

## 5. UD5-PR.01: Desarrollo de clases.

La práctica por realizar para esta unidad consiste en resolver los siguiente ejercicios:

5.1	Gestión de vehículos.....	1
5.2	CONSIDERACIONES A TENER EN CUENTA .....	3
5.3	IMPORTANTE .....	4
5.4	TAREA OPTATIVA .....	4
5.5	Formato de entrega .....	4
5.6	Recursos necesarios para realizar la tarea .....	5
5.7	Consejos y recomendaciones .....	5
5.8	Materiales .....	6
5.8.1	Recursos didácticos .....	6

### Detalles de la tarea de esta unidad.

En esta quinta unidad has ido aprendiendo a crear tus propias clases así como sus distintos miembros (atributos y métodos). Has experimentando con la encapsulación y accesibilidad (modificadores de acceso a miembros), has creado miembros estáticos (de clase) y de instancia (de objeto), has escrito constructores para tus clases, has sobrecargado métodos y los has utilizado en pequeñas aplicaciones. También has tenido tu primer encuentro el concepto de herencia, que ya desarrollarás en unidades más avanzadas junto con otros conceptos avanzados de la Programación Orientada a Objetos.

Una vez finalizada la unidad se puede decir que tienes un dominio adecuado del lenguaje Java como para desarrollar tus propias clases y utilizarlas en una aplicación final que sea capaz de manipular un conjunto de datos simple. Dada esa premisa, esta tarea tendrá como objetivo escribir una pequeña aplicación en Java empleando algunos de los elementos que has aprendido a utilizar.

### 5.1 Gestión de vehículos

Se trata de desarrollar una aplicación Java denominada TallerManolo que permita gestionar los

vehículos que le llegan a reparar. Mediante un menú que aparecerá en pantalla se podrán realizar determinadas operaciones:

- Nuevo Vehículo.
- Actualizar Kilómetros.
- Ver Matrícula.
- Ver Número de Kilómetros.
- Ver años de antigüedad.
- Ver propietario.
- Ver reparación.
- Ver importe factura.
- Salir.

La funcionalidad será la siguiente:

- Al iniciar la aplicación se mostrará el menú propuesto.
- Dependiendo de la opción seleccionada por el usuario:
  - **Nuevo Vehículo:** Se creará un nuevo Vehículo, si los datos introducidos por el empleado son correctos, que contendrá la siguiente información (marca, matrícula, número de kilómetros, fecha de matriculación, reparación, nombre del propietario, dni del propietario). Todos los datos serán solicitados por teclado y tan solo habrá que comprobar:
    - Que la fecha de matriculación es anterior a la actual: puedes solicitar por separado día, mes y año y construir un objeto LocalDate (tienes una referencia en el apartado Consejos y recomendaciones).
    - Que el número de kilómetros es mayor que 0.
    - Que el DNI del propietario es correcto.
    - Si no se cumple algunas de las condiciones se deberá mostrar el correspondiente mensaje de error. En ese caso habrá se mostrará de nuevo el menú principal.
  - **Actualizar Kilómetros:** Permitirá actualizar el número de kilómetros del vehículo. Habrá que tener en cuenta que solo se podrán sumar kilómetros.
  - **Ver Matrícula:** Mostrará la matrícula del vehículo por pantalla.

- **Ver Número de Kilómetros:** Mostrará el número de kilómetros por pantalla.
- **Ver años de antigüedad:** Mostrará por pantalla el número de años del vehículo desde que se matriculó, no la fecha de matriculación.
- **Ver propietario:** Mostrará por pantalla el nombre del propietario del vehículo junto a su DNI.
- **Ver reparación:** Mostrará la reparación del vehículo, incluyendo su matrícula y el número de kilómetros que tiene.
- **Ver importe factura:** se mostrará el importe total de la factura de la reparación del vehículo.
- **Salir:** El programa finalizará.

El proyecto de Netbeans constará de dos paquetes, donde se crearán las clases oportunas:

- tallerManolo: que contendrá la clase Vehículo y la clase Principal.
- utilidades: contendrá una clase con un método estático para realizar validaciones.

La clase Vehículo dispondrá de los siguientes métodos:

- Constructor o constructores.
- Métodos get y set para acceder a sus propiedades.
- Método get\_Años(): Devuelve un entero con el número de años que tiene el vehículo de antigüedad.

