

CASO PRÁCTICO 1 - ACCESO DE DATOS

JOSÉ CRUCES APARICIO 2º DAM

LECTURA DE FICHERO

Contexto

Luis está realizando una aplicación y debe almacenar un listado de datos. Como va a ser una cantidad pequeña de datos, decide almacenarlos en un fichero. Para comenzar las pruebas, almacena directamente los datos a mano con el bloc de notas. Ahora deberá comprobar si desde su aplicación puede leerlos correctamente.

Cuestiones a resolver

1. El siguiente código lee un fichero de texto línea a línea:

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class lecturajava {
    public static void leerfichero(String ruta) throws
FileNotFoundException, IOException {
        String salida="";
        FileReader fr = new FileReader(ruta);
        BufferedReader br = new BufferedReader(fr);

        //lectura del fichero mediante bucle while
        while((salida = br.readLine())!=null) {

        }
        br.close();
    }

    public static void main(String[] args) throws IOException {
        //llamamos a leerfichero con la ruta donde se encuentra
        leerfichero("c://prueba.txt");
    }
}
```

Crea una clase Java que lea un fichero de texto y lo muestre por pantalla. Puedes usar la misma clase mostrada. Ejecuta dicha clase mostrando el resultado.

2. Explica las líneas de códigos principales del código presentado. Por ejemplo:

- FileReader es...
- El bucle while se usa para...

3. Modifica dicha clase para que muestre por pantalla las líneas que contiene el fichero y el número de caracteres de cada línea. Intenta mostrar la línea y, seguidamente, el número de caracteres, tal y como se muestra a continuación:

```

Problems @ Javadoc Declaration Console
<terminated> lecturajava [Java Application] C:\Program Files (x86)\

Hola Sergio-->11
-->0
Estoy leyendo de un fichero en Java-->35
Puedes intentarlo desde otro lenguaje-->37
-->0
Sergio L.-->9
Fin del ejercicio-->17
el número de líneas es:7

```

Respuesta al ejercicio

Resolución del ejercicio, muestro aquí el código generado para resolver las cuestiones pedidas anteriormente y con muchos comentarios como se me pide explicando todo :

```

leeFichero.java
1 // Importamos las librerías necesarias para la lectura del archivo y la gestión de las excepciones
2 import java.io.BufferedReader;
3 import java.io.IOException;
4 import java.io.FileReader;
5 import java.io.FileNotFoundException;
6
7 // Clase leeFichero con las indicaciones de lo que hace
8 public class leeFichero {
9
10    // Método leeFichero que recibe la ruta que se quiere leer
11    // La ruta se la pasará desde el método main
12    public static void leeFichero(String ruta) throws IOException {
13        FileNotFoundException, IOException {
14            // Declaración de la variable tipo String para guardar ahí las líneas de texto
15            // que vamos a ir leyendo del archivo
16            String salida="";
17            // Iniciación de la variable con un número entero que marca el numero uno
18            // para que inicia el método desde la linea 1 del archivo que va a leer y luego irá creciendo conforme
19            // se vayan leyendo líneas a líneas
20            int numeroLineas=1;
21            // Creamos los objetos para poder leer el archivo
22            FileReader fr = new FileReader(ruta);
23            BufferedReader br = new BufferedReader(fr);
24
25        }
26
27        // Lectura del fichero mediante bucle while para que mientras haya líneas que leer
28        // siga contando y mostrando en pantalla las líneas y las cuentas de caracteres
29        while((salida = br.readLine())!=null) {
30            // Se guarda en una variable la cantidad de caracteres que tiene la linea leída
31            int cantidadCaracteres = salida.length();
32            // Mostrar en pantalla la linea que estamos recorriendo,
33            // con el numero de linea y la cantidad de caracteres que tiene cada linea
34            System.out.println("Linea " + numeroLineas + " (" + cantidadCaracteres + " caracteres): " + salida);
35            // Incrementa el número de líneas en uno para continuar con la siguiente
36            numeroLineas++;
37
38        }
39
40        // Cerramos el BufferedReader para liberar recursos y memoria
41        br.close();
42
43    }
44
45    // Main del programa que ejecuta la lectura de fichero con el método leeFichero()
46    public static void main(String[] args) throws IOException {
47        // Llamamos a leeFichero con la ruta donde se encuentra el archivo
48        // y ejecutamos leeFichero para que haga lo que queremos que haga
49
50        leeFichero("C:/Users/josechuait/Desktop/prueba.txt");
51    }
52
53}

```

Pongo aqui el codigo por que en la captura no se lee perfecto por que habia que comprimir los archivos en la plataforma, pero aqui se puede leer perfectamente :

