siguiente fórmula para calcular la similitud mediante TF-IDF.

$$TF\text{-}IDF_{i,j} = f_i^j * \log (N/n_i)$$

5.1.2 Similitud sintáctica

Al construir oraciones, lo hacemos valiéndonos de palabras que pertenecen a clases concretas (sustantivos, adjetivos, adverbios, verbos, etc.) y que se definen con criterios morfológicos y sintácticos (su combinatoria con otras palabras), y también valiéndonos de conjuntos coherentes de palabras que presentan una estructura interna (con núcleo, modificadores, etc.) y que constituyen categorías Sintácticas denominadas grupos o sintagmas. Dicho esto, parece evidente que conviene distinguir dos tipos de categorías: Las categorías llamadas tradicionalmente «clases de palabras»(sustantivo, adjetivo, determinativo, pronombre, verbo, adverbio, preposición y conjunción), y las categorías grupo o categorías sintagmáticas.
CLASES DE PALABRAS: son las diferentes categorías que sirven para agrupar a todas las palabras que comparten rasgos de forma, función o significado.
Comúnmente se distinguirán 8 clases: sustantivo, adjetivo, determinativo, pronombre, verbo, adverbio, preposición y conjunción. Además podemos se pueden incluir a las locuciones (conjunto de palabras que funcionan como una sola palabra)y a las perífrasis verbales (varios verbos que funcionan como uno).
LOS SINTAGMAS: Son la unidad mínima con capacidad para desempeñar una función sintáctica. Pueden estar formados por una sola palabra o por varias palabras que funcionan como una unidad. Existen los siguientes tipos de sintagmas:

1. Sintagma nominal (SN), precedido o no de preposición: unidad sintáctica que tiene como núcleo un sustantivo (o cualquier otra palabra que funcione como un sustantivo).
2. Sintagma verbal (SV): unidad sintáctica que tiene como núcleo un verbo.
3. Sintagma adjetivo (SAdj.): unidad sintáctica que tiene como núcleo un adjetivo.
4. Sintagma adverbial (SAdv.): unidad sintáctica que tiene como núcleo un adverbio.
5. Sintagma preposicional (SP). Se compone como mínimo de una preposición (u otro tipo de adposición), que hace de núcleo, y un sintagma como complemento obligatorio.

5.1.3 Similitud semántica

En el área del Procesamiento del lenguaje natural la similitud semántica de dos palabras es la medida de la interrelación existente entre ellas, siendo que determinar el grado de similitud semántica involucra el proceso de determinar un ranking para parejas que pertenecen a la misma clase semántica, además de la medida de similitud también se puede agregar una especificación en la cual aumentemos el peso de palabras muy específicas y disminuyamos el peso a palabras muy genéricas. Para calcular la similitud existes diversas métricas de las cuales algunas son:

- Leacock & Chodorow

Esta medida está basada en las longitudes de rutas usando la jerarquía “es-un” de WordNet, para las definiciones de sustantivos (Leacock et al.1998). La ruta más corta entre dos conceptos es aquella que incluye el menor número de conceptos intermedios.

- $Sim(W_i, W_j) = Max\left[-\log \frac{Dist(c_i, c_j)}{2xD}\right]$

  $Sim(W_i, W_j) = Max\left[\log 2D - \log Dist(c_i, c_j)\right]$, where $Dist(ci, c_j)$ is the shortest distance between concepts c_i and c_j.

- Resnik

Introduce una medida de relación basada en el concepto de "contenido de la información" más conocido en inglés como information content, el cual se trasluce como un valor que es asignado a cada concepto en una jerarquía basada en la evidencia encontrada en un corpus.

$$IC(concept) = -\log(P(concept))$$

- Lin

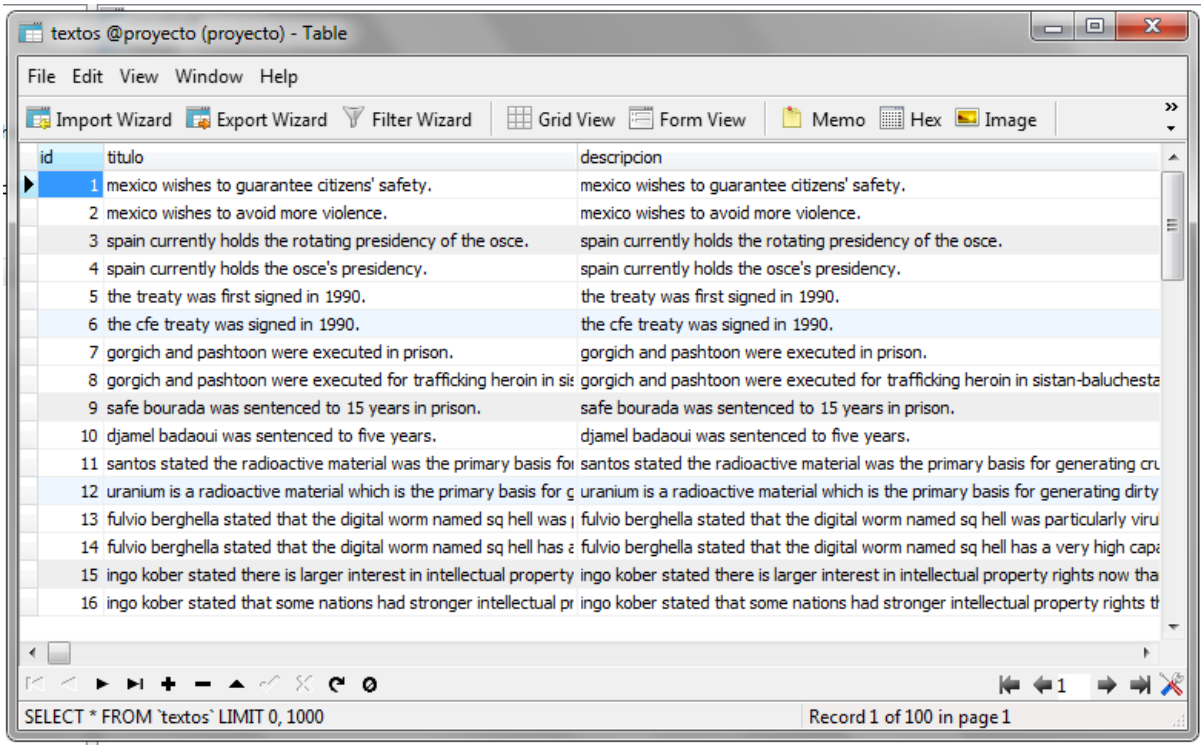
La medida de Lin (1997) está basada en su teorema de similitud. Este establece que la similitud de dos conceptos es medida por el ratio entre la cantidad de información necesaria para establecer la información común de ambos conceptos y la cantidad de información necesaria para describirlos.

$$related_{lin}(c_1, c_2) = \frac{2 \times IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

6.- Desarrollo del proyecto

6.1.- Módulo extractor

Este módulo será el encargado de extraer la información almacenada en la base de datos (Fig.1) de dos registros para así poder realizar la similitud entre estos. El resultado de este módulo servirá como entrada para el módulo de métricas (Fig. 2).



id	titulo	descripcion
1	mexico wishes to guarantee citizens' safety.	mexico wishes to guarantee citizens' safety.
2	mexico wishes to avoid more violence.	mexico wishes to avoid more violence.
3	spain currently holds the rotating presidency of the osce.	spain currently holds the rotating presidency of the osce.
4	spain currently holds the osce's presidency.	spain currently holds the osce's presidency.
5	the treaty was first signed in 1990.	the treaty was first signed in 1990.
6	the cfe treaty was signed in 1990.	the cfe treaty was signed in 1990.
7	gorgich and pashtoon were executed in prison.	gorgich and pashtoon were executed in prison.
8	gorgich and pashtoon were executed for trafficking heroin in sis	gorgich and pashtoon were executed for trafficking heroin in sistan-baluchesta
9	safe bourada was sentenced to 15 years in prison.	safe bourada was sentenced to 15 years in prison.
10	djamel badaoui was sentenced to five years.	djamel badaoui was sentenced to five years.
11	santos stated the radioactive material was the primary basis fo	santos stated the radioactive material was the primary basis for generating cru
12	uranium is a radioactive material which is the primary basis for g	uranium is a radioactive material which is the primary basis for generating dirty
13	fulvio berghella stated that the digital worm named sq hell was	fulvio berghella stated that the digital worm named sq hell was particularly viru
14	fulvio berghella stated that the digital worm named sq hell has a	fulvio berghella stated that the digital worm named sq hell has a very high capa
15	ingo kober stated there is larger interest in intellectual property	ingo kober stated there is larger interest in intellectual property rights now tha
16	ingo kober stated that some nations had stronger intellectual pr	ingo kober stated that some nations had stronger intellectual property rights th

Fig.1 Base de Datos con los registros almacenados

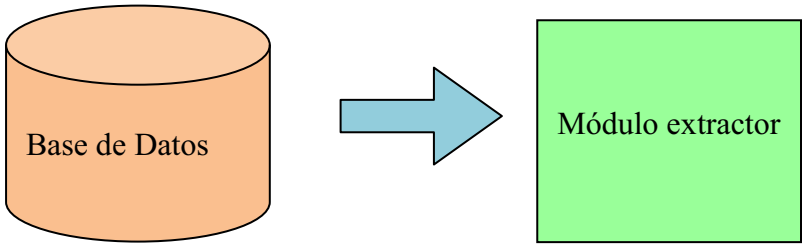
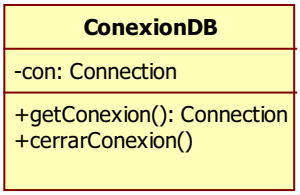
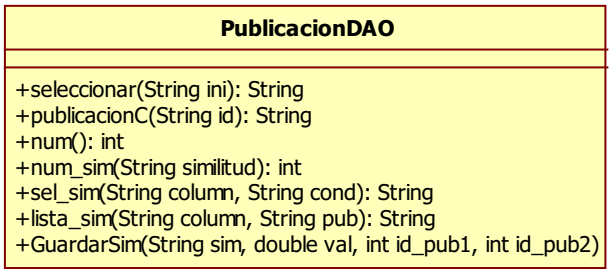


Fig.2 Arquitectura módulo extractor

Clases a utilizar



La clase “ConexionDB” es la encargada de conectarse con la base de datos, utilizando para ello la API “mysql-connector-java-5.1.18-bin.jar” (ver Anexo 10.1)



La clase “PublicacionDAO” es la encargada de realizar todas las consultas que se necesiten hacer a la base de datos ya sea para seleccionar, actualizar o insertar datos (ver Anexo 10.1).

6.2 Módulo de implementación de métricas

Este módulo utilizará tres técnicas de similitud (Tf-idf, Frases, Tripletas), con la finalidad de realizar el procesamiento de los textos de entrada y obtener el cálculo de similitud existente entre ellos. (Fig. 3)

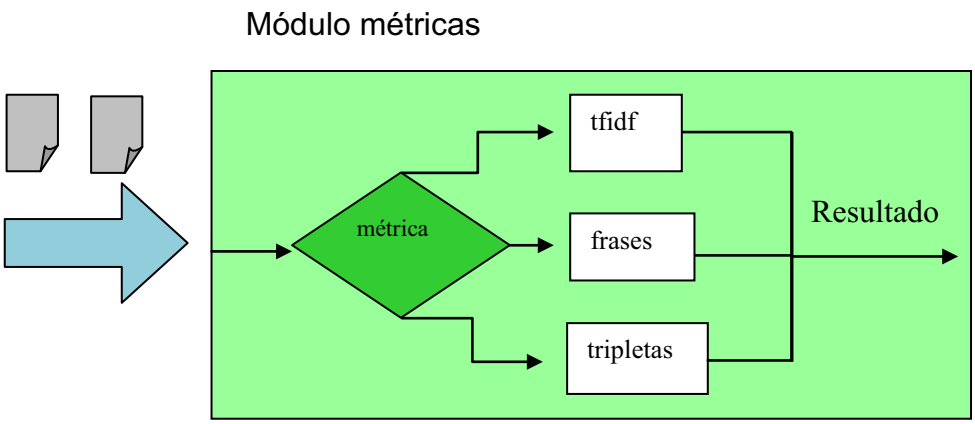


Fig. 3 Arquitectura módulo métricas

Clase utilizada

Similitud
+Similitud(String pub1, String pub2, String tipo_sim): double -token(String cad): ArrayList<String> -separar(String cad): String -quitar(String cad): String -sim_ws4j(ArrayList<ArrayList<ArrayList>> publi1, ArrayList<ArrayList<ArrayList>> publi2): double -csim_ws4j(ArrayList<ArrayList<ArrayList>> publi1, ArrayList<ArrayList<ArrayList>> publi2): double

La clase “Similitud” es la clase principal la cual recibe como parámetros las publicaciones a las cuales se calculará la similitud y el tipo de métrica a utilizar(ver Anexo 10.1).

TreeTaggerTest
+TreeTagger(ArrayList cadena): ArrayList<ArrayList>

La clase “TreeTaggerTest” realiza un etiquetado de las palabras colocando si las palabras son sustantivos, verbos, preposiciones etc. Esta clase utiliza las siguientes API’s: “org.annolab.tt4j-1.2.0-javadoc.jar”, “org.annolab.tt4j-1.2.0-sources.jar” y “org.annolab.tt4j-1.2.0.jar” con las cuales se realiza el etiquetado de las palabras obteniendo tres datos: token (palabra), pos (etiquetado) y lema (palabra sin conjugar)(ver Anexo 10.1).

SimilitudWS4J
-db: ILexicalDatabase -rcs: RelatednessCalculator
+WS4Jsim(String word1, String word2): double

La clase “SimilitudWS4J” calcula la similitud semántica entre dos palabras por lo cual se utilizará la API “ws4j-1.0.1.jar” utilizando la métrica ‘Path’ la cual devuelve un valor entre 0 y 1(ver Anexo 10.1).

Donde:

0.- las palabras son diferentes.

1.- las palabras son muy similares o iguales

Ejemplo.

Para ilustrar el funcionamiento se utilizaran los textos de la tabla 1:

Texto	Descripción
1	russia ratified the updated treaty in 2004 but the united states and other nato members have refused to do so arguing that moscow must first fulfill obligations to withdraw forces from georgia and from moldova's separatist region of trans-dniester.
2	russia has ratified the amended version but the united states and other nato members have refused to do so until the russian government withdraws troops from the former soviet republics of moldova and georgia.

Tabla 1.- Par de textos a utilizar de ejemplo.

