

# UF1\_ACTIVIDAD 1.

# MANEJO DE FICHEROS

Acceso a datos

Laura Gómez Bruno

INSTITUTO TECNOLÓGICO TELEFÓNICA 2º DAM

## Contenido

Declaración de variables .....	2
Funcionamiento del código .....	2
Funcionamiento del código 2.0 .....	2
CASO 1: NO EXISTE ARCHIVO COCHES.DAT .....	2
CASO 2: EXISTE EL ARCHIVO COCHES.DAT .....	3
1.    Añadir nuevo coche .....	3
2.    Borrar coche por Id .....	3
3.    Consultar coche por Id .....	4
4.    Listado de coches .....	4
5.    Exportar coches a archivo de texto .....	4
6.    Terminar el programa .....	5

## Declaración de variables

La declaración de las variables se ha realizado en la parte superior del código. Para mostrar que efectivamente están todas declaradas se exponen a continuación en forma de captura de pantalla.

```
//Se declaran las variables necesarias
Scanner scan = new Scanner(System.in);
String read;
Coche coche;
File file = new File("./coches.dat"); //Creamos el objeto File que se corresponde con el archivo "coches.dat"
boolean gof;
FileInputStream fichero; //Clase que lee un flujo de entrada de bytes. Se utiliza junto con la clase ObjectOutputStream
ObjectInputStream buffer; //Clase que procesa los datos de entrada. Se utiliza junto con la clase FileOutputStream
FileOutputStream fos; //Clase que lee un flujo de salida de bytes. Se utiliza junto con la clase ObjectInputStream
ObjectOutputStream oos = null; //Clase que procesa los datos de salida. Se utiliza junto con la clase FileInputStream
ArrayList<Coche> lista = new ArrayList<>();
int bufferBytes;
```

Screenshot 1

## Funcionamiento del código

Antes de mostrar nada al usuario, el programa va a realizar una búsqueda del archivo *coches.dat* para que, en caso de encontrarlo, lo lea y cargue la lista precargada que se debe de encontrar dentro de dicho archivo.

Para realizar la búsqueda del archivo vamos a ver primero si existe con el if.

En caso de existir, con las clases *FileInputStream* y *ObjectInputStream* van a acceder al fichero y leer la información de su interior para almacenarse en la variable lista (de tipo *ArrayList*).

```
if (file.exists()) {
    try {
        //Se abre el fichero para leerse
        fichero = new FileInputStream(file);
        buffer = new ObjectInputStream(fichero);
        bufferBytes = fichero.available(); //Variable que almacena los bytes que quedan aun por leerse
        if(bufferBytes > 0)
            try {
                lista = (ArrayList<Coche>) buffer.readObject();
            } catch (ClassNotFoundException ex) {
                Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
            }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Screenshot 2

## Funcionamiento del código 2.0

### CASO 1: NO EXISTE ARCHIVO COCHES.DAT

Ejecutamos el programa que hemos creado y nos aparece un menú con las opciones disponibles:

```
run:
Bienvenido, seleccione una opción
1.Añadir nuevo coche
2.Borrar coche por id
3.Consulta coche por id
4.Listado de coches
5.Exportar coches a archivo de texto
6.Terminar el programa
```

Screenshot 3

## CASO 2: EXISTE EL ARCHIVO COCHES.DAT

Si existe el archivo coches.dat, simplemente lo leerá y cargará el arraylist de su interior en el array listas.

Vamos a ir ejecutando una a una las diferentes opciones:

### 1. Añadir nuevo coche

Vamos a añadir un Fiat 500 Blanco, con matrícula 1234AAA.

```
6. Terminar el programa
1
A continuación se le pedirán los datos necesarios
para añadir un nuevo coche a la Base de Datos
Añada la matrícula, por favor:
1234AAA
Añada la marca, por favor:
Fiat
Añada el modelo, por favor:
500
Añada el color, por favor:
Blanco
¿Desea añadir el coche con las siguientes características?:
Id: 1
Matricula: 1234AAA
Marca: Fiat
Modelo: 500
Color: Blanco
si
Se ha añadido satisfactoriamente
```

Screenshot 4

```
1
A continuación se le pedirán los datos necesarios
para añadir un nuevo coche a la Base de Datos
Añada la matrícula, por favor:
Seat
Añada la marca, por favor:
Ibiza
Añada el modelo, por favor:
1234
Añada el color, por favor:
1234BBB
¿Desea añadir el coche con las siguientes características?:
Id: 1
Matricula: Seat
Marca: Ibiza
Modelo: 1234
Color: 1234BBB
no
El coche no se ha añadido
```

Screenshot 5

Además, vamos a añadir un Volkswagen Beetle Rojo con la matrícula 5678BBB (e id 2) y un Seat León Negro con matrícula 7654CCC (e id 3).

```
Id: 2
Matricula: 5678BBB
Marca: Volkswagen
Modelo: Beetle
Color: Rojo
```

Screenshot 6

```
Id: 3
Matricula: 7654CCC
Marca: Seat
Modelo: Leon
Color: Negro
```

Screenshot 7

Como podemos comprobar el id se asigna de forma interna sin que el usuario tenga que añadirlo de forma manual.

También nos proporciona los datos introducidos de nuevo para comprobarlos y, en caso de marcar “no” (Screenshot 5), los datos guardados no se almacenarán.

### 2. Borrar coche por Id

Para ello Ahora borraremos el Beetle.

```
6. Terminar el programa
2
Por favor, escriba el ID del coche que se eliminar:
2
Se ha encontrado el coche y se va a proceder a eliminar...
Se ha eliminado el coche indicado
```

Screenshot 8

En caso de buscar un id que no existe para borrarlo nos mostrará un mensaje de error:

```
6. Terminar el programa
2
Por favor, escriba el ID del coche que se eliminar:
6
Lo sentimos, el id proporcionado no corresponde con ningun id existente
```

Screenshot 9

### 3. Consultar coche por Id

Vamos a consultar el coche con id 1. Cuando escribamos el id, nos mostrará los datos que hemos pedido:

```
3
Por favor, escriba el ID del coche que see consultar:
1
El coche con el ID 1 tiene:
  Matricula: 1234AAA
  Marca: Fiat
  Modelo: 500
  Color: Blanco
```

Screenshot 10

En caso de buscar un id que no existe nos mostrará un mensaje de error:

```
5
Por favor, escriba el ID del coche que see consultar:
6
Lo sentimos, el coche con el id proporcionado no existe
```

Screenshot 11

### 4. Listado de coches

Al ejecutar esta opción aparecen todos los coches junto con todas sus especificaciones.

```
4
/////////////////////////////////
El coche con el ID 1 tiene:
Matricula: 1234AAA
Marca: Fiat
Modelo: 500
Color: Blanco
/////////////////////////////////
El coche con el ID 3 tiene:
Matricula: 7654CCC
Marca: Seat
Modelo: Leon
Color: Negro
/////////////////////////////////
```

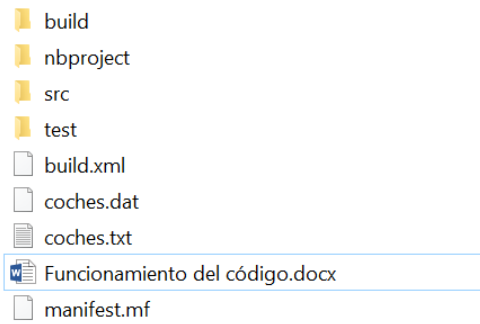
Screenshot 12

### 5. Exportar coches a archivo de texto

Cuando se elige esta opción se crea el archivo dentro la carpeta UF1\_1 (Screenshot 14) con el nombre *coches.txt* y aparecerá el siguiente mensaje (Screenshot 13):

```
6. Terminar el programa
$
Archivo creado satisfactoriamente
```

Screenshot 13



Screenshot 14

En el caso de hacer modificaciones sobre el programa (p.ej añadir un coche) y volver a crear el archivo, el archivo se sobrescribirá sobre el anterior.

## 6. Terminar el programa

Si se selecciona esta opción, el programa guarda el arraylist lista con los objetos coche que contenga en el archivo binario *coches.dat* (Screenshot 14) que servirá para que, cuando se vuelva a abrir el programa volvamos a tener todos los datos con los que habíamos trabajado anteriormente.

```
6. Terminar el programa
$
Lista de coches exportada al archivo coches.dat
BUILD SUCCESSFUL (total time: 51 minutes 32 seconds)
```

Screenshot 15