Universidad Nacional de La Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas (DIIT)

Código: C151

Título del Proyecto: Implementación de un Sistema de Recuperación de la Información

Programa de Investigación: *PROINCE*

Director del Proyecto: *Ryckeboer, Hugo Emilio Julio Ludovico*

Integrantes del Proyecto:

Spositto, Osvaldo Mario (Co-Director)

Barone, Miriam Andrea Teresa

Blanco, Gabriel Esteban

Etcheverry, Martín Esteban

Gargano, Cecilia Victoria

Procopio, Gastón Emanuel

Quintana, Fabio Hernán

Fecha de inicio: 01/01/2013

Fecha de finalización: 31/12/2014

RESUMEN

A partir de la década de los años 90 los avances tecnológicos de la informática permitieron un incremento exponencial en la generación y almacenamiento de la información y en la actualidad no se observa una desaceleración de esta tendencia. Esta gran cantidad de información almacenada hace que su búsqueda y recuperación sea cada vez más dificultosa y que debamos dedicarles gran cantidad de tiempo y esfuerzo. Esta situación motivó que se acelerara la evolución de la disciplina Recuperación de la Información (RI).

En la actualidad existen dos tendencias fundamentales en el desarrollo

de sistemas RI según el contexto y el ámbito de la fuente documental [1]:

La RI vertical que se enfoca en la indexación de fuentes documentales específicas (por ejemplo una biblioteca de Ciencias Jurídicas).

La RI horizontal que se enfoca en fuentes documentales generales (por ejemplo la Web).

El presente trabajo tiene por objetivo la implementación de un Sistema de Recuperación de la Información (SRI) de fuentes documentales específicas y la evaluación de su rendimiento.

Primero se estudiarán las técnicas de RI existentes para luego volcar los conocimientos adquiridos en el desarrollo de un sistema que utilice el modelo Indexación Semántica Latente (LSI por sus siglas en inglés), este modelo es una derivación del modelo Espacio Vectorial.

Se pretende diseñar un SRI que aplique el modelo LSI y que sea lo suficientemente abierto y flexible para ser utilizado en docencia e investigación.

La arquitectura del sistema deberá permitir que la visualización de los resultados y de las estructuras intermedias sea sencilla, como así también la modificación de la funcionalidad existente o el agregado de nueva funcionalidad, facilitando por consiguiente la experimentación.

Se espera que este proyecto contribuya a la formación de recursos humanos en RI y que el sistema desarrollado pueda servir de base para una transferencia de tecnología a las PYMEs de la región.

Palabras claves: Recuperación de la Información, Lematización Indexación Semántica Latente (LSI), Código abieto

Área de conocimiento: Ing. Comunicaciones y electrónica
Código de Área de conocimiento: 1800
Disciplina de conocimiento: Computación
Código Disciplina de conocimiento: 1802
Campo de Aplicación: Computación

Código Campo de Aplicación: 1802

Otras dependencias de la UNLaM que intervinieron en el Proyecto:

Otras instituciones intervinientes en el Proyecto: No Aplica

Otros proyectos con los que se relaciona:

Con estos desarrollos también se ha beneficiado un doctorando en sistemas de recuperación de información que necesita contar con una herramienta flexible para ensayar sus propias elaboraciones.

Recuperar documentos es un tema que se relaciona con la minería de datos, por lo tanto se piensa que se podrá corresponder con el proyecto de investigación pensado para el año 2015 de minería de datos

Implementación de un Sistema de Recuperación de la Información

Resumen

Durante los dos años del proyecto el grupo pudo construir y dejar operativo dos versiones de recuperador de información, mejor denominados clasificadores de documentos.

Uno de ellos, desarrollado en el primer año del proyecto y oportunamente informado, responde a las técnicas vectoriales con las debidas mejoras en los coeficientes. El segundo implanta la concepción conocida como indexación semántica latente, que resuelve satisfactoriamente dos aspectos del lenguaje: sinonimia y polisemia, que afecta a la calidad del recuperador u ordenador de documentos.

Se lo ha ofrecido a través de un servidor a toda la comunidad informática, en forma de código abierto. En el anexo I se describe como acceder a su uso.

Ambos buscadores están orientados a documentos en lengua española. El desarrollo también presupone que el corpus es estático o al menos de baja movilidad. Se puede identificar claras situaciones donde esto es aplicable, es más el proyecto nació como una extensión a la comunidad que tenía un corpus estático.

Con estos desarrollos también se ha beneficiado un doctorando en sistemas de recuperación de información que necesita contar con una herramienta flexible para ensayar sus propias elaboraciones.

Palabras claves: Recuperación de la Información, Lematización Indexación Semántica Latente (LSI), Código abierto.

Introducción:

A partir de la década de los años 90 los avances tecnológicos de la informática permitieron un incremento exponencial en la generación y almacenamiento de la información y en la actualidad no se observa una desaceleración de esta tendencia. Esta gran cantidad de información almacenada hace que su búsqueda y recuperación sea cada vez más dificultosa y que a ello se le deba dedicar gran cantidad de tiempo y esfuerzo. Esta situación motivó que se acelerara la evolución de la disciplina Recuperación de la Información (RI).

- **Definición del Problema**

En la actualidad existen dos tendencias fundamentales en el desarrollo de sistemas RI según el contexto y el ámbito de la fuente documental

- La RI 1 vertical que se enfoca en la indexación de fuentes documentales específicas (por ejemplo una biblioteca de Ciencias Jurídicas).
- La RI horizontal que se enfoca en fuentes documentales generales (por ejemplo la Web).

El presente trabajo tiene por objetivo la implementación de un SRI2 de fuentes documentales específicas y la evaluación de su rendimiento.

Durante el primer año se estudiaron las técnicas de RI básicas que utilizan la idea de búsqueda en un espacio vectorial

En este segundo año se volcaron los conocimientos adquiridos en el desarrollo de un sistema que utilice el modelo Indexación Semántica Latente (LSI por sus siglas en inglés), este modelo es una derivación del modelo Espacio Vectorial.

Se pretendió diseñar un SRI lo suficientemente abierto y flexible para ser

[1] RI vale por Recuperación de información.
[2] SRI vale por Sistema de Recuperación de la Información

utilizado en docencia e investigación. Ésta se espera realizar en un segundo proyecto que empalme con éste, utilizando el mismo grupo humano.

- **Justificación del Estudio**

Una institución de enseñanza de las disciplinas informáticas debe contar con un grupo humano que pueda transmitir las tecnologías involucradas con el aporte adicional de una experiencia personal en las mismas.

Esta necesidad de formar recursos humanos es una de las varias justificaciones de este proyecto

La tecnología de los recuperadores utiliza intensivamente estructuras de almacenamiento de datos, nociones estadísticas y teoría matricial que integran al menos sus formas elementales al plan de estudios de la carrera, con lo cual los docentes intervinientes en el proyecto quedaran particularmente capacitados en estas temáticas

La RI pasó a ser una herramienta tan fundamental en la informática como puede serlo una base de datos, un compilador o las herramientas de escritorio.

- **Limitaciones**

No encontramos ninguna limitación relacionadas con el cumplimiento de los objetivos.

Tal como se indicó en el protocolo los productos construidos están dirigidos exclusivamente para lengua Española y está establecido para corpus (biblioteca) de tipo cerrado y estático.

- **Alcances del Trabajo**

Los productos confeccionado y puesto a disposición de futuros usuarios cubren todas las etapas propias de un indexador y recuperación de la

información, como se describe a continuación en forma generalizada:

La entrada (input) consiste en los documentos adquiridos por el centro de documentación que maneja el sistema. Esto implica la existencia de criterios y políticas de selección, que a su vez implican un conocimiento detallado y exacto de las necesidades de información de la comunidad a la que se dirige el sistema. Una vez adquiridos los documentos, estos han de ser organizados y controlados de modo que puedan ser identificados y localizados en respuesta a los diferentes tipos de demandas de los usuarios. Las actividades de organización y control incluyen la clasificación, la categorización, la indización y el resumen. Dos elementos importantes son la descripción física del documento y la elección de los puntos de acceso para su inclusión en catálogos y bibliografías.

El proceso de indización implica dos fases intelectuales bastantes diferentes: el análisis conceptual de un documento y la traducción de aquel a un vocabulario determinado. Para efectuar un análisis conceptual adecuado, el indizador necesita no solo la comprensión de la materia del documento, sino también un buen conocimiento de las necesidades de los usuarios del sistema.

La segunda fase del proceso de indización es la traducción del análisis conceptual a un vocabulario determinado. Tras la indización, los documentos son almacenados de algún modo y los registros de indización se organizan en una segunda base de datos de forma que puedan ser buscados fácilmente en respuesta a distintos tipos de peticiones

Tras la indización, los registros se organizan en una base de datos de forma que puedan ser buscados fácilmente en respuesta a distintas peticiones.

Las fases de la salida del sistema (output) los documentalistas preparan estrategias de búsquedas para las peticiones pedidas. El análisis conceptual de la petición, traducido al lenguaje del sistema, es la estrategia de búsqueda.

Una vez terminada la estrategia de búsqueda se compara con las representaciones de los documentos de la base de datos. Las representaciones de los documentos que se ajustan a las estrategias de

búsquedas son recuperadas de las bases de datos y ofrecidas al usuario

Bajo el título de resultados se describe la ubicación del producto y su documentación

- **Objetivos**

El presente proyecto tiene como objetivo general el desarrollo e implementación de un SRI y los siguientes objetivos específicos:

- Aplicar el modelo LSI en el SRI.
- Disponer de una implementación que sirva de referencia para el estudio de RI.
- Formar recursos humanos en RI.

La arquitectura del sistema deberá permitir la visualización de los resultados y de las estructuras intermedias. También deberá ser sencilla la modificación de la funcionalidad existente o el agregado de nueva funcionalidad, facilitando por consiguiente la experimentación.

Se espera que este proyecto contribuya a la formación de recursos humanos en RI y que el sistema desarrollado pueda servir de base para una transferencia de tecnología a las Pymes de la región.

Desarrollo:

- **Material y Métodos**

Tratándose de construir un prototipo computacional, el método que explicamos a continuación consiste en analizar los algoritmos involucrados y las simplificaciones que pueden hacerse sobre los mismos con vistas a una menor complejidad computacional.

Componentes de un Sistema de Selección Ordenada de Documentos

Para poder devolver algo de una memoria es necesario haberlo

guardado antes. De esta verdad de Perogrullo surge que en estos sistemas habrá componentes destinados a la recepción de documentos y otros destinados a la entrega selectiva de algunos de los mismos.

En un sentido estricto no es necesario almacenar el documento ni entregarlo. Es satisfactorio un sistema que, tras procesar un documento en la etapa de ingreso, se limite a conservar solamente el lugar donde se encuentra almacenado y en las consultas entregue una lista ordenada con las ubicaciones de los documentos pertinentes.

Antes de diseñar uno u otro subsistema hay que precisar que se aporta a cada uno y como se relacionan para alcanzar su objetivo.

La etapa de *Memorización, alta* o *ingreso* recibe un conjunto de documentos. Salvo aplicaciones particulares, por ejemplo obras completas de Hegel o de Santo Tomás que son corpus [3] voluminosos y cerrados por definición, la mayoría crece incrementalmente. No queda claro que pudieran tener bajas, posibilidad que se desechó en este proyecto.

Las incorporaciones incrementales, en bien de la eficiencia, no son perfectas y es aconsejable reprocesar el corpus cuando éste ha sufrido un número importante de adiciones. Nuestro trabajo se ha centrado en el proceso de Corpus completos.

La *Etapa de Recuperación* recibe, en la mayoría de los sistemas, palabras que el Interrogante considera que deberían ser palabras claves de los documentos que él consideraría idóneos. Se podría decir que emite un documento minúsculo representativo de lo que espera obtener. Esta consulta podría constar de una o más oraciones. Usuarios expertos, teniendo alguna idea de lo que hace el sistema, omiten en esta escritura, palabras que saben que no influirán en la consulta limitándose a escribir una sucesión de reducidos sintagmas nominales o verbales.

Sistemas exitosos han enriquecido esto, con un metalenguaje que

[3] **Corpus**: Conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación (Real Academia Española)

aportan operadores precisos que amplían la precisión de la consulta. Algunos de esos operadores no son aplicables al desarrollo que se encara en esta investigación, lo que se destacará en los puntos donde tendría incidencia.

Dada las varias formas posibles de referirse a un mismo concepto a causa de las inflexiones morfológicas que puede sufrir la palabra para indicar género, número, persona, así como el agregado de sufijo para modificar su función gramatical, hay consenso unánime de realizar una operación inversa y reducir la palabra a su raíz de modo tal que consultas formuladas con una palabra derivada de la raíz encuentren documentos que utilizaron otra distinta. El ejemplo más sencillo es unir una palabra en plural con la misma palabra en singular.

Limitación de lengua

En primer lugar hay que notar que la etapa morfológica a la cual se refieren los párrafos precedentes, está orientada en su implantación a una lengua, y de querer combinar documentos de distinta lengua es necesario repetir esta tarea para cada lengua y finalmente unificar las raíces provenientes de distintas lenguas que representan conceptos idénticos. Dado el alto contenido lingüístico de esta tarea y el deseo de tener un sistema básico funcionando, a los fines del proyecto se decidió limitar esta tarea a la lengua castellana.

Los buscadores multilingües introducen varios problemas adicionales. En primer lugar, reconocer el idioma de un documento y el idioma de la consulta. En segundo lugar consultas formuladas en una lengua debe incluir también en la respuesta documentos escritos en otras lenguas: De lo cual se concluye que el sistema debe contener al menos una capacidad elemental de traducción.

Todas estas complicaciones que introduce el multilingüismo son ajenas al núcleo de la problemática de los buscadores, que es aportar una propuesta de documentos pertinentes en respuesta a una consulta. El grupo decidió, por lo tanto, no incluir multilingüismo por ahora.

La “E*tapa de Reducción”* de palabras a sus raíces, se conoce como *lexematización* y el trozo de código que lo realiza un *lexematizador*. Siendo este componente compartido con otras aplicaciones que procesan la lengua, se lo consigue como módulo separado para varias lenguas, incluido el español. Para Español hay solo uno denominado el *Porter Español*, por ser una adaptación de un lexematizador para la lengua española muy famoso denominado “Porter”

Fue obtenido de Internet, de la siguiente dirección electrónica:

http://sourceforge.net/projects/stemmer-es/files/ . Estando escrito en php se lo re-escribió en C# para unificar el código en un único dialecto. La lengua de codificación no tiene mucha importancia pero varios integrantes tienen más familiaridad con el C#· y la Institución lo privilegia. De allí su elección.

Algunas palabras no tienen una semántica propia sino que indican el modo con el cual la semántica de unas palabras modifica la de otras. Conjunciones y preposiciones no llevan asociada un concepto, pero son fundamentales para ligar a otras palabras en un concepto o aserción. Los técnicos los denominan “*stop-words*”. La lógica filosófica en su estudio del lenguaje les da un nombre más conceptual “*palabras sincategoremáticas*”. Estos se aportan al sistema explícitamente.

Es de notar que para realizar experimentos es necesario disponer de un conjunto de documentos, denominada el “corpus”. En este proyecto recurrimos a una base de documentos jurídicos, la comunidad que los ha reunido para sus propios intereses será un beneficiario directo de este proyecto.

Se impone a los usuarios del sistema agrupar todos los documentos del Corpus en un único directorio. Un sistema amplio debe admitir multiplicidad de formas de codificación. El grupo resolvió el problema de convertir archivos de extensiones .doc, .docx, .pdf, .htm y .html a *texto plano*. Una instrucción de bifurcación múltiple tipo case o switch, realiza esta distribución permitiendo de un modo sencillo agregar otras extensiones y sus correspondientes rutinas de conversión. Dadas las características del corpus jurídico del cual se dispuso, no

fue necesario pensar en más extensiones.

Hay un conjunto adicional de tareas que preceden al uso de un lexematizador y que tuvieron que ser programadas, algunas de ellas a la medida del proyecto. Las posibilidades al respecto quedan tan abiertas como sea la imaginación de los desarrolladores y campo de aplicación. Estas se agrupan en una rutina de prelematización.

Las tres primeras son de uso universal.

a) Supresión de signos de puntuación.
b) Conversión a minúsculas,
c) Eliminación de las palabras sincategoremáticas. Esta última tarea se realiza recurriendo a una lista confeccionada a mano y por conocimiento de la lengua. Si así no fuera se puede tener una buena aproximación mirando cuales son las palabras de uso más frecuentes en la lengua.
d) Al tratarse de un corpus jurídico aparecen dos elementos que requieren un tratamiento a medida: Las fechas y las citas de documentos jurídicos.

Reconocimiento de fechas y citas de legislación

Existen varias formas de escribir una fecha, las dos más frecuentes son:

a) El día en números seguido de “de” luego el mes como texto, otro “de” y el año escrito con 4 dígitos.

b) Día, mes y año escritos en forma numérica, separados por barras “/”. La más rígida de estas requiere dedicarle siempre dos dígitos al día y mes y cuatro al año.

La experiencia del grupo indica que al reconocerlo y unificar la imagen interna, conviene elegir la forma numérica dura y convertir las demás a ella. Así se hizo convirtiendo las textuales a esta última.

El código más elegante se obtuvo recurriendo a expresiones regulares traductores.

Nuevamente, una vez dominada esta técnica se deja para mejoras futuras encarar el reconocimiento de otras formas de escritura: dígitos cero faltantes, otros separadores, meses abreviados, etc.

Se tomó el debido cuidado para que la palabra ficticia que representa una fecha o cita jurídica quede codificada como un objeto inerte para el Lematizador.

Etapas del Proceso de Recuperación de la Información

La Disciplina RI busca proveer a los Usuarios, documentos en los cuales, posiblemente, encuentre una respuesta a la inquietud intelectual que motivó la consulta. Por lo tanto, el problema con el cual se encuentra la RI se puede definir como: “Dada una necesidad de conocimiento (consulta) y un conjunto de documentos, hay que ordenarlos de mayor a menor en su relevancia, facilitando el acceso a un subconjunto de los más relevantes”.

La solución de esta problemática se puede dividir en dos subproblemas:

Elegir un modelo para calcular la relevancia de un documento frente a una consulta.

Diseñar algoritmos y estructuras de datos que lo implanten eficientemente.

El proceso de RI se puede dividir en dos etapas: la **Indexación** y la **Búsqueda**.

Indexación de la Información:

La Indexación se puede entender como la identificación y asignación de descriptores4 representativos de los temas y que responden al modo intuitivo con el cual los usuarios del Sistema tratarán de describir la información que

[4] Término o símbolo válido y formalizado que se emplea para representar inequívocamente los conceptos de un documento o de una búsqueda. (www.rae.es)

buscan.

La etapa de la indexación se divide en las siguientes subetapas:

Parsing de documentos: modelización de los documentos a ser indexados.

Análisis: la forma más común de representar un documento de texto es por un sistema de términos indexados o palabras claves. Estos términos son extraídos con el siguiente proceso, donde se normalizan los términos del documento a ser indexado, por reducción a su raíz gramatical (Stemming) y se eliminan/descartan:

- ✓ Los signos de puntuación
- ✓ Las palabras denominadas en inglés “stopwords” (ej.: la, las y, los, en, etc.)
- ✓ Las palabras que se “escapan” de los umbrales superior e inferior.
- ✓ Eventualmente los acentos

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

Reducción de términos: Metodología para lograr una búsqueda más rápida y eficiente. En el caso de las Palabras frecuentes, presentes en la mayoría de documentos no aportan una ayuda a las búsquedas pero aumentan el volumen de las tablas y por consiguiente el tiempo de proceso. Es conveniente suprimirlas. Algo similar ocurre con palabras con una única aparición o poco más. El sistema confeccionado recurre a un par de umbrales para eliminar unos y otros.