6.2.1.- Tf-idf

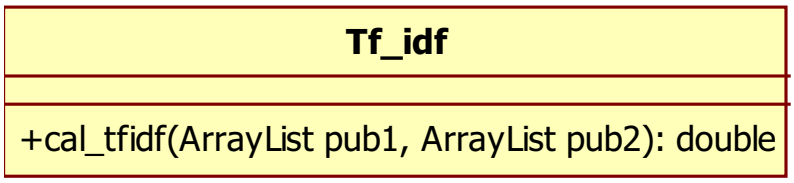
Es una técnica basada en frecuencia de aparición de las palabras en un texto. La forma implementada para calcular la similitud es la siguiente:

Pseudocódigo del algoritmo

```
1.- menor = texto más corto;
2.- mayor = texto más largo;
3.- ArrayList pal = new ArrayList();
4.- Para todas las palabras i de menor{
5.-         a = menor [i] existe en pal ?;
6.-         Si (a){
```

```
7.-                 Incrementar frecuencia de palabra en auxf1
8.-         }No{
9.-                 pal.add(menor[i]);
10.-                Inicializar frecuencia en auxf1 y auxf2
11.-        }
12.- }
13.- Para todas la palabras i de mayor {
14.-        a = mayor [i] existe en pal ?;
15.-        Si (a){
16.-                Incrementar frecuencia de palabra en auxf2
17.-        }
18.- }
19.- Para todas las frecuencias i de auxf1{
20.-        Prom_frec += val_menor(auxf1[i],auxf2[i])
            /val_mayor(auxf1[i],auxf2[i]);
21.- }
22.- Similitud= Prom_frec / sumar #palabras distintas de texto1 y
     texto2;
```

Clase utilizada



Esta clase recibe como parámetros dos ArrayList los cuales contienen los textos a calcular similitud y devuelve el resultado en tipo doublé (ver Anexo 10.1).

Ejemplo de la implementación del algoritmo en la Fig. 4 en donde se puede observar el ArrayList generado del texto más corto, las frecuencias de aparición de las palabras en los dos textos y la similitud calculada.

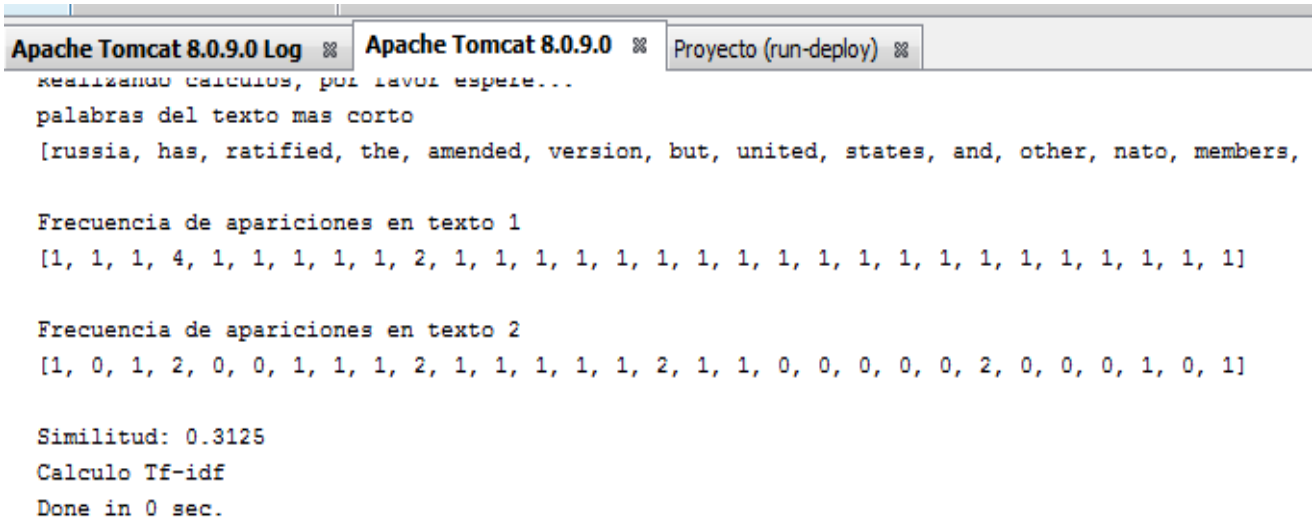


Fig. 4. Implementación del algoritmo Tf-idf.

6.2.2.- Frases

Esta técnica utiliza Frases Nominales (FN), Frases Preposicionales (FP) y Frases Verbales (FV) para realizar el cálculo de similitud.

Frases Nominales.- Están compuestas por sustantivos (N).

Frases Verbales.- Conformadas por verbos (V).

Frases Preposicionales.- Compuesta por una preposición (I) + sustantivos (N).

En el algoritmo se considerara FN si está compuesta por a lo mas cinco sustantivos consecutivos, FV si está compuesta por a lo mas tres verbos consecutivos y FP si está compuesta por una preposición y una FN.

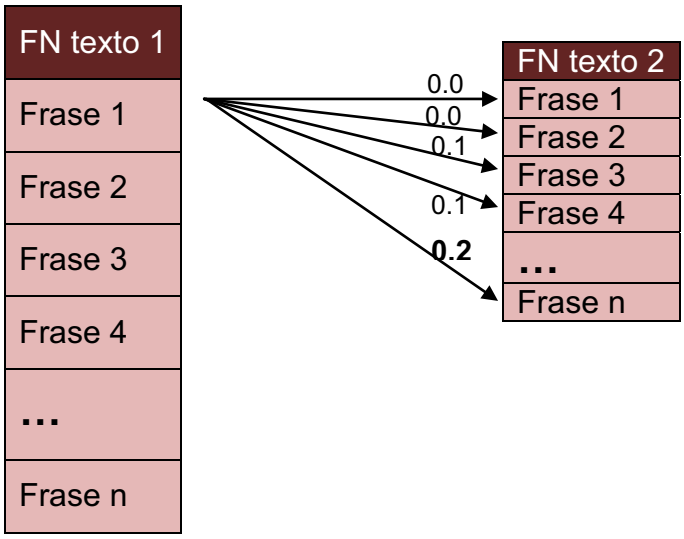
Calculo de similitud, pseudocódigo.

1.-	Etiquetar textos
2.-	Crear FV, FN, FP de los textos
3.-	Para cada FVi de Texto1 {

```
4.-     Para cada FVj de Texto2 {
5.-         s = cal_simi (FVi, FVj);
6.-         Si (s > sim [i]) sim [i] = s;  //obtenemos el máximo
7.-     }
8.- }
9.- p [0]= promedio (sim []);
10.- Para cada FNi de Texto1 {
11.-    Para cada FNj de Texto2 {
12.-        s = cal_simi (FNi, FNj);
13.-        Si (s > sim [i]) sim [i] = s;  //obtenemos el máximo
14.-    }
15.- }
16.- p [1]= promedio (sim[]);
17.- Para cada FPi de Texto1 {
18.-    Para cada FPj de Texto2 {
19.-        s = cal_simi (FPi, FPj);
20.-        Si (s > sim [i]) sim [i] = s;  //obtenemos el máximo
21.-    }
22.- }
23.- p [2]= promedio (sim []);
24.- Similitud = promedio (p []);
```

Representación del cálculo de la similitud

Cada frase del texto 1 calcula su similitud con cada frase del texto 2 y después se obtiene la similitud con el valor más alto y ese valor será la similitud de la frase del texto 1.



FN texto 1	Similitud
Frase 1	**0.2**
Frase 2	0.3
Frase 3	0.8
Frase 4	0.5
…	…
Frase n	1.0
Promedio	**0.56**

Una vez ya obtenidas todas las similitudes de cada frase se realiza un promedio para obtener la similitud total de la frase FN.
Lo mismo se realiza para las frases FV y FP.

FRASE	Similitud
FV	0.49
FN	**0.56**
FP	0.6
Promedio	**0.55**

Para calcular la similitud total que hay entre los textos se realiza un promedio entre las frases FV, FP, FN

Clase

Frases
-indp: ArrayList<ArrayList> -indp1: ArrayList<ArrayList> -indp2: ArrayList<ArrayList>
+cal_frase(ArrayList<ArrayList> tag): ArrayList<ArrayList<ArrayList>> -calcular_frase(String patron, int i, ArrayList frase, ArrayList<ArrayList> tag): int -setindp(ArrayList aux) -setindp1(ArrayList aux) -setindp2(ArrayList aux)

La clase “Frases” es la encargada de generar las tres frases: FV, FP, FV recibiendo como parámetro un ArrayList bidimensional con el texto ya etiquetado y devuelve un ArrayList de tres dimensiones compuesto con las Frases para poder realizar el cálculo de la similitud (ver Anexo 10.1).

Ejemplo. Las Fig.5 y 6 muestran el etiquetado de los textos realizado por la clase TreeTaggerTest.

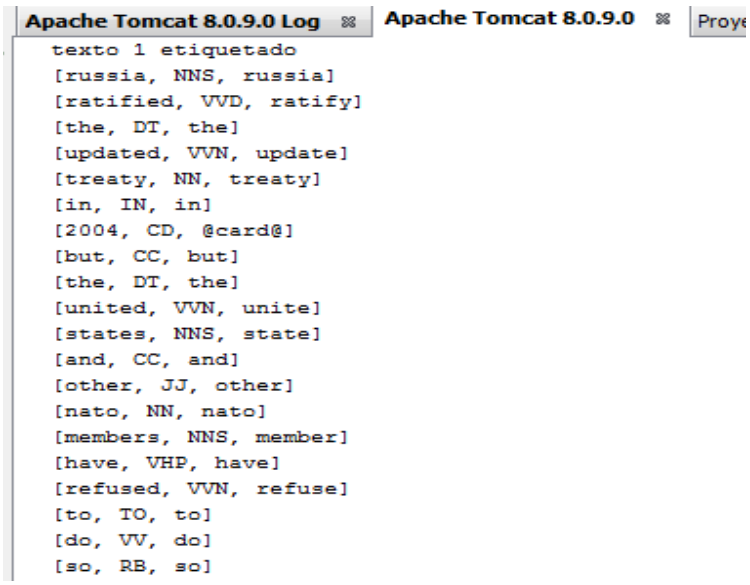


Fig.5 Etiquetado del texto 1 el cual servirá para crear las frases FV, FP y FN.

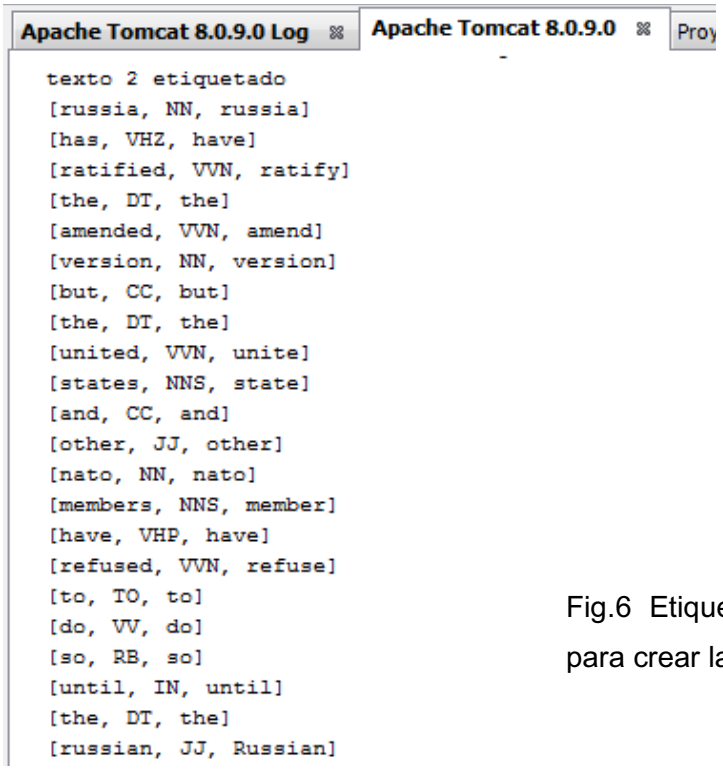


Fig.6 Etiquetado del texto 2 el cual servirá para crear las frases FV, FP y FN.

Las Fig. 7 y 8 muestran la creación de las frases FV, FN y FP las cuales fueron creadas por la clase Frases.

La primer fila contiene las FV con a lo más 3 verbos, la segunda fila contiene las FN con a lo más 5 sustantivos y la tercer fila tiene las FP (preposición + FN).

```
Realizando calculos, por favor espere...
[[ratify], [update], [unite], [have, refuse], [do], [argue], [fulfill], [withdraw]]
[[russia], [treaty], [state], [nato, member], [moscow], [obligation], [force], [dniester]]
[[from, Georgia], [from, moldovas, separatist, region], [of, trans]]
```

Fig. 7 FV, FN y FP creadas del texto1

```
[[have, ratify], [amend], [unite], [have, refuse], [do], [withdraw]]
[[russia], [version], [state], [nato, member], [government], [troop], [republic], [Georgia]]
[[of, Moldova]]
```

Fig. 8 FV, FN y FP creadas del texto 2

La Fig. 9 muestra la similitud de cada frase (FV, FN, FP respectivamente) entre los textos, así como la similitud total calculada para ellos.

