**INSTITUTO TECNOLÓGICO DE COSTA RICA I SEMESTRE 2020**

**INGENIERÍA EN COMPUTACIÓN**

**IC-8020 ELECTIVA TEXTO**

**TAREA PROGRAMADA 2**

**GUÍA DE DOCUMENTACIÓN**

1. Introducción

El problema consiste en indexar diferentes secciones o campos (específicamente <body>, <title>, <h?> y <a>) de de una colección de varios documentos html contenidos dentro de un archivo de texto para luego ser buscados de manera eficiente utilizando el poder del lenguaje de consultas que ofrece la librería Lucene de Apache. Además, se deben poder crear nuevos índices y actualizarlos con nuevos documentos. Finalmente, se debe poder navegar por los resultados mostrados en páginas de 20 ítems cada una.

1. Completar la siguiente tabla para describir el estado en que quedó la indización de la colección.

|  |  |  |
| --- | --- | --- |
| **Etapa** | **% de complet.** | **Comentario o aclaración** |
| **INDIZACIÓN** |  |  |
| **Campo "texto"** |  |  |
| Extrae adecuadamente del elemento <body> | 100 % |  |
| Separación en palabras (letras incluyendo eñe) | 100 % |  |
| Eliminación de stopwords | 100 % |  |
| Extracción de raíces (stemming) | 100 % |  |
| Eliminación de acentos, preserva eñe | 100 % |  |
| **Campo "ref"** |  |  |
| Extrae adecuadamente del elemento <a> | 100 % |  |
| Separación en palabras (letras incluyendo eñe) | 100 % |  |
| Conversión a minúsculas | 100 % |  |
| **NO** hay extracción de raíces (stemming) | 100 % |  |
| Eliminación de acentos, preserva eñe | 100 % |  |
| **Campo "encab"** |  |  |
| Extrae adecuadamente del elemento <h?> | 100 % |  |
| Separación en palabras (letras incluyendo eñe) | 100 % |  |
| Eliminación de stopwords | 100 % |  |
| Extracción de raíces (stemming) | 100 % |  |
| Eliminación de acentos, preserva eñe | 100 % |  |
| **Campo "titulo"** |  |  |
| Extrae adecuadamente del elemento <title> | 100 % |  |
| Separación en palabras (letras incluyendo eñe) | 100 % |  |
| Conversión a minúsculas | 100 % |  |
| **NO** hay extracción de raíces (stemming) | 100 % |  |
| Eliminación de acentos, preserva eñe | 100 % |  |
|  |  |  |
| **Indexado de la colecciones (wiki-[pg][12])** |  |  |
| ¿Cuánto tarda en indexar wiki-p1.txt? | 100 % | De 20 pruebas (10 en una computadora y 10 en otra) el promedio es |
| ¿Cuánto tarda en indexar wiki-p2.txt? | 100 % |  |
| ¿Cuánto tarda en actualizar índice wiki-p1.txt? | 100 % |  |
| ¿Cuánto tarda en indexar wiki-g1.txt? | 100 % |  |
| ¿Cuánto tarda en indexar wiki-g2.txt? | 100 % |  |
| ¿Cuánto tarda en actualizar índice wiki-g1.txt? | 100 % |  |
| **Consultas** |  |  |
| Permite usar lenguaje de consultas de Lucene | 100 % |  |

