



Segunda actividad entregable obligatoria

Concesionaria de autos y motos.

Desarrollar un programa con Java que simule el funcionamiento de una concesionaria de vehículos. Al ejecutarse, la aplicación deberá generar exactamente la siguiente salida por consola y finalizar su ejecución inmediatamente después:

Marca: Peugeot // Modelo: 206 // Puertas: 4 // Precio: \$200.000,00
Marca: Honda // Modelo: Titan // Cilindrada: 125c // Precio: \$60.000,00
Marca: Peugeot // Modelo: 208 // Puertas: 5 // Precio: \$250.000,00
Marca: Yamaha // Modelo: YBR // Cilindrada: 160c // Precio: \$80.500,50

=====

Vehículo más caro: Peugeot 208
Vehículo más barato: Honda Titan
Vehículo que contiene en el modelo la letra 'Y': Yamaha YBR \$80.500,50

=====

Vehículos ordenados por precio de mayor a menor:
Peugeot 208
Peugeot 206
Yamaha YBR
Honda Titan

=====

Vehículos ordenados por orden natural (marca, modelo, precio):
Marca: Honda // Modelo: Titan // Cilindrada: 125c // Precio: \$60.000,00
Marca: Peugeot // Modelo: 206 // Puertas: 4 // Precio: \$200.000,00
Marca: Peugeot // Modelo: 208 // Puertas: 5 // Precio: \$250.000,00
Marca: Yamaha // Modelo: YBR // Cilindrada: 160c // Precio: \$80.500,50

La solución debe cumplir con los siguientes requisitos:

- Diseñar una solución orientada a objetos, creando por lo menos una clase abstracta y al

menos una interfaz.

- Se debe hacer uso del framework Collections. Y API Stream.
- La salida es por consola y exactamente como se requiere.
- Sobrescribir el método toString().
- Se debe implementar la interfaz Comparable al menos en una clase.
- La lista de vehículos se debe cargar en un único método. No se debe solicitar ningún ingreso por pantalla.
- El algoritmo de impresión debe ser genérico, es decir, no debe depender de la cantidad, modelo o tipo de vehículos.
- Se espera que el proyecto se cree con Spring Boot. Que se gestione con Maven. Y que se utilicen herramientas Lombok para reducir el código repetitivo y seguir buenas prácticas de desarrollo.
- El entregable será la URL de un repositorio GitHub público. Será deseable que el repositorio incluya un README explicativo sobre la estructura del proyecto, las herramientas utilizadas, las instrucciones de ejecución y cualquier otra información que consideren pertinente.

Observación: El uso de buenas prácticas en el diseño, claridad del código, separación de responsabilidades, documentación y legibilidad serán considerados criterios fundamentales para la evaluación del trabajo. El mismo deberá defenderse en clase, respondiendo las consultas del docente y pudiendo explicar lo que se hizo en código.

Fecha de entrega 22/10/25