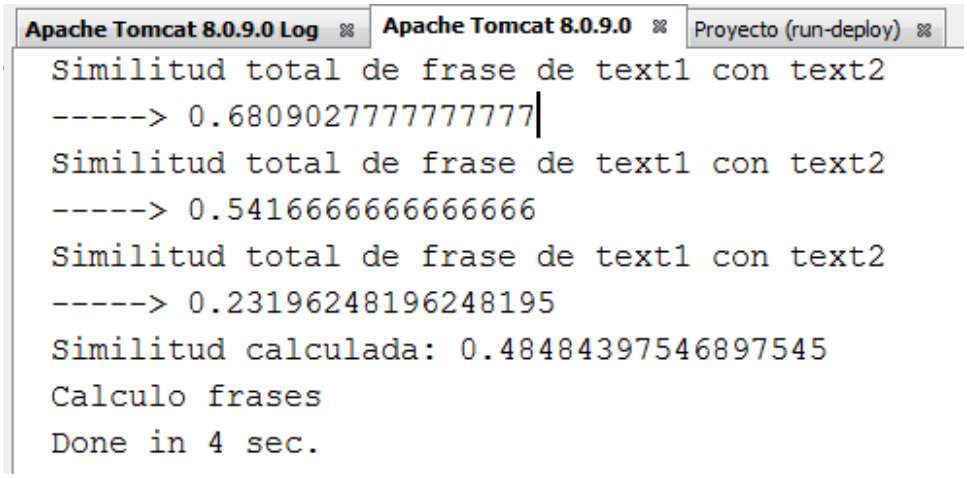


Fig. 9 calculo de similitud para las frases así como la total realizando el promedio de las anteriores

6.2.3.- Tripletas

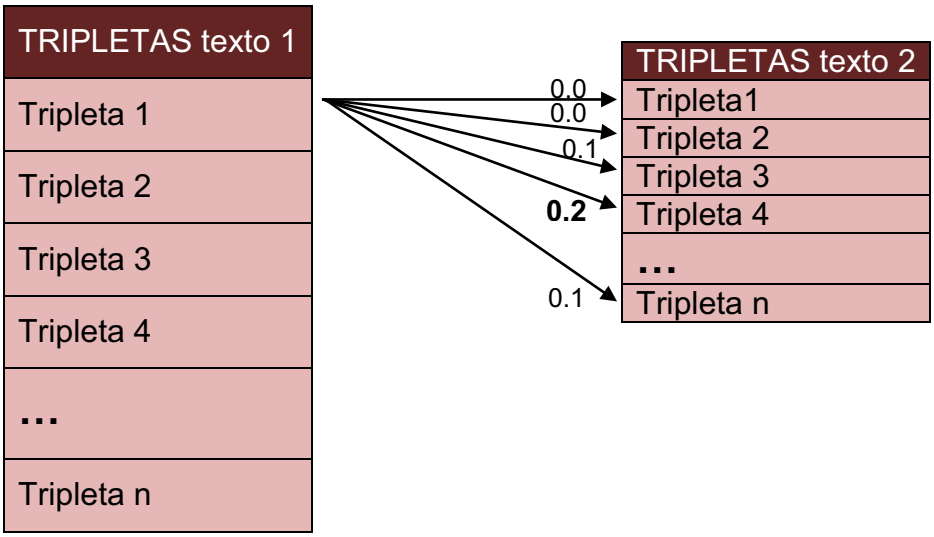
Esta técnica retoma la anterior con la diferencia que realiza la siguiente agrupación: FN FV (FN || FP) a la cual llamamos tripleta, por tanto para calcular la similitud las comparaciones se hacen sobre las tripletas.

Pseudocódigo

```
1.- Etiquetar textos
2.- Crear FV, FN, FP de los textos
3.- Crear Tripletas de los textos
4.- Para cada tripleta i de Texto 1{
5.-         Para cada tripleta j de Texto 2{
6.-                 s = calc_sim (tripleta i, tripleta j);
```

```
7.-                 Si (s > sim [i]) sim [i] = s;
8.-             }
9.-     }
10.-    similitud = promedio (sim []);
```

Representación del cálculo de la similitud



Para cada tripleta del texto 1 calcular similitud con cada tripleta del texto 2, después se obtendrá la similitud de mas grande encontrada.

TRIPLETAS texto 1	Similitud
Tripleta 1	**0.2**
Tripleta 2	0.3
Tripleta 3	0.8
Tripleta 4	0.5
…	…
Tripleta n	1.0
Promedio	**0.56**

Una vez ya obtenidas todas las similitudes de cada tripleta se realiza un promedio para obtener la similitud total entre los textos.

Clase

Tripleta
-indp: ArrayList<ArrayList> -indp1: ArrayList<ArrayList> -indp2: ArrayList<ArrayList> -ini_fin: ArrayList<ArrayList> -matriz_trip: ArrayList<ArrayList<ArrayList>>
+calc_tripleta(ArrayList<ArrayList> tag): ArrayList<ArrayList<ArrayList>> -calcular_frase(String patron, int i, ArrayList frase, ArrayList<ArrayList> tag): int

La clase "Tripleta" es la encargada de crear las tripletas recibiendo como parámetro un ArrayList bidimensional con el texto ya etiquetado y regresa un ArrayList tridimensional con las tripletas para realizar el cálculo de la similitud (ver Anexo 10.1).

Ejemplo

- Etiquetar textos.

Esta métrica utiliza el mismo etiquetador que la anterior por tanto los texto etiquetados son los mismo que los de las figuras 5 y 6.

- Crear Frases.

La creación de frases es igual a la anterior métrica por que las creadas de los textos se pueden visualizar en la figuras 7 y 8.

- Crear Tripletas.

La creación de tripletas la realiza la clase Tripletas la cual recibe un ArrayList bidimensional con el texto ya etiquetado y regresa un ArrayList tridimensional con las tripletas ya formadas como se puede ver en la Fig. 10 y 11

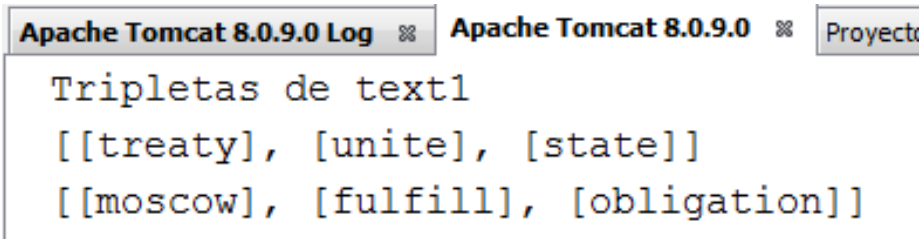


Fig. 10 Tripletas formadas de text1

```
Tripletas de text2
[[version], [unite], [state]]
[[government], [withdraw], [troop]]
```

Fig. 11 Tripletas formadas de text2

En la Fig. 12 se puede observar la similitud en cada tripleta y la similitud calculada para los textos la cual es un promedio de todas las similitudes de las tripletas.

```
Apache Tomcat 8.0.9.0 Log | Apache Tomcat 8.0.9.0 | Proyecto (run-deploy)
Similitud en tripletas
[tripleta 1 : 0.7222222222222223, tripleta 2 : 0.2888888888888889]
Similitud de los textos
----> 0.5055555555555556
Tripletas de text1
Done in 4 sec.
```

Fig. 12 Calculo de similitud entre los textos.

6.3.- Módulo sistema web, visualización y enriquecimiento

Los módulos sistema web y visualización son los encargados de realizar la interfaz web entre el usuario y el sistema, este sistema utiliza paginas JSP's, js (javascript) y html.Con las jsp's se implementa la conexión con las clases en java para realizar los cálculos de similitud, las consultas a la base de datos y el enriquecimiento de la base de datos con los valores de la similitud calculada para cada una de las métricas. Fig 13

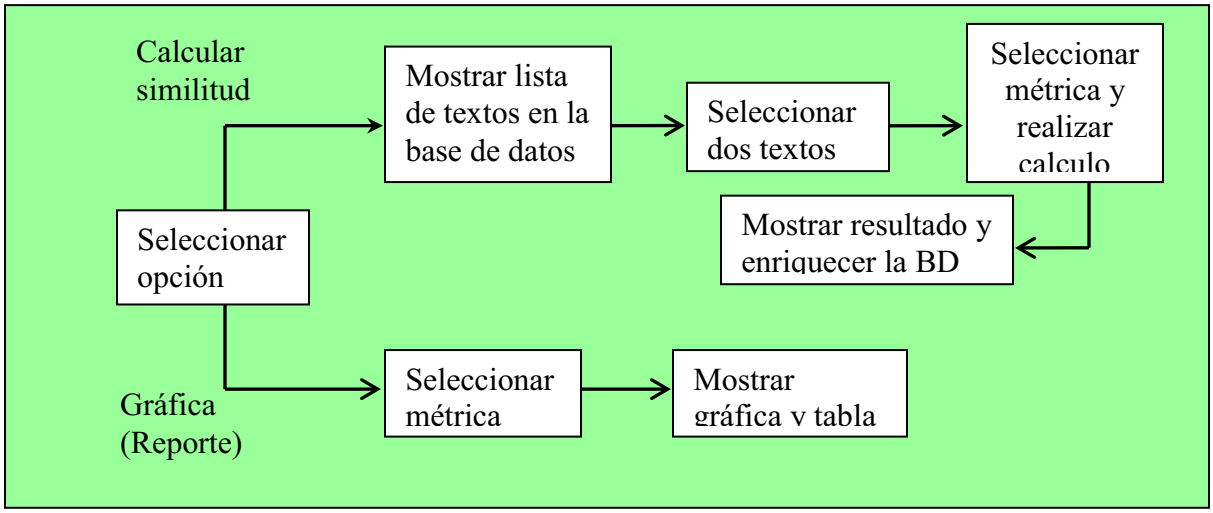


Fig.13 Arquitectura del sistema web, visualización y enriquecimiento

La página principal contiene dos opciones a realizar en el sistema los cuales son: Calcular Similitud (elige dos textos y realiza el cálculo con cualquiera de las métricas) o ver Gráfica de artículos similares (reporte de las similitudes realizadas). Fig.14 (ver Anexo 10.2)

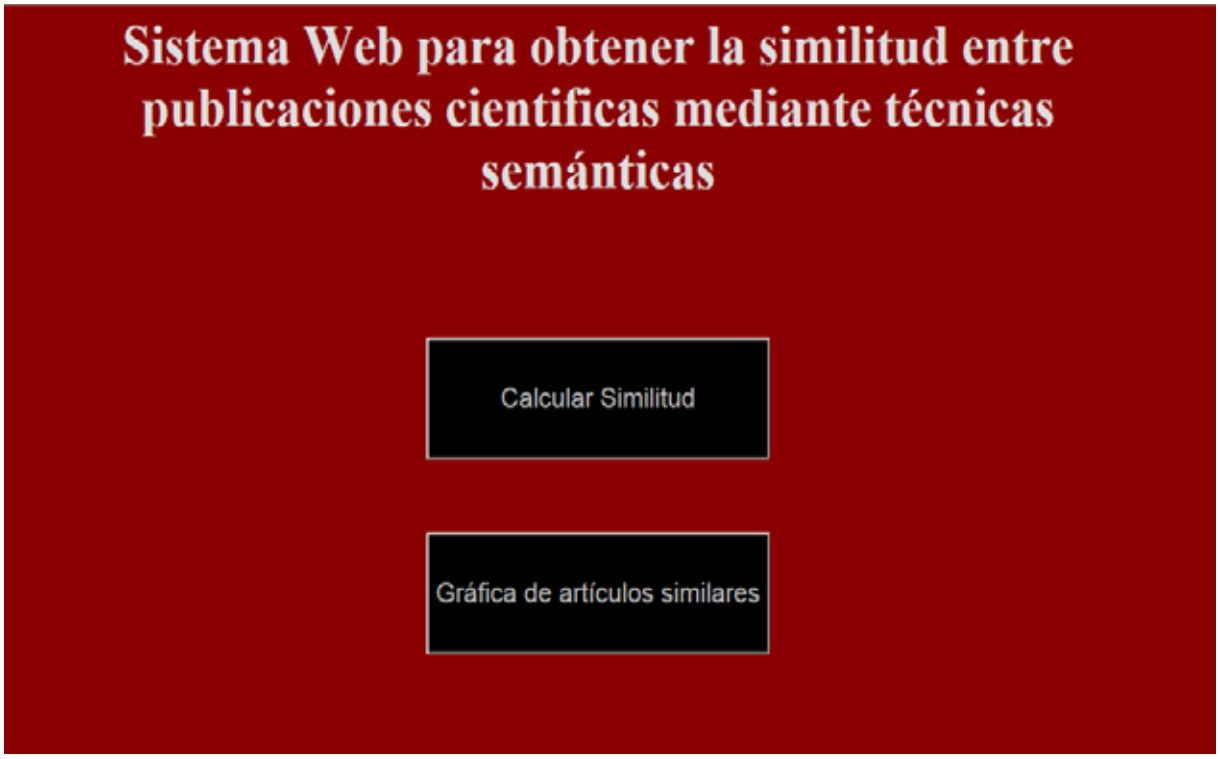


Fig. 14 Página principal de interfaz web la cual permite seleccionar dos opciones

6.3.1 Calcular Similitud

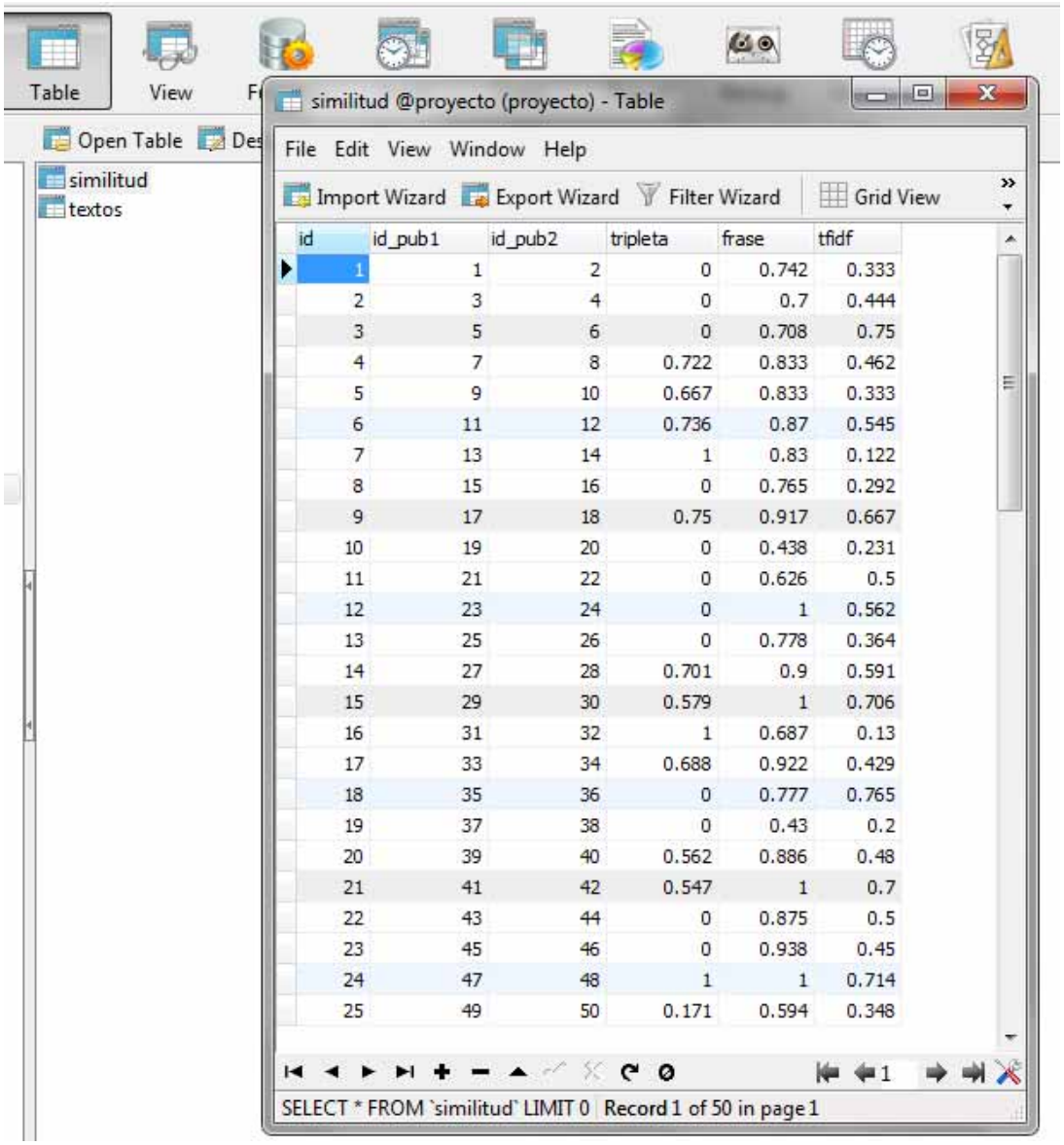
Esta sección nos presenta una lista de los textos que existen en la base de datos (Fig 15) de los cuales podemos seleccionar dos para calcular su similitud con cualquiera de las tres métricas: Tf-idf (Fig. 16), Frases (Fig. 17), Tripletas (Fig. 18), dependiendo del tamaño de los textos el cálculo puede tardar tiempo en completarse por lo que este procedimiento se realiza de manera asíncrona con ayuda de AJAX implementado con JQUERY, al terminar el cálculo el sistema muestra la similitud entre los textos y realiza el enriquecimiento a la base de datos guardando el valor calculado para estos dos textos (Fig. 19), el cual será consultado cuando se revise el reporte(ver Anexo 10.2).

