

Relación de ejercicios. Unidad 2. Manejo de ficheros.

- Ejercicio1.** Realiza un programa Java que muestre los ficheros de un directorio. El nombre del directorio se pasará al programa desde la línea de comandos al ejecutarlo (muestra el primer nivel).
- Ejercicio2.** Modifica el programa anterior o haz un nuevo programa que permita listar de forma recursiva los archivos y posibles subdirectorios del directorio indicado desde la línea de comandos.
- Ejercicio3.** Realiza un programa que elimine de forma recursiva un directorio. Se debe contemplar la posibilidad de que el directorio contenga subdirectorios a diferentes niveles de profundidad. El nombre del directorio se pasará al programa desde la línea de comandos al ejecutarlo.
- Ejercicio4.** Realiza un programa que liste todos los ficheros modificados entre dos fechas en un directorio dado. Con el fin de facilitar la elaboración del programa, considera las fechas y el directorio como valores dados en el programa. Para realizar el ejercicio:
- Debes crear un filtro con `FileFilter` para un intervalo de fechas, con un límite inferior y uno superior (valores tipo `long`).
 - Puedes utilizar la función `File.lastModified()` que devuelve un `long` representado los milisegundos transcurridos desde las 0:0:0:000 del 01/01/1970 hasta la fecha de modificación del fichero.
 - El método `Date.getTime()` convierte una fecha al valor `long` equivalente a los milisegundos transcurridos desde las 0:0:0:000 del 01/01/1970 hasta esa fecha.
- Ejercicio5.** Realiza un programa que añada a un fichero de texto, el contenido de otro fichero de texto.
- Ejercicio6.** Crea una aplicación donde pidamos la ruta de un fichero por teclado y un texto que queramos a escribir en el fichero. Deberás mostrar por pantalla el mismo texto pero variando entre mayúsculas y minúsculas, es decir, si escribo “Bienvenido” deberá devolver “BIENVENIDO”. Si se escribe cualquier otro carácter, se quedará tal y como se escribió. Deberás crear un método para escribir en el fichero el texto introducido y otro para mostrar el contenido en mayúsculas.
- Ejercicio7.** Crea una aplicación que pida por teclado un número de números aleatorios enteros positivos y la ruta de un fichero. Este fichero contendrá la cantidad de números aleatorios enteros positivos que se ha pedido por teclado. El rango de los números aleatorios estará entre 0 y 100, incluyendo el 100. Cada vez que ejecutemos la aplicación añadiremos números al fichero sin borrar los anteriores, es decir, si cuando creo el fichero añado 10 números y después añado otros 10 al mismo, en el fichero habrá 20 números que serán mostrados por pantalla.
- Ejercicio8.** A partir de un fichero binario de acceso aleatorio de empleados `AleatorioEmple.dat` y con estructura de registros como la que se muestra a continuación:

Relación de ejercicios. Unidad 2. Manejo de ficheros.

Apellido

Dep

Salario

Realiza un programa que introduzca datos en dicho fichero. Además realiza un programa que reciba un identificador desde la línea de comandos y visualice sus datos. Si el empleado no existe debe visualizar un mensaje indicándolo.

Ejercicio9. A partir del fichero binario de acceso aleatorio de empleados *EmpleAleatorio.dat*, realiza un programa Java que permita eliminar empleados. Para ello se hará un borrado lógico que consistirá en poner a valor -1 el id del empleado.

Ejercicio10. Crea una aplicación que almacene los datos básicos de un vehículo como la matrícula (String), marca (String), tamaño de depósito (double) y modelo (String) en ese orden. Vamos a realizar el mismo ejercicio con serialización, con los respectivos métodos get y un constructor que se invocará con todos los atributos. El atributo tamañoDeposito no se incluirá en el fichero (aun así debemos pedirlo). Los datos anteriores datos se pedirán por teclado y se irán añadiendo al fichero (no se sobrescriben los datos) cada vez que ejecutemos la aplicación. El fichero siempre será el mismo, en todos los casos.

Muestra todos los datos de cada coche en un cuadro de diálogo, es decir, si tenemos 3 vehículos mostrará 3 cuadros de dialogo con sus respectivos datos. Un ejemplo de salida de información puede ser este:

