

```

1 package Ejercicio01;
2
3 //librerias importadas
4 import java.io.File;
5 import java.io.FileNotFoundException;
6 import java.util.ArrayList;
7 import java.util.ListIterator;
8 import java.util.Scanner;
9 import java.util.concurrent.TimeUnit;
10
11 public class U08E01A {           //inicio clase
12     private static Scanner stdin; //para la lectura del fichero
13     private static String cadena = ""; //para coger cada token y manipularlo
14     private static ArrayList <Palabras> palabras = new ArrayList<>(); //arrayList donde guardamos Palabras
15     private static Palabras temp; //temporal donde iremos guardando Palabras para comparar
16     private static boolean flag; //bandera que nos permite saber si hay o no que almacenar nueva palabra
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                 //primero vemos si la lista esta vacia y aï¿adimos el primer elemento
42                 //al ser primer objeto le ponemos cero, porque en la primera comprobacion
43                 //se encontrará consigo mismo y por lo tanto se sumara 1 a su contador de apariciones.
44                 //posteriormente iremos poniendo el contador a 1 en cada objeto que creemos
45                 if (palabras.size() == 0) {
46                     palabras.add(new Palabras(cadena,0));
47                 }
48
49                 //nos creamos un objeto temporal con la cadena que hemos creado
50                 temp = new Palabras(cadena,0);
51

```

```

52         //corremos el arrayList para ver si la palabra existe
53         ListIterator<Palabras> li = palabras.listIterator();
54         while ( li.hasNext()){
55             flag = true;
56             Palabras aux = (Palabras)li.next();
57             //en este caso usamos el metodo equals de la clase String
58             //comparamos el atributo palabra de cada objeto del arrayList con el temporal
59             if(aux.getPalabra().equals(temp.getPalabra())) {
60                 aux.setContador(1);
61                 flag = false;
62                 break;
63             }
64         }
65
66         //en el caso de no encontrar la palabra, creamos el objeto correspondiente
67         if(flag == true) {
68             palabras.add(new Palabras(cadena,1));
69             flag = true;
70         }
71     }
72
73     } catch (FileNotFoundException e) {
74         System.out.println("Error: Fichero no encontrado");
75         System.out.println(e.getMessage());
76     } catch (Exception e) {
77         System.out.println("Error de lectura del fichero");
78         System.out.println(e.getMessage());
79     }
80
81
82     //creamos un para empezar la busqueda de la palabra mas larga
83     temp = new Palabras(" ",0);
84
85     //volvemos a correr el array para encontrar la palabra de mas de 5 letras con mas repeticiones
86     //llamamos al metodo comparaTo de Palabras para realizar la comparacion
87     //segun la variable valor, almacenamos los datos en la variable temp
88     //con la que seguimos corriendo el arrayList y comparando
89     ListIterator<Palabras> li = palabras.listIterator();
90     while ( li.hasNext()){
91         Palabras aux = (Palabras)li.next();
92         valor = aux.compara(temp);
93         if(valor > 0) {
94             temp.setPalabra(aux.getPalabra());
95             temp.ponContador(aux.getContador());
96         }
97     }
98
99     //paramos el crono y calculamos el tiempo transcurrido
100     long b = System.currentTimeMillis();
101     long total = b - a;
102     total = TimeUnit.MILLISECONDS.toSeconds(total);

```

```
103
104     //mostramos datos por pantalla
105     System.out.println("palabras localizadas: "+ palabras.size());
106     System.out.println("Palabra: " + temp.getPalabra());
107     System.out.println("Numero repeticiones: " + temp.getContador());
108     System.out.println("Tiempo transcurrido: " + total + " segundos");
109
110     /*for (Palabras i : palabras) {
111         System.out.println(i.getPalabra() + " " + i.getContador() + " " + i.getPalabra().length());
112     }*/
113
114 }
115
116
117
118 }
119
```