Fig. 15 interfaz para seleccionar los artículos a calcular similitud

Fig. 16 Cálculo de la similitud estadística (Tf-idf).

Fig. 17 Cálculo de la similitud por Frases.

Fig. 18 Cálculo de la similitud por tripletas.

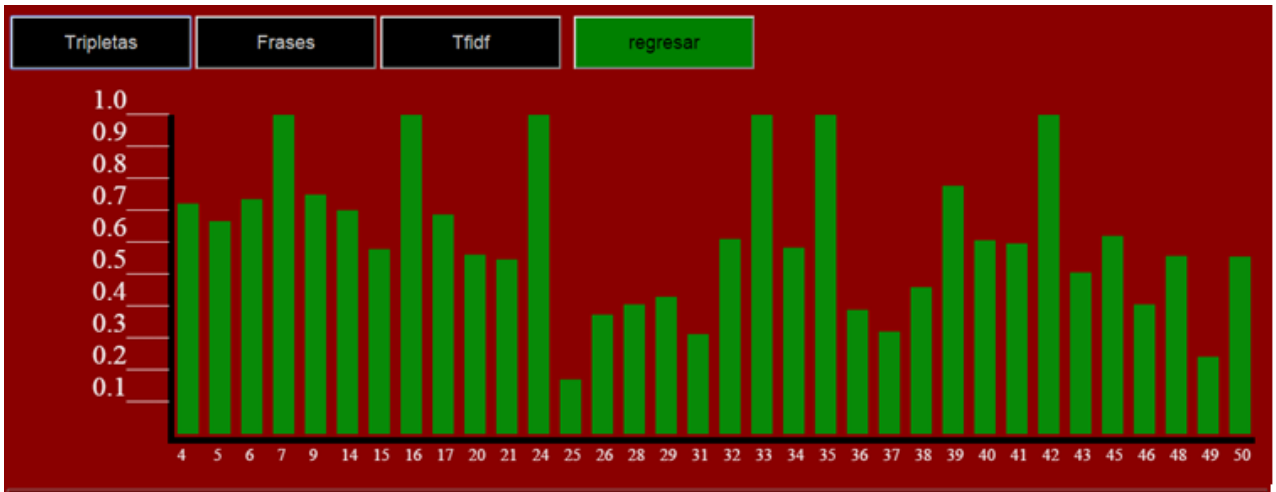


id	id_pub1	id_pub2	tripleta	frase	tfidf
1	1	2	0	0.742	0.333
2	3	4	0	0.7	0.444
3	5	6	0	0.708	0.75
4	7	8	0.722	0.833	0.462
5	9	10	0.667	0.833	0.333
6	11	12	0.736	0.87	0.545
7	13	14	1	0.83	0.122
8	15	16	0	0.765	0.292
9	17	18	0.75	0.917	0.667
10	19	20	0	0.438	0.231
11	21	22	0	0.626	0.5
12	23	24	0	1	0.562
13	25	26	0	0.778	0.364
14	27	28	0.701	0.9	0.591
15	29	30	0.579	1	0.706
16	31	32	1	0.687	0.13
17	33	34	0.688	0.922	0.429
18	35	36	0	0.777	0.765
19	37	38	0	0.43	0.2
20	39	40	0.562	0.886	0.48
21	41	42	0.547	1	0.7
22	43	44	0	0.875	0.5
23	45	46	0	0.938	0.45
24	47	48	1	1	0.714
25	49	50	0.171	0.594	0.348

Fig. 19 Tabla con las similitudes calculadas para cada par de textos

6.3.2.- Gráfica de artículos similares

Esta página presenta una grafica de barras creada con HTML5 y una tabla con la similitud (0 – 1] entre los textos mediante consultas a la base de datos utilizando JSP's. Las similitudes que se pueden consultar son (métricas): Tripletas, Frases y Tfidf.Fig 20 se muestra el ejemplo de similitud basada en Tripletas (ver Anexo 10.2).



Similitud tripleta

ID x	Texto 1	Texto2	Porcentaje
4	gorgich and pashtoon were executed in prison.	gorgich and pashtoon were executed for trafficking heroin in sistan-baluchestan province.	0.722
5	safe bourada was sentenced to 15 years in prison.	djamel badaoui was sentenced to five years.	0.667
6	santos stated the radioactive material was the primary basis for generating crude weapons of mass destruction and terrorism.	uranium is a radioactive material which is the primary basis for generating dirty weapons of mass destruction and terrorism.	0.736
7	fulvio berghella stated that the digital worm named sq hell was particularly virulent and replicated itself at the rate of 8000 times an hour.	fulvio berghella stated that the digital worm named sq hell has a very high capacity to replicate itself and the digital worm named sq hell is slowing down the poste italiane computer network and making some computers inoperable.	1.000
9	investigators claim the british company was a cia cover.	russian investigators stated that the british company was a cia cover.	0.750

Fig.20 Reporte donde se puede observar la similitud entre los textos por medio de una grafica y una tabla.

7.- Resultados

Las pruebas utilizadas por la herramienta fueron realizadas con 50 pares de textos, los cuales se muestras en la tabla 2.

No.	Texto 1	Texto2
1	mexico wishes to guarantee citizens' safety.	mexico wishes to avoid more violence.
2	spain currently holds the rotating presidency of the osce.	spain currently holds the osce's presidency.
3	the treaty was first signed in 1990.	the cfe treaty was signed in 1990.
4	gorgich and pashtoon were executed in prison.	gorgich and pashtoon were executed for trafficking heroin in sistan-baluchestan province.
5	safe bourada was sentenced to 15 years in prison.	djamel badaoui was sentenced to five years.
6	santos stated the radioactive material was the primary basis for generating crude weapons of mass destruction and terrorism.	uranium is a radioactive material which is the primary basis for generating dirty weapons of mass destruction and terrorism.
7	fulvio berghella stated that the digital worm named sq hell was particularly virulent and replicated itself at the rate of 8000 times an hour.	fulvio berghella stated that the digital worm named sq hell has a very high capacity to replicate itself and the digital worm named sq hell is slowing down the poste italiane computer network and making some computers inoperable.
8	ingo kober stated there is larger interest in intellectual property rights now than 20 or 30 years ago.	ingo kober stated that some nations had stronger intellectual property rights than others.
9	investigators claim the british company was a cia cover.	russian investigators stated that the british company was a cia cover.

10	russia and nato member countries will hold meetings this fall on the conventional forces in europe treaty (cfe).	russian parliament voted unanimously to suspend russia's participation in the conventional forces in europe (cfe) treaty.
11	the united states government and other nato members have refused to ratify the amended treaty until officials in moscow withdraw troops from the former soviet republics of moldova and georgia.	the united states and other nato members have refused ratify the amended treaty until russia completely withdraws from moldova and georgia.
12	russia would not immediately increase military strength along russian borders.	russia would not hesitate to increase military strength along russian borders if the need arises.
13	the center will be operational in august 2008.	the center will formally open in 2009.
14	brazilian police surrounding slums north of rio de janeiro to rid them of violent drug traffickers.	the police officers surrounded the slums north of rio de janeiro to rid them of violent drug traffickers and seize weapons and drugs.
15	regional and international non-proliferation issues should be addressed through dialogue and negotiations.	the second proposal was that regional and international non-proliferation issues should be addressed through dialogue and negotiations.
16	bombing occurred in afghanistan as attacks by anti-government insurgents increase in recent days.	a bombing occurred in afghanistan but police have not yet stated who is responsible.
17	georgian government websites are under intense cyber attack following russian military strikes against georgia late last week.	the cyber attacks follow russian military strikes launched against georgia late last week.

18	switzerland's trade and diplomatic relations with iran have been criticized in recent months after foreign minister micheline calmy-rey traveled to tehran in march 2008 to sign a deal with iran's state gas firm.	switzerland's trade and diplomatic relations with the islamic republic have been criticized in recent months after foreign minister micheline calmy-rey traveled to tehran in march 2008 to sign a gas deal.
19	the conventional forces in europe treaty limits the number of military aircraft, tanks and other non-nuclear heavy weapons in europe.	the conventional forces in europe treaty was signed by russian and nato members in 1990.
20	the african union has proposed a peacekeeping mission to help somalia's struggling transitional government stabilize somalia.	the african union has proposed a peacekeeping mission to aid the struggling transitional government in stabilizing somalia, particularly after the withdrawal of ethiopian forces.
21	reports state that the syrian government was concealing a nuclear facility at the attacked site.	u.n. experts have begun analyzing reports that state the syrian government was concealing a nuclear facility at the attacked site.
22	the drug is also known as ice or shabu.	methamphetamine is also known as ice.
23	national, regional and international efforts to end the illicit trade in small arms and light weapons.	the global community must cooperate to end illicit trade of small arms and light weapons.
24	religious extremism continues in pakistan despite the banning of militant groups.	religious extremism continues in pakistan despite the banning of militant groups by the pakistani government.
25	russian officials have called for a conference on the conventional forces in europe treaty to discuss ratification of the amended treaty.	antonov spoke the day before a conference on the conventional forces in europe treaty.

26	russian president putin signs decree suspending russia's application of european arms control treaty.	president vladimir putin signed a decree suspending russia's participation in the conventional forces in europe treaty.
27	myanmar was formerly known as burma.	myanmar was formerly called burma.
28	alto huallaga is located northeast of the capital lima.	puerto cabezas is located 557 km northeast of managua, nicaragua.
29	nato regrets russia's decision to suspend participation in the cfe.	russia's decision to suspend participation in the cfe is a step in the wrong direction.
30	human rights violations in myanmar include summary executions, torture and the recruitment of child soldiers.	these human rights violations include summary executions, torture and the recruitment of child
31	general nikolai n. urakov stated by telephone that the state scientific center of applied microbiology has quite reliable systems of protection in case of emergency.	general nikolai n. urakov is the longtime director of the the state scientific center of applied microbiology.
32	alstom is in competition with japanese and german countries for the contract.	alstom is competing against japanese company shinkansen and german company ice for the contract.
33	the legislation is the most recent effort by japan to ascribe more freedom to its tightly controlled military and would overturn a ban on the military use of space.	the legislation is the most recent effort by japan to ascribe more freedom to the tightly controlled military technically known as a self-defense force.
34	south korea launches new bullet train reaching 300 kph.	south korea has had a bullet train system since the 1980s.
35	russia is a member of the quartet on the middle east.	russia is a member of the nsg.
36	anatoly sokolov announced-- soviet-built a-135 missile defense system	soviet-built a-135 missile defense system around moscow is obsolete and inefficient.

	around moscow is obsolete and inefficient.	
37	the deal has been in process for several years.	the iaea has been investigating iran's nuclear activities for 4 years.
38	adherence to non-proliferation obligations will enhance mutual trust and foster international cooperation in nuclear energy.	zhang yan stated that all countries should abide by non-proliferation obligations in order to enhance mutual trust and create a sound environment for international cooperation in nuclear energy.
39	french energy and transport company alstom may sign a contract to build a high-speed rail link between beijing and shanghai.	alstom may sign a contract to build a high-speed rail link between beijing and shanghai.
40	no other drug has become as integral in decades.	the drug has been around in other forms for years.
41	the international atomic energy agency reached an agreement with iranian officials that stated there were no remaining issues and ambiguities regarding iran's nuclear program and activities.	the accord said that there were no remaining issues and ambiguities regarding iran's nuclear program and activities.
42	beijing police apprehend more than 20 people including 8 foreigners in suspicion of using and trafficking drugs in 2 pubs in a popular downtown area.	beijing police apprehended more than 20 people in suspicion of using and trafficking drugs in 2 downtown bars.
43	russia ratified the updated treaty in 2004 but the united states and other nato members have refused to do so arguing that moscow must first fulfill obligations to withdraw forces from georgia and from moldova's separatist	russia has ratified the amended version but the united states and other nato members have refused to do so until the russian government withdraws troops from the former soviet republics of moldova and georgia.

	region of trans-dniester.	
44	there is conflicting opinion as to the extent of damage suffered by the insurgency.	violence continues despite conflicting opinion on the extent of damage suffered by the rebel groups.
45	helmand province is the world's largest opium-producing region.	helmand province is the world's largest producer of opium.
46	police stated tsai wen-huang would be extradited to taiwan.	police stated that-- chung has been on the run.
47	2 of the individuals arrested were released.	2 of the zambians arrested are juveniles.
48	mexican president felipe calderon has sent 2800 special agents and soldiers to sinaloa due to the high degree of organized crime and drug trafficking.	mexican president felipe calderon has sent 2800 special agents and soldiers to sinaloa to fight drug trafficking.
49	berlin will host top-level meetings in october to save key european arms control treaty conventional forces in europe (cfe).	russia suspended its participation in a key european arms control treaty that governs deployment of troops in europe.
50	the osce meeting lasted two days.	the osce meeting was held in madrid.

Tabla 2. Textos utilizados en las pruebas

Las similitudes calculadas con las métricas Tf-idf, Frases y Tripletas se muestran en la tabla 3, además de tener la columna con la similitud ideal para cada par de textos.

Para cada par de textos se tiene la siguiente interpretación.

1.- Los textos son completamente equivalentes, significan lo mismo.

0.8.- Los textos son casi equivalentes, pero algo poco importante hace la diferencia.

0.6.- Los textos son más o menos equivalentes, pero alguna información importante difiere/falta.

0.4.- Los textos no son equivalentes, pero comparten detalles.

0.2.- Los textos son diferentes, pero están en el mismo tema.

0.- Los texto están en diferentes temas.

