



Acceso a datos

UT1 Manejo de ficheros

Objetivo

- Controlar el sistema de ficheros de nuestro equipo desde java
 - O, lo que es lo mismo, **acceder a los datos** del sistema de ficheros de nuestro equipo desde Java.
- Ejemplos:
 - Saber si un determinado fichero está presente en una carpeta.
 - Hacer un dir de una carpeta

¿Qué es un fichero?

- Reflexiona y expresa lo que entiendes por fichero
- Busca algunas definiciones “oficiales” de lo que es un fichero

¿Qué representa la clase `File`?



La clase File

- CONSTRUCTORES

File(File parent, String child)

Creates a new File instance from a parent abstract pathname and a child pathname string.

File(String pathname)

Creates a new File instance by converting the given pathname string into an abstract pathname.

File(String parent, String child)

Creates a new File instance from a parent pathname string and a child pathname string.

File(URI uri)

Creates a new File instance by converting the given file: URI into an abstract pathname.

La clase File pertenece al paquete `java.io` y representa una herramienta para manejar un fichero o directorio de nuestro sistema de archivos.

Esta clase puede pasarse como parámetro a las principales clases que manejan ficheros, como `FileReader`, `FileInputStream`, `FileWriter`, etc.

- ALGUNOS MÉTODOS INTERESANTES...

- boolean exists() → true si el archivo existe en una ruta dada
- String getPath()
- String getAbsolutePath()
- boolean isDirectory()
- String[] list() → es como el comando “dir” o “ls” (para objetos File que son directorios)

- Lista completa de métodos → **Consultar el API online!!**

En Programación de 1º ya se estudiaron estos métodos de la clase File

■ Clase *File*

■ Constructores

- `File(String ruta)`
- `File(String ruta, String nombre)`
- `File(File directorio, String nombre)`

■ Métodos

- `canRead()` comprueba si el fichero se puede leer
- `canWrite()` comprueba si el fichero se puede escribir
- `delete()` borra dicho fichero
- `getPath()` devuelve la ruta del fichero
- `mkdir()` crea un directorio con la ruta del objeto que lo recibe
- `isDirectory()` comprueba si dicho fichero es un directorio

■ Constructores de otras clases

- `FileReader(File fichero)`
- `FileWriter(File fichero)`

Este curso trabajaremos con algunos más...

NOMBRE	DESCRIPCIÓN	PARÁMETROS	DATO DEVUELTO
exists	Indica si existe o no el fichero.	Ninguno	boolean
isDirectory	Indica si el objeto File es un directorio.	Ninguno	boolean
isFile	Indica si el objeto File es un fichero.	Ninguno	boolean
isHidden	Indica si el objeto File esta oculto.	Ninguno	boolean
getAbsolutePath	Devuelve una cadena con la ruta absoluta del fichero o directorio.	Ninguno	String
canRead	Indica si se puede leer.	Ninguno	boolean
canWrite	Indica si se puede escribir.	Ninguno	boolean
canExecute	Indica si se puede ejecutar.	Ninguno	boolean
getName	Devuelve una cadena con el nombre del fichero o directorio.	Ninguno	String
getParent	Devuelve una cadena con el directorio padre.	Ninguno	String
listFiles	Devuelve un array de File con los directorios hijos. Solo funciona con directorios.	Ninguno	Array de File
list	Devuelve un array de String con los directorios hijos. Solo funciona con directorios.	Ninguno	Array de String
mkdir	Permite crear el directorio en la ruta indicada. Solo se creará si no existe.	Ninguno	boolean
makedirs	Permite crear el directorio en la ruta indicada, también crea los directorios intermedios. Solo se creará si no existe.	Ninguno	boolean
createNewFile	Permite crear el fichero en la ruta indicada. Solo se creará si no existe. Debemos controlar la excepcion con IOException.	Ninguno	boolean

Ejemplo 1: Averiguar si un determinado fichero está presente en la carpeta actual

