

10.D. Aplicaciones del almacenamiento de información en ficheros.

2. Utilización de los sistemas de ficheros.

2.1. Clase File.

La clase **File** proporciona una representación abstracta de ficheros y directorios.

Esta clase, permite examinar y manipular archivos y directorios, independientemente de la plataforma en la que se esté trabajando: Linux, Windows, etc.

Las instancias de la clase **File** representan nombres de archivo, no los archivos en sí mismos.

El archivo correspondiente a un nombre dado podría ser que no existiera, por ello, habrá que controlar las posibles excepciones.

Al trabajar con **File**, las rutas pueden ser:

- Relativas al directorio actual.
- Absolutas si la ruta que le pasamos como parámetro empieza por
 - La barra "/" en Unix, Linux.
 - Letra de unidad (C:, D:, etc.) en Windows.
 - UNC(universal naming convention) en windows, como por ejemplo:

```
File miFile=new File("\\\\mimaquina\\download\\prueba.txt");
```

A través del objeto **File**, un programa puede examinar los atributos del archivo, cambiar su nombre, borrarlo o cambiar sus permisos. Dado un objeto **file**, podemos hacer las siguientes operaciones con él:

- **Renombrar** el archivo, con el método **renameTo()**. El objeto **File** dejará de referirse al archivo renombrado, ya que el **String** con el nombre del archivo en el objeto **File** no cambia.
- **Borrar** el archivo, con el método **delete()**. También, con **deleteOnExit()** se borra cuando finaliza la ejecución de la máquina virtual Java.
- **Crear** un nuevo fichero con un nombre único. El método estático **createTempFile()** crea un fichero temporal y devuelve un objeto **File** que apunta a él. Es útil para crear archivos temporales, que luego se borran, asegurándonos tener un nombre de archivo no repetido.
- **Establecer** la fecha y la hora de modificación del archivo con **setLastModified()**. Por ejemplo, se podría hacer: **new File("prueba.txt").setLastModified(new Date().getTime());** para establecerle la fecha actual al fichero que se le pasa como parámetro, en este caso *prueba.txt*.
- **Crear** un directorio con el método **mkdir()**. También existe **mkdirs()**, que crea los directorios superiores si no existen.
- **Listar** el contenido de un directorio. Los métodos **list()** y **listFiles()** listan el contenido de un directorio **list()** devuelve un vector de **String** con los nombres de los archivos, **listFiles()** devuelve un vector de objetos **File**.
- **Listar** los nombres de archivo de la raíz del sistema de archivos, mediante el método estático **listRoots()**.

Autoevaluación

Indica si es verdadera o falsa la siguiente afirmación:

Un objeto de la clase **File** representa un fichero en sí mismo. ¿Verdadero o falso?

- ☐ Verdadero.
- ☐ Falso.

EducaMadrid - Vicepresidencia, Consejería de Educación y Universidades - [Ayuda](#)