No.	Tf-idf	Frases	Tripletas	Ideal	No.	Tf-idf	Frases	Tripletas	Ideal
1	0.333	0.742	0	0.8	**26**	0.273	0.669	0.374	0.64
2	0.444	0.7	0	0.84	**27**	0.571	0.889	0	0.88
3	0.75	0.708	0	0.84	**28**	0.267	0.722	0.406	1
4	0.462	0.833	0.722	0.64	**29**	0.4	0.8	0.43	0
5	0.333	0.833	0.667	0.12	**30**	0.75	0.735	0	0.76
6	0.545	0.87	0.736	0.72	**31**	0.346	0.719	0.313	0.84
7	0.122	0.83	1	0.72	**32**	0.471	0.803	0.611	0.48
8	0.292	0.765	0	0.4	**33**	0.484	0.689	1	0.72
9	0.667	0.917	0.75	0.88	**34**	0.25	0.611	0.584	0.76
10	0.231	0.438	0	0.28	**35**	0.455	0.867	1	0.4
11	0.5	0.626	0	0.92	**36**	0.812	0.8	0.389	0.44
12	0.562	1	0	0.4	**37**	0.333	0.718	0.321	0.92
13	0.364	0.778	0	0.44	**38**	0.387	0.944	0.46	0.12
14	0.591	0.9	0.701	0.76	**39**	0.737	0.871	0.778	0.64
15	0.706	1	0.579	0.72	**40**	0.267	0.611	0.607	0.8
16	0.13	0.687	1	0.52	**41**	0.519	0.907	0.597	0.4
17	0.429	0.922	0.688	0.96	**42**	0.56	0.805	1	0.68
18	0.765	0.777	0	0.76	**43**	0.312	0.485	0.506	0.76
19	0.2	0.43	0	0.56	**44**	0.421	0.821	0	0.84

20	0.48	0.886	0.562	0.72	**45**	0.636	0.755	0.621	0.64
21	0.7	1	0.547	0.84	**46**	0.118	0.645	0.406	1
22	0.5	0.875	0	0.68	**47**	0.4	0.671	0	0.36
23	0.45	0.938	0	0.8	**48**	0.609	0.9	0.558	0.32
24	0.714	1	1	0.8	**49**	0.241	0.44	0.242	0.92
25	0.348	0.594	0.171	0.8	**50**	0.3	0.622	0.556	0.36

Tabla 3. Similitudes calculadas para los textos con las 3 métricas

En la tabla 4 se muestra el cálculo del promedio de cada métrica de similitud y de la similitud ideal en la cual se puede observar que la métrica mas cercana al valor ideal es la de Frases mientras que la métrica de Tripletas fue la que menos se acerco al valor ideal.

Métrica	Promedio	Diferencia con Ideal	% respecto del ideal
Tf-idf	0.451	0.1922	70.19
Frases	0.771	0.1278	80.13
Tripletas	0.418	0.2252	64.99
Ideal	**0.6432**		

Tabla 4. Comparaciones de métricas con valor ideal

En la Fig. 21 se muestra la gráfica de las similitudes encontradas para la métrica Tf-idf.

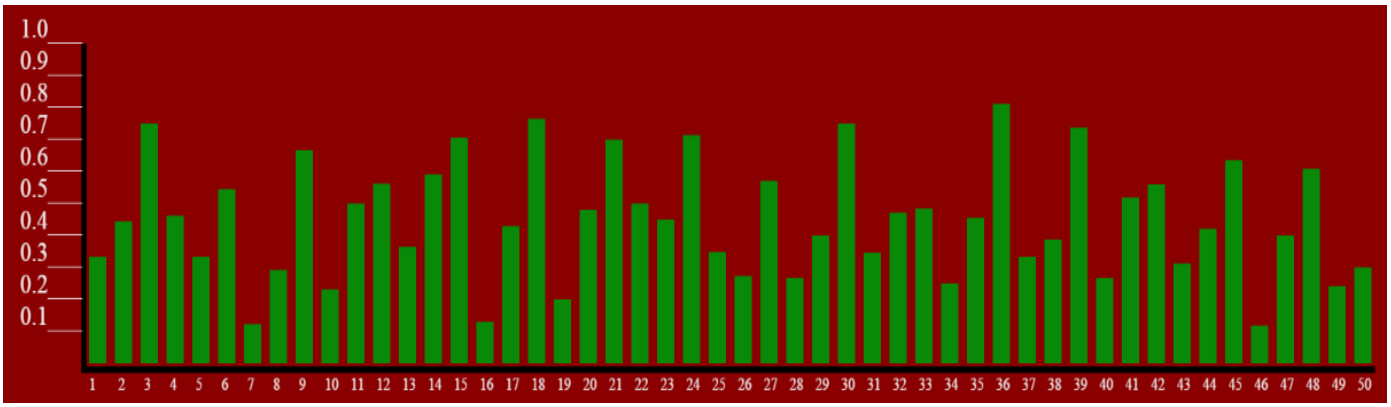


Fig. 21 Gráfica de la métrica Tf-idf

En la Fig. 22 se muestra la gráfica de las similitudes encontradas para la métrica Frases.

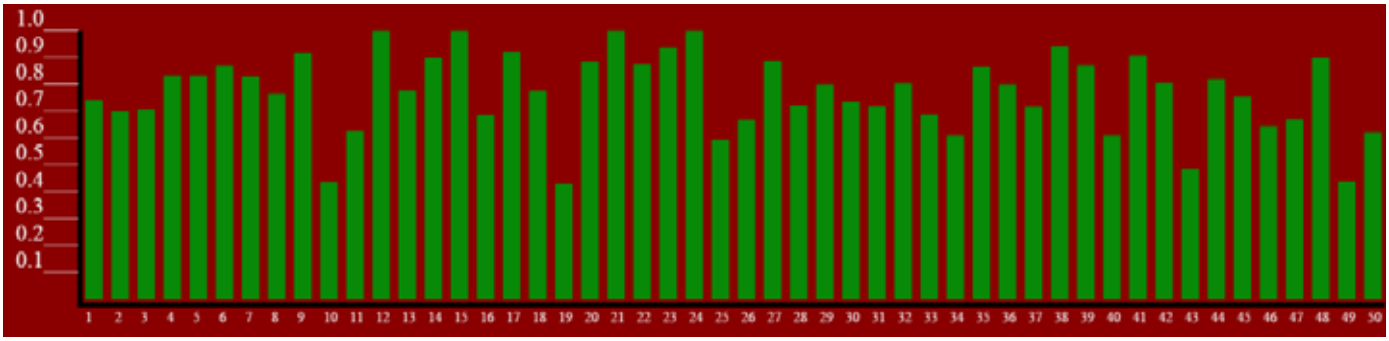


Fig. 22 Gráfica de la métrica Frases

En la Fig. 23 se muestra la gráfica de las similitudes encontradas para la métrica Tripletas.

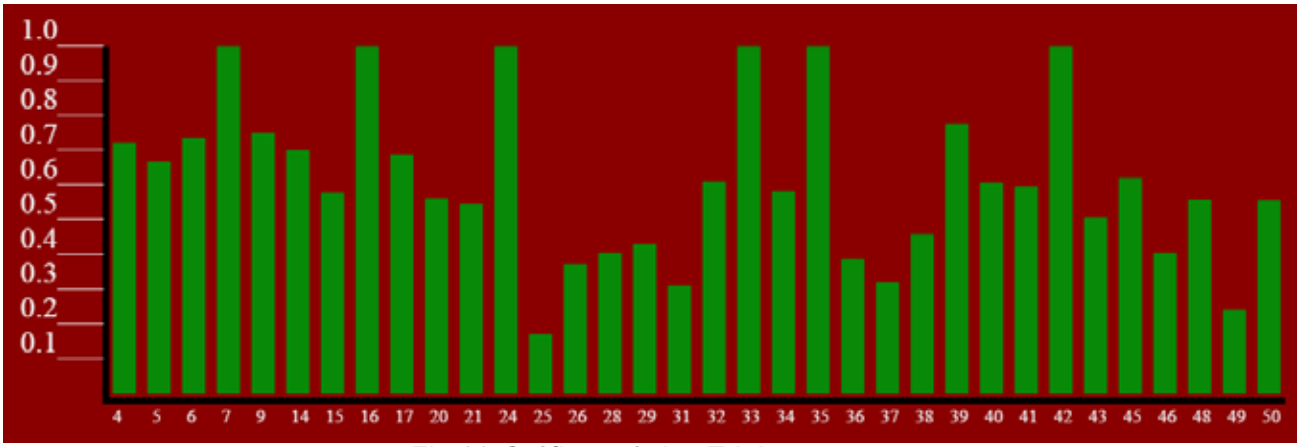


Fig.23 Gráfica métrica Tripletas

8.- Conclusiones

En este proyecto se implementó un sistema web, creado con JSP´s para la comunicación con clases java, el cual integra tres métricas de similitud (estadística, sintáctica y semántica) para el cálculo de estas, además de comunicarse con una base de datos de la cual obtiene los texto y realiza el almacenamiento de la similitud encontrada en cada par de textos, agregando la presentación de un reporte donde se observa las similitudes calculadas.

Con los 50 pares de textos que se utilizaron para realizar los cálculos de similitud se obtuvo el promedio de cada métrica y se comparó con el promedio de las similitudes ideales obteniendo así la métrica más cercana al valor ideal la cual fue la similitud por Frases, la peor métrica fue la de Tripletas. La métrica por Tripletas en este caso resultó ser la peor debido a que esta similitud requiere de una estructura mayor que las demás FN FV (FN || FP) (tripleta) y los textos al ser cortos no es suficiente para que se conforme la tripleta, la métrica de Frases resultó mejor que la métrica de Tf-idf debido a que solo trabaja con ciertas palabras (FV, FN, FP) las cuales son las que aportan un cambio al sentido del texto y Tf-idf al trabajar con todas las palabras si alguna es diferente resta similitud aunque se trate por ejemplo de un sinónimo.

La utilidad de este proyecto conlleva a muchos beneficios debido a que, al determinar el grado de similitud, se pueden clasificar textos científicos, detectar plagio entre ellos, facilitar la localización de referencias a trabajos similares y generar redes temáticas.

9.- Referencias Bibliográficas

[1].Morán Torres, Roberto Iván,*" Sistema de detección de plagio en archivos de texto"*, proyecto terminal, División de Ciencias Básica e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2013.

[2]. S. M. Ugalde Chávez, *"Sistema de recuperación de información semántico"*, proyecto terminal, División de Ciencias Básica e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[3].L. E. García Rodríguez, "*Obtención de una medida cuantitativa de similitud de códigos fuente escritos en lenguaje C*", proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2007.

[4].M. A. Mirón Bernal, *"ANÁLISIS DE SIMILITUD ENTRE PROGRAMAS DE ALTO NIVEL*", tesis, Centro de investigación en computación, Instituto Politécnico Nacional, México, 2008

[5].Chang, J.W, Lee M.C, WangT.I, Su C.Y, and Hsieh T.C.,*"Using grammar patterns to evaluate semantic similarity for short texts"*, Computing Technology and Information Management (ICCM), vol. 2, pp.548-553, 2012.

[6].iThenticate: Plagiarism Detection Software [online],http://www.ithenticate.com, 2014

[7].Turnitin - Originality Check [online],http://turnitin.com/en_us/features/originalitycheck, 2014

[8] Courtney Corley, Rada Mihalcea," *Measuring the Semantic Similarity of Texts*", Department of Computer Science,University of North Texas,pp. 13-18,2005.

[9] Saad, S.M. ; Kamarudin, S.S. ,"*Comparative analysis of similarity measures for sentence level semantic measurement of text*",Control System Computing and Engineering (ICCSCE), pp. 90 – 94,2013.

[10] Keliang JIA, Jibin FU, Xiaoyu JIANG, Jintao MAO,"*Semantic Similarity Computation Based on HowNet2008*",Natural Language Processing and Knowledge Engineering, pp. 1-5, 2008

[11] "P*rocesamiento de lenguaje natural".* [Online].Disponible: http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/232/A4.pdf?sequence=4

[12]Maximino J. Ruiz Rufino," *ANÁLISIS GRAMATICAL.TERMINOLOGÍA Y ABREVIATURAS*". [Online]. Disponible:https://www.hf.uio.no/ilos/tjenester/kunnskap/sprak/nettsprak/spansk/portal/spa2101/tekster/fellesundervisning/2011/apoyo1.pdf

[13]SULEMA TORRES RAMOS,"Optimización global de coherencia en la desambiguación del sentido de las palabras",doctorado,Centro de investigación en computación, Instituto Politécnico Nacional,Mexico,2009

