UNIVERSIDAD DE GRANADA

PRÁCTICA 1: PREPROCESADO DE DOCUMENTOS

Parser de documentos con TIKA

Anne Serrano Andrades
Pablo Huertas Arroyo

1. INTRODUCCIÓN

En esta práctica se nos pide extraer información de unos documentos a partir de un directorio raíz pasado como parámetro en la entrada del programa, para poder procesar dichos documentos para recuperar información o analizarlos.
Una vez organizados los documentos en un directorio procedemos a aclarar las opciones.

2. MAIN

Lo primero que nos encontramos en nuestro programa es la comprobación del número de parámetros, si hay menos de dos argumentos saltará un mensaje de error y otro mensaje aclarando su uso.
Para poder trabajar con los documentos que hay dentro del directorio creamos un objeto de tipo File con el nombre del directorio y guardamos todos sus archivos en un array de String, donde quedarán guardados los nombres de cada uno de los archivos.
Una vez creada la instancia de Tika, aumentamos el tamaño limitado de caracteres.
Creamos un objeto de tipo Metadata para guardar los metadatos de los documentos.

```
37  public class PracticaTika{
38      public static String identifyLanguage(String texto){
39          LanguageDetector identifier = new OptimaizeLangDetector().loadModels();
40          LanguageResult idioma = identifier.detect(texto);
41
42          return idioma.getLanguage();
43      }
44      public static void main (String[] args)throws Exception {
45
46
47          if(args.length < 2){
48              System.out.println("Número de parámetros incorrectos");
49              System.out.println("USO -> <[-d/-l/-t]> <directorio>");
50              return;
51          }
52
53
54
55          //DIRECTORIO
56          File directorio = new File(args[1]);
57          String[] ficheros = directorio.list();
58          System.out.println("Ficheros en el directorio actual: " + ficheros.length);
59          System.out.println("Directorio actual: " + directorio.getAbsolutePath());
60          System.out.println("Nombre de los ficheros del directorio actual:" + Arrays.toString(ficheros));
61          System.out.println("args[0] = " + args[0]);
62
63          Tika tika = new Tika();
64
65          //Para poder leer más caracteres
66          tika.setMaxStringLength(1000000000);
67          Metadata metadata = new Metadata();
68
69          //OPCIONES
70          String opcion=args[0];
71          if("-d".equals(opcion)){
72              crearTabla(ficheros, tika, metadata, args[1]);
73          }else if("-l".equals(opcion)){
74              extraerEnlaces(ficheros, tika,metadata, args[1]);
75          }else if ("-t".equals(opcion)){
76              crearCSV(ficheros, tika, metadata,args[1]);
77          }else{
78              System.out.println("OPCION NO VÁLIDA, USO -> <[-d/-l/-t]> ");
79          }
80
81      }
```

2.1. OPCIONES

2.1.1. -d

Se pide realizar una tabla de forma automática con el nombre del fichero, su tipo, codificación e idioma.
Aquí hemos usado la función auxiliar crearTabla, esta función abre el directorio y parsea todos sus archivos y sacamos cada uno de los datos necesarios:

- NOMBRE-> el nombre se extrae directamente del array.
- TIPO -> el uso de la función detect de Tika nos devuelve el tipo de contenido del archivo.
- IDIOMA-> para el idioma usamos una función auxiliar que ya nos proporcionó el profesor en un ejemplo.
- CODIFICACIÓN -> dentro del objeto metadata existe una clave específica llamada "Content-Encoding" donde encontraremos el tipo de codificación del archivo.