Búsqueda de Información

La Búsqueda de Información es una secuencia ordenada de pasos, ejecutados con la finalidad de analizar los documentos que contienen cierta información o de entregar datos/información concreta que responde a determinada pregunta5. La búsqueda bibliográfica se acota a la identificación, selección, ordenamiento y entrega de las referencias de los documentos y sus enlaces, si están disponibles, al texto completo de ellos.

Se considera a la Búsqueda de Información, como una las principales tareas a la hora de la investigación científica. Las principales etapas a la hora de la búsqueda de Información son: Planificación, ejecución y evaluación.

Planificación: Etapa donde se identifican conceptos y límites que comprenden la necesidad de Información requerida por el Usuario.

Ejecución: Etapa en donde el Usuario introduce su estrategia de búsqueda, observa los resultados y realiza los ajustes, de ser necesario.

Evaluación: Etapa en donde se evalúa la pertinencia o no de la información encontrada.

Cálculo de los pesos

Confección de los pesos de los lexemas.

Finalmente la palabra real o ficticia es enviada al lematizador que devuelve un código de lexema. Buscadores avanzados buscan palabras en cercanía. No siendo este nuestro objetivo, cada documento se puede simplificar como un conjunto de pares <lexema; frecuencia>. Estos pares deben ser transformados en coeficientes para una matriz, que inevitablemente será rala.

Una vez más se enfrentó el proyecto con multiplicidad de formas de

[5] De http://acimed.sld.cu/index.php/acimed/article/view/142/113

transformar frecuencias en pesos de los lexemas. Algo en lo cual coinciden todas las técnicas es dar peso nulo, si y sólo si, el lexema no figura en el documento.

La técnica más elemental asigna peso uno y cero. La mayoría introduce escalas logarítmicas. Esto significa que a doble frecuencia el peso aumenta en una constante. Fuera del argumento fáctico de que las cosas andan mejor así, el grupo encuentra una similitud con varios órganos sensoriales que responden en forma logarítmica al estímulo. Leer un documento y percibir una frecuencia de repetición es también una sensación.

Una de las fórmulas usadas da como Peso, el logaritmo de la frecuencia previamente incrementada en 1, lo que asegura el coeficiente 0 ante la ausencia. Otros aplican sólo a frecuencias no nulas un logaritmo e incrementan el resultado en 1, evitando ceros falsos.

La sencillez de tales fórmulas hizo inmediata la codificación de las mismas.

Finalmente es práctica corriente corregir estos pesos sobre la base de la presencia del lexema en todo el documento. La idea es que lexemas presentes en muchos documentos discriminan poco, llegado el momento de la selección. Estas correcciones globales explican porque el trabajo incremental de incorporación de documentos no es tan preciso como el global. Un documento nuevo aporta a las frecuencias globales para el futuro y se beneficia con las frecuencias globales de los ya incorporados, pero estos no conocieron los aportes del documento nuevo.

Este proyecto desarrolló un SRI utilizando el modelo LSI, el método arranca con la construcción de una matriz “A”, donde las filas representan los términos o palabras claves y las columnas los documentos que forman parte del Corpus. Cada elemento a_{ij} representa el peso del término i en el documento j. El peso de cada elemento se calcula como el producto del Peso Local del término en un documento l_{ij}, el peso global del término en la colección de documentos g_i y el factor de normalización del documento d_j: $a_{ij} = l_{ij}\, g_i\, d_j$

Tablas para el Cálculo de los Distintos Pesos.

En las siguientes tablas se pueden observar las fórmulas de los pesos locales y globales más utilizadas como así también las fórmulas de los factores de normalización más utilizadas.

Fórmulas de pesos locales

Nombre	Fórmula
Binaria	l_{ij} = 1 si el término existe en el documento, 0 si no existe
Log	$l_{ij} = \log (1 + tf_{ij})$ tf_{ij} = número de ocurrencias del término i en el documento j
Augnom	$l_{ij} = (x(tf_{ij}) + (tf_{ij} / \max_i (tf_{ij})) / 2$ $x(tf_{ij}) = 1$ si $tf_{ij} > 0$, 0 si $tf_{ij} = 0$
Frecuencia de Término	$l_{ij} = tf_{ij}$

Fórmulas de pesos globales

Nombre	Fórmula
Ninguna	$g_i = 1$
Entropy	$g_i = 1 + (\sum_j (p_{ij} \log(p_{ij})) / \log n)$ $p_{ij} = tf_{ij} / \sum_j tf_{ij}$ n = cantidad de documentos del corpus

IDF	$g_i = \log (n / df_i)$ df_i = cantidad de documentos en los cuales ocurre el término i
GFIDF	$g_i = \sum_j tf_{ij} / df_i$
Normal	$g_i = 1 / (\sum_j tf_{ij}^2)^{1/2}$
Probabilística Inversa	$g_i = \log ((n - df_i) / df_i)$

Fórmulas de factores de normalización

Nombre	**Fórmula**
Ninguna	$d_j = 1$
Coseno	$d_j = (\sum_i (g_i l_{ij})^2)^{1/2}$

- **Lugar y Tiempo de la Investigación**

El presente proyecto, por no tener tareas de campo, se desarrolló íntegramente en las instalaciones del DIIT de la UNLaM durante los años 2013 y 2014.

A través de distintos encuentros se fueron evaluando las pruebas y resultados. El equipo de trabajo realizaba las lecturas y resumen del material bibliográfico. El desarrollo del sistema se realizó por los programadores en el laboratorio utilizando PC, se adoptó por política institucional, el lenguaje C# para codificarlo.

Para la construcción el código fuente se utilizó el siguiente equipamiento:

Procesador de 1.6 GHz, 1 GB de RAM, 300 GB de espacio en disco duro y tarjeta de video compatible con DirectX 9 y una resolución de pantalla de 1024 x 768.

El software que se utilizó comprendía:

Microsoft Windows 7 SP1 (x86 o x64).

Microsoft Windows Server 2008 R2 SP1 (x64).

Microsoft .NET Framework 4.5.

Microsoft Visual Studio 2013.

Enlaces de descarga

Microsoft .NET Framework 4.5:

https://www.microsoft.com/es-ar/download/details.aspx?id=42642

Microsoft Visual Studio 2015 (Versiones gratuitas):

https://www.visualstudio.com/es-es/downloads/download-visual-studio-vs.aspx

- **Descripción del Objeto de Estudio**

Vastas colecciones de documentos son de escasa utilidad si no se dispone de un medio para extraer del conjunto aquellos que son afines a la inquietud del momento.

La catalogación multifacética de los documentos y la interfaz para especificar una necesidad documental, junto con la algoritmia asociada, constituyen las partes esenciales de los llamados recuperadores de información.

Lograr un sistema completo es el punto de partida para poder experimentar mejoras locales.

Archivos del corpus

El corpus (conjunto de textos almacenados de forma digital) a indexar puede estar conformado por archivos con las siguientes extensiones: ".txt", ".pdf", ".docx" y ".html". En los últimos tres casos el sistema realiza una conversión a formato ".txt" de manera de obtener una entrada normalizada. Para la conversión de archivos PDF se utiliza la biblioteca iTextSharp, la cual es de código abierto y permite la manipulación de este tipo de ficheros. Para la conversión de archivos DOCX se emplea la biblioteca de Microsoft "Microsoft.Office.Interop.Word". Por último, para la conversión de archivos HTML se usa un método propio que extrae el contenido relevante en forma de texto plano y elimina los metadatos contenidos.

El proceso de normalización del corpus genera como resultado un directorio donde se encuentran todos los ficheros en formato TXT (los que originalmente eran de este formato y los convertidos) y un subdirectorio donde se encuentran los archivos de otros formatos. La ubicación de este directorio puede configurarse a conveniencia del usuario.

Archivos de entrada

Tanto el sistema indexador como el de consulta necesitan además dos archivos. El archivo "StopWordsFinal.txt" contiene las palabras (677) sin peso semántico o sincategoremáticas. Este se utiliza para eliminar dichas palabras de la colección de términos que se obtiene durante la lematización del corpus a indexar. En el programa de consulta se usa para lo mismo, solo que aplicado al texto de búsqueda. El archivo "espa~nol.words" es un diccionario en español que contiene los lexemas (53.065) aceptados como válidos. Los términos que no se encuentran en este diccionario no se tienen en cuenta en el proceso de lematización, con esto se evita sobrecargar el sistema con palabras con errores de ortografía o tipeo.

Archivos de salida

Como resultado se obtiene seis archivos de texto:

Matrizskinv.txt	contienen las matrices que relacionan los documentos con los términos y sus pesos calculados dentro del corpus
MatrizTermDoc.txt	
Matrizuk.txt	
TablaDocumentos.txt	almacenan las claves, rutas de acceso y cantidad de documentos, además las claves, descripción y ocurrencia de los términos en cada fichero y en su totalidad
TablaTerminoDocumento.txt	
TablaTerminos.txt	

Estos se utilizan como entrada del sistema buscador, además del diccionario en español y el listado de palabras sincategoremáticas.

En el anexo se observa una explicación detallada del funcionamiento del sistema.

- **Descripción de Población y Muestra**

En el Anexo II hay una descripción del corpus particular utilizado en la experimentación de este trabajo.

- **Resultados**

Resultados en cuanto a servicios a la comunidad, tanto para informática como para legos en el tema:

Se logra el desarrollo de un motor de búsqueda de documentos suficientemente abierto y flexible.

Para los informáticos se les brinda un sistema abierto, con su respectiva

documentación.

Para los legos se les ofrece un sistema compilado listo para usar

Es inevitable que en la construcción de una herramienta soporte para futuras investigaciones los aportes originales genuinos queden distribuidos y no tengan suficiente entidad para justificar una publicación.

Resultados en cuanto a la formación de recursos humanos:

Los integrantes del equipo han adquirido una comprensión global y detallada de los componentes de un sistema de información y de las alternativas teóricas y de implementación que existen.

Por otro lado en esta investigación hay una transferencia concreta y es de útil importancia dado que uno de los integrantes del proyecto ha iniciado un doctorado en IR, para lo cual necesita disponer de un sistema funcionando alrededor del cual pueda ensayar sus propias contribuciones.

Resultados en cuanto a la difusión de resultados:

Se considera que los logros obtenidos serán de interés para la comunidad universitaria por la aplicación práctica de conocimientos teóricos y a la comunidad en general dado su condición de open sources.

Se tiene la intención de continuar trabajando en un nuevo proyecto, el cual será presentado a la comunidad de investigadores como medio de difundir los logros alcanzados.

Resultados en cuanto a transferencia hacia las actividades de docencia y extensión:

Se espera que el sistema de recuperación documental sea utilizado en labores docentes, de investigación.

- **Cronograma**

Actividades / Responsables Año 2014	**Mes 1**	***Mes 2***	***Mes 3***	**Mes 4**	**Mes 5**	**Mes 6**	**Mes 7**	**Mes 8**	**Mes 9**	**Mes 10**	**Mes 11**	**Mes 12**
Desarrollo y Documentación												
Testing												
Prueba final -												
Implementación												

- **Discusión**

Después de haber logrado en el primer año de trabajo del grupo, un recuperador convencional de información, que nosotros creemos que debiera denominarse un “clasificador de afinidad de documentos” el grupo se dedicó a estudiar e implantar el método conocido como Latent Semantic Indexing, LSI (Indexación Semántica Latente, en español), viendo el peso que está tomando, sobre todo por sus propiedades sobresalientes en resolver dos dificultades que afectan a los recuperadores de documentos: la Sinonimia6 y la Polisemia7 (o equivocidad o múltiple acepción).

Conceptos que son explicados a continuación junto con su incidencia en el funcionamiento de los buscadores.

Sinonimia

[6] **sinonimia.** F. Circunstancia de ser sinónimos dos o más vocablos. - http://www.rae.es
[7] **Polisemia**. Pluralidad de significados de una palabra o expresión - http://www.rae.es/

Dos palabras son sinónimas[8] si representan un mismo concepto o realidad. Esta relación establece una equivalencia y por lo tanto una partición de las palabras:

Ejemplos:

Terminar – Finalizar

Barrilete – Cometa

Los lenguajes técnicos evitan introducir sinónimos en las palabras que son propias de su disciplina, pero no pueden evitar que en el uso general de la lengua exista sinonimia.

Esta se ve agravada por la inevitable regionalización en comunidades muy grandes de hablantes. El español de méxico y de argentina difieren en el nombre que le dan en muchos casos a una misma creación técnica, p. ej.: ascensor – elevador.

Polisemia

Una palabra es equívoca si designa más de un concepto. También se habla de multiplicidad de las acepciones o de polisemia.

Ejemplos:

gato designa tanto un animal como una herramienta.

llama designa tanto un animal como una manifestación del fuego.

vela designa tanto un artefacto primitivo de iluminación como una parte de ciertas embarcaciones.

Un buen lector no se confunde con las equivocidades ya que el contexto le permite desambiguar el significado, idea que se verá subyacente en el LSI.

Analogía

La equivocidad no debe confundirse con los usos analógicos de un

[8] **Sinónimo**: Se dice de una palabra o expresión que, respecto de otra, tiene el mismo significado o muy parecido, como *empezar* y *comenzar*

concepto y su correspondiente palabra. En ello se perciben realidades en parte común, en parte diversa.

Ejemplos:

- *Los **brazos** de un río - Los **brazos** de una persona - los **brazos** de una lámpara.*
- Las **patas** de la silla - Las **patas** de un animal.
- *La **sanidad** de un animal - la **sanidad** de un clima - la **sanidad** de un alimento* (ARISTÓTELES).

Brazo utilizado en el primer ejemplo, abarca realidades distintas, ubicadas en un medio, ya sea acuoso, viviente o inerte, pero al mismo tiempo, encierra un concepto compartido: Múltiples elementos que concurren o se desprenden de un elemento común. Más evidente en el uso de "pata".

En estos ejemplos la palabra queda precisada por un complemento de especificación. Pero podría no estar tan cerca:

La lámpara del escritorio tiene roto un **brazo**.

En estos usos analógicos, analogías de atribución para mayor precisión, se puede distinguir un analogado principal al cual hacen referencia los demás.

En el caso de brazo, debe serlo el brazo de una persona, anterior a los demás usos tanto en el conocimiento humano como en sus necesidades de comunicación.

Como reconocer estas situaciones

Ayuda a distinguir estas 3 situaciones: sinonimia, equivocidad y analogía, un intento de traducción a otra lengua.

Un grupo de palabras sinónimas puede tener en otra lengua una única traducción u otro grupo sinonímico, no necesariamente de la misma cardinalidad.

Los equívocos dejan de serlo, p.ej. *vela* se traduce al inglés por *candle* o *sail*, según el concepto que se quiera referenciar. El traductor lo descubre por

el contexto. Si la equivocidad se mantiene, es de sospechar que el uso fue analógico.

La mayoría de los usos analógicos se traducen sin mayor inconveniente. Puede suceder, sin embargo, que en la otra lengua se hubiera creado un vocablo específico para uno de esos usos o que el significado común sea extraído de otro analogado principal. Sin ahondar si ello existe o no, se entendería si alguien hablara de las *piernas de un río*.

Incidencia en los buscadores

Debido a la *sinonimia*, documentos que hablan de un mismo tema, no aparecen como respuesta, cuando la evocación aporta uno solo de los sinónimos.

Una acción se llama *estacionar* o *aparcar*. Se puede utilizar indistintamente una u otra en distintos documentos. A la hora de recuperar información de los documentos, puede suceder que se use uno sólo de los términos. Si es así, con un sistema “clásico” de recuperación el que evoca sólo recuperará una parte de los documentos que le hubieran servido.

Esto no depende del método de dar peso a los coeficientes de la matriz lexema-documento. Si la consulta tiene una palabra el vector consulta tiene un único coeficiente no nulo y la respuesta se limita a los documentos que la contienen ordenados según el peso que haya tenido por frecuencia y otros considerandos.

Si la consulta tuviera dos palabras, una sola con sinonimia, los documentos que usan una alternativa saldrán muy desfavorecidos en el ordenamiento del resultado de la búsqueda.

La técnica LSI, a la que se dedica este año de trabajo, trata de registrar *una temática más que una palabra*, de modo tal que detecta que estos sinónimos son utilizables en contextos notablemente similares. Hablará de vehículos, de espacio, de orientación, de freno de mano y otros conceptos que no tienen mayormente sinónimos y que quedan involucrados en la acción descripta.

Documentos con tantas palabras comunes comparten una orientación en un espacio multidimensional.

Con la polisemia el fenómeno es el inverso de este. La palabra es única, pero las palabras que la acompañan son distintas en cada uno de los usos. Lo que se puede ilustrar con un par de ejemplos:

- Una *vela* (instrumento que alumbra) se enciende y quema, cosa que ningún navegante cuerdo, haría con el velamen de su nave.
- Del mismo modo, arriar y desplegar no se entiende bien como se ejecutaría con la vela de cera.

A diferencia de lo que ocurría con la sinonimia ahora los documentos se despliegan en dos direcciones distintas recurriendo a lexemas propios de una u otra temática y el buscador en cuanto la pregunta aporte un lexema adicional que la oriente tomará documentos de una u otra dirección.

Un nuevo ejemplo más complejo se tiene con la palabra *mango*:

- *Ayer comí una tarta de mango que estaba muy rica*
- *Cuando fui a buscar el martillo, descubrí que el mango estaba roto*.
- *No tengo un mango* (frase que se utiliza en *Argentina* para hacer referencia a no tener dinero).

Palabras como *rica, martillo,* son apropiadas con una de las acepciones y difícilmente forman frases inteligibles con las restantes. La tercera frase necesita un contexto más amplio para su desambiguación.

Como ilustración de la amplitud que puede tomar la polisemia se puede señalar que la real academia señala 12 acepciones para la palabra cabo.

Incidencia en el Usuario

Debido a la *sinonimia*, documentos que hablan de un mismo tema, no aparecen como respuesta, cuando la evocación aporta uno solo de los sinónimos.

La sinonimia no crea problemas en un sistema orientado a temáticas. Sí, en los sistemas que ahora se llamarían "*tradicionales*". En ellos, el usuario debe ser consciente de la sinonimia y aportar todos los sinónimos en la

consulta. Y si el lenguaje de consulta tiene operadores lógicos en modo disyuntivo. Consultas redactadas con texto extenso no sufren tanto con la sinonimia, pues las restantes palabras de la consulta pueden servir para incluir los documentos que usan los sinónimos, pero con un inmerecido descenso en el ordenamiento de especificidad.

Cualquier persona que desee recuperar documentos debe ser consciente de la existencia de palabras equívocas en su consulta, y adornar la misma con otras palabras que resuelvan la equivocidad para no recibir sugerencias de documentos en ambos sentidos.

Los sistemas de consulta interactivos ponen rápidamente en evidencia la equivocidad y llevan a que el consultante (ya sea positivamente, agregando temas o negativos, excluyéndolos), perfeccione su búsqueda.

La diferencia de comportamiento entre un sistema "*tradicional*" y uno orientado a temáticas, se nota en su comportamiento frente al desambiguador. Sea A una palabra polisémica y B una palabra agregada con intención de desambiguar. En un sistema tradicional, aparecerán mejor ubicados, documentos que poseen a ambos y luego entremezclados documentos con sólo A o sólo B. Se dice "entremezclados", pues, los que tienen A y B, tienen un propio ordenamiento, dependiendo de las veces que aparecen y el lugar: título, resumen, etc. Usos ocasionales de A y B pueden ceder frente a muchas citas de A o muchas citas de B.

En un sistema orientado a temáticas, la presencia conjunta prima por más tiempo en el ordenamiento.

En cuanto a los usos análogos de sustantivos, el agregado "… de NNN" ayuda a desambiguar el uso analógico. Esto lo debe saber quién consulta a un recuperador de documentos, tarea que rara vez se logra satisfactoriamente en un solo intento. Si aparecen usos analógicos no deseados se lo perfecciona agregando un determinante o con operadores de negación que incluyan a los otros.