10.- Anexos

10.1.- Clases Java

Clase Similitud

```
package proyecto;
import java.io.IOException;
import java.util.StringTokenizer;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.regex.Pattern;
import org.annolab.tt4j.TreeTaggerException;

/**
 *
 * @author LEG
 */
public class Similitud {
    public double Similitud(String pub1,String pub2,String tipo_sim) throws IOException, TreeTaggerException{
        long t0 = System.currentTimeMillis();
        ArrayList<ArrayList<ArrayList>> publi1=new ArrayList<ArrayList<ArrayList>>();
        ArrayList<ArrayList<ArrayList>> publi2=new ArrayList<ArrayList<ArrayList>>();
        ArrayList<ArrayList> tag1=new ArrayList<ArrayList>();
        ArrayList<ArrayList> tag2=new ArrayList<ArrayList>();
        ArrayList cadena=new ArrayList();
        TreeTaggerTest tagger=new TreeTaggerTest();
        double finalp=0;
        if(!tipo_sim.equals("tfidf")){
//************************para publicacion 1
          String nueva_cad = separar(pub1);
          cadena=token(nueva_cad);
          tag1=tagger.TreeTagger(cadena);
          //-----------------------para publicacion 2
```

```
            nueva_cad = separar(pub2);
            cadena=token(nueva_cad);
            tag2=tagger.TreeTagger(cadena);
            /*System.out.println("texto 1 etiquetado");
            imprimir1(tag1);
            System.out.println("texto 2 etiquetado");
            imprimir2(tag2);*/
          }
          System.out.println("Realizando calculos, por favor espere...");
switch(tipo_sim){
              case "frase":
System.out.println("Calculo frases");
Frases f1=new Frases();
                  Frases f2=new Frases();
                  publi1=f1.cal_frase(tag1);
                  publi2=f2.cal_frase(tag2);
                  //imprimir1(publi1);
                  //imprimir1(publi2);
                  finalp=sim_ws4j(publi1,publi2);
                  break;
              case "tripleta":
                  System.out.println("Calculo tripletas");
                  Tripleta t1=new Tripleta();
                  Tripleta t2=new Tripleta();
                  publi1=t1.calc_tripleta(tag1);
                  publi2=t2.calc_tripleta(tag2);
                  finalp=csim_ws4j(publi1,publi2);
                  /*System.out.println("Tripletas de text1");
                  imprimir1(publi1);
                  System.out.println("Tripletas de text2");
                  imprimir1(publi2);*/
                  break;
              default:
                  System.out.println("Calculo Tf-idf");
```

```
            Tf_idf tf1=new Tf_idf();
String nueva_cad = quitar(pub1);//quitar caracteres especiales
            String nueva_cad2 = quitar(pub2);//y colocar espacios
            String p="[ ]{1,}"; //patron para espacios
            Pattern p2=Pattern.compile(p);
            ArrayList aux1= new ArrayList(Arrays.asList(p2.split(nueva_cad)));//crear ArrayList a partir de espacios
ArrayList aux2= new ArrayList(Arrays.asList(p2.split(nueva_cad2)));
            finalp=tf1.cal_tfidf(aux1, aux2);
            break;
        }
        long t1 = System.currentTimeMillis();
        System.out.println("Resultado : "+finalp);
        System.out.println( "Done in "+(t1-t0)/1000+" sec." );
        return finalp;
    }
    private void imprimir1(ArrayList<ArrayList<ArrayList>> a1){
        int i;
        for(i=0;i<a1.size();i++){
            System.out.println(a1.get(i));
        }
    }
    private void imprimir2(ArrayList<ArrayList> a1){
        int i;
        for(i=0;i<a1.size();i++){
            System.out.println(a1.get(i));
        }
    }
    private static ArrayList<String> token (String cad){
        ArrayList cadena=new ArrayList();
        StringTokenizer st=new StringTokenizer(cad," ");//realizar separacion por 'espacio'
         while(st.hasMoreTokens()){
            cadena.add(st.nextToken());
         }
```

```
        return cadena;
    }
    private static String separar(String cad){
        char aux;
        String c=",?!.-_;'´*¨°|¡':{}[]+*=)(/&%$#\"";
        String nueva="";
         int i=0,j=0,f=0;
         for(i=0;i<cad.length();i++){
            aux=cad.charAt(i);
            for(j=0;j<c.length();j++){
                if(c.charAt(j)==aux){
                    f=1;break;
                }
            }
            if(f==1){
                if(aux!='\")
nueva=nueva+" "+aux+" "; //agregamos espacio
            }
            else{nueva=nueva+aux;}//simplemente agregamos
f=0;
         }
        return nueva;
    }
    private static String quitar(String cad){
        char aux;
        String c=",?!.-_;'´*¨°|¡':{}[]+*=)(/&%$#\"";
        String nueva="";
         int i=0,j=0,f=0;
         for(i=0;i<cad.length();i++){
            aux=cad.charAt(i);
            for(j=0;j<c.length();j++){
                if(c.charAt(j)==aux){
                    f=1;break;
                }
```

```
            }
            if(f==1){
               if(aux!='\'')
nueva=nueva+" "; //si es un caracter especial' lo reemplazamos
            }
            else nueva=nueva+aux;
            f=0;
}
       return nueva;
   }
   private double sim_ws4j(ArrayList<ArrayList<ArrayList>> publi1,ArrayList<ArrayList<ArrayList>>
publi2){
       int i,j,k,x,y;
       double val=0,prom=0,prom2=0,finalp=0;
       double frs[]={1,1,1};
       SimililudWS4J ws4j =new SimililudWS4J();
       for(i=0;i<3;i++){
            prom2=0;
            for(j=0;j<publi1.get(i).size();j++){
               ArrayList listval=new ArrayList();
               for(x=0;x<publi2.get(i).size();x++){
                  if(publi1.get(i).get(j).size()==publi2.get(i).get(x).size()){//calcular solo diagonal
                     for(k=0,prom=0;k<publi1.get(i).get(j).size();k++){
val=ws4j.WS4Jsim(publi1.get(i).get(j).get(k).toString(),publi2.get(i).get(x).get(k).toString());
                        prom+=val;
                     }
                     prom=(double)prom/(publi1.get(i).get(j).size());
                  }
                  else{
                    for(k=0,prom=0;k<publi1.get(i).get(j).size();k++){
                       for(y=0;y<publi2.get(i).get(x).size();y++){
val=ws4j.WS4Jsim(publi1.get(i).get(j).get(k).toString(),publi2.get(i).get(x).get(y).toString());
                          prom+=val;
                       }
```

```
                    }
                    prom=(double)prom/(publi1.get(i).get(j).size()*publi2.get(i).get(x).size());
                   }
                   listval.add(prom);
                   if(prom==1)break;
                 }//fin for x
                 if(publi2.get(i).size()!=0){
                   listval.sort(null);
                   prom2+=(double)listval.get(listval.size()-1);
}
             }//fin for j
                //System.out.println("Similitud total de frase de text1 con text2");
if(publi1.get(i).size()!=0 && publi2.get(i).size()!=0) frs[i]=prom2/publi1.get(i).size();
               //System.out.println("-----> "+frs[i]);
       }//fin for i
       finalp=(frs[0]+frs[1]+frs[2])/3;
//System.out.println("Similitud calculada: "+finalp);
return finalp;
   }
   private double csim_ws4j(ArrayList<ArrayList<ArrayList>> publi1,ArrayList<ArrayList<ArrayList>>
publi2){
       int i=0,j=0,k=0,w=0,x=0,y=0,c2=0;
       double val=0,prom1=0,max1=0,prom2=0,prom3=0;
       //ArrayList simtrip1=new ArrayList<>();
       SimililudWS4J ws4j =new SimililudWS4J();
       for(i=0;i<publi1.size();i++){
          ArrayList m1=new ArrayList<>();
          for(w=0;w<publi2.size();w++){
             prom2=0;c2=0;
             for(j=0;j<publi1.get(i).size();j++){
                ArrayList p1=new ArrayList<>();
                for(x=0;x<publi2.get(w).size();x++){
prom1=0;
                   //si tienen el mismo tamaño solo calculamos la diagonal
```

```
if(publi1.get(i).get(j).size()==publi2.get(w).get(x).size()){
                    for(k=0;k<publi1.get(i).get(j).size();k++){

val=ws4j.WS4Jsim(publi1.get(i).get(j).get(k).toString(),publi2.get(w).get(x).get(k).toString());
                        prom1=prom1+val;
                    }
                    prom1=prom1/publi1.get(i).get(j).size();//obtenemos promedio
}
                else{//si el tamaño es diferente
for(k=0;k<publi1.get(i).get(j).size();k++){
                        for(y=0;y<publi2.get(w).get(x).size();y++){
val=ws4j.WS4Jsim(publi1.get(i).get(j).get(k).toString(),publi2.get(w).get(x).get(y).toString());
                            prom1=prom1+val;
                        }
                    }
                    prom1=prom1/(publi1.get(i).get(j).size() * publi2.get(w).get(x).size());
                }
                p1.add(prom1);
                if(prom1==1)break;
            }//fin for x
            p1.sort(null);
            max1=(double) p1.get(p1.size()-1);//obtener el ultimo
            prom2=prom2+max1;c2++;
        }//fin for j
        m1.add(prom2/c2);
        if((prom2/c2)==1)break;
    }//fin for w
    if(publi2.size()!=0){
        m1.sort(null);
        //simtrip1.add("tripleta "+(i+1)+" : "+ m1.get(m1.size()-1));
        prom3=prom3 + (double)m1.get(m1.size()-1);
    }
}//fin for i
/*System.out.println("Similitud en tripletas");
```

```
        System.out.println(simtrip1);
System.out.println("Similitud de los textos");
System.out.println("----> "+prom3/publi1.size());*/
        if(publi1.size()!=0 && publi2.size()!=0)
            return prom3/publi1.size();
        else
            return 0;
}
}
```

Clase Tf_idf

```
package proyecto;
import java.util.ArrayList;
/**
 *
 * @author LEG
 */
public class Tf_idf {
    public double cal_tfidf(ArrayList pub1,ArrayList pub2){
        int i,ind,c=0;
        double prom=0;
        ArrayList menor=new ArrayList<>();
        ArrayList mayor=new ArrayList<>();
        ArrayList aux1=new ArrayList<>();
        ArrayList auxf1=new ArrayList<>();
        ArrayList auxf2=new ArrayList<>();

        if(pub1.size()<pub2.size()){menor=pub1;mayor=pub2;}
        else{menor=pub2;mayor=pub1;}

for(i=0;i<menor.size();i++){//recorremos menor
            String a=menor.get(i).toString();//obtenemos palabra de menor
            ind=aux1.indexOf(a.toLowerCase());//buscamos si ya existe en el arraylist aux1
            if(ind<0){//no existe en aux1
                aux1.add(a.toLowerCase());//agregamos en minusculas la palabra
                auxf1.add(1);//primera vez que lo agregamos
                auxf2.add(0);//inicializacion del otro array
```

```
            c++;//contador
          }
          else {//ya existe en aux1
auxf1.set(ind,(int)auxf1.get(ind)+1);//incrementamos
}
      }
    /*System.out.println("palabras del texto mas corto");
System.out.println(aux1);*/
for(i=0;i<mayor.size();i++){//recorremos mayor
String a=mayor.get(i).toString();//obtenemos palabra de mayor
          ind=aux1.indexOf(a.toLowerCase());//buscamos si existe en aux1
if(ind>=0){//si existe
            auxf2.set(ind,(int)auxf2.get(ind)+1);//incrementamos
}
          else{c++;}
      }
/*System.out.println("\nFrecuencia de apariciones en texto 1");
      System.out.println(auxf1);
      System.out.println("\nFrecuencia de apariciones en texto 2");
System.out.println(auxf2);*/
      for(i=0;i<auxf1.size();i++){//recorremos el array
          if((int)auxf1.get(i)<(int)auxf2.get(i)){//buscamos cual es el menor y mayor
            prom+=(int)auxf1.get(i)/(int)auxf2.get(i);
          }
          else{
            prom+=(int)auxf2.get(i)/(int)auxf1.get(i);
          }
      }
      System.out.println("\nSimilitud: "+prom/c);
return prom/c;
   }
}
```

Clase Frases

```
package proyecto;

import java.util.ArrayList;
```

```
import java.util.HashSet;
import java.util.Iterator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 *
 * @author LEG
 */
public class Frases {
    private ArrayList<ArrayList> indp =new ArrayList<ArrayList>();
    private ArrayList<ArrayList> indp1=new ArrayList<ArrayList>();
    private ArrayList<ArrayList> indp2=new ArrayList<ArrayList>();

    private void setindp(ArrayList aux){
        ArrayList aux1=new ArrayList();
        aux1=(ArrayList)aux.clone();
        indp.add(aux1);
    }
    private void setindp1(ArrayList aux){
        ArrayList aux1=new ArrayList();
        aux1=(ArrayList)aux.clone();
        indp1.add(aux1);
    }
    private void setindp2(ArrayList aux){
        ArrayList aux1=new ArrayList();
        aux1=(ArrayList)aux.clone();
        indp2.add(aux1);
    }
    private int calcular_frase(String patron,int i,ArrayList frase,ArrayList<ArrayList> tag){
        Pattern p=Pattern.compile(patron);
        if(i<tag.size()){
            Matcher m=p.matcher((String)tag.get(i).get(1));
            if(m.matches()){
                frase.add(tag.get(i).get(2));
                return calcular_frase(patron,i+1,frase,tag);
            }
```

```
            else
                return i;
        }
        else return i+1;
    }
    public ArrayList<ArrayList<ArrayList>> cal_frase(ArrayList<ArrayList> tag){
        ArrayList<ArrayList<ArrayList>> matriz=new ArrayList<ArrayList<ArrayList>>();
        String p_v="^V.{0,}";
        Pattern r_v=Pattern.compile(p_v);
        String p_n="^N.{0,}";
        Pattern r_n=Pattern.compile(p_n);
        String p_p="^I.{0,}";
        Pattern r_p=Pattern.compile(p_p);
        int i,j;
        for(i=0;i<tag.size();i++){
            ArrayList fr=new ArrayList();
            Matcher m_v=r_v.matcher(tag.get(i).get(1).toString());
            Matcher m_n=r_n.matcher(tag.get(i).get(1).toString());
            Matcher m_p=r_p.matcher(tag.get(i).get(1).toString());

            if(m_v.matches()){
                fr.add(tag.get(i).get(2));
j=calcular_frase(p_v,i+1,fr,tag);
if((j-i)<4){setindp(fr);}
                i=j-1;
            }else   if(m_n.matches()){
                        fr.add(tag.get(i).get(2));
j=calcular_frase(p_n,i+1,fr,tag);
if((j-i)<6){setindp1(fr);}
                    i=j-1;
                }
                else    if(m_p.matches()){
                            fr.add(tag.get(i).get(2));
                            i++;
                            Matcher m_n2=r_n.matcher((String)tag.get(i).get(1));
                            if(m_n2.matches()){
                                fr.add(tag.get(i).get(2));
```

```
j=calcular_frase(p_n,i+1,fr,tag);
if((j-i)<6){setindp2(fr);}
                                i=j-1;
                            }
                            else i--;
                        }
        }
        matriz.add(indp);//FV
        matriz.add(indp1);//FN
        matriz.add(indp2);//FP
return matriz;
    }
}
```

