

CLASE FILE

Esta clase se encuentra en la librería **java.io**. Con esta clase no podemos acceder a la información que contiene el fichero, tan solo a sus características. Tenemos tres constructores:

- ✓ `File(String nombre directorio)`, también puede aparecer el nombre del fichero incluido.

File fp = new File("d:/carpeta/fichero");

- ✓ `File(String ruta, String fichero)`

File fp = new File("d:/carpeta/", "fichero");

- ✓ `File(File ruta, String fichero)`

File fp = new File("d:/carpeta/");
File pf = new File(fp, "fichero");

Veamos los métodos para directorios y ficheros:

- ✓ **boolean canRead()**, informa si podemos leer la información que posee.
- ✓ **boolean canWrite()**, informa si podemos escribir.
- ✓ **boolean exists()**, indica si el fichero o directorio existe.
- ✓ **boolean isFile()**, nos dice si es un fichero.
- ✓ **boolean isDirectory()**, nos dice si es un directorio.
- ✓ **long lastModified()**, nos devuelve la fecha de la última modificación en milisegundos y ese valor lo tenemos que pasar a formato fecha.
- ✓ **String getName()**, nos devuelve el nombre del fichero o directorio.
- ✓ **String getPath()**, nos devuelve el nombre de la ruta.
- ✓ **String getParent()**, devuelve el directorio padre.

Si capturamos por teclado el nombre del fichero en los métodos `getName`, `getPath`, `getParent` no tienen mucho sentido, ya que sabemos esa información.

Métodos solo para ficheros:

- ✓ **boolean delete()**, borra el fichero.
- ✓ **long length()**, nos da el tamaño del fichero en bytes.
- ✓ **boolean renameTo(File)**, cambia el nombre del fichero. Pero en el objeto puede mantener el nombre antiguo.
- ✓ **void deleteOnExit()**, borra el archivo cuando finaliza la ejecución del programa. Es útil para los ficheros temporales o de trabajo.
- ✓ **boolean createNewFile()**, nos crea un fichero vacío de tamaño 0 bytes.

Métodos solo para directorios:

- ✓ **boolean mkdir()**, crea un directorio.
- ✓ **File [] listFiles()**, devuelve los nombres de los ficheros y directorios que se encuentra en el directorio indicado.

<https://docs.oracle.com/javase/7/docs/api/java/io/File.html>
<https://www.discoduroderoer.es/clase-file-y-sus-metodos/>

<https://tinchicus.com/2019/05/08/java-file-class/>

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 creara si no existe.	Ninguno.	boolean
mkdirs	Permite crear el directorio en la ruta indicada, también crea los directorios intermedios. Solo se creara si no existe.	Ninguno.	boolean
createNewFile	Permite crear el fichero en la ruta indicada. Solo se creara si no existe. Debemos controlar la excepción con IOException.	Ninguno.	boolean

Tenemos otros métodos como:

long getFreeSpace(), devuelve el número de bytes no ocupados en la partición o HD nombrada por este nombre de ruta abstracto.

long getTotalSpace(), devuelve el número total bytes de la unidad indicada o partición.

static File[] listRoots(), listado de unidades del sistema de archivos disponibles.

```
import java.io.File;
import java.util.Scanner;
/* capturamos el nombre de un fichero y si existe indicamos su tamaño
                                                                    File1.java
                                                                    */

class File1
{
    public static void main(String [] args)
    {
        String nombre;
        Scanner dato = new Scanner(System.in);
        File fichero;
        System.out.println("Escribe el nombre del fichero ");
        nombre = dato.next();
        fichero = new File(nombre);
        if ( fichero.exists())
        {
            System.out.println("EL FICHERO OCUPA "+fichero.length() +" BYTES");
        }
        else
            System.out.println("EL FICHERO NO EXISTE ");
    }
}
```