

```

43 <body <?php body_class()>
44 <div id="fb-root"></div>
45 <script>(function(d, s, id) {
46   var js, fjs = d.getElementsByTagName(s)[0];
47   if (d.getElementById(id)) return;
48   js = d.createElement(s); js.id = id;
49   js.src = "//connect.facebook.net/en_US/sdk.js#xfbml=1&version=v2.6&appId=2866444444444444";
50   fjs.parentNode.insertBefore(js, fjs);
51   }(document, 'script', 'facebook-jssdk'));</script>
52 <div id="page" class="site">
53   <a class="skip-link screen-reader-text" href="#content"><?php esc_html_e( 'Skip to content', 'urduube' ); ?></a>
54   <header id="masthead" class="site-header" role="banner">
55     <div class="site-branding">
56       <div class="navBtn pull-left">
57         <?php if(is_home() && $xpanel['homepage-style'] == 1) { ?>
58           <a href="#" id="openMenu"><i class="fa fa-bars fa-3x"></i></a>
59           <?php } else { ?>
60             <a href="#" id="openMenu2"><i class="fa fa-bars fa-3x"></i></a>
61             <?php } ?>
62         </div>
63         <div class="logo pull-left">
64           <a href="<?php echo esc_url( home_url() ) ?>">
65             
66           </a>
67         </div>
68         <div class="search-box hidden-xs hidden-sm pull-left ml-10">
69           <?php get_search_form(); ?>
70         </div>
71         <div class="submit-btn hidden-xs hidden-sm pull-left ml-10">
72           <a href="<?php echo get_page_link($xpanel['submit-link']) ?>" class="header-submit-btn"><i class="fa fa-search fa-3x"></i></a>
73         </div>
74         <div class="user-info pull-right mr-10">
75           <?php
76           if ( is_user_logged_in() ) {

```

# [ACCESO A DATOS]

[UNIDAD 1 TAREA 1]

[Javier Ortega Castillo]

## EJERCICIO: PLANTEAMIENTO DEL PROBLEMA

### 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 y procesarlos desde la aplicación.

### RESUELVE

Crea una clase Java llamada Fichero para gestionar los accesos a ficheros de texto usando `BufferedReader`.

Deberás escribir dos métodos públicos:

- `Fichero.leer(String nombre)`, al que se le pase como parámetro el nombre del archivo a leer y lo muestre por pantalla línea a línea.
- `Fichero.estadistica(String nombre)`, al que se le pase como parámetro el nombre del archivo a leer y muestre por pantalla el total de caracteres y el número de líneas.

Ejecuta los métodos desde la clase principal para probarlos.

Enviar el proyecto y un documento pdf en el que se muestre capturas de pantalla mostrando la correcta ejecución del programa y todo aquello que consideres necesario.

## RESOLUCIÓN DEL EJERCICIO

Tal y como dice el ejercicio, lo primero sería crear la clase Fichero, con los 2 métodos que se nos pide.

El primero quedaría de la siguiente forma:

```
// LEE EL ARCHIVO Y LO MUESTRA POR PANTALLA
public void leer(String nombre) throws IOException {

    File file = new File("C:/EjerciciosProgramacion/" + nombre);
    BufferedReader leer = null;

    if (file.exists()) {

        //EL BUFFERREADER SE VA A ENCARGAR IR RECORRIENDO TODO EL ARCHIVO
        leer = new BufferedReader(new FileReader(file));

        String linea = "";

        //RECORRERÁ TODO EL ARCHIVO LÍNEA A LÍNEA Y LAS IMPRIME POR PANTALLA
        while ((linea = leer.readLine()) != null) {

            System.out.println(linea);
        }

    }

    leer.close();
}
```

El código es simple pero aun así voy a explicar lo que hace.

Primero abrimos el archivo el cual se encuentra en la carpeta C:/EjerciciosProgramacion, y lo lee línea por línea, comprobando previamente que dicho archivo existía. Cuando lee la primera, al imprime por pantalla, y luego pasa a la siguiente, así hasta el final.