Los términos tendrán vectores ricos en coeficientes no nulos ya que aparecerán con diverso grado de pertenencia a más de una dimensión.

Estacionar que apareció en un ejemplo de la dimensión automovilística no es exclusivo de ella ya que aparece también en una dimensión culinaria.

De la exposición siguiente surgirá que las de dimensión tampoco quedarán tan claramente asociados a macro conceptos como *automovilístico*, *culinario*, sino que las matemáticas subyacentes al método harán su propia propuesta de dimensiones fundamentales.

Fundamentos matemáticos del LSI

Dos documentos que comparten una temática y sólo difieren en el uso de una u otra de dos palabras sinónimas tendrán una gran similitud en su descripción vectorial, lo que se traducirá en un ángulo pequeño entre sus vectores y de usar como medida el coseno, un coseno elevado. Medir similitud de documentos no requiere ningún agregado a las técnicas originales.

La dificultad se presenta cuando se efectúan consultas aportando muy pocos lemas y uno o más de ellos tiene sinonimia. El vector de consulta tendrá unos pocos documentos que usan exactamente las mismas palabras, y tendrán una ubicación privilegiada en la respuesta frente a aquellos que usaron uno o más sinónimos. Si todas las palabras de la consulta admiten sinónimos, habrá documentos relevantes que no aparecerán en la respuesta.

De esto último surge la idea de recodificar los documentos, no en función de lemas sino de orientaciones temáticas. Si se hace lo mismo con los lemas, estos quedarán codificados por su grado de pertenencia a las distintas orientaciones temáticas que se haya detectado en el Corpus.

Visto como problema matemático, es un cambio de un sistema de coordenadas por otro.

Rotar documentos ubicados en un espacio de lemas, tiene la misma complejidad computacional que la cantidad de lemas (que pueden ser decenas de miles). Y enfrentar consultas con documentos, tendrá la misma complejidad. Del análisis matricial que sigue, se verá que se pueden reducir las orientaciones temáticas a unos pocos cientos, sin afectar sensiblemente a la

calidad de las recuperaciones, con lo cual se evita aumentar el espacio de almacenamiento del Corpus procesado.

La enseñanza tradicional del análisis espectral en el cálculo matricial, está orientado a las matrices cuadradas. Conceptos como *autovalor*, *autovector fila, columna* son conocidos por profesionales de computación.

Es menos conocido que estos conceptos han sido extendidos a matrices rectangulares ya que la propiedad básica que involucra a *autovalor* y *autovector*: $Ax = \lambda x$, carece de sentido cuando A es una matriz rectangular de $m \times n$, con $m \neq n$, puesto que el miembro izquierdo exige que x sea de dimensión n, generando un vector de dimensión m y el resultado del producto indicado en el miembro derecho tiene la misma dimensión que x, o sea n, lo que hace imposible satisfacer la igualdad indicada.

Pero si se trabaja con dos juegos de vectores $\{ x_i \}$ e $\{ y_j \}$ se puede lograr soluciones al par de ecuaciones: $A\,x = \lambda\,y$ y $A^t\,y = \lambda\,x$.

En la segunda ecuación se usa la matriz transpuesta.

Combinando las dos ecuaciones se pueden obtener una ecuación para x y otra para y: $A^t\,A\,x = \lambda^2\,x$ y $A\,A^t\,y = \lambda^2\,y$

Las matrices $A^t\,A$ y $A\,A^t$ son cuadradas y simétricas, la primera de $n \times n$ y la segunda, una matriz de $m \times m$.

Arbitrariamente, supongamos $m > n$ el rango de A puede ser, a lo más n, y por lo tanto, la matriz $A\,A^t$ tiene rango a lo más n, lo que indica que tendrá $m - n$ autovalores nulos. Sus autovalores λ^2 son no nulos y positivos, coincidentes con los del producto más pequeño como surge del presente análisis.

Las matrices simétricas tienen autovectores fila y columna coincidentes.

Un juego de ellas hará la función de x en el planteo inicial y el otro el papel de y.

Una propiedad igualmente importante de los autovalores es la posibilidad de descomponer una matriz en factores $Y\Lambda X$, donde Λ es una matriz diagonal (bajo algunas condiciones que las matrices simétricas satisfacen) construida con los autovalores, X construido con los autovectores

columna en el mismo orden que el usado en Λ e Y como una matriz construida por yuxtaposición de los autovectores fila.

Esta descomposición admite una similar para las matrices rectangulares. El factor Y medirá m x m, el factor Λ m x n y el factor X n x n, de modo tal que el resultado que debe ser igual a A mida m x n.

Manteniendo sin pérdida de generalidad que m > n se construye Λ a partir de una matriz nula instalando en el cuadrado superior las raíces cuadradas de los autovalores encontrados en A^t A Sin pérdida de generalidad se puede tomar valores positivos.

Y es una matriz rectangular de mxm con los m vectores x obtenidos en A A^t

e X , una matriz cuadrada de nxn con los n vectores y obtenidos en A^t A

Esta escritura matricial se puede reescribir de esta otra forma:

A = $\sum_{i=1}^{n} x_i \lambda_i y_i$ lo cual pone mejor de relieve el papel de los λ_i

La matriz A quedó expresada como suma de n matrices de rango 1 y módulo 1 cada una con un factor de peso λ_i. Si se elimina el aporte de los autovalores más pequeños no se afecta sensiblemente el resultado pero se achica notablemente el volumen de la descripción de los documentos y el trabajo computacional de enfrentar consultas con documentos.

Surge entonces la idea aplicada con éxito en las implantaciones de LSI de buscar la representación tras una cantidad moderada de valores de λ_i

La matriz A, a la que queremos aplicar esta teoría y reescribir de esta manera es la matriz lexema-documento que en ambas dimensiones puede ocupar algunos miles.

Este reformuleo encierra un gran volumen de cálculo ya que se necesita obtener los autovalores de matrices simétricas voluminosas. Esto ha obligado al grupo a incursionar en la algoritmia matricial.

Existen métodos iterativos y métodos llamados exactos. Son exactos (dejando de lado la finitud de los sistemas numéricos computacionales) en cuanto construyen exactamente el polinomio característico. O mejor dicho son

exactos porque sólo recurren a operaciones racionales. Pero, hallar las raíces del polinomio característico si este supera a grado 4, forzosamente recurre a técnicas aproximadas.

- **Conclusiones**

El objetivo del presente proyecto fue construir íntegramente un prototipo de un sistema de organización de documentos para su posterior recuperación mediante un buscador.

Se hicieron dos prototipos, uno basado en lo que ahora se pueden llamar métodos clásicos y otro según el método LSI. En el Anexo II se describe el muestreo.

Ambos prototipos se hicieron en forma modular de modo tal de permitir futuras mejoras locales con su consiguiente experimentación.

Estos se almacenaron con código abierto a fin de facilitar a nuevos grupos interesados en esta tecnología para su iniciación en el tema y la base sobre la cual efectuar experimentos propios.

La dirección es: http://ftpingenieria.unlam.edu.ar/FTPIngenieria/

En el anexo I se describe su instalación y uso.

Nuestro sistema trabaja sobre un corpus cerrado, motivado por una necesidad de extensión a la sociedad, tal como explicamos en nuestra presentación del proyecto.

Se pueden señalar múltiples aplicaciones de similar característica. Autores muy prolíferos como Cervantes, Tomás de Aquino, Hegel han dejado numerosos escritos para la posteridad. Por definición estos son corpus cerrados.

Con solo cargar los escritos de estos autores dispondrían de una herramienta para su propia labor de estudio.

La unidad documental no necesariamente deba ser un libro completo sino, a criterio de quienes manejan los textos, pueden ser capítulos, secciones, etc.

- **Bibliografía:**

Se utiliza la misma bibliografía que fue presentada en el informe de avance

1] Cleverdon, C.W., 'Progress in documentation. Evaluation of information retrieval systems', Journal of Documentation, 26, 55-67, (1970).

[2] Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A. (1990) "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407.

[3] Lancaster, F.W., Information Retrieval Systems: Characteristics, Testing and Evaluation, Wiley, New York (1968).

[4] C. J. van Rijsbergen B.Sc., Ph.D., M.B.C.S. Information Retrieval (1979) 2da. Edición

[5] Salton, G., 'Automatic text analysis', Science, 168, 335-343 (1970).

[6] Sparck Jones, K., Automatic Keyword Classification for Information Retrieval, Butterworths, London (1971).

[7]. Winograd, T., Understanding Natural Language, Edinburgh University Press, Edinburgh (1972).

[8] Minsky, M., Semantic Information Processing, MIT Press, Cambridge, Massachusetts (1968).

[9] Baeza-Yates, R. y Ribeiro-Neto, B. "*Modern Information Retrieval*". ACM Press. Addison Wesley. 1999

[10] Salton, G. y Mc Gill, M.J. "Introduction to Modern Information Retrieval". New York. Mc Graw-Hill Computer Series. 1983.

[11]Tolosa Gabriel H. y Bordignon Fernando R.A. "Introducción a la Recuperación de Información - Conceptos, modelos y algoritmos básicos".

Universidad Nacional de Lujan. Creative Commons Atribución-No Comercial-Compartir Obras Derivadas Igual 2.5 Argentina License. 2007

Berry Michael W. y Browne Murray. Understanding Search Engines Mathematical Modeling and Text Retrieval Second Edition, SIAM. 2005.

Dominich, S. The Modern Algebra of Information Retrieval. Springer Verlag, Berlin Heidelberg, 2008.

Epifanio Tula, Luis Gerónimo y Medeot, Matías Daniel. “Sistema de Recuperación de Información - Motor de Búsqueda: Innuendo”. UTN Regional Córdoba. 2008

Fernández Luna Juan Manuel, Huete Guadix Juan Francisco, Pérez Vázquez Ramiro, Rodríguez Cano Julio César y Torres López Carmen. “Empleo de motores de búsqueda de código abierto para la recuperación de información vertical”. RCCI Vol. 3, No. 1-2 ENERO- JUNIO, 2009 p. 41-47.

Publicaciones

Para el año 2016 se piensa presentar el tema y los desarrollos efectuados en WICC para de esta forma tomar contacto con gente que tenga necesidades similares. Si bien el volumen de trabajo de programación de estos prototipos no encierra un desarrollo original en las ideas, sí en el código, existiendo una cuota de originalidad y avance.

Es de esperar que en un nuevo proyecto que empalme esto se puedan mostrar avances genuinos.

Actividades tecnológicas

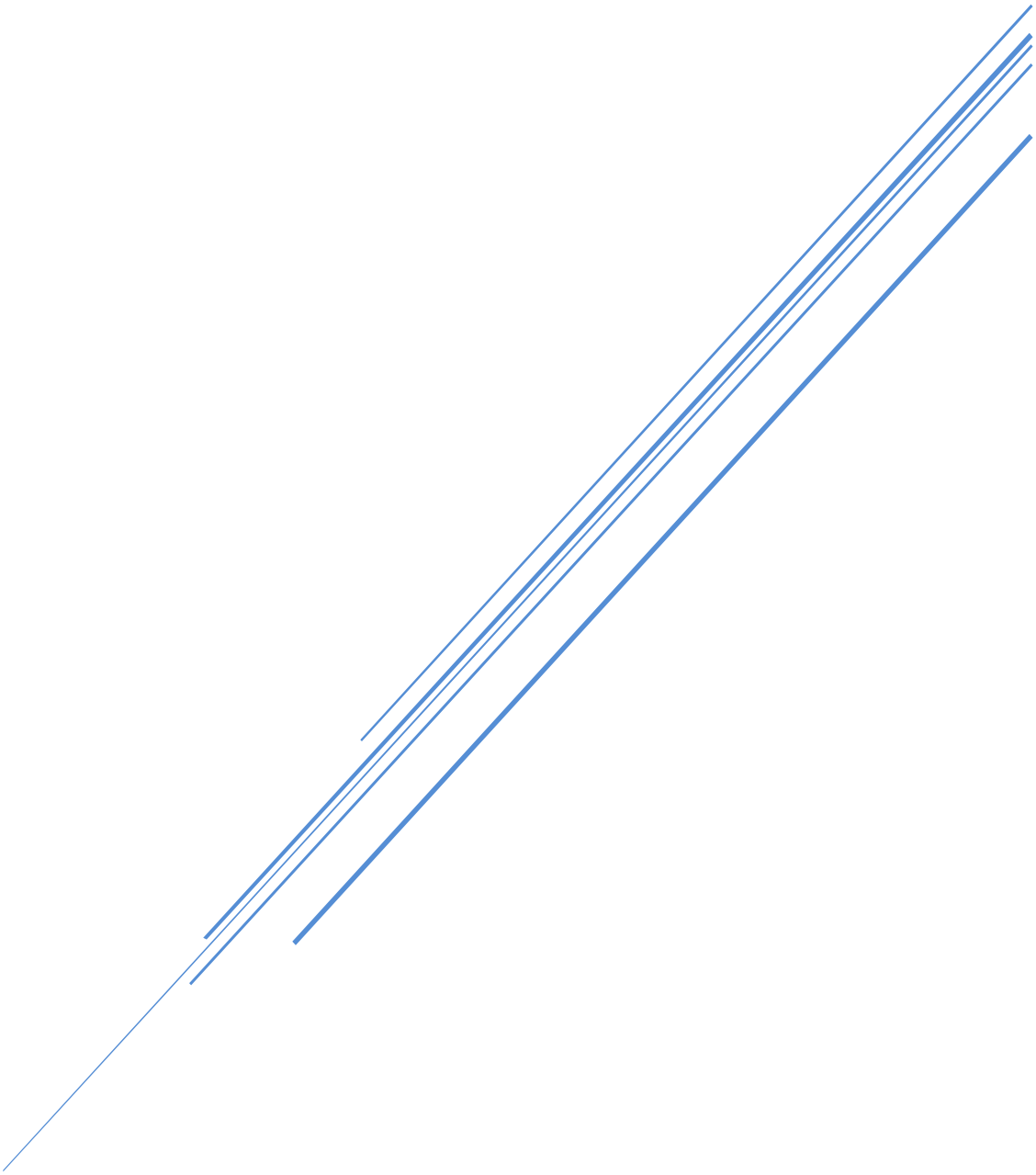
Se encuentra disponible este software para su visualización de código y uso del mismo, dado que está dentro de las características de software

distribuido y desarrollado libremente, libremente se refiere al poder modificar la fuente del programa sin restricciones de licencia, ya que muchas empresas de software encierran su código, ocultándolo y restringiéndose los derechos a sí misma.

Por lo tanto el equipo de investigación se encuentra disponible a inquietudes que puedan llegar a enriquecer lo producido, dado que se tiene pensado continuar con la temática.

ANEXO I

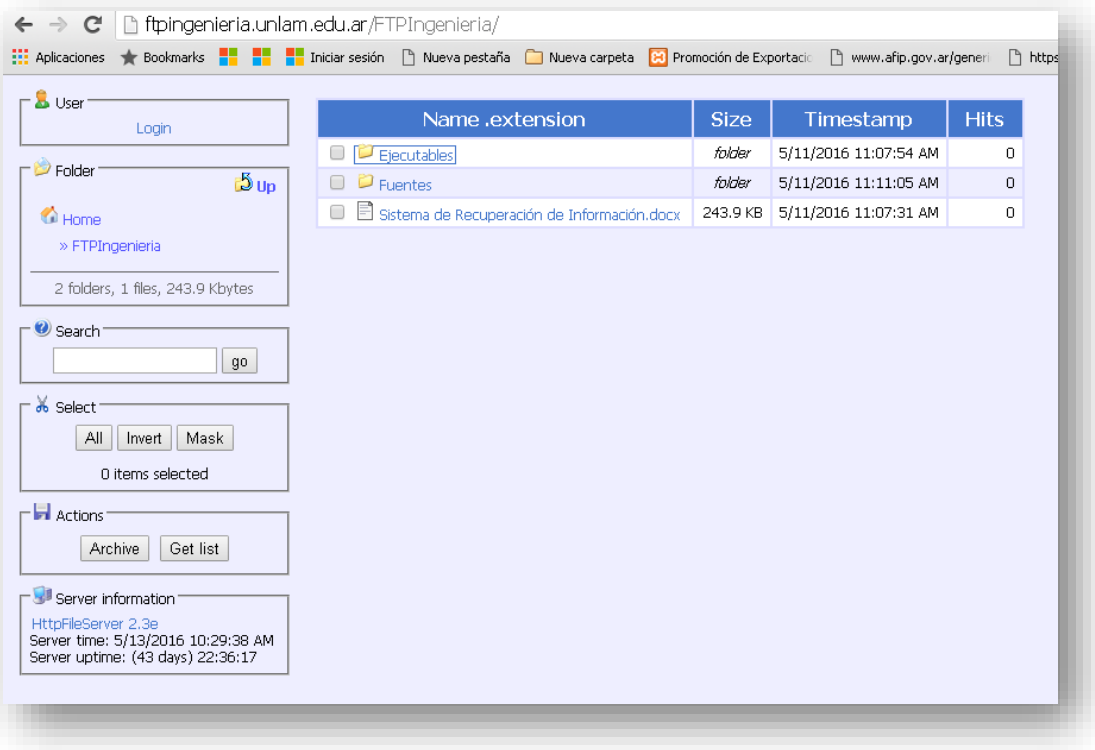
SISTEMA DE RECUPERACIÓN DE INFORMACIÓN- RI INSTALACIÓN Y USO



Instalación del sistema

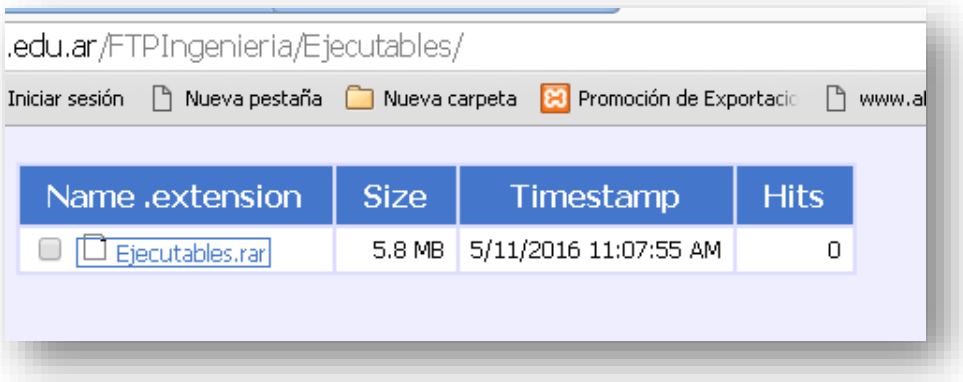
Al ingresar a la dirección: http://ftpingenieria.unlam.edu.ar/FTPIngenieria/

El sistema muestra la siguiente pantalla:

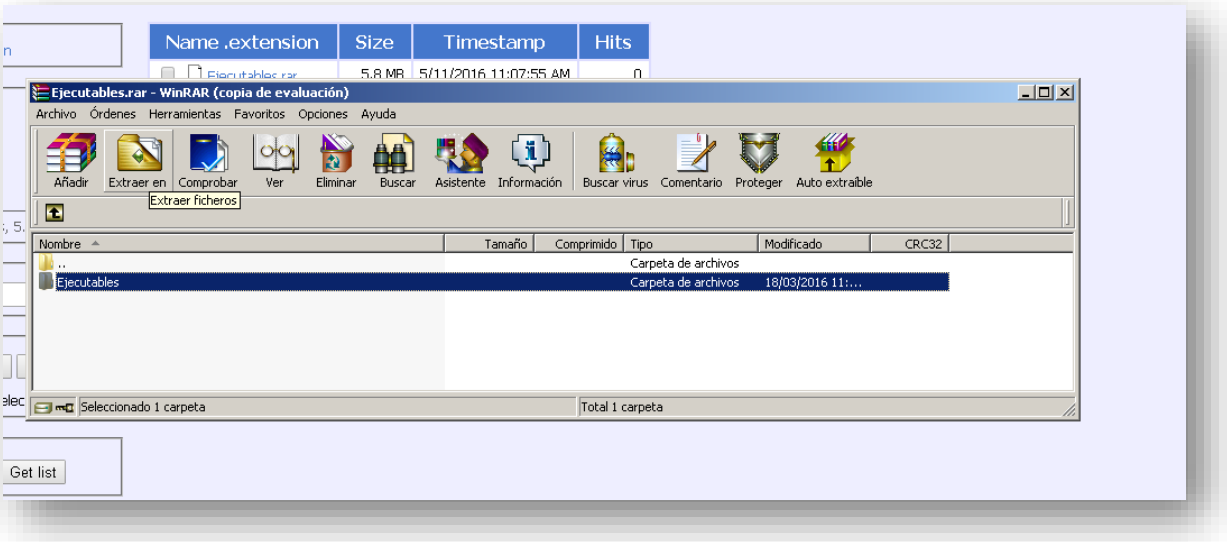


En esta pantalla se visualizan 2 carpetas: Ejecutables y Fuentes.