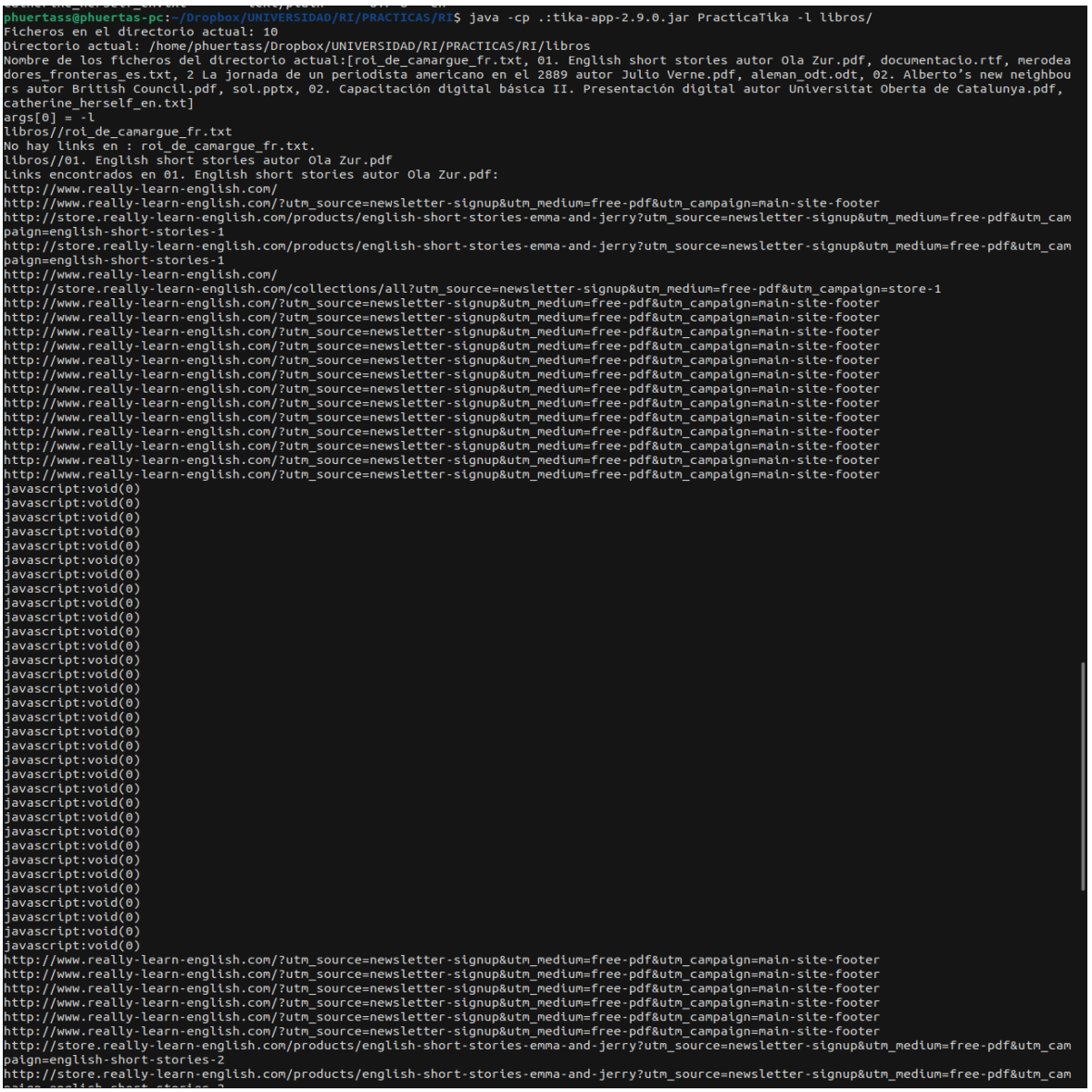
```java
//OPCIÓN -D CREAR TABLA
public static void crearTabla(String[] ficheros, Tika tika, Metadata metadata, String args) throws Exception{
    if (ficheros== null || ficheros.length==0){
        System.out.println("El directorio está vaciío");
    }
        //Encabezados
        System.out.println("Nombre\tTipo\tCodificación\tIdioma");
        for(String nombre : ficheros){

            String nameFile = args+ "/"+ nombre;
            System.out.println(nameFile);
            File archivo = new File(nameFile);

            if (!archivo.exists() || !archivo.isFile()) {

                System.out.println("El archivo " + nombre + " no existe o no es un archivo válido.");
                continue; // Skip to the next file
            }
                tika.parse(archivo,metadata);

                //TIPO
                String tipo = tika.detect(archivo);

                //IDIOMA
                String contenido = tika.parseToString(archivo);
                String idioma= identifyLanguage(contenido);

                //CODIFICACIÓN
                String codificacion = metadata.get("Content-Encoding");

                //IMPRIMIR LOS DATOS
                System.out.println(nombre + "\t"+tipo + "\t"+ codificacion+"\t"+idioma);
    }

}
```

```
phuertass@phuertas-pc:~/Dropbox/UNIVERSIDAD/RI/PRACTICAS/RI$ java -cp .:tika-app-2.9.0.jar PracticaTika -d libros/
Ficheros en el directorio actual: 10
Directorio actual: /home/phuertass/Dropbox/UNIVERSIDAD/RI/PRACTICAS/RI/libros
Nombre de los ficheros del directorio actual:[roi_de_camargue_fr.txt, 01. English short stories autor Ola Zur.pdf, documentacio.rtf, merodeadores_fronteras_es.txt, 2 La jornada de un periodista americano en el 2889 autor Julio Verne.pdf, aleman_odt.odt, 02. Alberto's new neighbours autor British Council.pdf, sol.pptx, 02. Capacitación digital básica II. Presentación digital autor Universitat Oberta de Catalunya.pdf, catherine_herself_en.txt]
args[0] = -d
Nombre  Tipo    Codificación    Idioma
libros//roi_de_camargue_fr.txt
roi_de_camargue_fr.txt  text/plain      UTF-8   fr
libros//01. English short stories autor Ola Zur.pdf
