**Trabajo Práctico Único: Creación de un Vocabulario**

**•  Enunciado:**

A los efectos de gestionar la información producida por el trabajo de un grupo de investigación en lingüística, se necesita un programa que sea capaz de procesar un conjunto de *archivos de texto* y construir un *vocabulario* *con todas las palabras diferentes* descubiertas en los archivos, de forma de determinar además la *frecuencia de aparición* de cada palabra en el conjunto de archivos.

Se pide desarrollar un programa que sea capaz de procesar de a uno los archivos indicados (puede descargarlos desde [aquí](http://uv.frc.utn.edu.ar/pluginfile.php/81671/mod_assign/intro/Libros%20%5BTP%20Unico%20TSB%202014%5D.zip" \t "_blank)), obteniendo de cada uno las distintas palabras detectadas. Notar aquí que para obtener las palabras se deberá analizar (o "parsear") el documento, limpiar el texto de los signos de puntuación, palabras numéricas o alfanuméricas, dígitos y todo otro símbolo que no forme parte de una palabra normalmente entendida; y una vez realizado este trabajo (o a medida que se va realizando) se procese cada palabra agregándola al vocabulario y contando su frecuencia de aparición.

Por otro lado, además de la frecuencia de aparición de cada palabra, se pide también informar el/los documentos del conjunto de documentos procesados donde la palabra fue encontrada (es decir que cada palabra debe mantener información acerca de en qué documentos apareció).

**•  Requerimientos:**

El programa a desarrollar debe cumplir con los siguientes requerimientos:

**a.)**Contar con interfaz de usuario enteramente basada en ventanas, cuadros de diálogo y componentes visuales.

**b.)**Permitir elegir el/los archivos a procesar: utilizando algún cuadro de diálogo que permita navegar en el sistema de archivos del sistema operativo, se debe poder elegir el/los archivos a procesar y a cada uno de los archivos elegidos enviarlos a la *cola de tareas* del sistema para ser procesados. Esto es:  el sistema los debe procesar de a uno pero debemos poder cargar el conjunto de archivos en una cola de forma que luego el sistema los procese. Una acción adicional interesante aquí es mostrar algún feedback del avance del proceso (ver para eso el uso del control *SwingWorker* con los profesores del práctico).

**c.)**Prestar especial atención en cuanto a factores de eficiencia:  pensar en un diseño eficiente para no caer en casos en los que el proceso de los archivos incurra en una cantidad exagerada de tiempo y vuelva nuestro programa una alternativa poco viable de ser utilizada en la realidad.

**d.)**Presentar una pantalla que liste todas las palabras del vocabulario (que estarán almacenadas en una base de datos), en una grilla mostrando la palabra, la frecuencia total de aparición y la cantidad de documentos de la muestra donde fue observada.

**e.)**La pantalla antes mencionada debe soportar la funcionalidad de filtrar el vocabulario en base a una raíz de palabra escrita en un cuadro de texto. Al decir raíz indicamos que la única posibilidad de filtro es mediante la búsqueda de los primeros caracteres de la palabra y no una sub cadena contenida en cualquier parte de la palabra.

**f.)      Punto bonus**: El extra a implementar en la interfaz será la posibilidad de ir filtrando la lista a medida que se van agregando letras en el cuadro de texto de filtro del vocabulario, es decir al escribir la ‘B’ filtre todas las palabras que comiencen con ‘B’/’b’ luego si agregamos la ‘a’ filtre todas las que comiencen con ‘Ba’/’ba’ si luego agregamos ‘r’ las que comiencen con ‘Bar’/’bar’ y así sucesivamente.

**g.)**El vocabulario y la relación entre las palabras y los documentos en los que se encontró la misma deberán ser almacenados en una base de datos.  **Nota:**utilizar aquí un *dbms* que no requiera instalación. De acuerdo a los vistos en clase puede ser *Java DB* en modo embedded o *SQLite* (por ejemplo). No se aceptaran trabajos presentados donde la base de datos deba ser instalada en un motor independiente de la aplicación.

**h.)**A losefectos de corregir el trabajo, debe ser posible agregar documentos nuevos a una base de datos donde ya hay un vocabulario construido. Es decir: al presentar el trabajo debe incluirse con el conjunto de archivos brindados con el presente enunciado ya procesados,  pero el programa debe soportar que se procese uno o varios archivos más en el momento de la corrección y se actualice el vocabulario con las nuevas frecuencias o nuevas palabras y sus frecuencias de acuerdo a el/los nuevos archivos procesados.

**•  Entrega:**

El trabajo es de *realización grupal*. Debe ser realizado en grupos de no menos de 2(dos) personas y no más de 3(tres) como máximo.

**Cada integrante** del grupo debe subir al aula virtual su copia del trabajo en un archivo comprimido conteniendo el proyecto del práctico y la base de datos. Obviamente, todos los integrantes de un mismo grupo deberán subir la misma copia… pero TODOS deben hacerlo.

El archivo comprimido mencionado debe ser nombrado incluyendo los apellidos de los integrantes del grupo separados por guiones por ejemplo si el práctico lo presentaran los jtp de la cátedra debería llamarse *Serrano-Steffolani-Teicher.zip* (o .rar o 7z o cualquier tipo de compresión que se pueda descomprimir con 7Zip).

**•  Entrega:**

Además de la evidente evaluación de resultados correctos y funcionamiento de acuerdo a lo solicitado tendrán especial atención en la calificación los siguientes puntos:

-   La eficiencia del algoritmo de proceso de los archivos.

-   El correcto e independiente uso de la base de datos para la persistencia, es decir que no usemos la base de datos por toda la aplicación.

-   El diseño de la interfaz de usuario, uso de diálogo para la selección de los archivos a procesar, y uso de grilla para mostrar cada palabra del vocabulario.

-   La eficiencia del proceso de filtrado en la interfaz de usuario.

-   La nota máxima alcanzada al presentar el trabajo *sin* la implementación del punto bonus será 8(ocho) y *con* la implementación del punto bonus la nota máxima  será de 10(diez).