Presionando sobre la carpeta **Ejecutable** se visualiza el archivo **Ejecutables.rar**

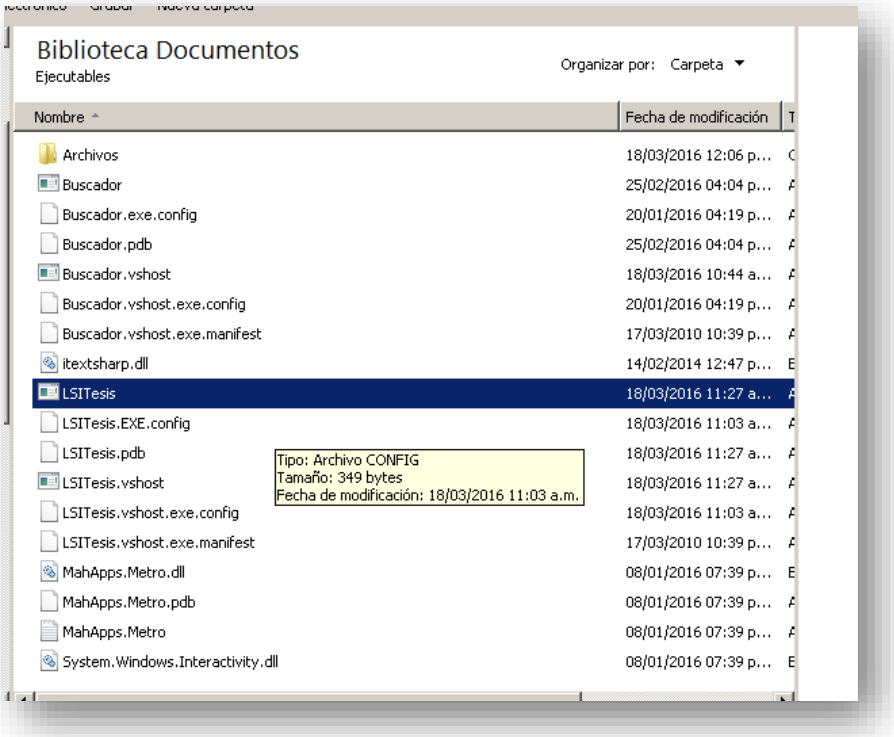


Al hacer doble clic sobre el mismo se ejecuta el programa win.rar para bajar el archivo mostrando la carpeta EJECUTABLES



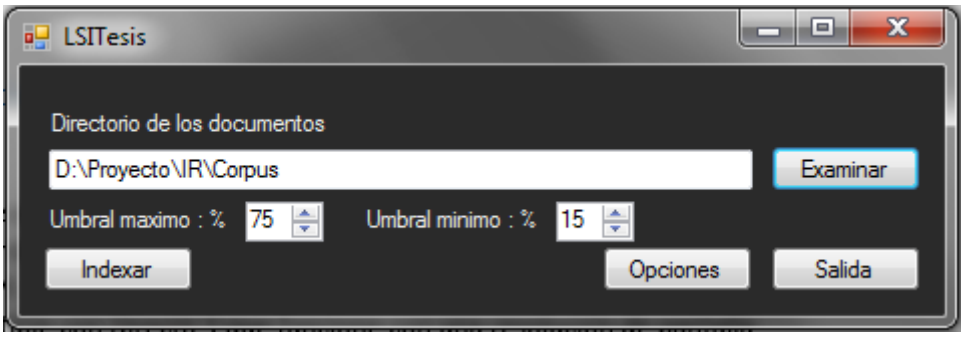
Se presiona sobre el icono “Extraer en” y se selecciona la carpeta donde se guardarán los archivos extraídos.

Luego entrar a la carpeta seleccionada y ejecutar el archivo llamado LSITesis



Uso del Sistema Indexador

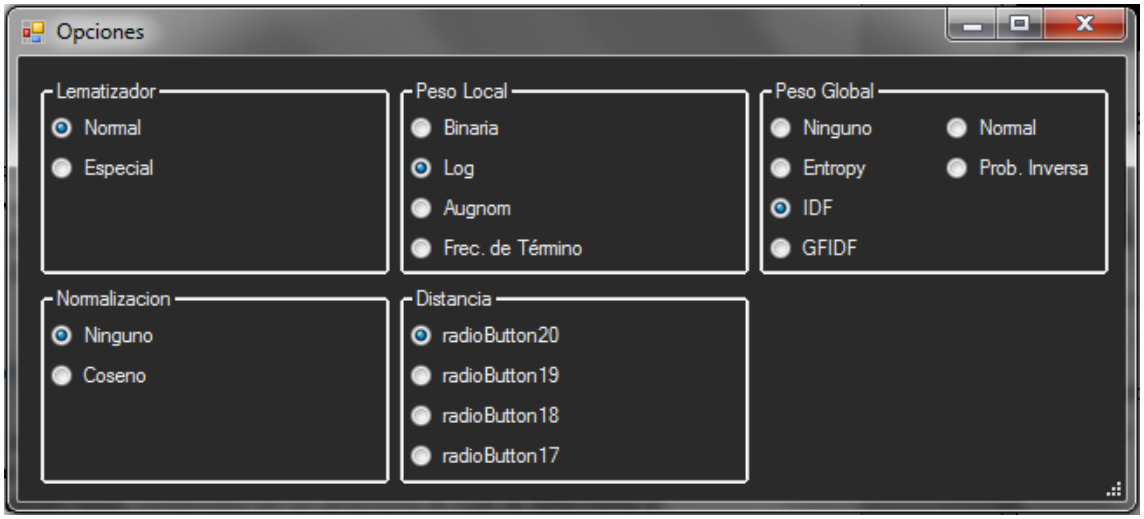
Los archivos que conforman el corpus deben estar contenidos en un directorio el cual se selecciona al comenzar la ejecución del programa presionando el botón “Examinar” o se puede ingresar su ruta de acceso en el cuadro de texto.



El umbral actúa como filtro, que toma como válidos los términos cuyo porcentaje calculado en base a su respectivo número de repeticiones se encuentre comprendido entre el mínimo y el máximo especificado. El objetivo es deshacerse de las palabras que sean poco representativas del contenido del corpus, ya sea porque están contenidas en muy pocos documentos o en casi todos. De esta manera se pretende mejorar la calidad del resultado obtenido al realizar una búsqueda, además de reducir el número de términos relacionados a los documentos y así mejorar el rendimiento del sistema durante el proceso de indexación.

Opciones de indexación

El botón “Opciones” proporciona acceso a la configuración de los parámetros de indexación. Al presionarlo se exhibe la siguiente ventana:



Configuración de lematizador: La opción “Normal” pone el sistema en un modo genérico en el cual la lematización se realiza sin un contexto definido (administrativo, jurídico, etc.), es decir, el corpus puede contener textos de múltiples temas y esto no afectaría el resultado obtenido. La opción “Especial” configura el proceso de lematización con un contexto específico (opción en desarrollo).

Fórmulas de los pesos locales:

Nombre	**Fórmula**
Binaria	l_{ij} = 1 el término existe en el documento l_{ij} = 0 el término no existe en el documento
Log	$l_{ij} = \log (1 + tf_{ij})$ tf_{ij} = nº de ocurrencias del término i en el documento j
Augnom	$l_{ij} = (x(tf_{ij}) + (tf_{ij} / \max_i (tf_{ij}))) / 2$ $x(tf_{ij}) = 1$ si $tf_{ij} > 0$, $x(tf_{ij}) = 0$ si $tf_{ij} = 0$
Frecuencia de termino	$l_{ij} = tf_{ij}$

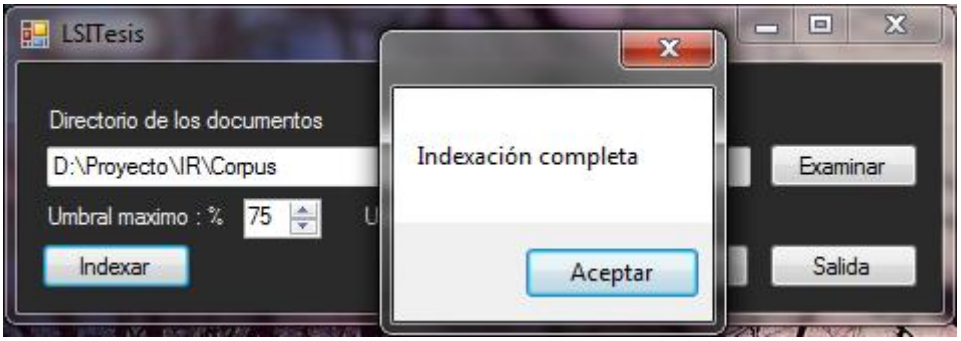
Fórmulas de los pesos globales:

Nombre	**Fórmula**
Ninguna	$g_i = 1$
Entropy	$g_i = 1 + (\sum_j (p_{ij} \log(p_{ij})) / \log n)$ $p_{ij} = tf_{ij} / \sum_j tfij$ n = cantidad de documentos del corpus
IDF	$g_i = \log (n / df_i)$ df_i = cant. de documentos en los que ocurre el término i
GFIDF	$g_i = \sum_j tf_{ij} / df$
Normal	$g_i = 1 / (\sum_j tf_{ij}^2)^{1/2}$
Probabilística inversa	$g_i = \log ((n - df_i) / df_i)$

Fórmulas de los factores de normalización:

Nombre	**Fórmula**
Ninguno	$d_j = 1$
Coseno	$d_j = (\sum_j (g_i l_{ij})^2)^{1/2}$

Una vez que se establece la configuración deseada se debe presionar el botón “Indexar” para comenzar el proceso de indexación. Al finalizar este, se muestra el siguiente mensaje:

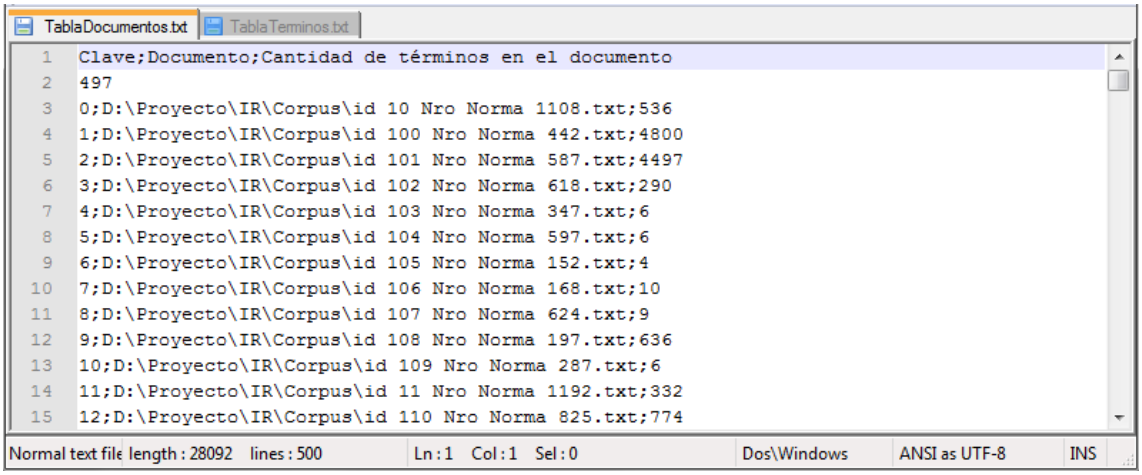


Archivos de salida

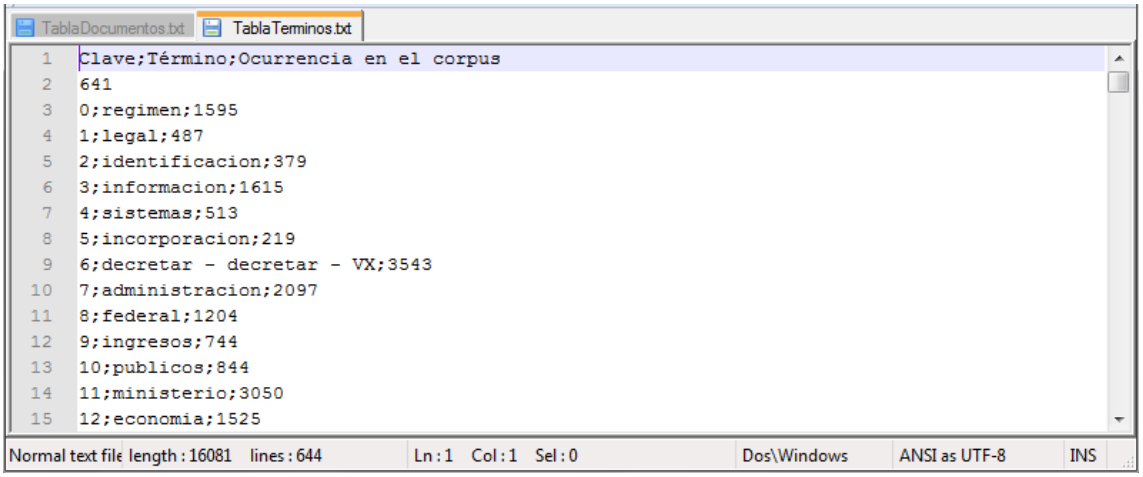
Como resultado se obtiene seis archivos de texto: “Matrizskinv.txt”, “MatrizTermDoc.txt”, “Matrizuk.txt”, “TablaDocumentos.txt”, “TablaTerminoDocumento.txt” y “TablaTerminos.txt”. Los primeros tres contienen las matrices que relacionan los documentos con los términos y sus pesos calculados dentro del corpus. Los tres restantes almacenan las claves, rutas de acceso y cantidad de documentos, además las claves, descripción y ocurrencia de los términos en cada fichero y en su totalidad. Estos se utilizan como entrada del sistema buscador, además del diccionario en español y el listado de palabras sincategoremáticas. Los archivos de salida se ubican en el mismo directorio donde se encuentran los archivos de entrada (explicado con anterioridad).

A continuación se muestra un ejemplo de tres de los archivos de obtenidos al finalizar el proceso de indexación. TablaDocumentos detalla la ruta de acceso, nombre, la cantidad de términos y la clave de identificación de cada archivo del corpus. TablaTerminos muestra el detalle, la ocurrencia total y la clave de identificación de cada término en el corpus. Por ultimo en TablaTerminoDocumento se encuentra la ocurrencia de cada término en cada documento y la relación entre ellos mediante el uso de sus respectivas claves.

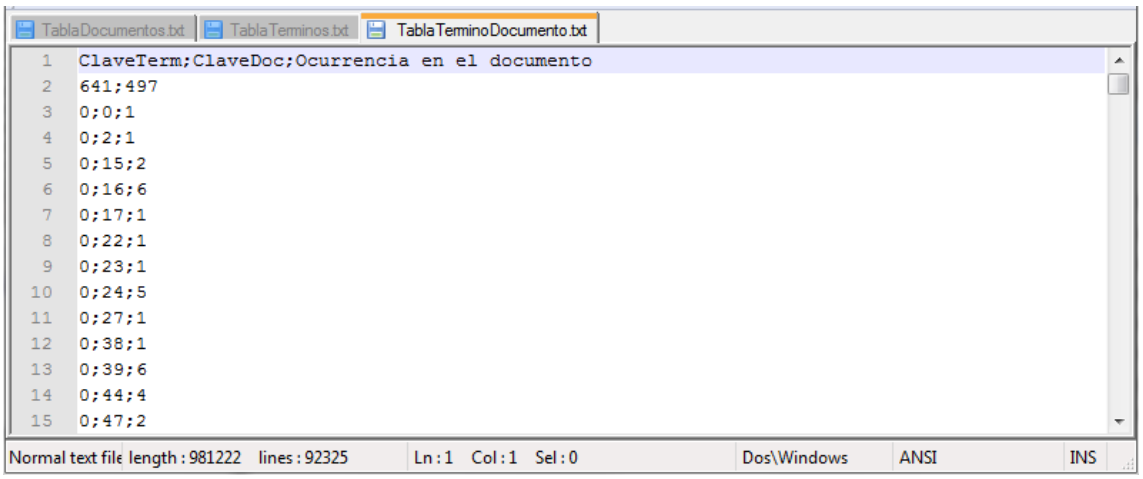
TablaDocumentos.txt:



TablaTerminos.txt:

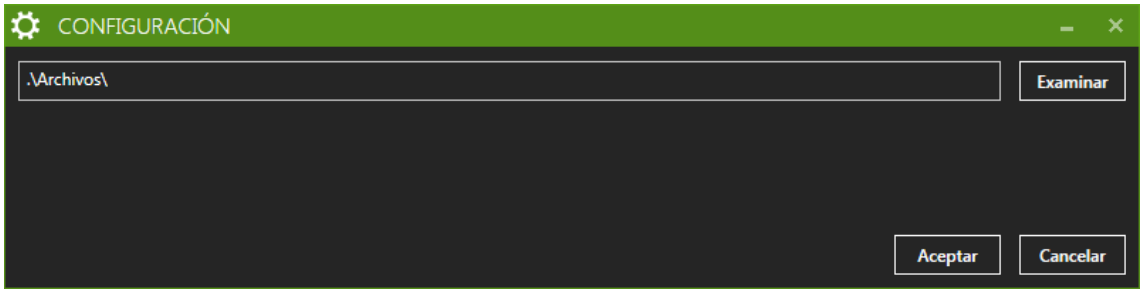


TablaTerminoDocumento.txt:



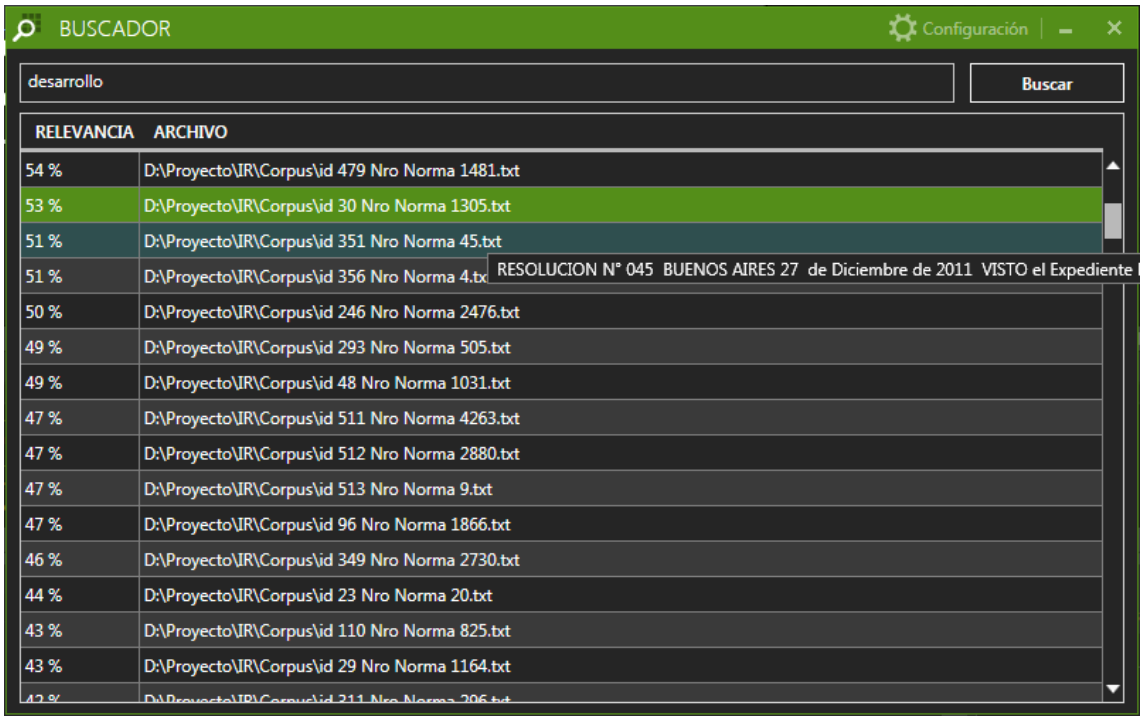
Uso del Sistema Buscador

La ubicación de los archivos de entrada que emplea el sistema de búsqueda se puede establecer presionando el botón “Configuración”, que se encuentra en la barra de títulos de la ventana principal. Se presenta el siguiente dialogo:



Para realizar una búsqueda se debe ingresar una palabra o frase en el cuadro de texto ubicado en la parte superior de la ventana y presionar la tecla Entrar o el botón “Buscar”. Si la búsqueda no arroja un resultado satisfactorio se muestra el mensaje correspondiente. De lo contrario, se listan los archivos cuyo contenido se relaciona con el texto ingresado y la correspondiente relevancia expresada como porcentaje.

