

Ejercicios Tema 1. Acceso a ficheros

Todos los ejercicios de este boletín deben estar dentro de un **package** llamado **tema01** (Siguiendo las reglas que hemos visto en clase, ej: `com.germangascon.tema01`).

Para la realización de los siguientes ejercicios se recomienda crear una carpeta específica donde copiaremos algunos directorios y archivos de tu disco duro (los que tu quieras) a modo de ejemplo. Todas las rutas deben hacer referencia, única y exclusivamente, a archivos ubicados dentro de dicha carpeta.

Un ejemplo de ruta absoluta para la carpeta de los ejemplos podría ser:

`/home/usuario/tema11/ficheros`

Para evitar problemas, no utilices espacios en blanco en la ruta absoluta de la carpeta.

La carpeta debe contener al menos: varios directorios y subdirectorios, varios archivos de texto plano (*.txt) y varios archivos binarios (*.exe, *.bin).

1. Utilizando la clase File comprueba que la carpeta que has creado para este boletín de ejercicios existe y que se trata de un directorio.
2. Muestra los archivos y directorios que están en la carpeta del ejercicio. Solo los de primer nivel (no recursivo).
3. Obtén las siguientes propiedades de la carpeta del ejercicio:
 - a) El nombre de la carpeta
 - b) La ruta absoluta
 - c) Si se puede leer
 - d) Si se puede escribir
4. Obtén las siguientes propiedades de un archivo que esté contenido dentro de la carpeta del ejercicio (el que tú quieras):
 - a) El nombre del archivo
 - b) La ruta absoluta
 - c) Si se trata de un archivo oculto
 - d) Si se puede leer

- e) Si se puede escribir
- f) La fecha de la última modificación (ayúdate de `DateTimeFormatter`). Luego cambia la fecha de modificación a la hora actual y vuélvela a comprobar.
- g) El tamaño en bytes, en KB y en MB.

5. Crea los siguientes métodos en una clase llamada `GestionArchivos`.

- a) `boolean crearArchivo(String directorio, String archivo)`: creará un archivo con el nombre `archivo`.
- b) `void listarDirectorio(String directorio)`: visualizará el contenido del directorio mostrando el tipo (fichero "f" , directorio "d"), el tamaño y los permisos de lectura y escritura. Ejemplo:
`hola.txt f 10 bytes rw`
- c) `void verInfo(String directorio, String archivo)`: visualizará el nombre, la ruta absoluta, si se puede escribir, si se puede leer, el tamaño, si es un directorio y si es un archivo.

Los parámetros que necesitan los métodos serán solicitados al usuario desde el programa principal.

6. Añade un método a la clase anterior que permita leer y mostrar el contenido de un archivo de texto.

7. Añade un método a la clase anterior que permita ver el contenido de un archivo binario en hexadecimal agrupados en bytes (de dos en dos). Ejemplo de salida:

```
6d 20 1b 5b 33 37 6d 5b 32 31 3a
3a 5d 0d 0a 1b 5b 31 6d 1b 5b 33
5b 30 30 6d 1b 5b 4b 1b 5b 3f 31
```

.....

Recuerda que la opción de formato `"%X"` podemos representar un valor numérico en hexadecimal.

8. Crea un programa que permita leer y mostrar el contenido de un archivo pasado como parámetro al llamar al programa. Para ello debes utilizar el parámetro `String[] args` de la función `main`.

9. Crea un método que reciba como parámetros dos archivos, compare su contenido e indique si son iguales o no.

10. Haz un método llamado `concat` que reciba como parámetro 2 archivos y cree un tercer fichero,

cuyo contenido, se obtendrá uniéndolo la información de los 2 archivos indicados. Primero irá el contenido del archivo recibido como primer parámetro y después el contenido del archivo recibido como segundo parámetro.

11. Haz un método llamado `concatLines` que reciba como parámetro 2 archivos y cree un tercer archivo, el contenido se obtendrá uniéndolo la información de los 2 archivos. Cada línea del archivo creado estará formado por la unión de la misma línea de los dos archivos leídos.