El segundo método quedaría de la siguiente forma:

```
// LEE EL ARCHIVO Y MUESTRA LA CANTIDAD DE CARACTERES Y LINEAS QUE TIENE
public void estadistica(String nombre) throws IOException {

    //INDICAMOS LA RUTA + NOMBRE DEL ARCHIVO
    File file = new File("C:/EjerciciosProgramacion/" + nombre);
    BufferedReader leer = null;

    int contadorCaracteres = 0;
    int contadorLineas = 0;

    if (file.exists()) {

        //LEER VA A SER LA VARIABLE QUE VA A IR LEYENDO EL ARCHIVO
        leer = new BufferedReader(new FileReader(file));

        String linea = "";

        //MIENTRAS QUEDE ALGUNA LÍNEA POR LEER, SUMA LOS CARACTERES Y LAS LÍNEAS, Y LAS MUESTRA
        //POR PANTALLA
        while ((linea = leer.readLine()) != null) {

            contadorCaracteres += linea.length();
            contadorLineas++;

        }

        System.out.println("El archivo tiene " + contadorCaracteres + " caracteres.");
        System.out.println("El archivo tiene " + contadorLineas + " líneas.");

    }

    leer.close();
}
```

Este segundo método funciona muy parecido, pero en vez de mostrar el archivo sin más, cuenta los caracteres y las líneas que contiene. Para saber el número de caracteres podemos aprovechar el código anterior, y hacer el `length()` de línea. Las líneas de código es tan simple como sumar cada vez que entre en el bucle.

He creado este archivo de texto como ejemplo:



\*accesoADatos.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Hola

me llamo Juan

y soy estudiante

en el ciclo online DAM de cesur

Una vez hecho esto, vamos a crear un main y comprobar que todo funciona bien.

```
import java.io.IOException;

public class main {

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

        String nombreArchivo = "accesoADatos.txt";
        Fichero fichero = new Fichero();

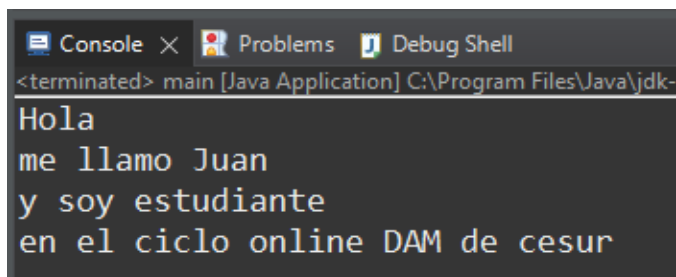
        fichero.leer(nombreArchivo);
        fichero.estadistica(nombreArchivo);

    }

}
```

Una vez lo tenemos terminado, vamos a comprobar que funciona. Estos son los resultados de ambas líneas de código.

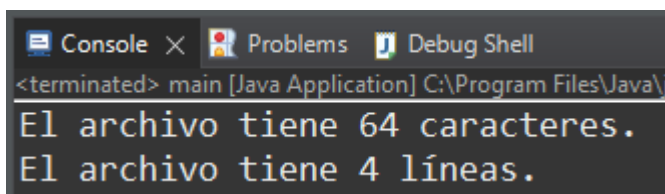
- fichero.leer(nombreArchivo);



The screenshot shows a Java IDE console window with the following output:

```
<terminated> main [Java Application] C:\Program Files\Java\jdk-
Hola
me llamo Juan
y soy estudiante
en el ciclo online DAM de cesur
```

- fichero.estadistica(nombreArchivo);



The screenshot shows a Java IDE console window with the following output:

```
<terminated> main [Java Application] C:\Program Files\Java\j
El archivo tiene 64 caracteres.
El archivo tiene 4 líneas.
```

Como podemos observar, ambos códigos están funcionando correctamente.

## CONCLUSIÓN

En resumen, creo que ha sido una buena primera tarea. Me ha gustado que hayamos trabajado con los File, algo que en el primer curso vimos, pero un poco de reojo, sin centrarnos mucho en eso, y creo que es algo que hoy en día es muy útil para una gran cantidad de trabajos, como puede ser por ejemplo que otra persona nos de cierta información en un bloc de notas, y nosotros luego podemos adaptar el código para que lea esa información y podamos convertirla en datos que podamos usar fácilmente.