Si se pone el cursor sobre el nombre de un archivo, el sistema muestra un breve resumen su contenido. Además se puede abrir cualquier archivo haciendo doble clic sobre la fila de la lista correspondiente.



Archivos del corpus

El corpus (conjunto de textos almacenados de forma digital) a indexar puede estar conformado por archivos con las siguientes extensiones: “.txt”, “.pdf”, “.docx” y “.html”. En los últimos tres casos el sistema realiza una conversión a formato “.txt” de manera de obtener una entrada normalizada. Para la conversión de archivos PDF se utiliza la biblioteca iTextSharp, la cual es de código abierto y permite la manipulación de este tipo de ficheros. Para la conversión de archivos DOCX se emplea la biblioteca de Microsoft “Microsoft.Office.Interop.Word”. Por último, para la conversión de archivos HTML se usa un método propio que extrae el contenido relevante en forma de texto plano y elimina los metadatos contenidos.

El proceso de normalización del corpus genera como resultado un directorio donde se encuentran todos los ficheros en formato TXT (los que originalmente eran de este formato y los convertidos) y un subdirectorio donde se encuentran los archivos de otros formatos. La ubicación de este directorio puede configurarse a conveniencia del usuario.

Archivos de entrada

Tanto el sistema indexador como el de consulta necesitan además dos archivos. El archivo “StopWordsFinal.txt” contiene las palabras (677) sin peso semántico o sincategoremáticas. Este se utiliza para eliminar dichas palabras de la colección de términos que se obtiene durante la lematización del corpus a indexar. En el programa de consulta se usa para lo mismo, solo que aplicado al texto de búsqueda. El archivo “espa~nol.words” es un diccionario en español que contiene los lexemas (53.065) aceptados como válidos. Los términos que no se encuentran en este diccionario no se tienen en cuenta en el proceso de lematización.

Ubicación de los archivos utilizando el código fuente

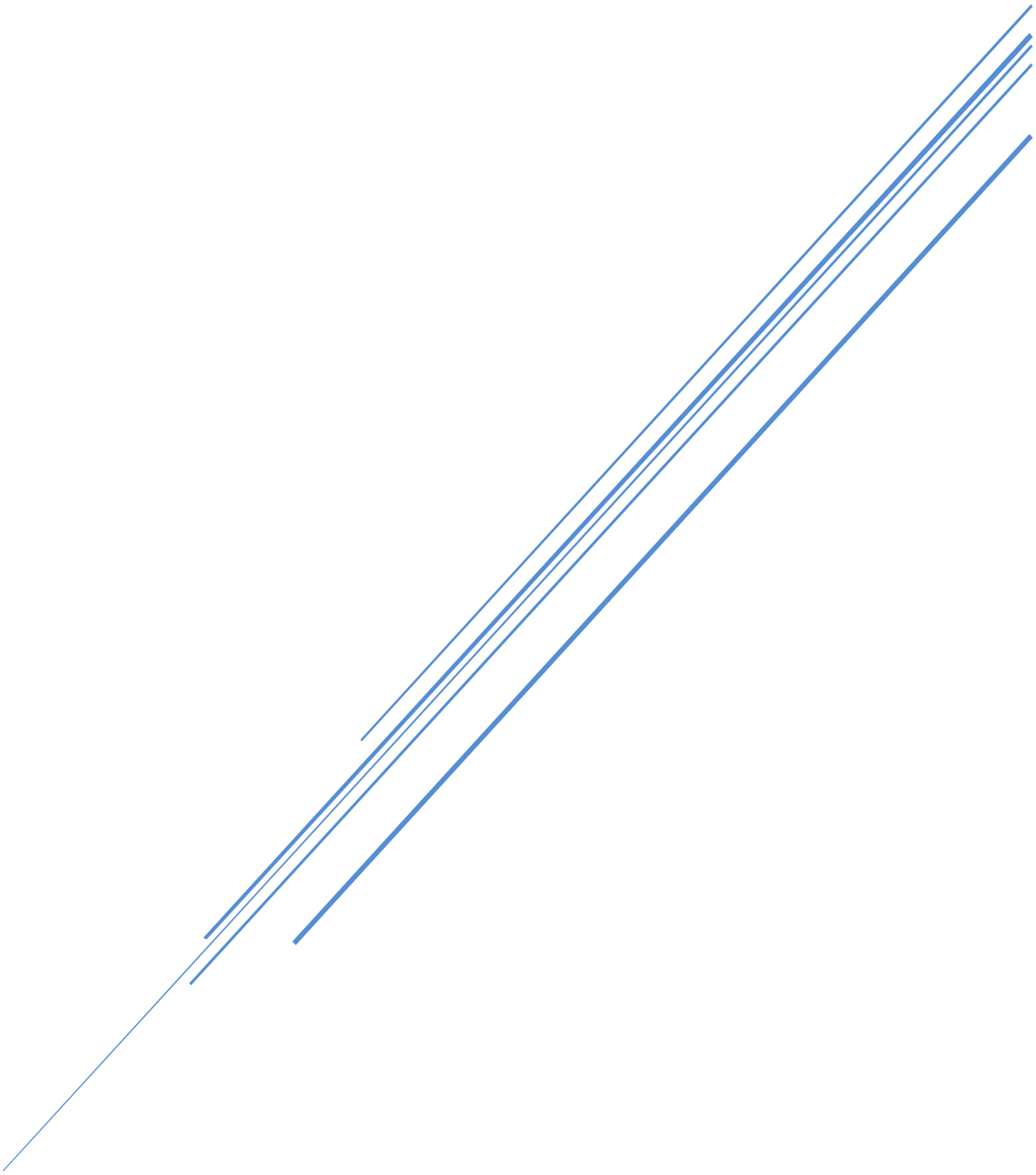
Se encuentran en el directorio "…\bin\Debug\Archivos", ubicado en la carpeta que contiene el proyecto del sistema indexador y de la misma forma para el sistema buscador. En el primer caso este directorio puede modificarse en el archivo "app.config". En el segundo caso puede modificarse en el archivo "app.config" o desde el menú de configuración.

Ubicación de los archivos utilizando los ejecutables

Están en el directorio "Archivos", el cual se encuentra en la misma carpeta que contiene los ejecutables y archivos adicionales. En el sistema indexador no puede modificarse la ruta de acceso a estos archivos. En el sistema buscador puede cambiarse desde el menú de configuración.

ANEXO II

SISTEMA DE RECUPERACIÓN DE INFORMACIÓN- RI



Comparación del sistema con y sin LSI

Especificaciones técnicas

Sistema Operativo: Windows Seven Profesional.

Procesar: Intel(R) Core(TM) i7-330 CPU 3.40GHz.

Memoria (RAM): 8 GB.

Disco Rígido: 500 GB.

Placa de video: GeForce 1GB.

Mouse.

Teclado.

Monitor 18'' (pulgadas).

Ámbito de la prueba

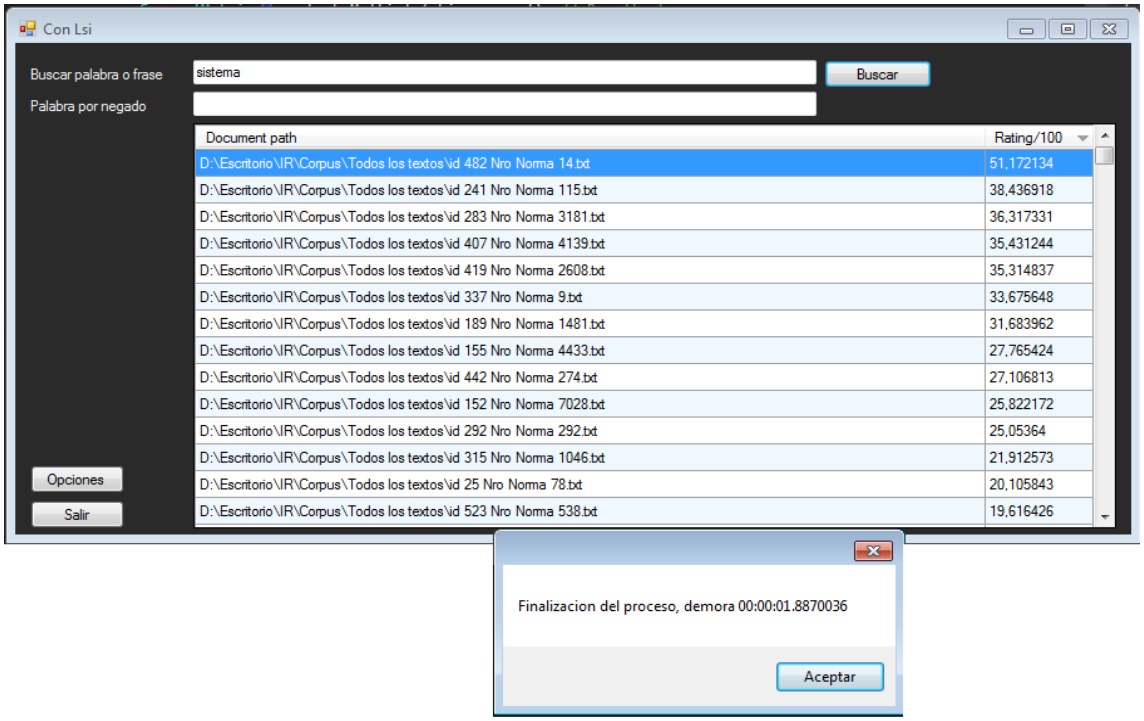
Se realizó con un corpus de 497 (cuatrocientos noventa y siete) documentos.
Se utilizó para el cálculo de peso local: Log, y para el peso global: IDF, a su vez se utilizó el método de normalización por coseno.
En cuanto al umbral, se utilizó 15% en el inferior y un 75% en el superior.
Los tiempos obtenidos serán en segundos, y serán truncados en 2 dígitos en caso de que este dé con coma.

Prueba Nº1

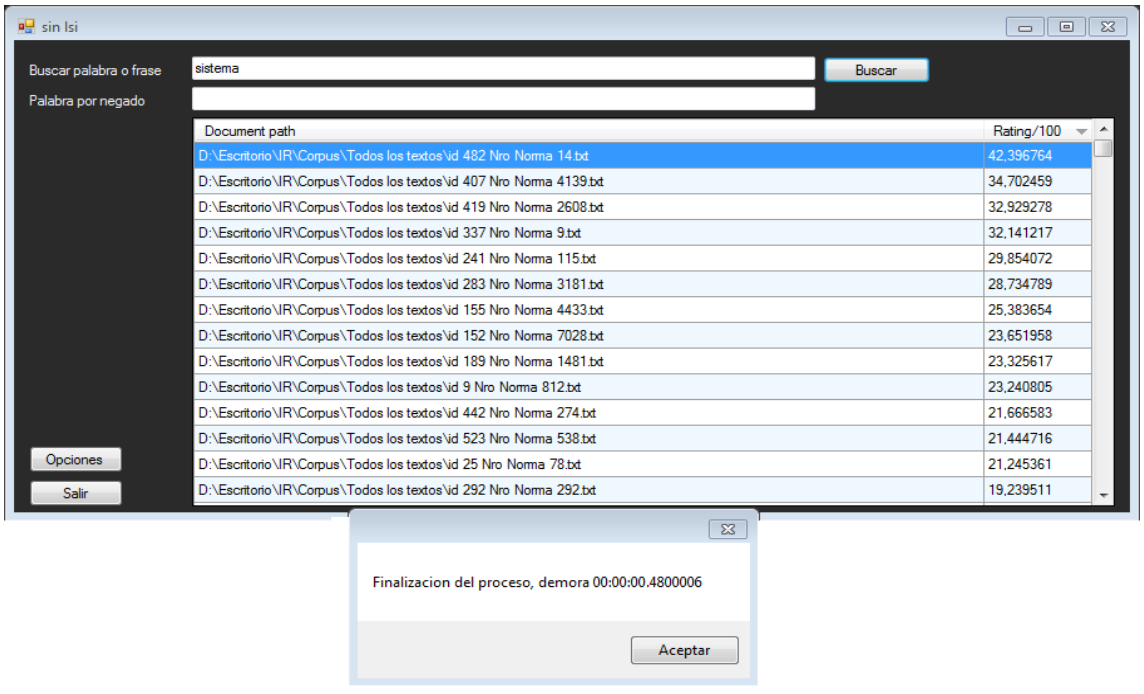
Realiza una búsqueda en el cual la palabra se encuentre en el corpus.

La palabra a buscar es: “sistema”

Con Lsi



Sin Lsi



Resultado de la prueba Nº1

Con Lsi:

Tiempo: 1,89 segundos.

Sin Lsi:

Tiempo 0,48 segundos.

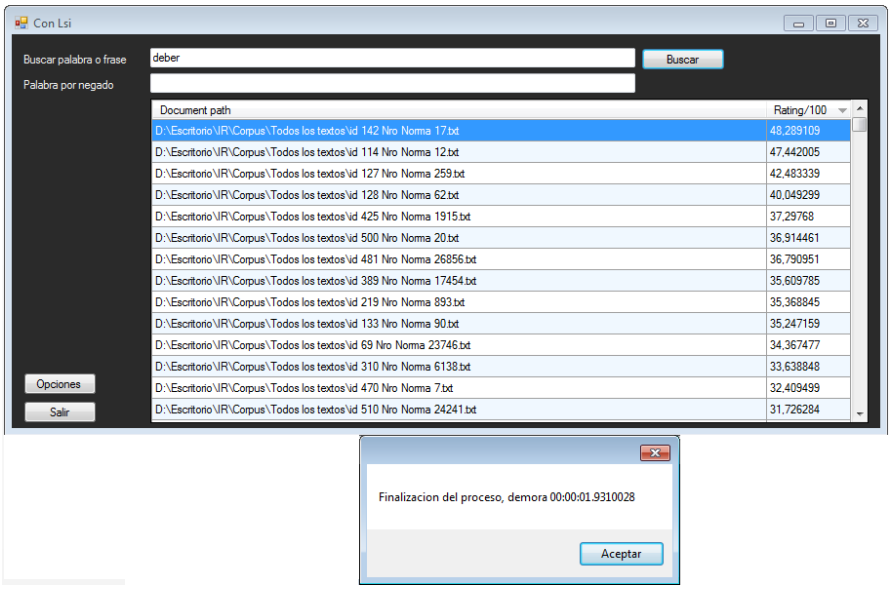
Se puede observar la diferencia de tiempos, utilizar lsi tardó más que sin lsi, también se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos.

Prueba Nº2

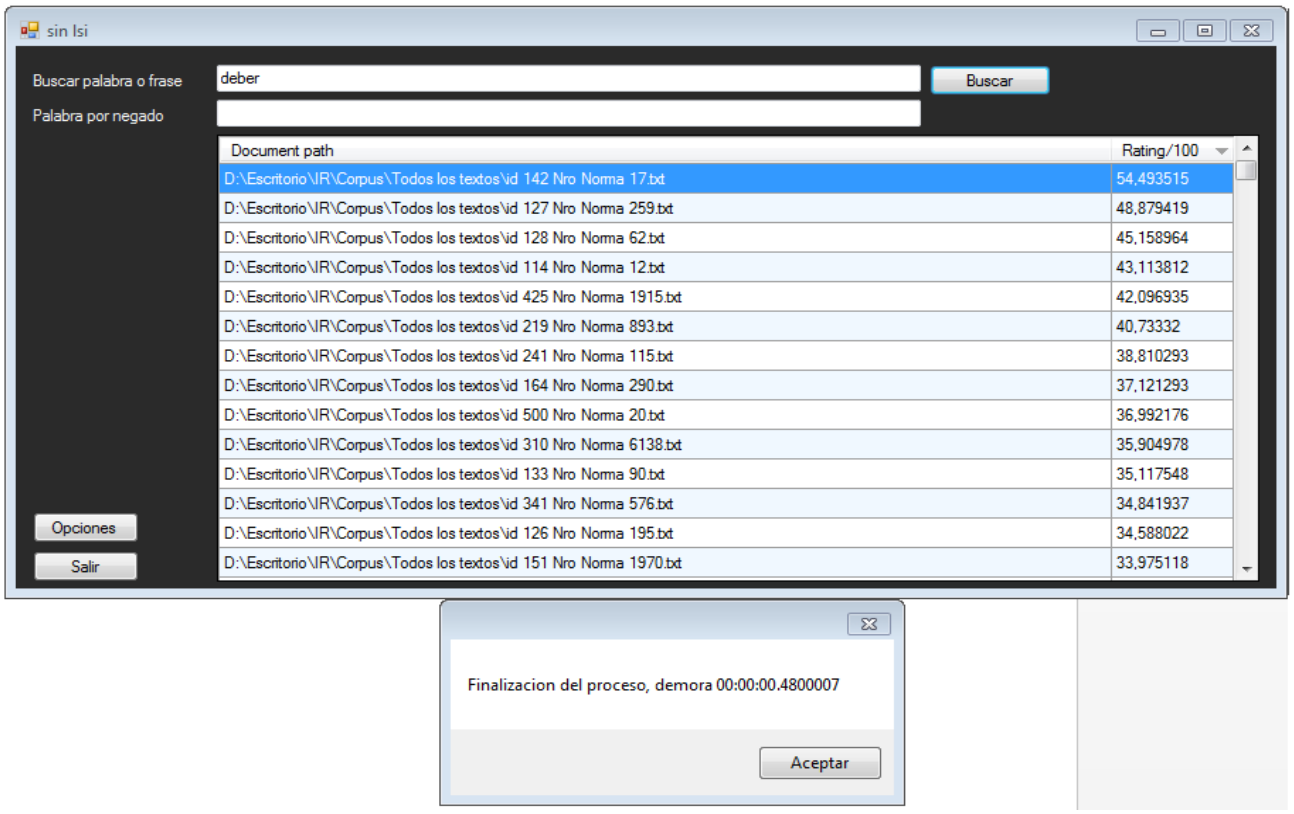
Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea la más mencionada y cumpla con el umbral.