01. English short stories autor Ola Zur.pdf     application/pdf UTF-8   en
libros//documentacio.rtf
documentacio.rtf        application/rtf UTF-8   es
libros//merodeadores_fronteras_es.txt
merodeadores_fronteras_es.txt   text/plain      UTF-8   es
libros//2 La jornada de un periodista americano en el 2889 autor Julio Verne.pdf
WARN  [Apache Tika: roi_de_camargue_fr.txt] 17:25:17,873 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif for TimesNewRomanPSMT
WARN  [Apache Tika: roi_de_camargue_fr.txt] 17:25:17,876 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif-Bold for TimesNewRomanPS-BoldMT
WARN  [main] 17:25:17,894 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif for TimesNewRomanPSMT
WARN  [main] 17:25:17,894 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif-Bold for TimesNewRomanPS-BoldMT
WARN  [main] 17:25:18,038 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSans-Bold for Verdana-Bold
2 La jornada de un periodista americano en el 2889 autor Julio Verne.pdf        application/pdf UTF-8   es
libros//aleman_odt.odt
aleman_odt.odt  application/vnd.oasis.opendocument.text UTF-8   de
libros//02. Alberto's new neighbours autor British Council.pdf
02. Alberto's new neighbours autor British Council.pdf  application/pdf UTF-8   en
libros//sol.pptx
INFO  [Apache Tika: roi_de_camargue_fr.txt] 17:25:18,688 org.apache.tika.parser.ocr.TesseractOCRParser Tesseract is installed and is being invoked. This can add greatly to processing time.  If you do not want tesseract to be applied to your files see: https://cwiki.apache.org/confluence/display/TIKA/TikaOCR#TikaOCR-disable-ocr
sol.pptx        application/vnd.openxmlformats-officedocument.presentationml.presentation       UTF-8   es
libros//02. Capacitación digital básica II. Presentación digital autor Universitat Oberta de Catalunya.pdf
02. Capacitación digital básica II. Presentación digital autor Universitat Oberta de Catalunya.pdf      application/pdf UTF-8   es
libros//catherine_herself_en.txt
catherine_herself_en.txt        text/plain      UTF-8   en
phuertass@phuertas-pc:~/Dropbox/UNIVERSIDAD/RI/PRACTICAS/RI$
```

