|  |  |  |
| --- | --- | --- |
| |  | | --- | | **EJERCICIO 1** | | case 1:  System.*out*.println("File Name?: ");  *nameFolder* = *keyboard*.next();  *p1\_CreateFolder*(*nameFolder*);  break;  private static void ***p1\_CreateFolder***(String nameFolder) {  String path = System.*getProperty*("user.dir");  System.*out*.println(path);  *separator* = File.*separator*;  *pathFolder* = path + *separator* + nameFolder;  System.*out*.println(*pathFolder*);  File folder = new File(*pathFolder*);//poner en formato file  System.*out*.println(folder);  if (!folder.exists()) {  folder.mkdir();  }  } | |
| |  | | --- | | **EJERCICIO 2** | | case 2:  System.*out*.print("File name?: ");  String fileName = *keyboard*.next();  System.*out*.print("Text?: ");  String text = *keyboard*.next();  fileName = *pathFolder* + *separator* + fileName + ".txt";  *p2\_createFile*(fileName, text);  break;  private static void ***p2\_createFile***(String fileName, String text) throws IOException {  //1- abrir ficheor  //fileName = pathFolder + separator + fileName + ".txt";  FileWriter fw = new FileWriter(fileName);  BufferedWriter bw = new BufferedWriter(fw);  //2- escribir en el fichero  bw.write(text);  //3- cerrar fichero  bw.flush();//borrar memoria  bw.close();//cerrar el fichero y la conexion  } | |
| |  | | --- | | **EJERCICIO 3** | | case 3:  *showFiles*();  System.*out*.println("File ?:");  int nFile = *keyboard*.nextInt();  fileName = *DocumentsList*[nFile - 1];  fileName = *pathFolder* + *separator* + fileName;  text = *readFiles*(fileName);  System.*out*.println(text);  break;  private static String ***readFiles***(String fileName) throws FileNotFoundException, IOException {  String text = "";  String line;  //1- abrir ficheor  FileReader fr = new FileReader(fileName);  BufferedReader br = new BufferedReader(fr);  //2- leer el fichero  while ((line = br.readLine()) != null) {  System.*out*.println(line);  text = line;  }  //3- cerrar fichero  br.close();  return text;  } | |
| |  | | --- | | **EJERCICIO 4** | | case 4:  *showFiles*();  System.*out*.println("File ?:");  nFile = *keyboard*.nextInt();  fileName = *DocumentsList*[nFile - 1];  fileName = *pathFolder* + *separator* + fileName;  *p4\_deleteFiles*(fileName);  break;  private static void ***p4\_deleteFiles***(String fileName) {  File file = new File(fileName);  file.delete();  } | |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **EJERCICIO 5** | | case 5:  *showFiles*();  System.*out*.println("File ?:");  nFile = *keyboard*.nextInt();  fileName = *DocumentsList*[nFile - 1];  fileName = *pathFolder* + *separator* + fileName;  text = *p5\_modifyFiles*(fileName);  System.*out*.print("Text: ");  text = *keyboard*.next();  FileWriter fw = new FileWriter(fileName);  BufferedWriter bw = new BufferedWriter(fw);  bw.write(text);  bw.flush();  bw.close();  break;  private static String ***p5\_modifyFiles***(String fileName) throws IOException {  String text = "";  String line;  FileReader fr = new FileReader(fileName);  BufferedReader br = new BufferedReader(fr);  while ((line = br.readLine()) != null) {  System.*out*.println(line);  text += line;  }  br.close();  return text;  } | |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **EJERCICIO 6** | | case 6:  *showFiles*();  System.*out*.println("File ?:");  nFile = *keyboard*.nextInt();  fileName = *DocumentsList*[nFile - 1];  fileName = *pathFolder* + *separator* + fileName;  text = *readFiles*(fileName);  int nChar = *p6\_characterCounter*(text);  System.*out*.println("Number of Char= " + nChar);  break;  private static int ***p6\_characterCounter***(String txt) {  return txt.length();  } | |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **EJERCICIO 7** | | case 7:  *showFiles*();  System.*out*.println("File ?:");  nFile = *keyboard*.nextInt();  fileName = *DocumentsList*[nFile - 1];  fileName = *pathFolder* + *separator* + fileName;  text = *readFiles*(fileName);  nChar = *p7\_wordsCounter*(text);  System.*out*.println("Number of Char= " + nChar);  break;  private static int ***p7\_wordsCounter***(String fileName) {  StringTokenizer st = new StringTokenizer(fileName);  return st.countTokens();  } | |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **EJERCICIO 8** | | case 8:  *showFiles*();  System.*out*.println("File ?:");  nFile = *keyboard*.nextInt();  fileName = *DocumentsList*[nFile - 1];  fileName = *pathFolder* + *separator* + fileName;  System.*out*.println("What word do you want to replace?: ");  String replace = *keyboard*.next();  System.*out*.println("You replace it with:");  String replace1 = *keyboard*.next();  *p8\_Replace*(fileName, replace, replace1);  break;    private static String ***p8\_Replace***(String fileName, String replace, String replace1) {  return fileName.replaceAll(replace, replace1);  } | |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **EJERCICIO 9** | | case 9:  *showFiles*();  System.*out*.println("File ?:");  nFile = *keyboard*.nextInt();  fileName = *DocumentsList*[nFile - 1];  fileName = *pathFolder* + *separator* + fileName;  *p9\_Print*(fileName);  break;  private static void ***p9\_Print***(String fileName) throws IOException {  File fileToPrint = new File(fileName);  Desktop.*getDesktop*().print(fileToPrint);  }  } | |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **EJERCICIO 10** | | case 10:  *showFiles*();  System.*out*.println("Quieres salir? Y/N");  String res = *keyboard*.next();  if (res.equals("Y")) {  System.*out*.println("Byee");  } else {  option = 90;  System.*out*.println("Continue: ");  }  break;  default:  System.*out*.println("Opcion no valida");  }//fin switch  } while (option != 10);  } | |