

FICHEROS:

Soporte de **memoria externa que almacena y registra datos**, para mantenerlos persistentemente en todas la ejecuciones.

PAQUETE:

package java.io Contiene las clases para manejar **ficheros**

TIPOS DE RUTAS:

ABSOLUTA: C:/Pro gram as /un ida d6/ apa rta dol /ej emp los /Fi leO bte ner Rut a.java

RELATIVA: Aparta dol /ej emp los /Fi leO bte ner Rut a.java

RUTA:

Ruta Unix: /usr/bin

Ruta Wind C:\Win do w s \S ystem32

ows:

CLASE FILE:

Ficheros y carpetas Operaciones sobre rutas de ficheros existentes o no

Importar e inicializar: file F = new file (String ruta)

ARRAY FILE:

E006listado.java

File[] listFiles(Donde se enumeran todos los elementos de la carpeta

Ejemplo de inicialización:

```
File carpeta = new File("C :/T emp ");
```

```
File [] arrayE lementos = carpet a.l ist Fil es();
```

FILE BYTE:

La diferencia fundamental es que no utiliza la representación en forma de texto sino que usa el formato binario. La ventaja es que ocupan me

E022binario.java

INICIALIZACION:

Clase RandomAccessFile → package java.io

```
Random Acc essFile raf = new Random Acc ess Fil e(Fi:ode );
```

Leer un archivo:

```
Random Acc essFile raf = new Random Acc ess Fil e(Fi:ode );
```

Escribir en un fichero:

```
Random Acc essFile raf = new Random Acc ess Fil e(Fi:ode );
```

METODOS:

E022binario.java

```
raf.wr ite Int();
```

E023sobreescribirBin.java

setLength(long medida) Modifica la medida del fichero Sino es mayor, se borra el exceso. Si es mayor, se

long getFilePointer(); Devuelve el desplazamiento del cursor. E0.java

..

FILE METODOS:

E001file.java

getParent() Devuelve el padre del file o null

getName() Devuelve el nombre del objeto dado

getAbsolutePath() Muestra ruta absoluta

E002estado.java

isFile() Booleano si la ruta del archivo abstracto es un archivo o no.



By kattorres

cheatography.com/kattorres/

Not published yet.

Last updated 7th June, 2022.

Page 1 of 2.

Sponsored by [CrosswordCheats.com](https://crosswordcheats.com)

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

FILE METODOS: (cont)

`isDirectory()` **Boolean** comprueba si el fichero instanciado es un directorio o no

`.exists();` **Boolean** comprueba si existe un fichero o directorio instanciado con la clase File

E003lecturaModificacion.java

`long length()` La función devuelve un valor largo que representa el número de bits; de lo contrario, devuelve 0L

`long lastModified()` Evalúa la última fecha de edición del elemento.

`Date fecha = new Date(documento.lastModified())`
E004gestionFicheros.java

`mkdir()` crear un nuevo directorio indicado por la ruta de acceso abstracta. La función devuelve BOOLEAN

`delete()` Permite eliminar ficheros y carpetas, devuelve un boolean

E005rename.java

`renameTo(File destino)`

METODOS EXT:

E018vocales.java

`.toLowerCase();` Convierte una cadena en minúscula.

`.toUpperCase();` Convierte una cadena en mayúscula.

E019enterosOrdenados.java

`.sort();` Ordena un arreglo, array.

LECTURA / ESCRITURA FILE

LECTURA:

Clase Scanner → package java.util (→ lectura)

INICIALIZACIÓN:

`Scanner lectura = new Scanner(File f);`

E012lectura.java

`f.close();` Al terminar una lectura, cerrar siempre.

`lectura.next();` Lee caracteres no separados por espacios o saltos.

`lectura.nextInt();` lee una frase completa hasta el siguiente salto de línea

ESCRITURA:

Clase PrintStream → package java.io (→ escritura)

INICIALIZACIÓN:

`PrintStream escribir = new PrintStream(File f);`

E014escribeEnteros.java

`escribir.println()` Hace salto de línea.

;

`escribir.print();` El puntador avanza un valor.

SENTENCIA TRY/CATCH

Todas las instrucciones vinculadas a trabajar con ficheros harán uso de esta sentencia.

TRY {} Mientras no haya excepciones se ejecuta este bloque de código

catch (Exception e) {} Se ejecuta cuando existen excepciones

