

## **Simulador De Comportamiento De Automóvil Con Excepciones**

María Paz Puerta Acevedo

Heily Yohana Rios Ayala

Universidad Autónoma

Programación Orientada a Objetos

Carlos Andres Rojas

1/04/2025

Manizales, Caldas, Colombia

## Objetivos

- Utilizar en el diseño e implementación de la solución los conceptos fundamentales de la programación orientada a objetos.
- Utilizar Netbeans como herramienta para el desarrollo de la solución computacional.
- Utilizar las facilidades de lectura y escritura de archivos que provee Java para implementar la persistencia de la aplicación.
- Manipular las excepciones para el paso y control de mensajes de error en la aplicación.

## Grupo de trabajo

El taller se realizará en grupos de dos estudiantes.

## Enunciado

Se desea desarrollar un simulador teórico de conducción de un vehículo simple. Para esto se utilizará Java como lenguaje de programación y el paradigma orientado a objetos para el diseño e implementación de la solución. La solución propuesta obedecerá a los siguientes requerimientos.

El **vehículo** podrá realizar las siguientes acciones:

- Encender
- