PRÁCTICA 1

Recuperación de la información

Juan Carlos González Quesada y Pedro Jiménez Alférez

Índice

¿Qué tenemos que realizar?

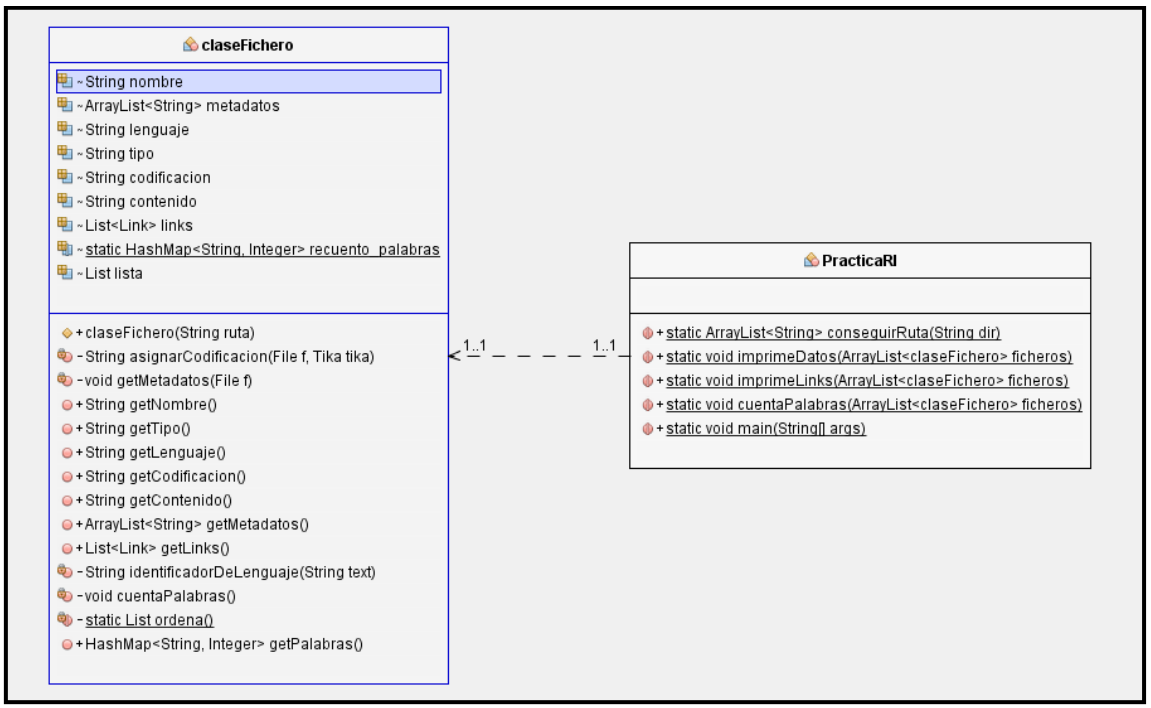
Nos hemos encargado de un programa que, dado unos ficheros, realiza las siguientes operaciones:

- Con la opción "-d" obtenemos la información de cada fichero.
- Con la opción "-l" obtenemos todos los enlaces de los ficheros.
- Con la opción"-t" obtenemos un ".csv" con el recuento de palabras.

Como se hizo:

Hemos pensado que la mejor forma de realizar este proyecto era individualizando cada fichero. De esta manera podemos tratar la información de cada fichero (cada metadato) de forma individual. Para ello hemos creado dos clases. Una será la clase Fichero, en donde trataremos las funciones relacionadas con los ficheros y sus metadatos, y la otra será la clase principal, donde se harán las funciones para cada una de las opciones que nos piden.

- Diagrama de clase



- Clase claseFichero:

Tenemos el constructor de la clase, varios get que nos ayudan para obtener más rápidamente el valor de los atributos, entre ellos se encuentran el del lenguaje, el de los link y metadatos y codificación, que nos hemos ayudado de clases ya creadas en Java.