2.1.2. -l

Con la opción -l obtendremos todos los enlaces que se pueden extraer de un documento.
Una vez obtenido los archivos que contiene el directorio creamos las distintas estructuras necesarias para almacenar y analizar los enlaces. A través de flujo de entrada paramos el archivo y obtenemos los enlaces. Guardamos todos esos enlaces en una lista y los imprimimos por pantalla, si no encuentra ningún enlace mostrará un mensaje indicándolo.

```java
//OPCION -L OBTENER ENLACES
public static void extraerEnlaces(String[] ficheros, Tika tika, Metadata metadata, String args)throws Exception{

    for(String nombre : ficheros){
        String nameFile = args+ "/"+ nombre;

            System.out.println(nameFile);
            File archivo = new File(nameFile);

            LinkContentHandler linkHandler = new LinkContentHandler();

            ParseContext parseContext = new ParseContext();
            AutoDetectParser parser = new AutoDetectParser();

            InputStream input = new FileInputStream(archivo);
            parser.parse(input,linkHandler,metadata,parseContext);

            List<Link> enlaces = linkHandler.getLinks();
            if(enlaces.isEmpty()){
                System.out.println("No hay links en : " +archivo.getName()+".");
            }else {
                System.out.println("Links encontrados en " +archivo.getName()+":");
                for(Link enlace : enlaces){

                    System.out.println(enlace.getUri());
            }

            }
    }
}
```



2.1.3. -t

Aquí se pide que para cada documento generamos un fichero con formato CSV que contenga la ocurrencia de cada uno de los términos en el mismo en orden decreciente de frecuencia, con el siguiente formato: término y frecuencia (número de ocurrencias del término en el documento) separados por ';'.

Hemos creado una clase nueva "FrecuenciaPalabras.java" para ver y ordenar el número de ocurrencias de una palabra, es básicamente una clase auxiliar

```java
import java.util.*;

public class FrecuenciaPalabras{
    public int ocurrencias;
    public String palabra;

    public FrecuenciaPalabras(String p, int o){
        palabra=p;
        ocurrencias=o;
    }
}

class OrdenarPorOcurrencias implements Comparator<FrecuenciaPalabras>{
    public int compare(FrecuenciaPalabras a, FrecuenciaPalabras b) { return b.ocurrencias - a.ocurrencias; }
}
```