## 5.2 CONSIDERACIONES A TENER EN CUENTA

- En la clase vehículo no debe solicitar datos por teclado ni escribir datos en pantalla. Esas operaciones se realizarán en la clase Principal.
- En la clase Vehículo no se deben hacer validaciones de datos. Los datos se validan en la clase Principal y si son correcto, se instancia el objeto Vehículo.
- Debes incluir una excepción para la validación del DNI. Es decir, cuando no sea válido, se lanzará una excepción que se gestionará en la clase Principal, desde donde se mostrará el correspondiente mensaje de error.
- La aplicación solo trabajará con un vehículo, por lo tanto, solo utilizará una referencia a un objeto de tipo Vehículo en la clase Principal. Si existe un vehículo y el usuario selecciona Nuevo Vehículo en el menú, se perderá la información del vehículo existente y se

guardará la del nuevo.

- No será necesario realizar comprobaciones de tipo en los datos solicitados por teclado.
- No se podrán mostrar datos de un vehículo si aún no se ha creado: en ese caso habrá que mostrar un mensaje por pantalla.

Piensa en los modificadores de acceso que debes utilizar tanto en atributos y métodos de la clase.

Además del programa deberás escribir también un informe con todas las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea.

### 5.3 IMPORTANTE

- En la cabecera de las clases añade documentación indicando autor y descripción de la clase.
- En la cabecera de cada método añade documentación indicando la funcionalidad que implementa y el valor que devuelve.
- El código fuente Java de esta clase debería incluir comentarios en cada atributo (o en cada conjunto de atributos) y método (o en cada conjunto de métodos del mismo tipo) indicando su utilidad.

### 5.4 TAREA OPTATIVA

- Incluir el código necesario para que cuando un dato solicitado no es correcto, nuestra aplicación muestre un mensaje y vuelva a solicitarlo hasta que se introduzca correctamente.

### 5.5 Formato de entrega

Se deben entregar el proyecto de Netbeans completo. Para empaquetar un proyecto en Netbeans, utiliza la opción **File - Export Project de Netbeans**: generarás un fichero .zip con el contenido completo del proyecto.



Para la entrega de esta tarea debes generar un único fichero .zip/.rar que contenga todos los ficheros anteriormente indicados y que debes nombrar de la siguiente forma:

- Ciclo-módulo: DAM-PROG-
- número de unidad (UD5)
- identificación de la práctica (PR.01)
- apellidos, seguidos de tu nombre, separados por guión bajo.

“DAM-PROG-UD5-PR.01-apellido1\_apellido2\_nombre.xxx”

Ejemplo: DAM-PROG-UD5-PR.01-Fernandez\_Lopez\_Maria.pdf



**Nota:** Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas sería:

DAM-PROG-UD5-PR.01-Sanchez\_Manas\_Begona

## 5.6 Recursos necesarios para realizar la tarea

- Ordenador personal
- Sistema operativo Windows o Linux
- Conexión a Internet
- JDK y JRE de Java
- Entorno NetBeans

## 5.7 Consejos y recomendaciones

Para realizar la aplicación te realizamos la siguiente serie de recomendaciones:

- Básate en los diferentes ejemplos que has tenido que probar durante el estudio de la unidad. Algunos de ellos te podrán servir de mucha ayuda, así que aprovéchalos.
- Puedes crear las clases que creas conveniente para llegar a una solución estructurada (por ejemplo, crear la clase DNI).

- El ejercicio resuelto de la clase DNI, en el cual se hacen comprobaciones de entrada, puede servirte de base para lanzar excepciones cuando no se cumplen ciertas condiciones. Además puedes utilizar ese código para realizar la validación el DNI del propietario del vehículo.
- En la Unidad 2, recuerda que hicimos una pequeña introducción al trabajo con cadenas de caracteres en Java.
- Si utilizas la clase Scanner, después de leer un entero lee una nueva línea para evitar errores de salto de línea. Es decir, para leer un entero siempre se deben ejecutar estas dos instrucciones:

```
valor=sca.nextInt();  
  
sca.nextLine();
```

- Para trabajar con fechas, utiliza las clases disponibles en el API Java a partir de la versión 8 (LocalDate y LocalTime). En el siguiente enlace tienes ejemplos de uso que te servirán para desarrollar la tarea.

<https://experto.dev/java-8-fechas-horas/>

<https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/time/LocalDate.html>

## 5.8 Materiales

### 5.8.1 Recursos didácticos

- Apuntes en el aula virtual.
- Ordenador personal y conexión a internet
- Manual Java IES San Clemente: <https://manuais.iessanclemente.net/index.php/Java>
- Aprenda Java como si estuviera en primero (Universidad de Navarra): <http://ocw.uc3m.es/cursos-archivados/programacion-java/manuales/java2-U-Navarra.pdf/view>
- Libro "Thinking in Java 4th edition" Bruce Eckel (Disponible en recursos del módulo)