La palabra a buscar es: “deber”

<u>Con Lsi</u>



Sin Lsi



Resultado de la prueba Nº2

Con Lsi:

Tiempo: 1,93 segundos.

Sin Lsi:

Tiempo 0,48 segundos.

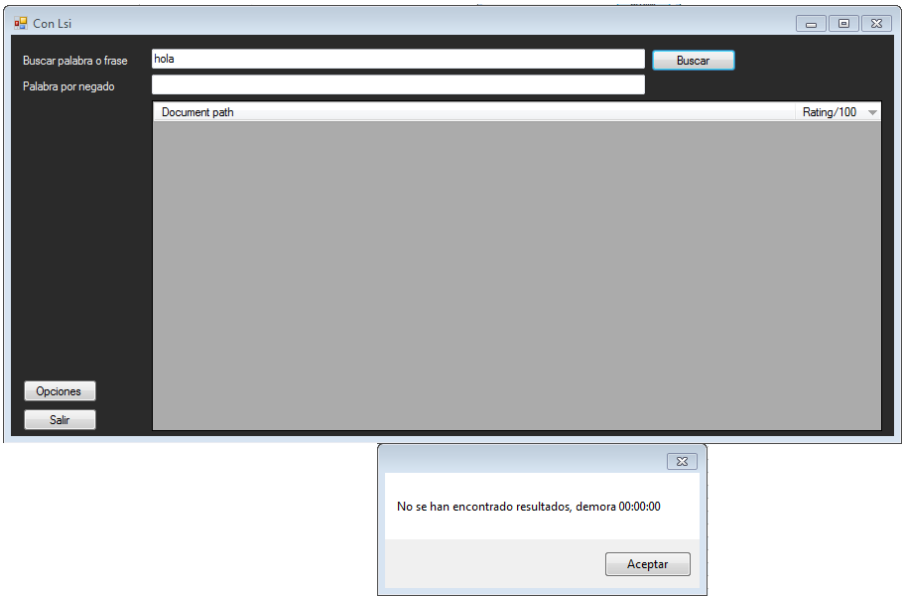
Se mantiene la premisa realizada anteriormente, con lsi tardó más que sin lsi. A diferencia de la primera prueba, se observa que la utilización del lsi en el ranqueo fue más proporcional que sin lsi.

Prueba Nº3

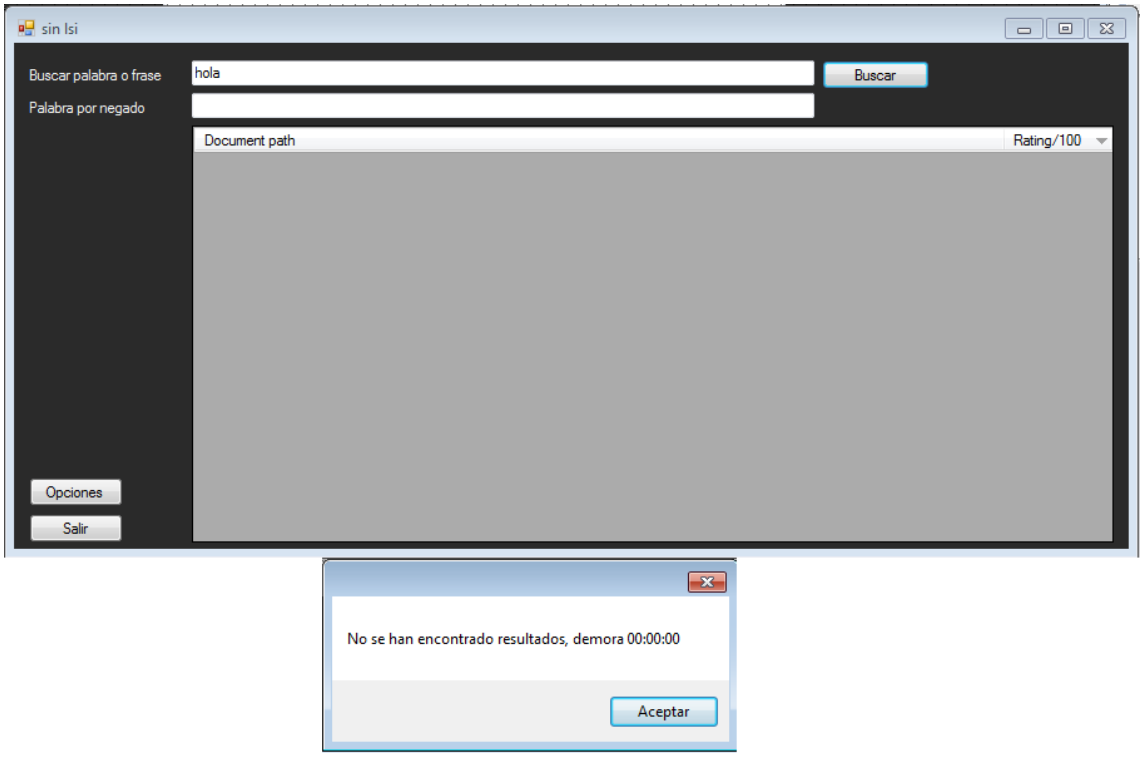
Realiza una búsqueda en el cual la palabra no se encuentre en ningún corpus.

La palabra a buscar es: “hola”

Con Lsi



Sin Lsi



Resultado de la prueba Nº3

Con Lsi:

Tiempo: 0 segundos.

Sin Lsi:

Tiempo 0 segundos.

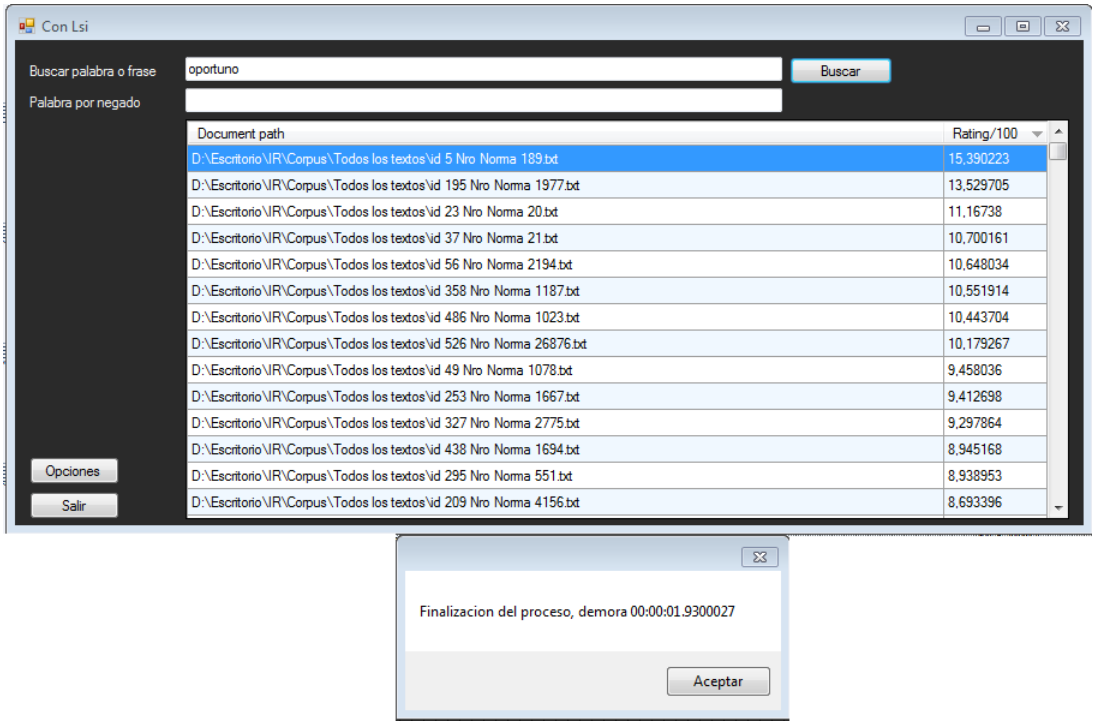
Se obtuvo el mismo resultado en ambos sistemas, ningún documento encontrado.

Prueba Nº4

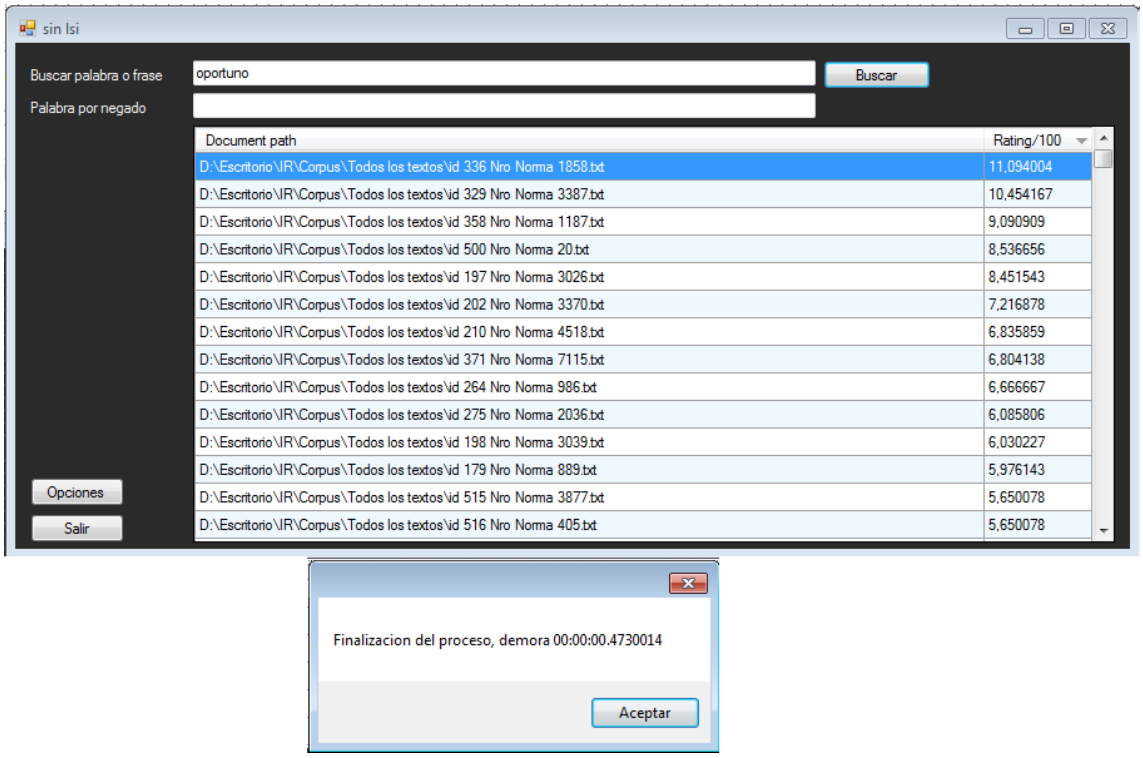
Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea la menos mencionada y cumpla con el umbral.

La palabra a buscar es: “oportuno”

Con Lsi



Sin Lsi



Resultado de la prueba Nº4

Con Lsi:

Tiempo: 1,93 segundos.

Sin Lsi:

Tiempo 0,47 segundos.

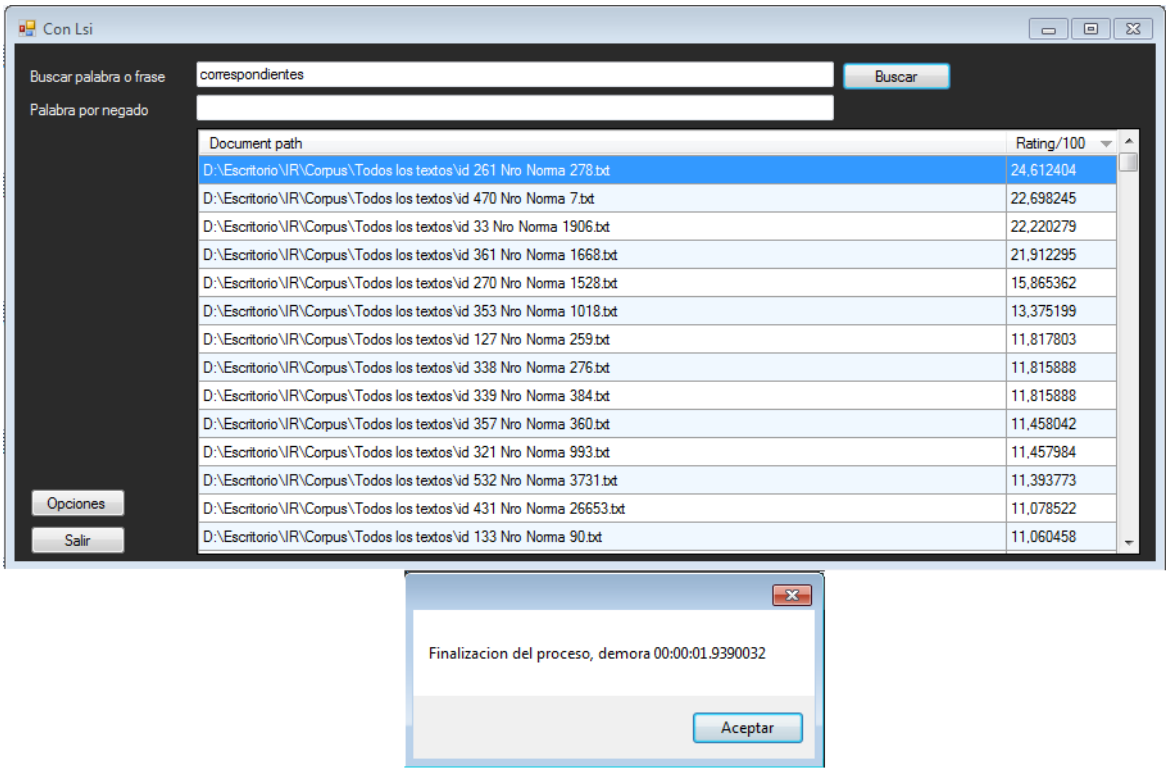
Los tiempos siguen siendo similares a la prueba Nº1 y Nº2, a diferencia de las pruebas mencionadas, ocurre que hay un poco más de proporción en el raqueo en el caso del sin lsi que en el que contiene el lsi. En este último ocurre esa distribución a partir del 3 documento en adelante.

Prueba Nº5

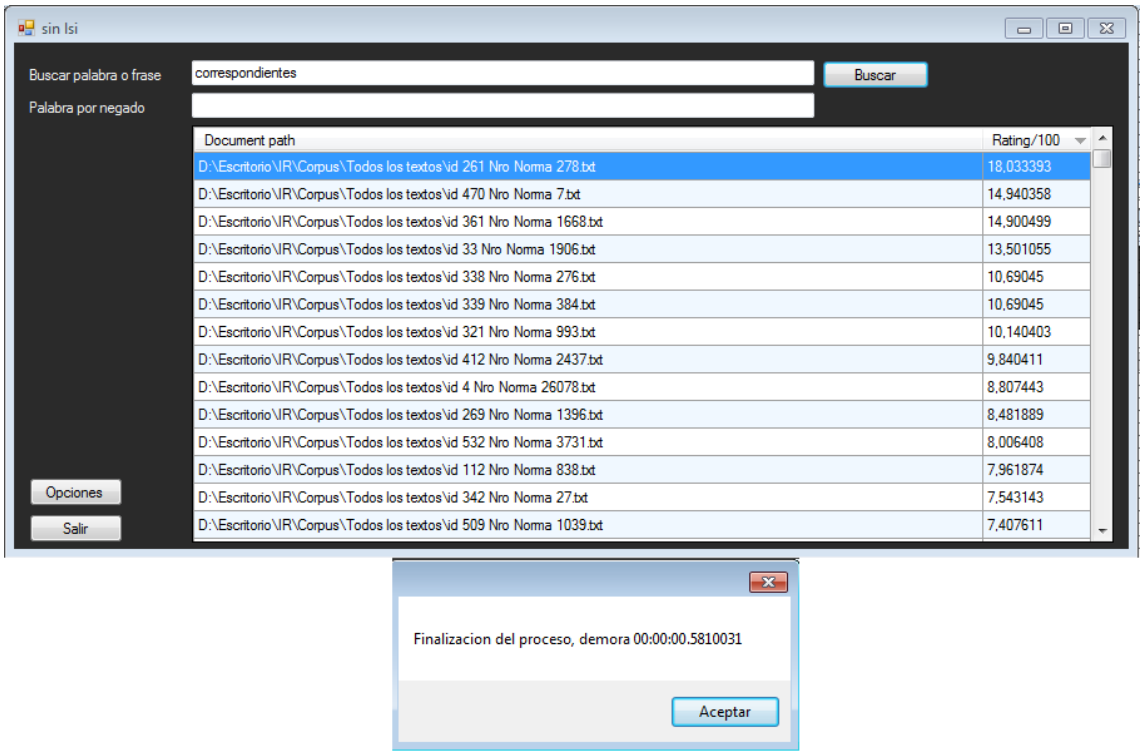
Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea la palabra más larga.

La palabra a buscar es: “correspondientes”

Con Lsi



Sin Lsi



Resultado de la prueba Nº5

Con Lsi:

Tiempo: 1,93 segundos.

Sin Lsi:

Tiempo 0,58 segundos.

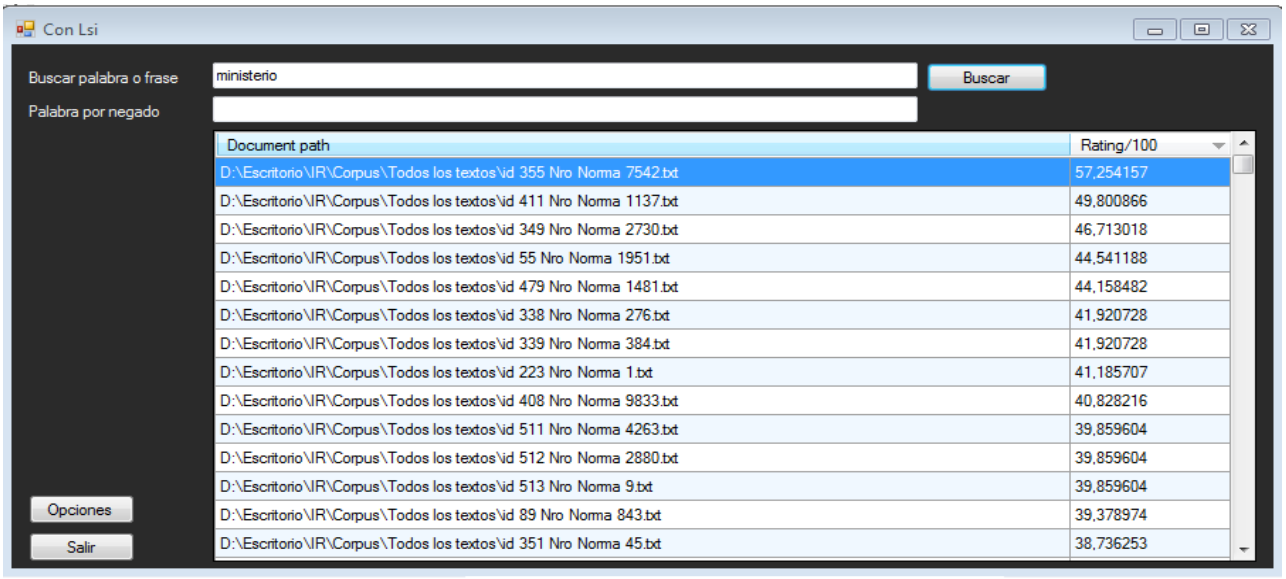
Tiempo similar a las obtenidas en las pruebas anteriores. Y una diferencia en el ranqueo en los documentos.

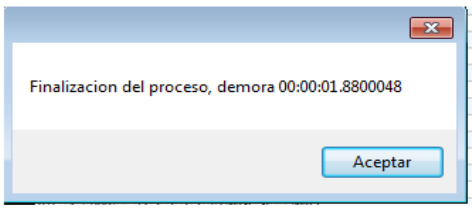
Prueba Nº6

Realiza una búsqueda en el cual la palabra se encuentre en el corpus.

