```java
package Ejercicio05;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Arbol {

    private static int contador = 0;

    Nodo raiz;

    public Arbol(){
        raiz=null;
    }

    public void insertar(int i, NodoS palabra){
        Nodo n = new Nodo(i);
        n.contenido = palabra;

        if(raiz == null) {
            raiz=n;
        } else {
            Nodo aux = raiz;
            while ( aux!=null){
                n.padre=aux;
                if(n.llave >= aux.llave){
                    aux=aux.derecha;
                }else{
                    aux=aux.izquierda;
                }
            }

        if(n.llave < n.padre.llave){
            n.padre.izquierda = n;
        } else {
            n.padre.derecha = n;
        }
    }

}

    public void recorrer(Nodo n){
        if(n != null) {
            while(contador < 10){
                recorrer(n.izquierda);
                if(n.contenido.getNCaracteres() > 5){
                    int apariciones = n.contenido.buscarPalabra(n.contenido.getPalabra());
                    System.out.println("Indice " + n.llave + " palabra: " + n.contenido.getPalabra() + " aparece " +
```

```
                    apariciones + " veces.");
52                  contador++;
53              }
54              recorrer(n.derecha);
55          }
56      }
57  }
58
59
60
61      private class Nodo {
62
63          //variables miembro
64          public Nodo padre;
65          public Nodo derecha;
66          public Nodo izquierda;
67          public int llave;
68          public NodoS contenido;
69
70          public Nodo(int indice){
71              llave=indice;
72              derecha=null;
73              izquierda=null;
74              padre=null;
75              contenido=null;
76          }
77      }
78
79  }
80
81  class Principal{
82
83      public static void main(String[] args) {
84
85          Arbol miArbol = new Arbol();
86          boolean puedoInsertar = false;
87          String cadena;
88          NodoS temp;
89          int indice = 1;
90          List<String> paraules = new ArrayList<>();
91
92          //fichero desde el cual vamos a trabajar
93          File fichero = new File("C:\\temp\\quijote.txt");
94
95          try {
96              //indicamos desde vamos a coger los datos, en este caso el fichero
97              Scanner stdin = new Scanner(fichero);
98
99              while(stdin.hasNext()){     //mientras haya tokens
100                 //leemos token
101                 cadena = stdin.next();
```

```
                    //limpiamos token
                    cadena = cadena.replaceAll("[\\.\\,\\:\\/\\-\\*\\?\\¿\\¡\\!\\)\\(\\'\\@\\;\\n\\<<\\>>]","");
                    //quitamos espacios
                    cadena = cadena.trim();
                    //pasamos token a minusculas
                    cadena = cadena.toLowerCase();
                    //añadimos esa cadena al arrayList
                    if (!paraules.contains(cadena)){
                        paraules.add(cadena);
                        puedoInsertar = true;
                        indice++;
                    }
                    //con la cadena limpia, creamos el objeto NodoS
                    temp = new NodoS(cadena);
                    //si puedo insetar, creamos el nodo con un indice y el objeto
                    if (puedoInsertar) miArbol.insertar(indice, temp);
                    //ponemos el flag a false
                    puedoInsertar = false;
                }

            } catch (FileNotFoundException e) {
                System.out.println("Error: Fichero no encontrado");
                System.out.println(e.getMessage());
            } catch (Exception e) {
                System.out.println("Error de lectura del fichero");
                System.out.println(e.getMessage());
            }

            miArbol.recorrer(miArbol.raiz);


        }
    }

    class NodoS implements Comparable<NodoS>{

        //variables miembro
        String palabra;
        int nCaracteres;

        //constructor
        public NodoS(String pal){
            this.palabra = pal;
            this.nCaracteres = pal.length();
        }

        //getters
        public String getPalabra() { return this.palabra; }
        public int getNCaracteres() { return this.nCaracteres;}

        //setters
```

```
153       public void setPalabra(String pal) { this.palabra = pal;}
154       public void setNCaracteres(int cont) { this.nCaracteres = cont;}
155
156       @Override
157       public int compareTo(NodoS sub){
158           int valor = this.getPalabra().compareTo(sub.getPalabra());
159           return valor;
160       }
161
162       public int buscarPalabra(String paraula){
163           String cadena;
164           int contador=0;
165
166           //fichero desde el cual vamos a trabajar
167           File fichero = new File("C:\\temp\\quijote.txt");
168
169           try {
170               //indicamos desde vamos a coger los datos, en este caso el fichero
171               Scanner stdin = new Scanner(fichero);
172
173               while(stdin.hasNext()){     //mientras haya tokens
174                   //leemos token
175                   cadena = stdin.next();
176                   //limpiamos token
177                   cadena = cadena.replaceAll("[\\.\\,\\:\\/\\-\\*\\?\\¿\\¡\\!\\)\\(\\'\\@\\;\\n\\<<\\>>]","");
178                   //quitamos espacios
179                   cadena = cadena.trim();
180                   //pasamos token a minusculas
181                   cadena = cadena.toLowerCase();
182                   //añadimos esa cadena al arrayList
183                   if (paraula.equals(cadena)){
184                       contador++;
185                   }
186               }
187
188           } catch (FileNotFoundException e) {
189               System.out.println("Error: Fichero no encontrado");
190               System.out.println(e.getMessage());
191           } catch (Exception e) {
192               System.out.println("Error de lectura del fichero");
193               System.out.println(e.getMessage());
194           }
195
196           return contador;
197
198       }
199   }
200
```