Para el código de la función crearCSV vemos si está creado el directorio CSV y si no se crea.
Parseamos los ficheros y esta vez usamos también toLowerCase() ya que el guión expresa que debemos pasar las palabras a minúsculas. Luego creamos un array donde se separan las palabras según los diferentes caracteres que vaya encontrando. Se crea un array de la clase FrecuenciaPalabras donde iremos almacenando el número de ocurrencias de la palabra. Vamos añadiendo tanto la palabra como su número de ocurrencias. Lo ordenamos de forma decreciente (gracias a OrdenarPorOcurrencias). Para terminar, se generan los .csv para cada documento.

```java
//OPCION -T CREAR CSV
public static void crearCSV(String[] nombresArchivos, Tika tikaInstancia, Metadata metadatos, String directorio)throws Exception{
    //CREAR DIRECTORIO CSV
    File carpetaCSV = new File("CSV");
    if(!carpetaCSV.exists()){
        carpetaCSV.mkdir();
    }

    for(String nombreArchivo: nombresArchivos){
        String rutaArchivo = directorio + "/" + nombreArchivo;

        System.out.println(rutaArchivo);
        File archivo = new File(rutaArchivo);
        tikaInstancia.parse(archivo, metadatos);
        String contenidoTexto = tikaInstancia.parseToString(archivo).toLowerCase();

        //Separar las palabras
        String[] palabras = contenidoTexto.split("\\s+|[\\.|\\,|\\;|\\:|\\¿|\\?|\\¡|\\!|\\(|\\)|\\{|\\}|\\*|\\$|\\^|\\=|\\'|\\'|\\-|\\_|\\-|\\<|\\>|\\#|\\`|\\~|\\€]");

        //Creamos un ArrayList de frecuencia de palabras
        ArrayList<FrecuenciaPalabras> frecuenciaPalabrasLista = new ArrayList<FrecuenciaPalabras>();

        //Se rellena el ArrayList
        for(String palabra : palabras){
            int contador = 0; //inicializamos un contador a 0
            Boolean esta = false;

            while(contador < frecuenciaPalabrasLista.size() && !esta){ //mientras que el contador no sea mayor que el num de palabras y no se encuentre
                if(frecuenciaPalabrasLista.get(contador).palabra.compareTo(palabra)==0){ //si la encontramos por primera vez
                    esta = true;
                }else{ //si no, incrementamos el num de veces que se encuentra
                    contador++;
                }
            }

            if(esta){
                frecuenciaPalabrasLista.get(contador).ocurrencias++; //aumentamos el num de veces que aparece
            }else{
                frecuenciaPalabrasLista.add(new FrecuenciaPalabras(palabra, 1)); //añadimos 1 para que no desaparezca
            }
        }
```

```java
            //Ordenamos en orden decreciente
            Collections.sort(frecuenciaPalabrasLista, new OrdenarPorOcurrencias());

            //Quitamos las palabras que no sirven
            for(int i=0; i<frecuenciaPalabrasLista.size(); i++){
                if(frecuenciaPalabrasLista.get(i).palabra.compareTo("")==0){
                    frecuenciaPalabrasLista.remove(i);
                }
            }

            //Generamos los CSV para cada documento
            String nombreFicheroCSV = "./CSV/" + archivo.getName() + ".csv";
            System.out.println("Directorio CSV creado para el documento " + archivo.getName());

            // Añadimos la información al fichero CSV
            String contenidoCSV = "";

            for(int i=0; i<frecuenciaPalabrasLista.size(); i++){
                contenidoCSV += frecuenciaPalabrasLista.get(i).palabra + ";" + frecuenciaPalabrasLista.get(i).ocurrencias + "\n";
            }

            //Usamos PrintWriter ya que nos permite imprimir representaciones formateadas de una salida de stream de texto
            PrintWriter escritor = new PrintWriter(nombreFicheroCSV);
            escritor.print(contenidoCSV);
            escritor.close();
        }
    }

}
```

