

# Informe PROG06

Para crear el programa solicitado he dividido la tarea en tres partes.

- Estructura inicial
- Implementación de todos los requisitos
- Comprobación y corrección

## Estructura inicial

Inicialmente se generan las cuatro clases (`Principal`, `Concesionario`, `Vehiculo` y `Validar`) y se introducen los datos básicos en cada una de ellas.

En `Vehiculo` se aprovecha lo creado en la UD5 añadiendo dos métodos:

- `getDatosResumido()` devuelve únicamente una cadena de texto con la matrícula, la marca y el precio del vehículo
- `getDatosExtendido()` devuelve, además de lo anterior, el kilometraje y la descripción del vehículo.

En `Concesionario` se incluyen las variables `listaVehiculos`, que es un array de vehículos, `numeroVehiculos`, que llevará la cuenta de cuántos vehículos hay creados.

En `Validar` se reutilizan métodos de la UD5 adaptándolo a las instrucciones de la entrega, por lo que la validación del DNI del titular se hará mediante patrón en el método `esDniCorrecto()`; se añaden también la validación mediante patrón en `esMatriculaCorrecta()` y mediante métodos de la clase `String` en `esNombreCorrecto()`.

En `Principal` se escriben las instrucciones para instanciar el concesionario. Para mostrar el menú se emplea un método propio, así como para las distintas opciones del mismo.

## Implementación de todos los requisitos

En esta parte se verifican todos los requisitos del enunciado.

Primeramente, se crean todos los métodos necesarios para el concesionario: `buscarVehiculo()`, `insertarVehiculo()`, `listarVehiculos()` y `actualizarKms()`, así como un único constructor que contendrá la creación del array de vehículos limitado a 50 y el contador de los mismos iniciado en 0.

En la clase `Validar`, todos los métodos devolverán únicamente verdadero o falso, por lo que edito los métodos creados en la unidad anterior para seguir el mismo esquema. También elimino el método `obtenerEnteroPorTeclado()` que validaba lo introducido por

teclado dividiéndolo en dos: `esNumeroEntero()` para los números enteros y `esNumeroDecimal()` para el precio de los vehículos.

En `Principal` desarrollo todos los apartados del menú, optando en la mayor parte de los casos por métodos propios para cada uno de los apartados. En algunos casos opto por subdividirlo, de forma que `pedirFechaMatriculacion()` se encargue de la solicitud de datos y `nuevoVehiculoFechaMatriculacion()` de comprobar que la fecha cumple los requisitos y puede ser almacenada en el vehículo que se está creando.

Relacionado con lo mencionado previamente del método `obtenerEnteroPorTeclado()`, se crean los métodos `enteroPorTeclado()` y `doublePorTeclado()` para solicitar números enteros o decimales.

En `Concesionario`, la principal tarea novedosa fue el trabajo con los arrays. Me obligó a modificar en numerosas ocasiones los bucles para evitar encontrarme con `NullPointerException` ya que intentaba acceder a vehículos que todavía no estaban creados, por lo que en todos ellos empleo bucles `while` para salir de ellos sin necesidad de recorrer todos los vehículos.

Por último, opto por incluir la opción adicional de permitir eliminar un vehículo almacenado mediante el método `eliminarVehiculo()` creado en la clase `Concesionario`. En este caso, primeramente el bucle `while` localiza el vehículo a eliminar y el bucle `for` reposiciona los vehículos almacenados en las siguientes posiciones del array.

## Comprobación y corrección

Una vez creado el programa, se han ejecutado diversas comprobaciones introduciendo datos erróneos para verificar que el comportamiento es el correcto.

Al comprobar la validación de los datos introducidos, evitando que salten excepciones, observo que el método `esNumeroDecimal()`, creado inicialmente con patrón, devuelve un `NumberFormatException` al lidiar con números con coma en ellos. Es por ello que opto por validar el dato mediante `Double.parseDouble()` y capturando la excepción en caso de producirse.

Indicar que en el código se incluye lo que considero que es una mejora desde el punto de vista del usuario (que en este caso también era yo): al introducir una matrícula en las opciones de buscar el vehículo o modificar el kilometraje, la matrícula introducida puede ir con las letras en minúscula ya que empleo previamente `toUpperCase()` para que conste como mayúscula igualmente. Este método también se ha empleado al almacenar el DNI del propietario del vehículo.

Una vez comprobado todo, se añadió la documentación requerida, se comentó el código y se escribió este informe.