1. Completar la siguiente tabla para describir el resultado obtenido para algunas consultas de prueba.

| **Consulta** | **Colección** | **Resultado esperado** | **Obervaciones sobre el resultado obtenido** |
| --- | --- | --- | --- |
| magnoel | wiki-p1.txt | Éxito | Éxito, 1 resultado. |
| titulo:magnoel | wiki-p1.txt | Éxito | Éxito, 1 resultado. |
| encab:magnoel | wiki-p1.txt | Éxito | Éxito, 1 resultado. |
| cartago | wiki-p1.txt | Fallo | Fallo |
| cartago | wiki-g1.txt  wiki-g2.txt | Éxito |  |
| alejandro AND magno | wiki-p1.txt  wiki-p2.txt  wiki-g2.txt | Fallo | wiki-p1.txt = Éxito, 1 resultado (stemming) |
| alejandro AND magno | wiki-g1.txt | Éxito |  |
| ref:alejandro AND ref:magno | wiki-g1.txt | Éxito |  |
| amintas | wiki-p1.txt  wiki-p2.txt  wiki-g2.txt | Fallo |  |
| amintas | wiki-g1.txt | Éxito |  |
| tenistas de macedonia | wiki-p1.txt  wiki-p2.txt  wiki-g2.txt | Fallo |  |
| tenistas de macedonia | wiki-g1.txt | Éxito |  |
| ref:"tenistas de macedonia" | wiki-g1.txt | Éxito |  |
| turrialba | wiki-p1.txt  wiki-p2.txt  wiki-g2.txt  wiki-g1.txt | Fallo |  |

Éxito: encontró al menos un documento en la colección especificada.  
Fallo: no encontró ningún documento en la colección especificada.

1. Describa brevemente cómo usó las facilidades de Lucene y de otras herramientas para realizar las siguientes tareas:

* Indexar <body>.

Primero se configura el analizador de Lucene con el que se determina las características que definarán cada token que se ingrese en los campos.

Seguidamente utilizando un parser de archivos HTML (JSoup) se extrae el texto del elemento <body>. Luego de los 3 analizadores configurados, uno de ellos de ellos tiene varios filtros (TokenFilter) entre ellos un CharTokenizer que determina los caracteres que formaran tokens, para ello se sobreescribió la función isTokenChar() donde se valida que el caracter pertenezca a la expresión regular [A-Za-zÁÉÍÓÚÜáéíóúüÑñ].

Posteriormente, el resultado de ese tokenStream se pasa a un filtro que convierte todos los caracteres a minúsculas (LowerCaseFilter), el tokenStream resultante del se pasa a un filtro de stop words (StopFilter) alimentado previamente con la lista de stop words indicada para el proyecto, seguidamente el tokenStream sin stop words se pasa a un filtro que recibe un Snowball Stemmer en español para extraer las raíces de las palabras. Una vez hecho todo lo anterior ese texto se guarda para ser procesado una vez más removiendo los acentos.

Concluyendo, el texto producto de todo ese procesamiento es guardado en una campo indexable (IndexableField) con nombre de campo “texto”. Seguidamente, ese campo junto con los otros campos generados (<a>,<h?>,<title>, etc.) es agregado a un documento de Lucene, que finalmente es indexado por el IndexWriter de la librería Lucene.

* Extraer y mostrar el contenido de una página.

Para extraer el contenido HTML del documento y cargarlo en el navegador durante la lectura del archivo se fue guardando la línea de inicio de cada nuevo documento HTML y el número de líneas que componían al documento, estos valores se guardaban en un tipo abstracto de dato para ser luego accesados durante la indexación para generar 2 campos en el documento de Lucene donde se guardaran estas variables.

De esta forma, cuando se realiza una consulta y se retornan los documentos de Lucene relacionados se puede acceder a esos dos datos a través de los campos y con ellos volver a abrir el archivo texto y extrear únicamente esa porción específica, con la cual se escribe un nuevo archivo, con el título del HTML, guardado en una ubicación temporal desde la cual se carga en el navegador.

1. Comentarios finales (estado del programa)

En nuestra opinión el estado final del programa es muy satisfactorio, no solamente se logró implementar correctamente las funcionalidades de la librería Lucene para indexar y consultar los documentos sino que también logramos hacerlo con tiempos de indexación razonables y utilizando una interfaz de usuario amigable. Durante el desarrollo hubo varias limitaciones por la falta de conocimiento de la librería, pero con la experimentación y el error (y leer mucha documentación) se llegó a comprender cómo funciona.