

Un taller mecánico nos ha pedido que informaticemos su gestión de taller. Para ello el programa debe llevar un registro de los mecánicos y vehículos así como de las reparaciones que se efectúan. Según nos dice el jefe de taller, una reparación es efectuada por solo un único mecánico y solo afecta a un vehículo; los vehículos se registran en el sistema la primera vez que llegan al taller y las sucesivas reparaciones se adjuntan para poder llevar un historial; lo mismo ocurre con los mecánicos, son dados de alta al entrar en el taller y es necesario guardar su historial de reparaciones. Los datos que se vaya añadiendo al programa se guardan en un fichero en disco que se carga en cada ejecución.

El programa deberá permitir desde menú:

- Consultar y listar los vehículos
- Consultar y listar los mecánicos
- Eliminar vehículos y reparaciones
- Cargar y grabar ficheros con la información de vehículos, mecánicos y reparaciones

Deberá realizarse un programa Java con arreglo a los siguientes

REQUISITOS:

1. Información a tratar

Vehículos: Se denomina así, genéricamente, a cualquier vehículo automóvil reparable en el taller. De cada uno se guarda la siguiente información:

- Matrícula. String. Actúa como identificador del vehículo
- Marca y modelo. String
- Tipo (camión, tractor, turismo, moto, etc.). String
- Nombre y apellidos del propietario. String
- Teléfono del propietario. String que puede contener espacios, puntos y guiones

Mecánicos:

- DNI. String. Identificador
- Nombre. String
- Apellidos. String

Reparaciones:

- Código. Entero. Es el identificador de la reparación
- DNI-mecánico. String
- Matrícula. String
- Fecha. String. Con formato aaaa/mm/dd
- Importe. Real. En euros
- Descripción del trabajo. String

2. Opciones de menú:

- Listar todos los vehículos **(0,5 puntos)**
 - Una fila por cada vehículo en la que aparecerán todos sus datos excepto las reparaciones
- Listar todos los mecánicos **(0,5 puntos)**
 - Una fila por cada mecánico en la que aparecerán sus datos y todas sus reparaciones.
- Realizar una reparación **(0,5 puntos)**
 - Se pide por teclado la matrícula del vehículo, el DNI del mecánico, código, fecha y descripción.
 - La reparación se añade al historial del mecánico y a la del vehículo
- Eliminar una reparación **(0,5 puntos)**

- Se pide por teclado la matrícula del vehículo y el DNI del mecánico, y el código de la reparación.
- La reparación se debe eliminar del vehículo y del mecánico
- Ver reparaciones de un vehículo. **(0,5 puntos)**
 - Se pide la matrícula por teclado
 - Se mostrará una fila por cada reparación del vehículo con toda su información incluyendo el dni del mecánico

3. Situaciones de error a controlar:

- Se considerarán errores no recuperables los problemas genéricos de entrada/salida y los de ficheros no existentes. El programa terminará de forma ordenada y mostrará un mensaje explicativo al usuario. **(0,5 puntos)**
- Son situaciones que no provocarán la terminación de la ejecución las que provengan de incoherencias (errores de lógica) en los datos suministrados por el usuario. Si hay errores en los datos suministrados por el usuario el programa no debe terminar y debe informar al usuario con un mensaje en pantalla alusivo. **(0,5 puntos)**
- Si hay errores en los datos leídos desde fichero se dejará constancia de ellos en un fichero de log pero el programa arrancará con normalidad, habiendo cargado los datos correctos y descartado los erróneos o incoherentes. **(0,5 puntos)**
- Se deben controlar como mínimo las siguientes situaciones:
 - Añadir una reparación a un mecánico y vehículo pudiendo no existir alguno de los dos **(0,5 puntos)**
 - Borrado de reparación cuyo código no existe **(0,5 puntos)**
 - Ver las reparaciones de un vehículo cuya matrícula no existe **(0,5 puntos)**

4. Carga de datos desde fichero (2 puntos)

- El programa al iniciarse cargará los datos desde fichero
- El fichero estará en formato texto plano
- Cada registro contiene información relativa a un vehículo, un mecánico o una reparación. Se usarán estos formatos de registro:

Registro Vehículo:
1@<matricula>@<marca_modelo>@<tipo>@<nombre_apellidos>@<telef>
Registro Mecánico
2@<dni_mecanico>@<nombre>@<apellidos>
Reparación Mecánico
3@<codigo>@<dni_mecanico>@<matricula>@<fecha>@<importe>@<descripcion>

```

1@1043XLH@VOLKSWAGEN POLO GP-MATCH@Turismo@MORAL ARIAS, PABLO MANUEL@860430244
2@87780714@DIEGO@ARGÜELLES PEREZ
3@1008@87780714@7213AUQ@2008/08/11@285.57@Cambio 6V (Td /Tdi)
3@1017@65459353@4762CNQ@2004/02/16@125.25@Cigüeñal
...

```

- Los registros pueden aparecer en cualquier orden pero una reparación aparecerá después de su mecánico y su vehículo
- El fichero puede contener líneas en blanco, mal formadas o con información incoherente. El programa debe descartar las líneas mal formadas o en blanco pero debe cargar las siguientes

- No se debe informar al usuario de los errores pero deben quedar registrados en un fichero de log llamado log.txt.
- Son líneas mal formadas las que :
 - Son de tipo desconocido (no empiezan por 1, 2 ó 3)
 - No tienen el número de campos correcto para el tipo de registro
 - Contienen números mal formados (p.e. "1d3,2" no es un número)
- Las incoherencias que pueden aparecer en el fichero son del estilo vehículo repetido, mecánico repetido, etc.

5. Grabación de fichero de salida: (1 puntos)

Al finalizar el programa se grabarán de forma automática los datos contenidos en memoria en un fichero .csv que tendrá el mismo formato que el de entrada. El fichero se llamará salida.csv

6. Diseño:

- Debe haber una única clase que actúe de interfaz con el usuario y:
 - Será la única que podrá leer de teclado y escribir en pantalla
 - No contendrá lógica de la aplicación
- Todas las situaciones de error deben controlarse utilizando excepciones. **(0,5 puntos)**
- La carga y grabación de ficheros debe realizarse desde una clase específica a tal fin
- Todos los incidentes producidos en la carga deben ser informados en un de log **(1 puntos)**