```
// importamos las librerias necesarias para la lectura del archivo y la gestion de las excepciones
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

// Clase leerfichero con las indicaciones de lo que hace
public class leeFichero {

    // Metodo leerfichero que recibe la ruta que se quiere leer
    // La ruta se la pasará desde el método main
    public static void leerfichero(String ruta) throws
        FileNotFoundException, IOException {
        // Declaracion de la variable tipo String para guardar ahí las líneas de texto
        // que vamos a ir leyendo del archivo
        String salida="";
        // Declaracion de la variable con un numero entero que marca el numero uno
        // para que inicie el metodo desde la linea 1 del archivo que va a leer y luego ira creciendo conforme
        // se vayan leyendo linea a linea
        int numeroLinea=1;
        // Creamos los objetos para poder leer el archivo
        FileReader fr = new FileReader(ruta);
        BufferedReader br = new BufferedReader(fr);

        //lectura del fichero mediante bucle while para que mientras haya líneas que leer
        // siga contando y mostrando en pantalla las líneas y las cuentas de caracteres
        while((salida = br.readLine())!=null) {
            // Guarda en una variable la cantidad de caracteres que tiene la linea leída
            int cantidadCaracteres = salida.length();
            // Mostrar en pantalla la linea que estamos recorriendo ,
            // con el numero de linea y la cantidad de caracteres que tiene cada linea
            System.out.println("Línea " + numeroLinea + " (" + cantidadCaracteres + " caracteres): " + salida);
            // Incrementa el número de linea en uno para continuar con la siguiente
            numeroLinea++;

        }
        // Cerramos el BufferedReader para liberar recursos y memoria
        br.close();
    }

    // main del programa que ejecuta la lectura de fichero con el metodo leerfichero()
    public static void main(String[] args) throws IOException {
        //Llamo a leerfichero con la ruta donde se encuentra el archivo
        // y ejecuto leerfichero para que haga lo que quiero que haga

        leerfichero("/Users/josechuair/Desktop/prueba.txt");
    }
}
```

Y en las siguientes capturas de pantalla muestro el resultado tras la ejecución del programa:

The screenshot shows the IntelliJ IDEA interface. The left panel displays the project structure for 'leeFichero' with files like 'leeFichero.java', 'leeFichero.out', and 'leeFichero.iml'. The right panel shows the code for 'leeFichero.java' which reads a file named 'privada.txt'. The 'Run' tab at the bottom is selected, showing the output of the program. The output window displays the following text:

```

Línea 1 (11 caracteres): Estoy leyendo de un fichero en Java
Línea 2 (0 caracteres):
Línea 3 (35 caracteres): Puedes intentarlo desde otro lenguaje
Línea 4 (37 caracteres): Puedes intentarlo desde otro lenguaje
Línea 5 (0 caracteres):
Línea 6 (9 caracteres): Sergio L.
Línea 7 (17 caracteres): Fin del ejercicio
Process finished with exit code 0

```

This screenshot shows a terminal window with the title 'Run' and the tab 'leeFichero' selected. The terminal output shows the command used to run the Java application and the resulting text output. The output is identical to the one shown in the IntelliJ IDEA run output.

```

/Library/Java/JavaVirtualMachines/jdk-23.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52731:/Applications/IntelliJ IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/josechuair/IdeaProjects/leeFichero/out/production/leeFichero leeFichero
Línea 1 (11 caracteres): Hola Sergio
Línea 2 (0 caracteres):
Línea 3 (35 caracteres): Estoy leyendo de un fichero en Java
Línea 4 (37 caracteres): Puedes intentarlo desde otro lenguaje
Línea 5 (0 caracteres):
Línea 6 (9 caracteres): Sergio L.
Línea 7 (17 caracteres): Fin del ejercicio
Process finished with exit code 0

```

Conclusión: El programa funciona correctamente, del archivo texto.txt hemos conseguido que muestre las líneas de texto, que cuente la cantidad de caracteres de cada línea y que también muestre esa información en pantalla.

Bibliografía:

UD1 Temario Acceso datos
Documentación oficial Java:

<https://docs.oracle.com/javase/8/docs/api/java/io/FileReader.html>