Clase Tripleta

```
package proyecto;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 *
 * @author LEG
 */
public class Tripleta {
    private ArrayList<ArrayList> indp=new ArrayList<ArrayList>();
    private ArrayList<ArrayList> indp1=new ArrayList<ArrayList>();
    private ArrayList<ArrayList> indp2=new ArrayList<ArrayList>();
    private ArrayList<ArrayList> ini_fin=new ArrayList<ArrayList>();

    private ArrayList<ArrayList<ArrayList>> matriz_trip=new ArrayList<ArrayList<ArrayList>>();

    public ArrayList<ArrayList<ArrayList>> calc_tripleta(ArrayList<ArrayList> tag){
        int i,j;
        boolean flag=false;
```

```
        String patron_fv="^V.{0,}";
        String patron_fn="^N.{0,}";
        String patron_fp="^I.{0,}";
        Pattern rn=Pattern.compile(patron_fn);
        Pattern rv=Pattern.compile(patron_fv);
        Pattern rp=Pattern.compile(patron_fp);
        for(i=0;i<tag.size();i++){
           flag=false;
           ArrayList fn=new ArrayList<>();
          Matcher mn=rn.matcher((String)tag.get(i).get(1));
          Matcher mp0=rp.matcher((String)tag.get(i).get(1));
          if(mn.matches()){
             fn.add(tag.get(i).get(2));//agregamos pos
j=calcular_frase(patron_fn,i+1,fn,tag);
if((j-i)<6){
              for(i=j;i<tag.size();i++){
                 ArrayList fv=new ArrayList<>();
                 Matcher mv=rv.matcher((String)tag.get(i).get(1));
                 Matcher mn2=rn.matcher((String)tag.get(i).get(1));
                 Matcher mp=rp.matcher((String)tag.get(i).get(1));
                 if(mv.matches()){
                    fv.add(tag.get(i).get(2));//agregamos pos
j=calcular_frase(patron_fv,i+1,fv,tag);
if((j-i)<4){
                       for(i=j;i<tag.size();i++){
                          ArrayList fp=new ArrayList<>();
                          ArrayList fn2=new ArrayList<>();
                          ArrayList<ArrayList> tripleta=new ArrayList<ArrayList>();
                          Matcher mv3=rv.matcher((String)tag.get(i).get(1));
                          Matcher mp2=rp.matcher((String)tag.get(i).get(1));
                          Matcher mn3=rn.matcher((String)tag.get(i).get(1));
                          if(mp2.matches() && (i+1)<tag.size()){
                             fp.add(tag.get(i).get(2));//agregamos pos
                             i++;
                             Matcher mn4=rn.matcher((String)tag.get(i).get(1));
                             if(mn4.matches()){
                                fp.add(tag.get(i).get(2));//agregamos pos
```

```
j=calcular_frase(patron_fn,i+1,fp,tag);
if((j-i)<6){
                                    i=j-1;
                                    //crear Tripleta
                                    tripleta.add(fn);
                                    tripleta.add(fv);
                                    tripleta.add(fp);
                                    matriz_trip.add(tripleta);
                                    flag=true;break;
                                }
                                 else {i=j-1;}
                            }else i--;
                        }else if(mn3.matches()){
                              fn2.add(tag.get(i).get(2));//agregamos pos
j=calcular_frase(patron_fn,i+1,fn2,tag);
if((j-i)<6){
                                    i=j-1;
                                    //crear Tripleta
                                    tripleta.add(fn);
                                    tripleta.add(fv);
                                    tripleta.add(fn2);
                                    matriz_trip.add(tripleta);
                                    flag=true;break;
                                }
                                else {i=j-1;}
                            }
                            else if(mv3.matches()){
                                ArrayList fv2=new ArrayList<>();
j=calcular_frase(patron_fv,i+1,fv2,tag);
if((j-i)<4){
                                    i=j-1;
                                    flag=true;break;
                                }
                                else{i=j-1;}
                            }
                    }//fin for
                }
```

```
                    else {i=j-1;}
                }//fin if(mv)
                else if(mn2.matches()){i--;break;}
                    else if(mp.matches() && (i+1)<tag.size()){
                            ArrayList fp=new ArrayList<>();
                            i++;
                            Matcher mn4=rn.matcher((String)tag.get(i).get(1));
                            if(mn4.matches()){
                                j=calcular_frase(patron_fn,i+1,fp,tag);
                                if((j-i)<6){i=j-1;break;}
                                else{i=j-1;}
                            }
                            i--;
                        }
                if(flag)break;
}//fin for
            }
            else {i=j-1;}//ya que el ciclo for aumentará en uno
}//fin if(fn)
        else if(mp0.matches() && (i+1)<tag.size()){
                ArrayList fp=new ArrayList<>();
                i++;
                Matcher mn0=rn.matcher((String)tag.get(i).get(1));
                if(mn0.matches()){
                    j=calcular_frase(patron_fn,i+1,fp,tag);
                    i=j;
                }
                i--;
             }
    }//fin for
    return matriz_trip;
  }
  private int calcular_frase(String patron,int i,ArrayList frase,ArrayList<ArrayList> tag){
    Pattern p=Pattern.compile(patron);
    if(i<tag.size()){
      Matcher m=p.matcher((String)tag.get(i).get(1));
      if(m.matches()){
```

```
            frase.add(tag.get(i).get(2));
            return calcular_frase(patron,i+1,frase,tag);
          }
          else
            return i;
}
      else return i+1;
   }
}
```

Clase ConexionDB

```
package proyecto;
import java.sql.*;
public class ConexionDB {
        private Connection con = null;
        public ConexionDB() throws InstantiationException, IllegalAccessException {
                try {
                        Class.forName("com.mysql.jdbc.Driver").newInstance();
                }
                   catch (ClassNotFoundException e)
                   {
                        e.printStackTrace();
                        }
                try{
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/proyecto",”root","");
                }
                catch (SQLException e)
                {
                        e.printStackTrace();
                }
        }
        public Connection getConexion(){
                return con;
        }
        public void cerrarConexion(){
                try {
```

```
                con.close();
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
}
```

Clase Publicacion

```
package proyecto;

/**
 *
 * @author LEG
 */
public class Publicacion {
    private String titulo;
    private String descripcion;
    private String id;
    private String porcentaje="a";

   public void settitulo(String tit){
       this.titulo=tit;
   }
   public void setdescripcion(String des){
       this.descripcion=des;
   }
   public void setid(String idc){
       this.id=idc;
   }
   public void setporcentaje(String p){
       this.porcentaje=p;
   }

   public String toString(){
       if("a".equals(porcentaje)){
```

```
                return "<tr onMouseOver='color(this)' onMouseout='color_a(this)' id='"+id+"' onclick='sel_radio(this.id)' ><td><input type='radio' name='publicaciones' id='r"+id+"'></td><td><div id='t"+id+"'>"+ titulo + "</div></td></tr>";//<td>" + descripcion+"</td>
      }
      else
          return porcentaje;
  }
  public String toPublicacion(){
return descripcion;
  }
}
```

Clase PublicacionDAO

```
package proyecto;
import java.io.IOException;
import java.io.Serializable;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Vector;
import org.annolab.tt4j.TreeTaggerException;
import org.apache.xmlrpc.XmlRpcClient;
/**
 *
 * @author LEG
 */
public class PublicacionDAO implements Serializable {

    public String seleccionar(String ini) throws InstantiationException, IllegalAccessException{
        ArrayList<Publicacion> ap=new ArrayList<Publicacion>();
            try {
                ConexionDB c=new ConexionDB();
                Connection con=c.getConexion();
                if(con!=null){
```

```
                    Statement st;
                    st = con.createStatement();
                ResultSet r=st.executeQuery("SELECT id,titulo,descripcion FROM textos Where id >='"+ini+"' limit 20;");//solo obtenermos 20 registros
                    while(r.next()){
                Publicacion p=new Publicacion();
                p.settitulo(r.getString("titulo"));
                p.setdescripcion(r.getString("descripcion"));
                p.setid(r.getString("id"));
                ap.add(p);//agregamos al array
            }
                    st.close();
                }
                c.cerrarConexion();
            } catch (SQLException e) {
                e.printStackTrace();
            }

        return ap.toString();

    }
    public String publicacionC(String id) throws InstantiationException, IllegalAccessException{
        Publicacion p=new Publicacion();
        try {
                ConexionDB c=new ConexionDB();
                Connection con=c.getConexion();
                if(con!=null){
                    Statement st;
                    st = con.createStatement();
                ResultSet r=st.executeQuery("SELECT descripcion FROM textos Where id='"+id+"' limit 1;");
                    while(r.next()){
                p.setdescripcion(r.getString("descripcion"));
            }
                    st.close();
                }
                c.cerrarConexion();
            } catch (SQLException e) {
```

```
                e.printStackTrace();
            }

        return p.toPublicacion();
    }

    public int num() throws InstantiationException, IllegalAccessException{//unicamente obtenemos cantidad de registros
        int n=0;
            try {
                ConexionDB c=new ConexionDB();
                Connection con=c.getConexion();
                if(con!=null){
                    Statement st;
                    st = con.createStatement();
                ResultSet r=st.executeQuery("SELECT id FROM textos;");
                r.last();
                n=r.getRow();
                    st.close();
                }
                c.cerrarConexion();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        return n;
    }

    public int num_sim(String similitud) throws InstantiationException, IllegalAccessException{//unicamente obtenemos cantidad de registros
        int n=0;
            try {
                ConexionDB c=new ConexionDB();
                Connection con=c.getConexion();
                if(con!=null){
                    Statement st;
                    st = con.createStatement();
                ResultSet r=st.executeQuery("SELECT id FROM similitud where "+similitud+"!='0';");
```

```
                    r.last();
                    n=r.getRow();
                                st.close();
                        }
                        c.cerrarConexion();
                } catch (SQLException e) {
                        e.printStackTrace();
                }
            return n;
        }
    public String sel_sim(String column,String cond) throws InstantiationException, IllegalAccessException{
            ArrayList<Publicacion> ap=new ArrayList<Publicacion>();
                try {
                        ConexionDB c=new ConexionDB();
                        Connection con=c.getConexion();
                        if(con!=null){
                                Statement st;
                                st = con.createStatement();
                    ResultSet r=st.executeQuery("SELECT "+column+" FROM similitud Where "+cond+" !='0';");
                        while(r.next()){
                    Publicacion p=new Publicacion();
                    p.setporcentaje(r.getString(column));
                    ap.add(p);
                }
                st.close();
                        }
                        c.cerrarConexion();
                } catch (SQLException e) {
                        e.printStackTrace();
                }
        return ap.toString();
        }
    public String lista_sim(String column,String pub) throws InstantiationException, IllegalAccessException{
            ArrayList ap=new ArrayList<>();
                try {
```

```
                    ConexionDB c=new ConexionDB();
                    Connection con=c.getConexion();
                    if(con!=null){
                            Statement st;
                            st = con.createStatement();
                ResultSet r=st.executeQuery("SELECT id_pub1,id_pub2 FROM similitud
Where "+column+" !='0';");
                 while(r.next()){
               if(pub.equals("pub1")){
                  Statement st1;
                  st1=con.createStatement();
                  ResultSet r2=st1.executeQuery("SELECT titulo FROM textos WHERE
id="+r.getString("id_pub1")+";");
                  while(r2.next()){
                     ap.add(r2.getString("titulo"));
                  }
                  st1.close();
               }
               else{
                  Statement st1;
                  st1=con.createStatement();
                  ResultSet r2=st1.executeQuery("SELECT titulo FROM textos WHERE
id="+r.getString("id_pub2")+";");
                  while(r2.next()){
                     ap.add(r2.getString("titulo"));
                  }
                  st1.close();
               }
            }
            st.close();
             }
                    c.cerrarConexion();
             } catch (SQLException e) {
                    e.printStackTrace();
             }
        System.out.println(ap);
    return ap.toString();
      }
```

```
  public void GuardarSim(String sim,double val,int id_pub1,int id_pub2) throws InstantiationException, IllegalAccessException{
      int n=0;
      try {
          ConexionDB c=new ConexionDB();
          Connection con=c.getConexion();
          if(con!=null){
              Statement st,st2,st3;
              st = con.createStatement();
              st2 = con.createStatement();
              st3 = con.createStatement();
              ResultSet r=st.executeQuery("Select id from similitud Where id_pub1="+id_pub1+" OR id_pub2="+id_pub2+";");
              r.last();
              n=r.getRow();
              System.out.println("regs"+n);
              if(n<1){
                 String ins="Insert into similitude (id_pub1,id_pub2,"+sim+")VALUES("+id_pub1+","+id_pub2+","+val+");";
                 System.out.println(ins);
                 //ResultSet r2=st2.executeQuery(ins);
                 System.out.println(st2.executeUpdate(ins));
              }
              else{
                 r.first();
                 String upd="UPDATE similitud SET "+sim+"="+val+" Where id="+r.getString("id")+";";
                 System.out.println(upd);
                 System.out.println(st3.executeUpdate(upd));
}
            st.close();
            st2.close();
            st3.close();
}
          c.cerrarConexion();
      } catch (SQLException e) {
          e.printStackTrace();
      }
```

```
   }
}
```

Clase SimilitudWS4J

```
package proyecto;
import edu.cmu.lti.lexical_db.ILexicalDatabase;
import edu.cmu.lti.lexical_db.NictWordNet;
import edu.cmu.lti.ws4j.RelatednessCalculator;
import edu.cmu.lti.ws4j.impl.HirstStOnge;
import edu.cmu.lti.ws4j.impl.JiangConrath;
import edu.cmu.lti.ws4j.impl.LeacockChodorow;
import edu.cmu.lti.ws4j.impl.Lesk;
import edu.cmu.lti.ws4j.impl.Lin;
import edu.cmu.lti.ws4j.impl.Path;
import edu.cmu.lti.ws4j.impl.Resnik;
import edu.cmu.lti.ws4j.impl.WuPalmer;
import edu.cmu.lti.ws4j.util.WS4JConfiguration;
public class SimililudWS4J {

   private static ILexicalDatabase db = new NictWordNet();

      private static RelatednessCalculator[] rcs = {

         //  ,,new JiangConrath(db),
           new Path(db)
            //new Lesk(db),
           //new Lin(db)
            //new WuPalmer(db)
            //new HirstStOnge(db)
         //new LeacockChodorow(db)
         //new Resnik(db)
      };
      //private static void run( String word1, String word2 ) {
      public double WS4Jsim(String word1,String word2){
            WS4JConfiguration.getInstance().setMFS(false);
            double s=0;
            for ( RelatednessCalculator rc : rcs ) {
```

```
            s = rc.calcRelatednessOfWords(word1, word2);
            if(s>1)s=1;
              //System.out.println( rc.getClass().getName()+"\t"+s );
        }
        return s;
    }
    /*public static void main(String[] args) {

        long t0 = System.currentTimeMillis();

        run("horse","house");

        long t1 = System.currentTimeMillis();

        System.out.println( "Done in "+(t1-t0)+" msec." );

}*/
}
```

Clase TreeTaggerTest

```
package proyecto;
import java.io.IOException;
import java.util.ArrayList;
import org.annolab.tt4j.*;
import static java.util.Arrays.asList;
public class TreeTaggerTest {
        public ArrayList<ArrayList> TreeTagger(ArrayList cadena) throws IOException,
TreeTaggerException {
        ArrayList<ArrayList>res=new ArrayList<ArrayList>();
        System.setProperty("treetagger.home",
"C:\\Users\\LEG\\Documents\\NetBeansProjects\\ProyectoIntegracion\\TreeTagger");
        TreeTaggerWrapper<String> tt = new TreeTaggerWrapper<String>();
        try {

tt.setModel("C:\\Users\\LEG\\Documents\\NetBeansProjects\\ProyectoIntegracion\\TreeTagger\\mo
delos\\english.par:iso8859-1");
            tt.setHandler(new TokenHandler<String>() {
```

```
                    public void token(String token, String pos, String lemma) {
                        ArrayList dat=new ArrayList();
                        dat.add(token);
dat.add(pos);
                        dat.add(lemma);//palabra sin conjugar
res.add(dat);
                        //System.out.println(token + "\t" + pos + "\t" + lemma);
                    }
                });
                tt.process(cadena);
            }
            finally {
                tt.destroy();
}
        return res;
            }
}
```