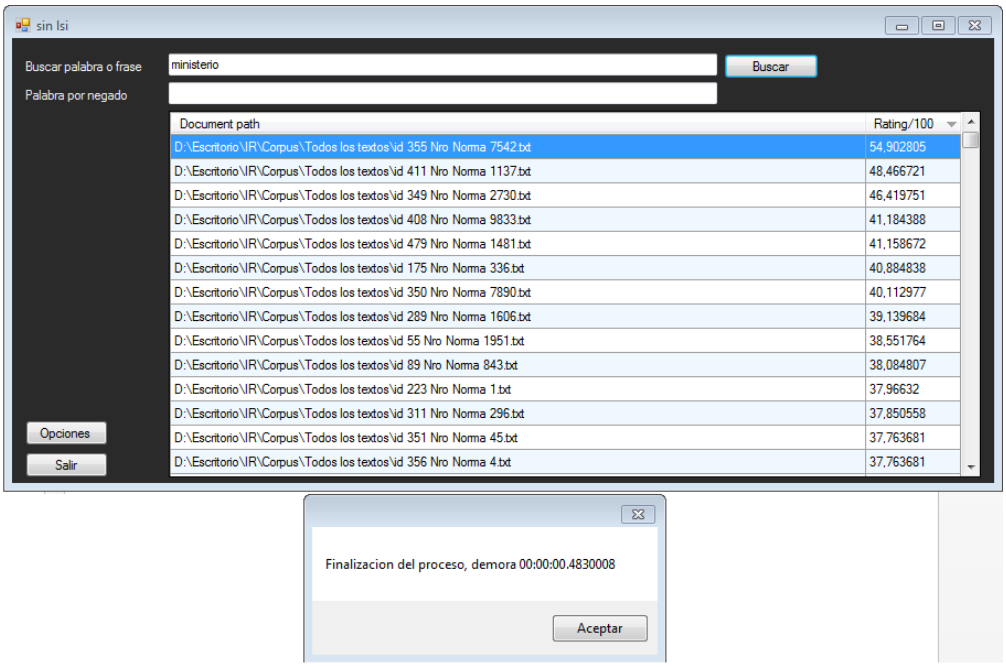
La palabra a buscar es: “ministerio”

Con Lsi





Sin Lsi



Resultado de la prueba Nº6

Con Lsi:

Tiempo: 1,88 segundos.

Sin Lsi:

Tiempo 0,48 segundos.

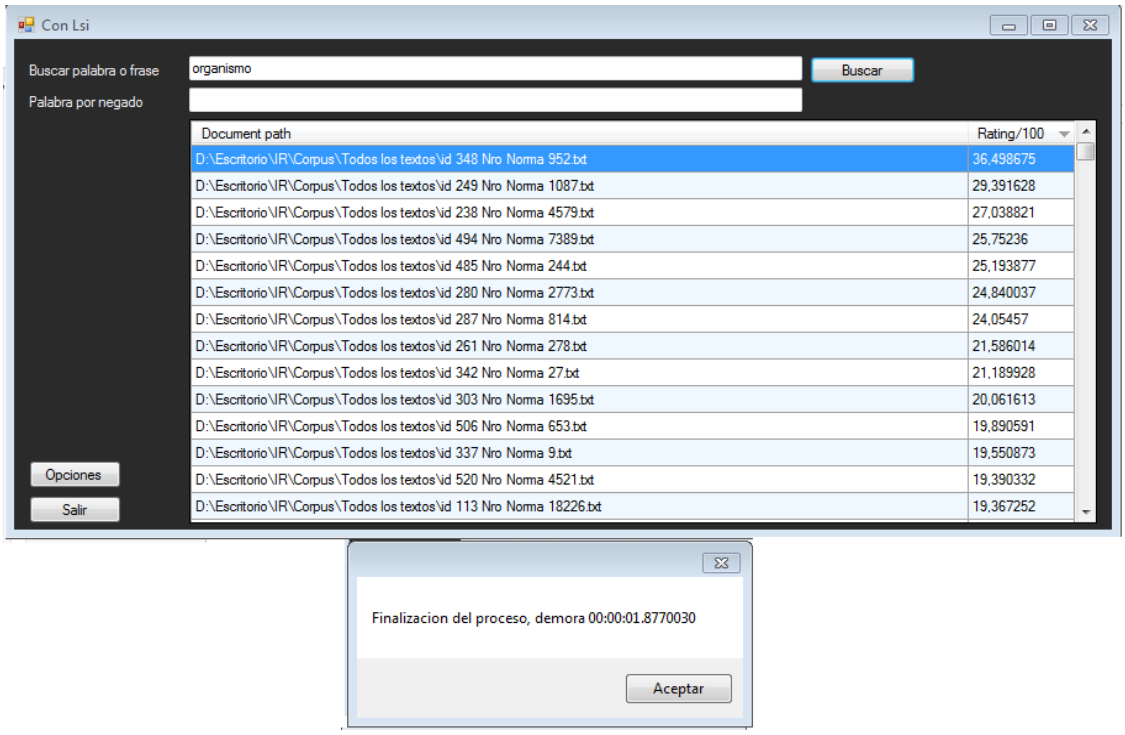
Se puede observar la diferencia de tiempos, utilizar lsi tardó más que sin lsi, también se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos.

Prueba Nº7

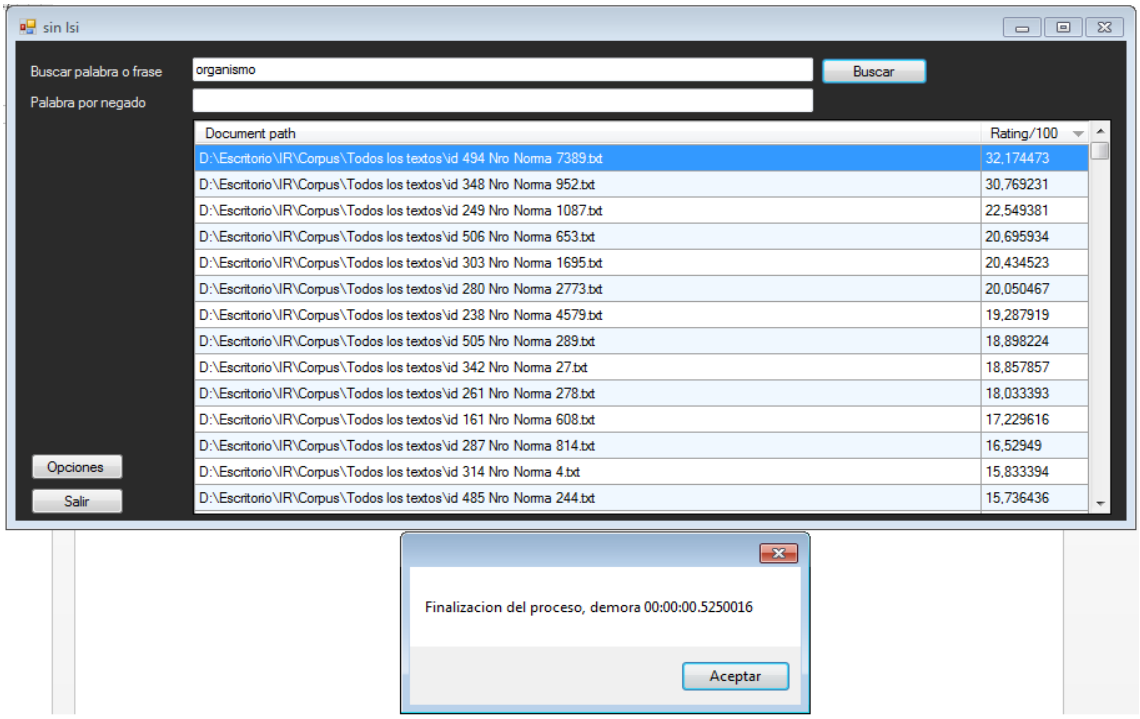
Realiza una búsqueda en el cual la palabra se encuentre en el corpus.

La palabra a buscar es: “organismo”

Con Lsi



Sin Lsi



Resultado de la prueba Nº7

Con Lsi:

Tiempo: 1,88 segundos.

Sin Lsi:

Tiempo 0,52 segundos.

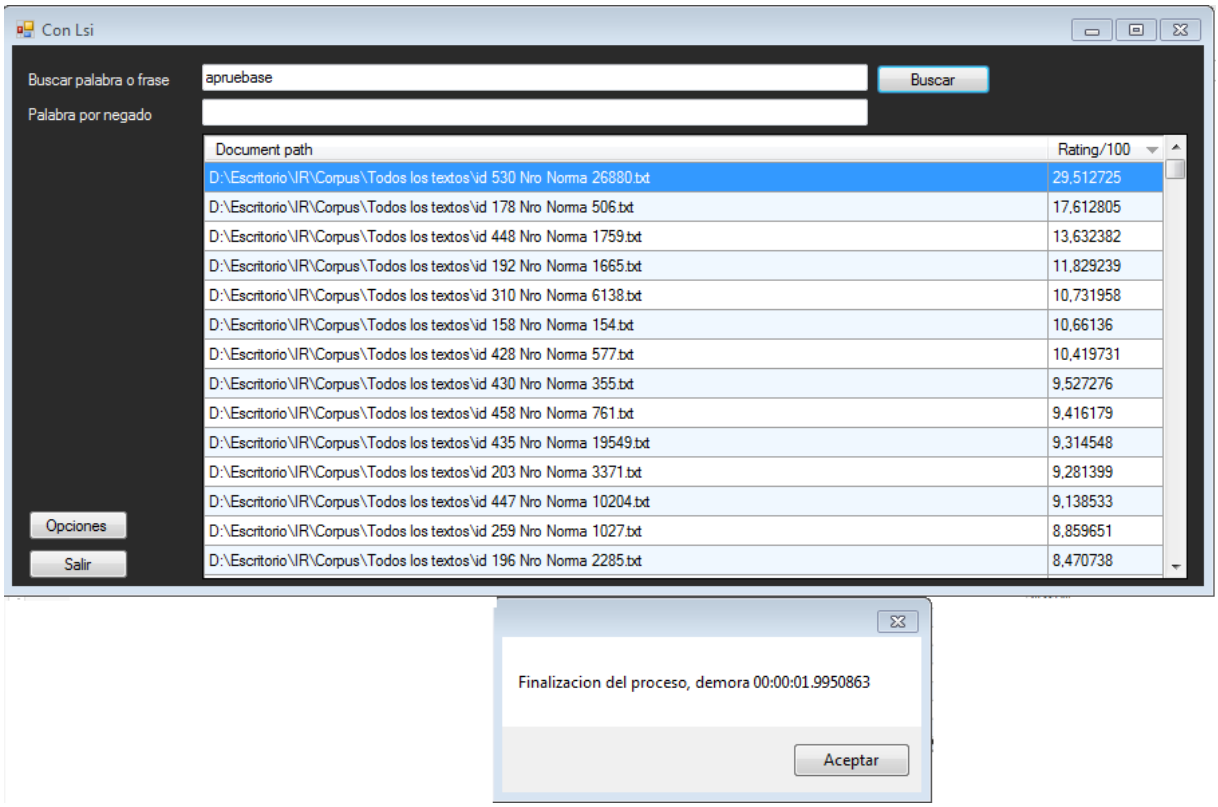
Se puede observar la diferencia de tiempos, utilizar lsi tardó más que sin lsi, también se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos.

Prueba Nº8

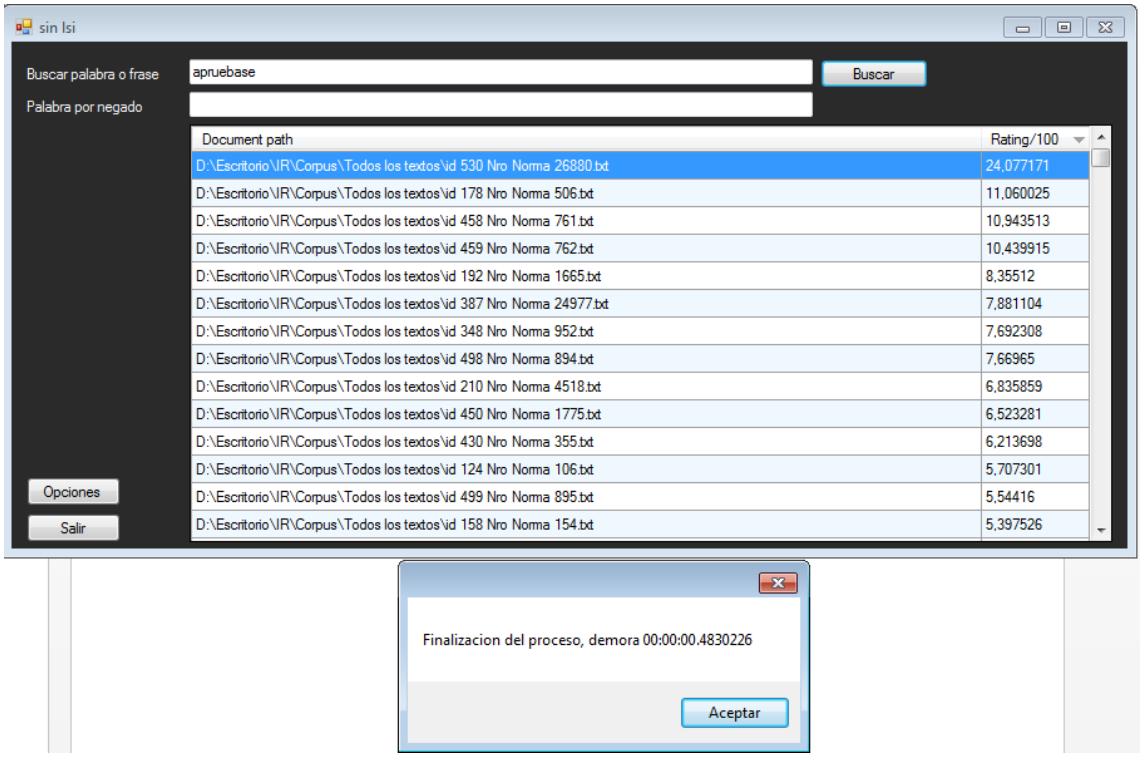
Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea una de las menos mencionadas y cumpla con el umbral.

La palabra a buscar es: “apruebase”

Con Lsi



Sin Lsi



Resultado de la prueba Nº8

Con Lsi:

Tiempo: 1,99 segundos.

Sin Lsi:

Tiempo 0,49 segundos.

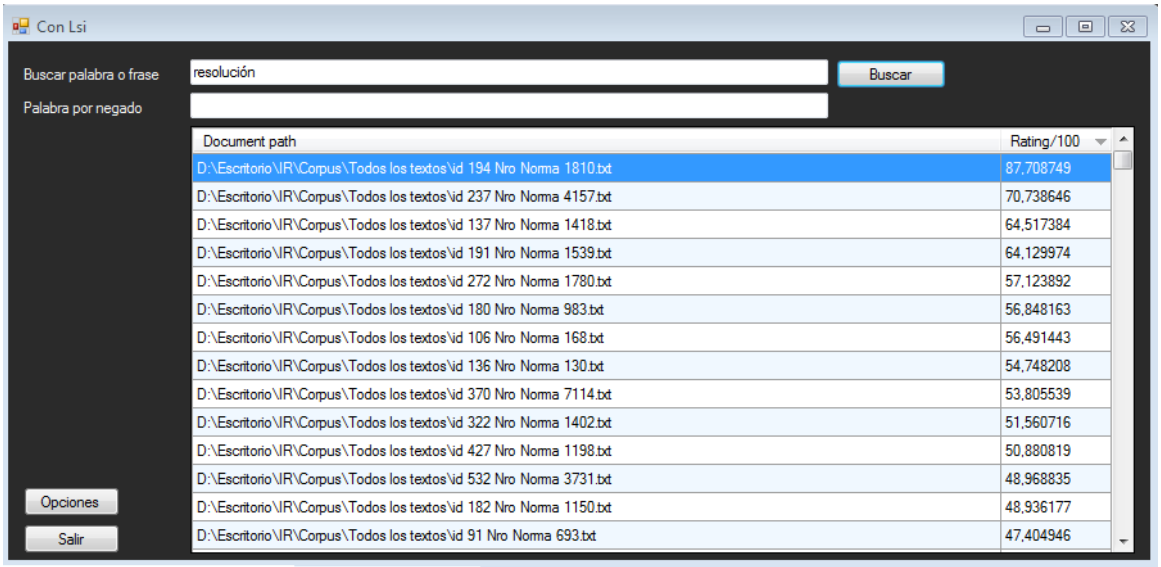
Se puede observar la diferencia de tiempos, utilizar lsi tardó más que sin lsi, también se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos, lo cual no había sucedido con la prueba Nº4 que hubo una distribución proporcional en el raqueo.

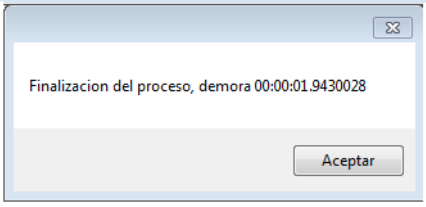
Prueba Nº9

Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea una de las más mencionadas y a su vez esté en la mayoría de los documentos y cumpla con el umbral.

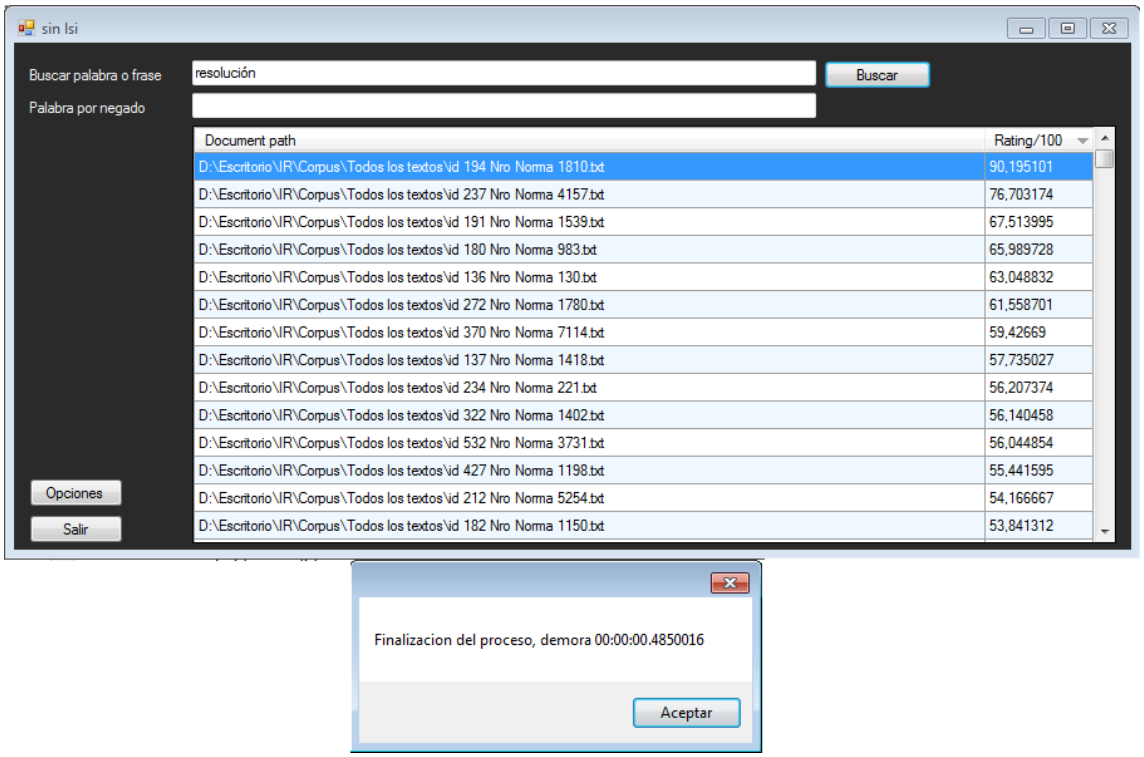
La palabra a buscar es: “resolución”

Con Lsi





Sin Lsi



Resultado de la prueba Nº9

Con Lsi:

Tiempo: 1,94 segundos.