Para obtener la lista de palabras, hemos utilizado la finción "split", para separar cada vez que nos encontremos un espacio. Hemos eliminado todos los caracteres que no sean una letra. (Hemos añadido palabras con tildes o signos de puntuación del alfabeto aleman o francés) y los hemos almacenados en nuestro HashMap.

La función ordena(), nos ordena de forma dereciente el HAshMap. Para realizar esta función nos hemos ayudado del foro StackOverFloat.

- Clase PracticaRI:

Para el desarrollo de esta clase, hemos tenido varios problemas, que hemos solucionado como explicaremos más adelante. La función de "conseguirRutas()", obtiene los paths de cada archivo.

Para la primera opción, utilizaremos la función "imprimirDatos", que recorre la lista de ficheros y de cada uno con los gets de la clase anterior, muestra en forma de tabla los metadatos.

Para la segunda opción, apoyándonos en la clase Link, mostramos todos los links obtenidos de cada fichero.

Para la última opción:

```
public static void cuentaPalabras(ArrayList<claseFichero> ficheros) throws FileNotFoundException, UnsupportedEncodingException {
      File archivoAux = new File("CSV");
      String ruta = archivoAux.getAbsolutePath();
      ruta=ruta+"\\";
        for(int i=0; i<ficheros.size(); i++) {
                PrintWriter writer = new PrintWriter(ruta+ficheros.get(i).getNombre()+ ".csv", "UTF-8");
                System.out.println(ruta+ficheros.get(i).getNombre()+ ".csv");
                Iterator it = new ReverseListIterator(ficheros.get(i).lista);
                writer.println("Text/Size");
                while(it.hasNext()) {
                      String valor=it.next().toString();//Al usar la función de una librería. Nos añade las palabras separadas por un =
                      String aux=valor.replace('=', ';');
                      if(aux.charAt(0)!=';'){//Eliminamos el espacio, que lo cuenta como caracter. Ya que se encuentra dentro del Código ASCII
                          writer.println(aux);
                      }
                }
                writer.close();
        }
}
```

Creamos una carpeta llamada CSV, donde se almacenarán todos los archivos ".csv". El primer problema, era que aunque obteníamos la ruta de esa carpeta, luego no guardaba los archivos dentro, por tanto, tuvimos que añadir una barra (que se indica en Windows como \\) para que no hubiese ese inconveniente.

Luego dentro del for y para cada archivo, creábamos el archivo. El segundo problema es que nos separaba las palabras de los números (Juan = 563) por un igual. Así que antes de añadirlo al csv, reemplazamos el = por un ;. El último problema, es que almacena también los caracteres vacíos, por tanto, si la palabra empieza por un ;, no la añadimos al fichero.

Luego, el propio main será un bucle while que sólo parará cuando se introduzca por teclado alguna tecla diferente a las opciones pedidas por el programa.