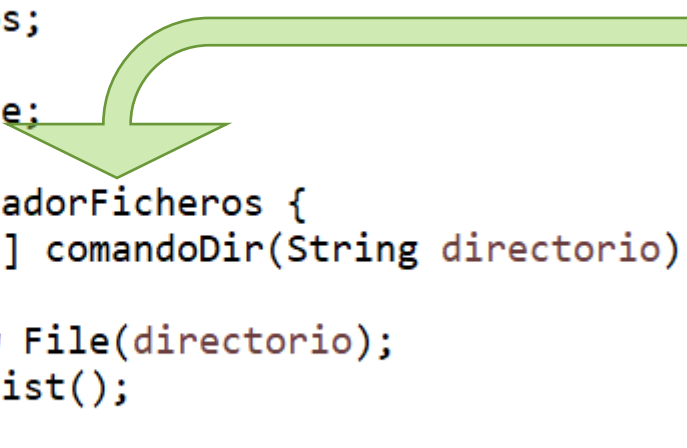
```
1 package UT1Ficheros;
2 import java.io.*;
3
4 public class Ejemplo1
5 {
6     public static void main(String[] args)
7     {
8         System.out.println("¿Está inventado.txt en esta carpeta?");
9         File f = new File("inventado.txt");
10        System.out.println(f.exists());
11
12    }
13 }
```

Si estamos trabajando con Eclipse la carpeta actual se corresponde con la carpeta del proyecto en que estemos, dentro del workspace que hayamos definido.

La carpeta actual se identifica por el símbolo punto (.)

Ejemplo 2: Mostrar la lista de ficheros en un directorio pasado como parámetro

```
*ManejadorFicheros.java ✖
1 package UT1Ficheros;
2
3 import java.io.File;
4
5 public class ManejadorFicheros {
6     public String[] comandoDir(String directorio)
7     {
8         File f=new File(directorio);
9         return f.list();
10    }
11
12 }
```



```
*TestManejadorFicheros.java ✖
1 package UT1Ficheros;
2
3 public class TestManejadorFicheros {
4     public static void main(String[] args)
5     {
6         ManejadorFicheros m = new ManejadorFicheros();
7
8         String directorio = "C:/app2019/Prueba";
9         String[] ficheros=m.comandoDir(directorio);
10
11         for(int i=0; i<ficheros.length; i++)
12         {
13             System.out.println(ficheros[i]);
14         }
15     }
16 }
```

Ejercicio 1

- Crea un método que te diga si un determinado fichero pasado como parámetro está presente dentro de un directorio también pasado como parámetro. Llama a este método “estaEn”
- Escribe el código necesario para poder probar el citado método.

Ejercicio 2

- **Investiga** cómo pasar parámetros al método main **a través de la línea de comandos**. Modifica el ejemplo 2 para que muestre la lista de ficheros que hay dentro de un directorio pasado a través de la línea de comandos.

Ejercicio 3

- Crea un método que te diga si un determinado fichero pasado como parámetro es un directorio.
- En caso afirmativo devolverá true y, adicionalmente, mostrará por consola los ficheros que contiene.

Ejercicio 4

- Escribir el código necesario para mostrar por consola la siguiente información de un fichero que se le pasa a través de la línea de comandos (comprobar primero que el fichero está presente en el directorio en el que estamos)
 - Nombre
 - Ruta absoluta
 - Se puede leer
 - Tamaño
 - Decir si se trata de directorio o de un fichero

Ejemplo de E/S de datos desde un fichero

Estudiado en
Programación de 1º

```
try {
    BufferedReader reader =
        new BufferedReader(new
    FileReader("nombrefichero"));
    String linea = reader.readLine();
    while(linea != null) {
        // procesar el texto de la línea
        linea = reader.readLine();
    }
    reader.close();
}
catch(FileNotFoundException e) {
    // no se encontró el fichero
}
catch(IOException e) {
    // algo fue mal al leer o cerrar el fichero
}
```