```java
package Ejercicio03;

import java.io.File;
import java.io.FileFilter;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.attribute.UserPrincipal;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;


public class U6E03 {
    //variables globales
    public static String extension;     //almacena una extension de archivo

    public static FileFilter filter = new FileFilter() {
        public boolean accept(File file) {
                String tmp = file.getName().toLowerCase();
                if (tmp.endsWith(extension)){
                    return true;
                }
                    return false;
                }
            };



    public static void main(String[] args) throws IOException {

        //variables locales
        String carpetaActual;   //almacena la carpeta actual
        String carpeta;         //almacena una carpeta


        Scanner stdin = new Scanner(System.in); //para la entrada de datos

        boolean flag=false;

        File e = null;

        //carpeta actual
        carpetaActual = System.getProperty("user.dir");
        System.out.println("Carpeta actual: " + carpetaActual);

        //creamos el objeto File
        File d = new File(carpetaActual);
        File[] listado = d.listFiles();

        //preguntamos por la carpeta a examinar
```

```
52          System.out.print("Carpeta a examinar: ");
53          carpeta = stdin.nextLine();
54
55          //preguntamos por la extension
56          System.out.print("Explorar ficheros que acaban en: ");
57          extension = stdin.nextLine();
58
59          //vemos si existe la carpeta en el directorio actual
60          for(int i = 0; i < listado.length; i++) {
61              if(listado[i].isDirectory()) {
62                  if(listado[i].getName().equals(carpeta)) flag = true;
63                  e = new File(listado[i].getAbsolutePath());
64              }
65          }
66
67          //si la carpeta a examinar no existe
68          if (!flag) System.out.print("La carpeta no existe");
69          //si existe
70          else {
71              listado = e.listFiles();
72              //imprimimos la primera linea
73              System.out.println("[" + carpeta + "]");
74
75              //llamada a funcion
76              recorreFichero(listado,2);
77
78              //vemos si hay ficheros con la extension
79              File[] listaFicheros = e.listFiles(filter);
80              //si hay archivos con la extension
81              if(listaFicheros.length > 0) {
82                  for(int i = 0; i < listaFicheros.length; i++) {
83                      //imprimir el nombre del fichero
84                      System.out.print(listaFicheros[i].getName()+ " ");
85                      System.out.print(listaFicheros[i].length() + " bytes ");
86                      // imprimir la fecha de la ultima modificacion
87                      long millisec = listaFicheros[i].lastModified();
88                      Date dt = new Date(millisec);
89                      System.out.print(" Ul.Mod ");
90                      System.out.print(new SimpleDateFormat("dd-MM-yyyy hh:mm:ss ").format(dt));
91                      //imprimir los permisos
92                      System.out.print("Permisos (RWX):");
93                      if(listaFicheros[i].canRead())System.out.print("R");
94                      else System.out.print("-");
95                      if(listaFicheros[i].canWrite())System.out.print("W");
96                      else System.out.print("-");
97                      if(listaFicheros[i].canExecute())System.out.print("X");
98                      else System.out.print("-");
99                      //imprimir usuario
100                     Path p = listaFicheros[i].toPath();
101                     UserPrincipal userPrincipal = null;
102                     try {
```

```
                        userPrincipal = Files.getOwner(p);
                    } catch (IOException f) {
                        // TODO Bloque catch generado automáticamente
                        f.printStackTrace();
                    }
                    System.out.print(" Propietario: " + userPrincipal);

                }
            }

        }

    }

    public static void recorreFichero(File[] listado, int espacios) {
        File[] listadoTemporal = null;
        //corremos el directorio
        if(listado.length > 0) {
            for(int i = 0; i < listado.length; i++) {
                if(listado[i].isDirectory()) {
                    for(int e = 0; e < espacios; e++) System.out.print(" ");
                    System.out.println("[" + listado[i].getName() + "]");
                    File temp = new File(listado[i].getAbsolutePath());
                    listadoTemporal = temp.listFiles();
                    File[] archivosTemporal = temp.listFiles(filter);
                    if(archivosTemporal.length > 0) {
                        //hay ficheros con la extension
                        for(int j = 0; j < archivosTemporal.length; j++) {
                            for(int e = 0; e < espacios; e++) System.out.print(" ");
                            //imprimir nombre
                            System.out.print(archivosTemporal[j].getName()+ " ");
                            //imprimir peso
                            System.out.print(archivosTemporal[j].length() + " bytes ");
                            // imprimir fecha modificacion
                            long millisec = archivosTemporal[j].lastModified();
                            Date dt = new Date(millisec);
                            System.out.print(" Ul.Mod ");
                            System.out.print(new SimpleDateFormat("dd-MM-yyyy hh:mm:ss ").format(dt));
                            //imprimir permisos
                            System.out.print("Permisos (RWX):");
                            if(archivosTemporal[j].canRead())System.out.print("R");
                            else System.out.print("-");
                            if(archivosTemporal[j].canWrite())System.out.print("W");
                            else System.out.print("-");
                            if(archivosTemporal[j].canExecute())System.out.print("X");
                            else System.out.print("-");
                            //imprimir usuario
                            Path p = archivosTemporal[j].toPath();
                            UserPrincipal userPrincipal = null;
                            try {
                                userPrincipal = Files.getOwner(p);
```

```
154                             } catch (IOException e) {
155                                 // TODO Bloque catch generado automáticamente
156                                 e.printStackTrace();
157                             }
158                             System.out.print(" Propietario: " + userPrincipal);
159                             System.out.print("\n");
160                         }
161                     }
162                     recorreFichero(listadoTemporal,espacios+2);
163 
164                 }
165 
166             }
167         }
168     }
169 
170 }
171 
```