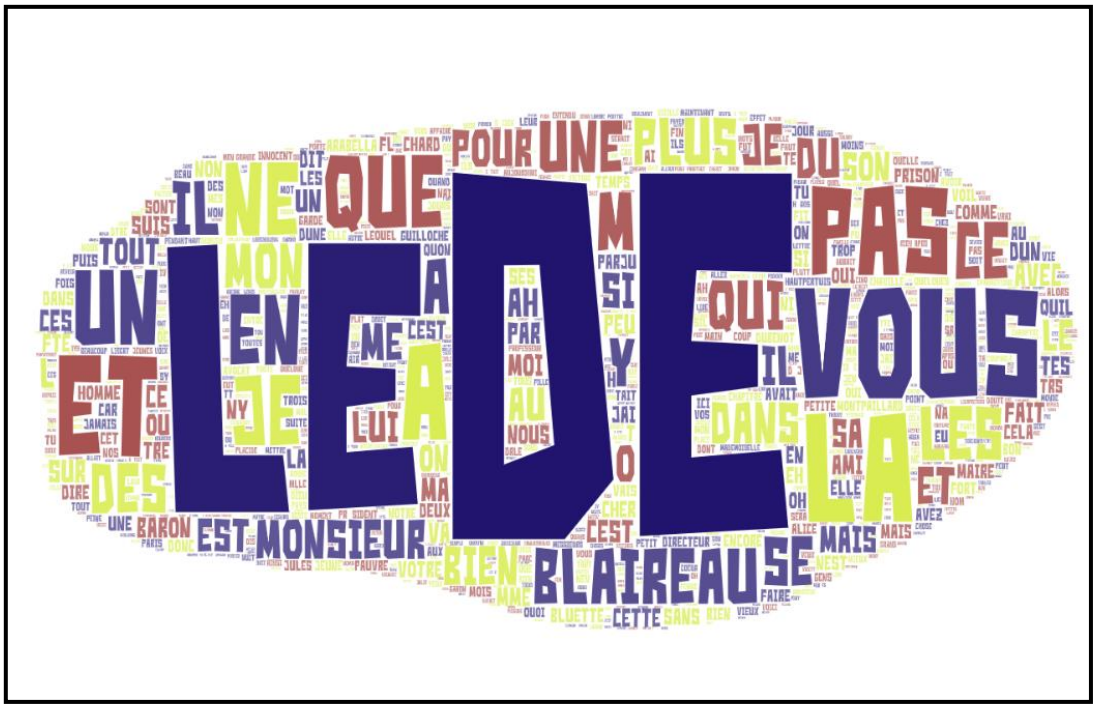
¿Cómo se ejecuta el programa?

Está hecho para ser ejecutado y compilado en NETBEANS y WINDOWS. En el fichero que se entrega, está el proyecto que se puede añadir a Netbeans y la memoria. En

la carpeta “documentos” se encuentran sobre 50 archivos con los que hemos probado. Varios pdfs, varios libros en distintos idiomas, más de tres formatos distintos y de varios tipos. Una vez ejecutada la opción “-t”, dentro de la carpeta “CSV” se tendrán todos los recuentos de las palabras.

Nube de palabras:

Nosotros la hemos realizado con el libro: *Allais, Alphonse - L'Affaire Blaireau*, que está en francés. Y este es el resultado:



Text/Size
de;1261
le;699
vous;598
à;548
la;544
pas;440
que;439
et;370
un;352
ne;328
Blaireau;327
en;302
je;296
les;280
une;263
monsieur;212
plus;205
qui;204
a;203
des;199
dans;194
ce;192
il;184
bien;173
pour;171
mon;167
se;164
Je;158
du;152

Trabajo en grupo

En esta práctica, en un principio estuvimos hablando sobre cuál era el mejor entorno posible para la realización de la práctica. Decidimos desarrollarlo desde Netbeans debido a que nos ha sido muy sencillo trabajar con distintas clases y funciones además de poder incorporar librerías de manera inmediata. Hemos trabajado cada uno desde su casa y usando Google Meet para poder trabajar y poder comunicarnos fluidamente.

El método de trabajo ha sido sencillo: primero pensábamos en cómo podíamos implementar nuestras ideas en el proyecto. A continuación, buscábamos la información que nos fuese necesaria para poder implementarla ya fuese desde el PDF de la práctica o desde Internet. Por último, hemos creado el código para la práctica y hemos corregido los errores que nos ha producido.

BIBLIOGRAFÍA

https://stackoverflow.com/questions/3227524/how-to-detect-language-of-user-entered-text

https://es.stackoverflow.com/questions/45846/c%C3%B3mo-separar-un-string-en-java-c%C3%B3mo-utilizar-split

https://docs.oracle.com/javaee/7/api/javax/ws/rs/core/Link.html

https://docs.oracle.com/javase/8/docs/api/java/lang/Class.html

https://stackoverflow.com/questions/17821895/getting-started-with-apache-tika