10.2.- Páginas Sistema Web

calcular.jsp

```
<%@page language="java" import="proyecto.PublicacionDAO" import="proyecto.Similitud" pageEncoding="ISO-8859-1"%>
<jsp:useBean id="pDAO" class="proyecto.PublicacionDAO" scope="page">
<jsp:setProperty name="pDAO" property="*"></jsp:setProperty>
</jsp:useBean>
<jsp:useBean id="Sim" class="proyecto.Similitud" scope="page">
<jsp:setProperty name="Sim" property="*"></jsp:setProperty>
</jsp:useBean>

<%
   String p1=pDAO.publicacionC(request.getParameter("p1"));
   String p2=pDAO.publicacionC(request.getParameter("p2"));
   double res= Sim.Similitud(p1,p2, request.getParameter("s"));
out.println("La similitud entre publi1 y publi2  es de: "+ res);
   //guardar similitud en base de datos
//pDAO.GuardarSim(request.getParameter("s"),
res,Integer.parseInt(request.getParameter("p1")),Integer.parseInt(request.getParameter("p2")));
```

```
   /*Integer i,aux;
   String a,b,param;
   for(i=1;i<=100;i+=2){
      a= i.toString();
      String p1=pDAO.publicacionC(a);
      aux=i+1;
      b= aux.toString();
      String p2=pDAO.publicacionC(b);

      double res= Sim.Similitud(p1,p2,"frase");
      pDAO.GuardarSim("frase", res,i,i+1);
      res= Sim.Similitud(p1,p2,"tripleta");
      pDAO.GuardarSim("tripleta", res,i,i+i);
      res= Sim.Similitud(p1,p2,"tfidf");
      pDAO.GuardarSim("tfidf", res,i,i+i);

}
   out.println("LISTO");*/
%>
```

similitud.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<jsp:useBean id="JSONRPCBridge" scope="session" class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="proyecto1" scope="session"  class="proyecto.PublicacionDAO"/>
<% JSONRPCBridge.registerObject("proyecto", proyecto1); %>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script type="text/javascript" src="../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src="../js/jsonrpc.js"></script>
<script type="text/javascript" src="../js/mostrar.js"></script>
<script type="text/javascript" src="../js/similitud.js"></script>
```

```
<style type="text/css">
        body{
            background-color: #8a0000;
        }
        td{
            color: #bbbbbb;
            font-weight: bold;
            font-size: 22px;
        }
        #titulo{
            width: 80%;
            font-size: 44px;
            font-weight: bold;
            color:#dddddd;
        }
        input[type="button"]{
            cursor: pointer;
            width: 170px;
            height: 50px;
            font-size: 18px;
            color:#bbbbbb;
            background-color:black;
        }
        input[type="submit"]{
            cursor: pointer;
            font-size: 14px;
            font-weight: bold;
            width: 100px;
            height: 40px;
            background-color: green;

        }
        #pub1,#pub2,#porcentaje{
            color: yellow;
        }

</style>
```

```
<title>Proyecto de integraci&oacute;n</title>
</head>
<body onLoad="json()" link="#eeeeee">
<center>
<div id="titulo" align="center">Calcular Similitud</div><br>
<div id="publicaciones"></div>
<div id="avance"></div>
<div id="temp"></div><script>document.getElementById("temp").style.display="none";</script>
<br>
</center>
<input type="button" value="Seleccionar" onclick="seleccion()"><br>
<div id="pub">
<table>
<tr><td colspan="2"> </td></tr>
<tr>
<td>
                Publicaci&oacute;n uno:
</td>
<td>
<div id="pub1"></div><input type="hidden" id="tpub1" value="">
</td>
<td>
<input type="hidden" id="tpub1" value="">
</td>
</tr>
<tr><td colspan="2"> </td></tr>
<tr>
<td>
                Publicaci&oacute;n dos:
</td>
<td>
<div id="pub2"></div><input type="hidden" id="tpub2" value="">
</td>
<td>
<input type="hidden" id="tpub2" value="">
</td>
</tr>
```

```
<tr><td colspan="2"> </td></tr>
<tr>
<td>
<input type="button" value="limpiar" onclick="limpiar()">
</td>
<td colspan="2">
<div id="sim">
<input type="button" value="Similitud Frases" onclick="ServSim('frase')">
<input type="button" value="Similitud Tripletas" onclick="ServSim('tripleta')">
<input type="button" value="Similitud Estadistica" onclick="ServSim('tfidf')">
</div>
<script>document.getElementById("sim").style.display="none";</script>
</td>
</tr>
<tr><td> </td></tr>
<tr>
<td colspan="3">
<div id="porcentaje"></div>
</td>
</tr>
<tr><td> </td></tr>
<tr><td colspan="3" align="center"><input type="submit" value="Regresar" onclick="location.href='../'"></td></tr>
<tr><td> </td></tr>
</table>
</div>
</body>
</html>
```

reporte.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<jsp:useBean id="JSONRPCBridge" scope="session" class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="reporte1" scope="session"  class="proyecto.PublicacionDAO"/>
<% JSONRPCBridge.registerObject("proyecto", reporte1); %>
```

```
<html>
<head>
<script type="text/javascript" src="../js/jsonrpc.js"></script>
<script>
        function json(){
           jsonrpc = new JSONRpcClient("/Proyecto/jsonrpc");
}
        function grafica(similitud){
           var id=new Array();
           filas=jsonrpc.proyecto.num_sim(similitud);//obtener cantidad de registros
           //obtener tamaño de pantalla
           var a=20,e=10;//ancho de rectangulo,espacio entre rectangulos
           var ancho= filas*(a+e);  //ancho=#filas*(ancho+espacio) entre rectangulos
           var alto= 300; //alto
           var crear="<canvas id='graf' width='"+(ancho+200)+"' height='"+(alto+100)+"'>Necesita
actualizar su navegador para ver este reporte</canvas>";
           document.getElementById('principal').innerHTML=crear;//creamos etiqueta canvas con
medidas
           var canvas=document.getElementById('graf');
           if(canvas.getContext){
              var reg=jsonrpc.proyecto.sel_sim(similitud,similitud);
              reg=reg.replace(/[\[\]]/g,'');//eliminar corchetes
              var porcent=new Array();
porcent=reg.split(",");//separar en token's
var xi=160,yi=40;//coordenadas de linea vertical de plano
              var al=6;//ancho linea
              var ctx=canvas.getContext('2d');
              //crear x e y
              ctx.lineWidth=al;//ancho de linea
              ctx.beginPath();//indicamos que empezaremos trazo
              ctx.moveTo(xi-al,yi);//x,y punto inicial
              ctx.lineTo(xi-al,alto+yi+al);//linea vertical
              ctx.lineTo(ancho-al+xi,alto+yi+al);//linea horizontal
              ctx.stroke();

              //crear etiquetas en ejes
              ctx.fillStyle="rgb(240,240,240)";
              var aux=alto/10;
```

```
var et1=new Array("0.1___","0.2___","0.3___","0.4___","0.5___","0.6___","0.7___","0.8___","0.9___","1.0___");
                for(i=0;i<10;i++){
                    ctx.font="26px Times New Roman";
                    ctx.fillText(et1[9-i],xi-80,yi+(aux*i)-al);
}



                var reg2=jsonrpc.proyecto.sel_sim("id",similitud);
reg2=reg2.replace(/[\[\]]/g,'');//eliminar corchetes
                id=reg2.split(",");//separar en token's
//fin etiquetas
                //ctx.fillStyle="rgb(1,223,1)";
ctx.font="16px Times New Roman";

                for(i=1;i<=filas;i++){
                    ctx.fillStyle="#088A08";//#04B404
                    ctx.fillRect(xi,alto+yi,a,-porcent[i-1]*alto);//x,y,ancho,alto
                    ctx.fillStyle="#eeeeee";
                    ctx.fillText(id[i-1],xi,alto+yi+25)
xi=xi+a+e;// ancho + espacio entre rectangulos
                }
            }else{
                alert('Su navegador no soporta canvas');
            }
            var pub1=new Array();
            var reg3=jsonrpc.proyecto.lista_sim(similitud,"pub1");
reg3=reg3.replace(/[\[\]]/g,'');//eliminar corchetes
            pub1=reg3.split(",");//separar en token's

var pub2=new Array();
            var reg4=jsonrpc.proyecto.lista_sim(similitud,"pub2");
reg4=reg4.replace(/[\[\]]/g,'');//eliminar corchetes
            pub2=reg4.split(",");//separar en token's

            var tabla="<font color='white' size=4><table border=1 align=center cellpadding=10>\n\
<tr><th colspan='4'>Similitud "+similitud+"</th></tr>\n\
```

```
<tr><th>ID x</th><th>Texto 1</th><th>Texto2</th><th>Porcentaje</th></tr>";
            for(i=1;i<=id.length;i++){
                tabla+="<tr><td>"+id[i-1]+"</td><td>"+pub1[i-1]+"</td><td>"+pub2[i-
1]+"</td><td>"+porcent[i-1]+"</td></tr>";
            }
            tabla+="</table></font>";
            document.getElementById("lista").innerHTML=tabla;


        }


</script>
<style type="text/css">
        body{
            background-color: #8a0000;
        }
        input[type="button"]{
            cursor: pointer;
            width: 170px;
            height: 50px;
            font-size: 18px;
            color:#bbbbbb;
            background-color:black;
        }
        #reg{
            color:black;
            background-color:green;
        }
        table{
            text-align: center;
        }
        th{
            font-size: 24px;
        }
</style>
</head>
<body onload="json()">
<table>
```

```
<tr>
<td><input type="button" value="Tripletas" onclick="grafica('tripleta')"></td>
<td><input type="button" value="Frases" onclick="grafica('frase')"></td>
<td><input type="button" value="Tfidf" onclick="grafica('tfidf')"></td>
<td> </td>
<td><input type="button" value="regresar" onclick="location.href='../'" id="reg" name="reg"></td>
</tr>
</table>
<div id="principal">
</div>
<br><br><br>
<div id="lista">
</div>
</body>
</html>
```

Index.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Proyecto de integraci&oacute;n</title>
<script>
        document.getElementByTagName("input").style.cursor
</script>
<style type="text/css">
        body{
            background-color: #8a0000;
        }
        #titulo{
            width: 80%;
            font-size: 44px;
            font-weight: bold;
            color:#dddddd;
        }
        input[type="button"]{
            cursor: pointer;
            width: 300px;
```

```
            height: 100px;
            font-size: 22px;
            color:#bbbbbb;
background-color:black;

        }
</style>

</head>
<body>
<center>
<div id="titulo">Sistema Web para obtener la similitud entre publicaciones cientificas
            mediante t&eacute;cnicas sem&aacute;nticas</div><br><br><br><br>
<table border="0" cellpadding="20" cellspacing="20" width="50" id="menu">
<tr>
<td><input type="button" name="similitud" value="Calcular Similitud" onclick="javascript:location.href='jsonrpc/similitud.jsp'"></td>
</tr>
<tr>
<td><input type="button" name="reporte" value="Gr&aacute;fica de art&iacute;culos similares" onclick="javascript:location.href='jsonrpc/reporte.jsp'"></td>
</tr>
</table>
</center>
</body>
</html>
```

mostrar.js

```
function json()
{

   jsonrpc = new JSONRpcClient("/Proyecto/jsonrpc");
filas=jsonrpc.proyecto.num();//obtener cantidad de registros
   var i;
   var avance="<table border='0' cellspacing='10' align='center'><tr>";//imprimiremos cantidad de 'hojas' para navegacion
for(i=1;i<=(filas/20);i++){
```

```
        avance+="<td id='"+i+"' onclick='mostrar(this.id)'><a href='javascript:void(0)'>"+i+"</a></td>";
    }
    if((filas%20)!=0)avance+="<td id='"+i+"' onclick='mostrar(this.id)'><a
href='javascript:void(0)'>"+i+"</a></td>";
    avance+="</tr></table>";
    document.getElementById("avance").innerHTML=avance;
    var resul="<table border='1' cellpadding='10' align='center'>";
resul+=jsonrpc.proyecto.seleccionar(1);//primer grupo de resultados
resul+="</table>";
    var resul2=resul.replace(/[,\[\]]/g,'');
document.getElementById("publicaciones").innerHTML=resul2;
}
function mostrar(ini){
    var val2=0;

    //val2---->consulta a partir de este # de id
    //casos base   ----- se mostraran 20 registros
    //si ini===1 permanece igual
if(ini=='2')val2=ini*10;
    if(parseInt(ini)>2){
        var val1=parseInt(ini)-2;
        val2=(parseInt(ini)+val1)*10;
    }
    var resul="<table border='1' cellpadding='10' align='center'>";
resul+=jsonrpc.proyecto.seleccionar(val2+1);
resul+="</table>";
    var resul2=resul.replace(/[,\[\]]/g,'');
document.getElementById("publicaciones").innerHTML=resul2;
}
```

similitud.js

```
var sim=1;
function color(fila){
    fila.style.background="#111111";
}
function color_a(fila){
```

```
    fila.style.background="#8a0000";
}
function sel_radio(radio){
  rad=document.getElementById("r"+radio);
  rad.click();
  var temp=document.getElementById("temp");
  //temp.innerHTML=document.getElementById("t"+radio).innerHTML;
  temp.innerHTML=radio;
}
function seleccion(){
//var temp=document.getElementById("temp");
    var temp=document.getElementById("temp").innerHTML;
    var publica=document.getElementById("pub"+sim);
    if(sim<3){
        publica.innerHTML=document.getElementById("t"+temp).innerHTML;//temp.innerHTML;
        document.getElementById("tpub"+sim).value=temp;//hidden con el valor de id
if(sim==2)document.getElementById("sim").style.display="block";
sim++;
    }
}
function limpiar(){
    var pub1=document.getElementById("pub1");
    var pub2=document.getElementById("pub2");
    var porc=document.getElementById("porcentaje");
    var temp=document.getElementById("temp");
    var tpub1=document.getElementById("tpub1");
    var tpub2=document.getElementById("tpub2");
    pub1.innerHTML="";
pub2.innerHTML="";
    porc.innerHTML="";
    //temp.innerHTML="";
    tpub1.value="";
    tpub2.value="";
    sim=1;
    document.getElementById("sim").style.display="none";
}
function ServSim(simi){//similitud a realizar
```

```
    var res=document.getElementById("porcentaje");
    //obtenr id's de las publicaciones
    var pub1=document.getElementById("tpub1").value;
    var pub2=document.getElementById("tpub2").value;

    res.innerHTML="<table border=0 cellpadding='10'><tr><td><img src='../img/reloj.gif' width='110px' height='120px' title='Procesando...' alt='Procesando...'></td></tr><tr><td>Calculando Similitud, Por favor espere</td></tr></table>";
$.post("calcular.jsp",{p1:pub1,p2:pub2,s:simi},function(data){
res.innerHTML=data;
    });
}
```