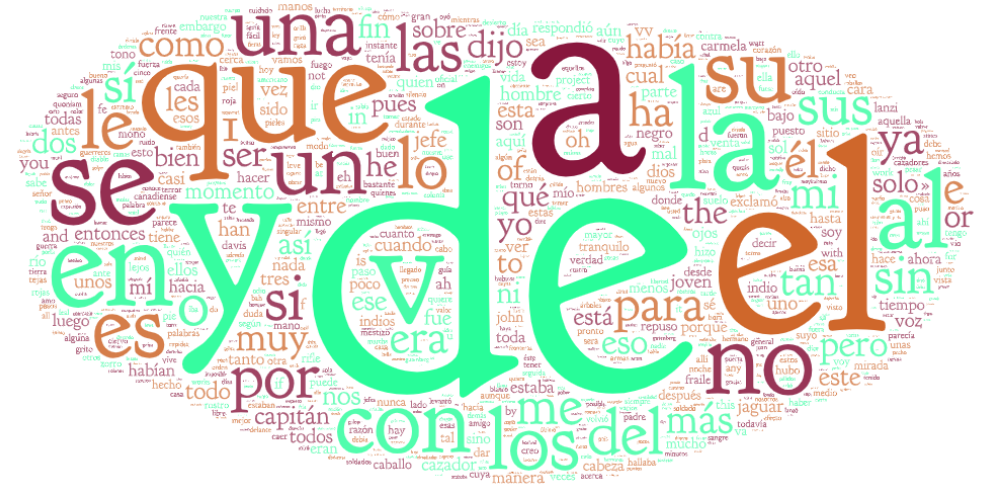
```
phuertass@phuertas-pc:~/Dropbox/UNIVERSIDAD/RI/PRACTICAS/RI$ java -cp .:tika-app-2.9.0.jar PracticaTika -t libros/
Ficheros en el directorio actual: 10
Directorio actual: /home/phuertass/Dropbox/UNIVERSIDAD/RI/PRACTICAS/RI/libros
Nombre de los ficheros del directorio actual:[roi_de_camargue_fr.txt, 01. English short stories autor Ola Zur.pdf, documentacio.rtf, merodea
dores_fronteras_es.txt, 2 La jornada de un periodista americano en el 2889 autor Julio Verne.pdf, aleman_odt.odt, 02. Alberto's new neighbou
rs autor British Council.pdf, sol.pptx, 02. Capacitación digital básica II. Presentación digital autor Universitat Oberta de Catalunya.pdf,
catherine_herself_en.txt]
args[0] = -t
libros//roi_de_camargue_fr.txt
Directorio CSV creado para el documento roi_de_camargue_fr.txt
libros//01. English short stories autor Ola Zur.pdf
Directorio CSV creado para el documento 01. English short stories autor Ola Zur.pdf
libros//documentacio.rtf
Directorio CSV creado para el documento documentacio.rtf
libros//merodeadores_fronteras_es.txt
Directorio CSV creado para el documento merodeadores_fronteras_es.txt
libros//2 La jornada de un periodista americano en el 2889 autor Julio Verne.pdf
WARN  [Apache Tika: roi_de_camargue_fr.txt] 17:26:27,796 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif f
or TimesNewRomanPSMT
WARN  [Apache Tika: roi_de_camargue_fr.txt] 17:26:27,799 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif-B
old for TimesNewRomanPS-BoldMT
WARN  [main] 17:26:27,813 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif for TimesNewRomanPSMT
WARN  [main] 17:26:27,814 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSerif-Bold for TimesNewRomanPS-BoldMT
WARN  [main] 17:26:27,913 org.apache.pdfbox.pdmodel.font.PDTrueTypeFont Using fallback font LiberationSans-Bold for Verdana-Bold
Directorio CSV creado para el documento 2 La jornada de un periodista americano en el 2889 autor Julio Verne.pdf
libros//aleman_odt.odt
Directorio CSV creado para el documento aleman_odt.odt
libros//02. Alberto's new neighbours autor British Council.pdf
Directorio CSV creado para el documento 02. Alberto's new neighbours autor British Council.pdf
libros//sol.pptx
INFO  [Apache Tika: roi_de_camargue_fr.txt] 17:26:28,441 org.apache.tika.parser.ocr.TesseractOCRParser Tesseract is installed and is being i
nvoked. This can add greatly to processing time.  If you do not want tesseract to be applied to your files see: https://cwiki.apache.org/con
fluence/display/TIKA/TikaOCR#TikaOCR-disable-ocr
Directorio CSV creado para el documento sol.pptx
libros//02. Capacitación digital básica II. Presentación digital autor Universitat Oberta de Catalunya.pdf
Directorio CSV creado para el documento 02. Capacitación digital básica II. Presentación digital autor Universitat Oberta de Catalunya.pdf
libros//catherine_herself_en.txt
Directorio CSV creado para el documento catherine_herself_en.txt
phuertass@phuertas-pc:~/Dropbox/UNIVERSIDAD/RI/PRACTICAS/RI$
```

3. GRÁFICAS

- TEXTO EN INGLÉS

- TEXTO EN FRANCÉS

- TEXTO EN ESPAÑOL



4. COMPILACIÓN Y EJECUCIÓN

- **COMPILACIÓN:**

  javac -cp tika-app-2.9.0.jar PracticaTika.java FrecuenciaPalabras.java

- **EJECUCIÓN:**

  java -cp .:tika-app-2.9.0.jar PracticaTika -[opción] [directorio]