

```

1 package Ejercicio01;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.util.Iterator;
6 import java.util.Scanner;
7 import java.util.Set;
8 import java.util.TreeSet;
9 import java.util.concurrent.TimeUnit;
10
11 public class U08E01B {
12
13     private static Scanner stdin;           //para la lectura del fichero
14     private static String cadena = "";      //para coger cada token y manipularlo
15     private static Set <Palabras> palabras = new TreeSet<>(); //TreeSet donde guardamos Palabras
16     private static Palabras temp;          //temporal donde iremos guardando Palabras para comparar
17     private static int valor;              //para la busqueda de la palabra con mas repeticiones
18
19     public static void main(String[] args) throws FileNotFoundException { //inicio main
20
21         //ponemos en marcha el crono
22         long a = System.currentTimeMillis(); //empezamos a contar
23
24         //fichero desde el cual vamos a trabajar
25         File fichero = new File("C:\\temp\\quijote.txt");
26
27         try {
28             //indicamos desde vamos a coger los datos, en este caso el fichero
29             stdin = new Scanner(fichero);
30
31             while(stdin.hasNext()){ //mientras haya tokens
32                 //leemos token
33                 cadena = stdin.next();
34                 //limpiamos token
35                 cadena = cadena.replaceAll("[\\.\\,\\:\\:\\/\\-\\*\\|\\?\\|\\¿\\|\\;\\|\\!\\|\\)\\|\\(|\\'|\\|@\\|\\;\\|\\n\\|\\<\\|\\>]", "");
36                 //quitamos espacios
37                 cadena = cadena.trim();
38                 //pasamos token a minusculas
39                 cadena = cadena.toLowerCase();
40
41                 //con la cadena limpia, creamos el objeto Palabras
42                 temp = new Palabras(cadena, 0);
43
44                 //añadimos el objeto a la coleccion
45                 palabras.add (new Palabras(cadena,1));
46
47             }
48
49         } catch (FileNotFoundException e) {
50             System.out.println("Error: Fichero no encontrado");
51             System.out.println(e.getMessage());

```

```

52     } catch (Exception e) {
53         System.out.println("Error de lectura del fichero");
54         System.out.println(e.getMessage());
55     }
56
57     //creamos un para empezar la busqueda de la palabra mas larga
58     temp = new Palabras(", ", 0);
59
60
61     //volvemos a correr el array para encontrar la palabra de mas de 5 letras con mas repeticiones
62     //llamamos al metodo comparaTo de Palabras para realizar la comparacion
63     //segun la variable valor, almacenamos los datos en la variable temp
64     //con la que seguimos corriendo el arrayList y comparando
65     Iterator<Palabras> it = palabras.iterator();
66     while (it.hasNext()){
67         Palabras aux = (Palabras)it.next();
68         valor = aux.compara(temp);
69         if(valor > 0) {
70             temp.setPalabra(aux.getPalabra());
71             temp.ponContador(aux.getContador());
72         }
73     }
74
75
76     //paramos el crono y calculamos el tiempo transcurrido
77     long b = System.currentTimeMillis();
78     long total = b - a;
79     total = TimeUnit.MILLISECONDS.toSeconds(total);
80
81     //mostramos datos por pantalla
82     System.out.println("palabras localizadas: " + palabras.size());
83     System.out.println("Palabra: " + temp.getPalabra());
84     System.out.println("Numero repeticiones: " + temp.getContador());
85     System.out.println("Tiempo transcurrido: " + total + " segundos");
86
87 }
88
89
90
91 }
92
93
94

```