Sin Lsi:

Tiempo 0,49 segundos.

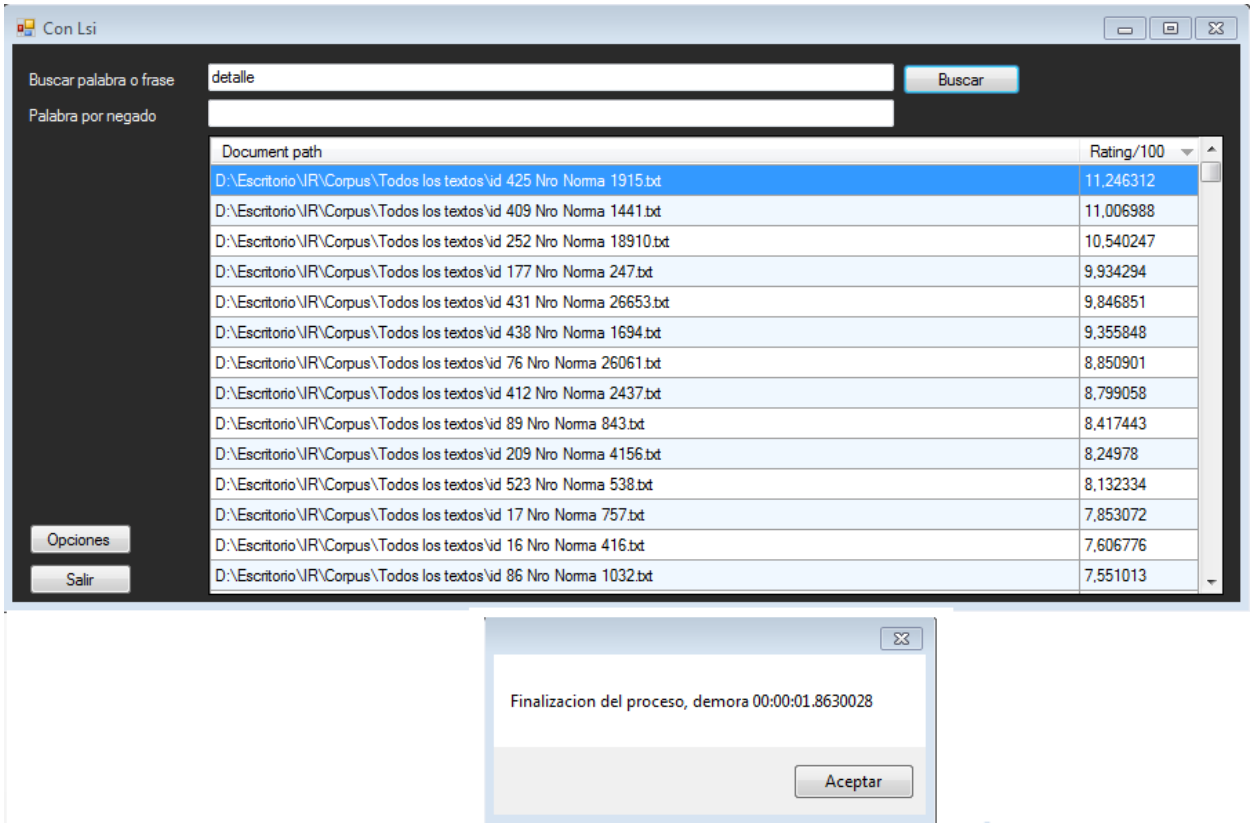
Se puede observar la diferencia de tiempos, utilizar lsi tardó más que sin lsi, también se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos, lo cual no había sucedido con la prueba Nº2 que hubo una distribución proporcional en el raqueo.

Prueba Nº10

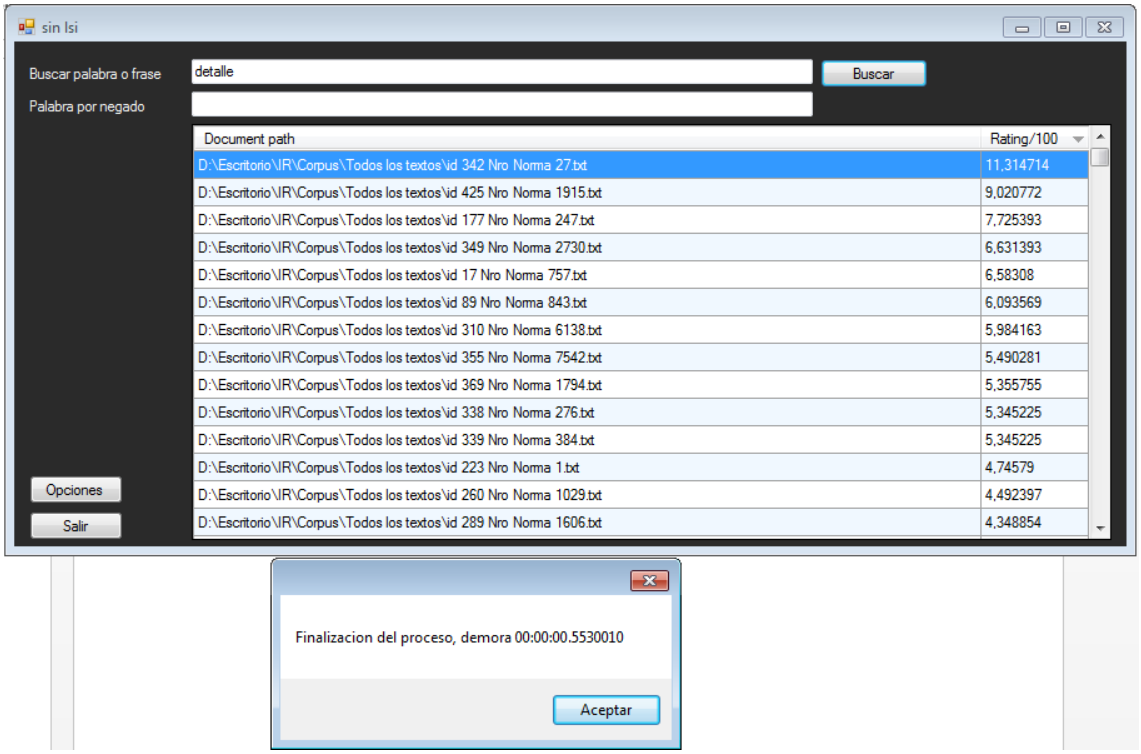
Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea una de las que en menos corpus se encuentre y cumpla con el umbral.

La palabra a buscar es: “detalle”

Con Lsi



Sin Lsi



Resultado de la prueba Nº10

Con Lsi:

Tiempo: 1,86 segundos.

Sin Lsi:

Tiempo 0,55 segundos.

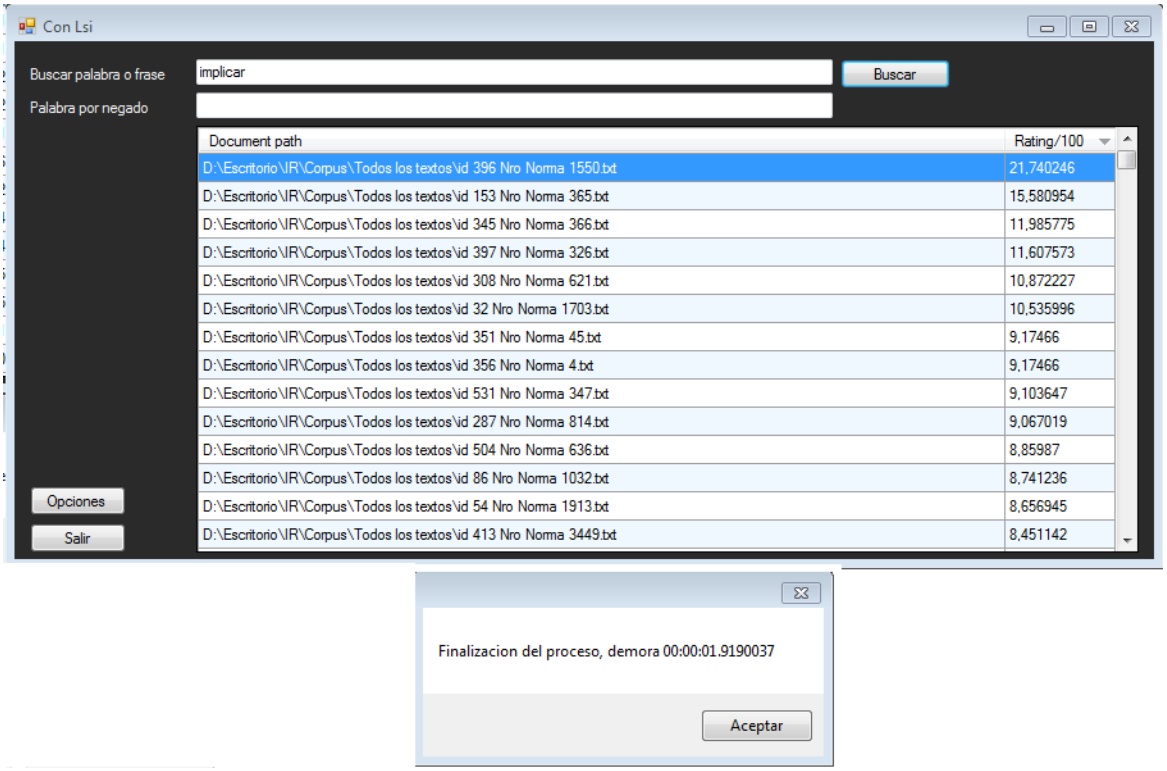
Se puede observar la diferencia de tiempos, utilizar lsi tardó más que sin lsi, también se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos, en cuanto a la distribución, el lsi lo hizo más proporcional que sin el lsi.

Prueba Nº12

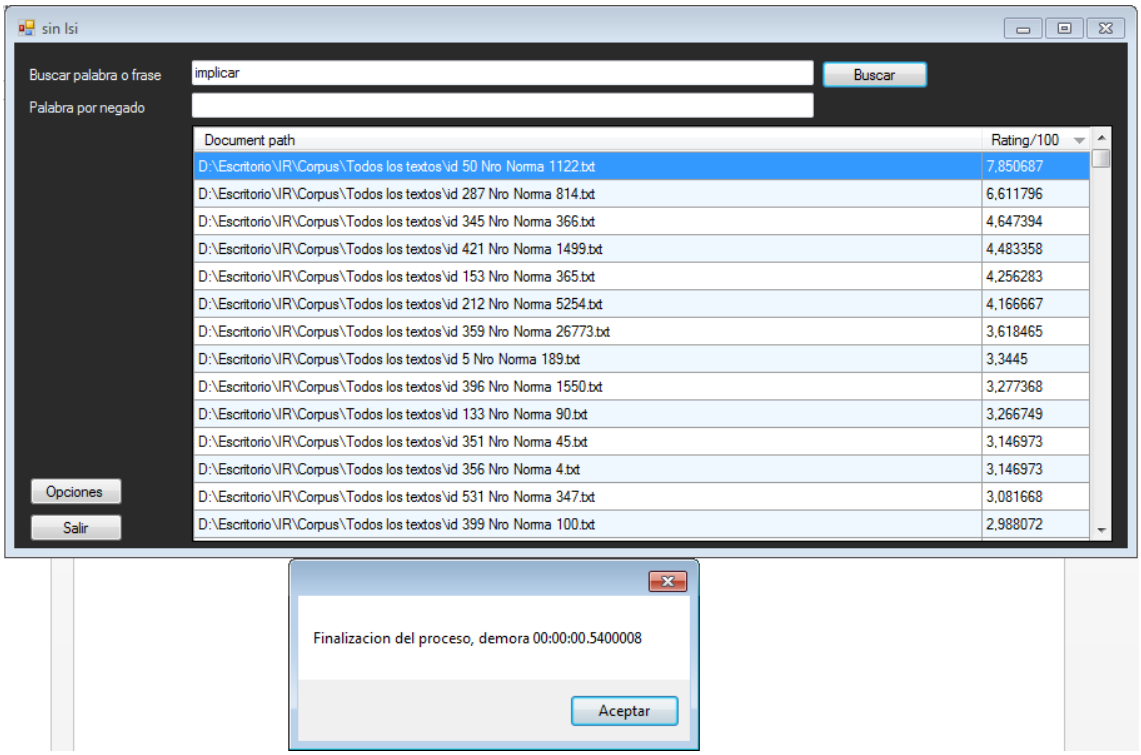
Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea una de las que en menos corpus se encuentre e igual cantidad de documentos en la que se encuentre como en la prueba anterior y cumpla con el umbral.

La palabra a buscar es: “implicar”

Con Lsi



Sin Lsi



Resultado de la prueba Nº12

Con Lsi:

Tiempo: 1,91 segundos.

Sin Lsi:

Tiempo 0,54 segundos.

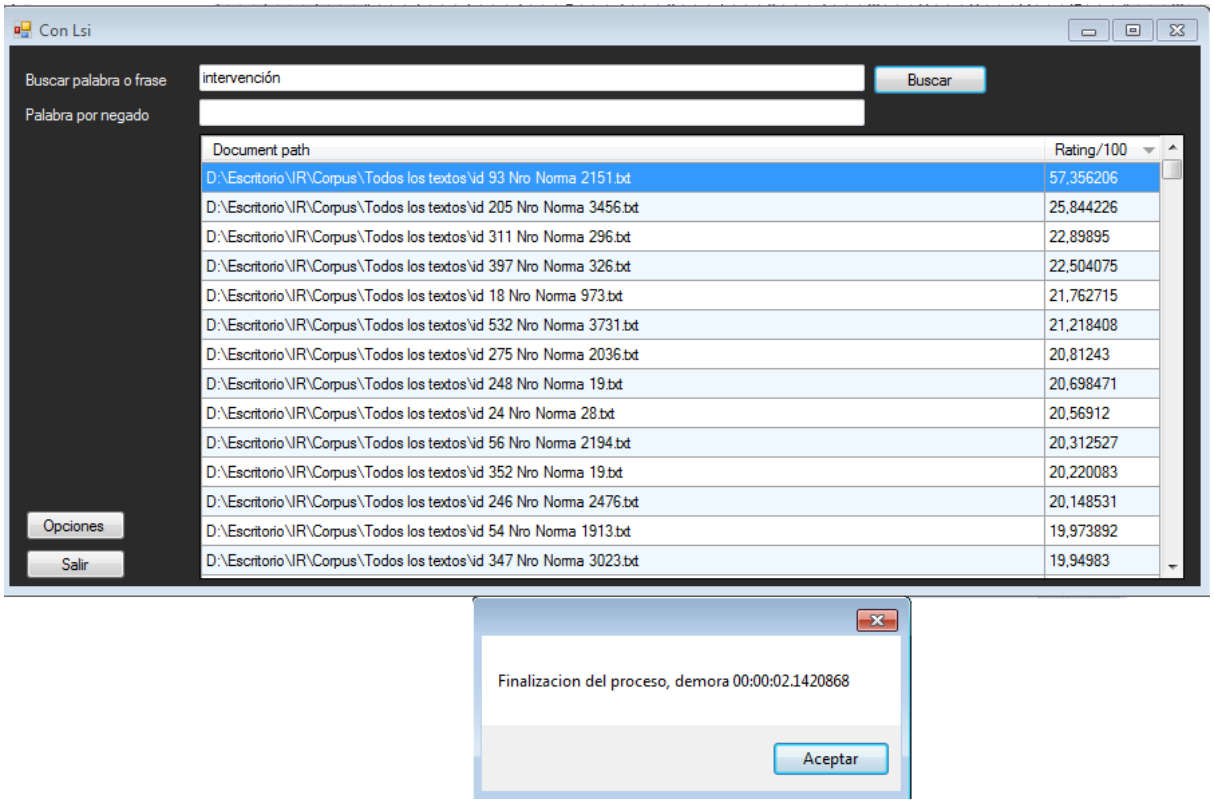
Se puede observar la diferencia de tiempos, utilizar lsi tardó más que sin lsi, también se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos, en cuanto a la distribución, ocurrio lo inverso en el caso anterior, el sin lsi fue más distribuido que con lsi.

Prueba Nº12

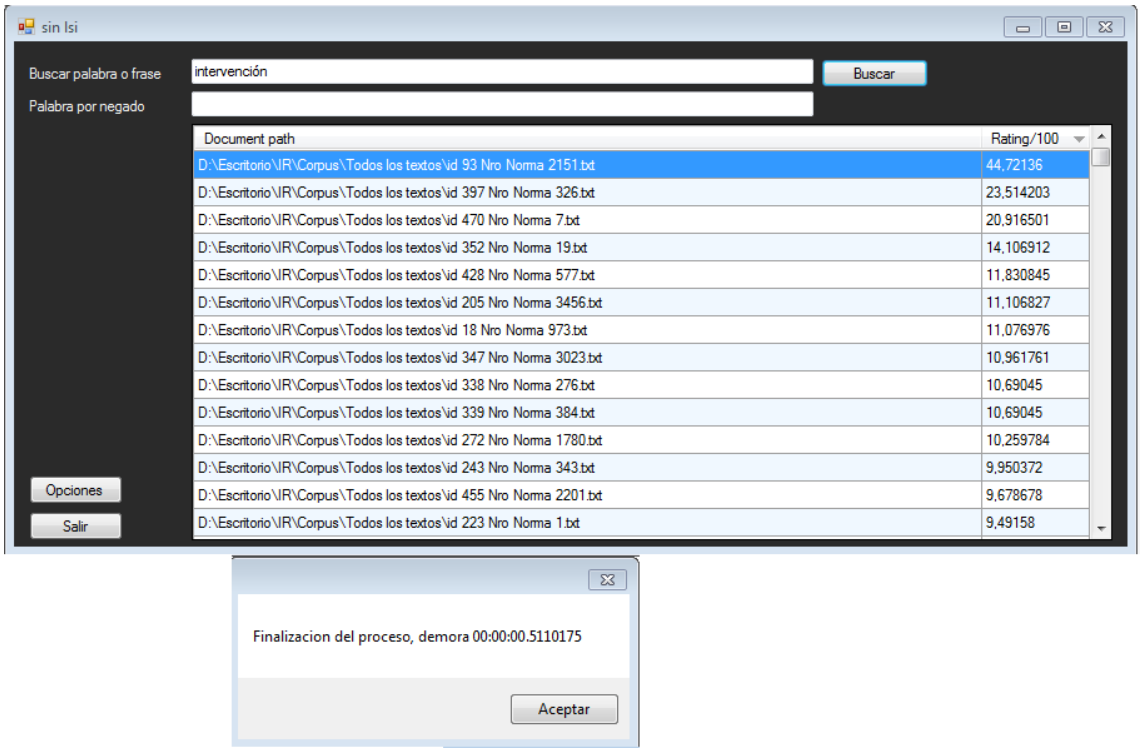
Realiza una búsqueda en el cual la palabra se encuentre en los corpus y esta sea una de las que en más corpus se y cumpla con el umbral.

La palabra a buscar es: “intervención”

Con Lsi



Sin Lsi



Resultado de la prueba Nº12

Con Lsi:

Tiempo: 2,14 segundos.

Sin Lsi:

Tiempo 0,51 segundos.

Se puede observar la diferencia de tiempos y un aumento en el tiempo con el lsi, mientras que el sin lsi mantuvo el tiempo. También se puede observar que hay diferencia en el ranqueo realizado por cada uno de los métodos, en cuanto a la distribución.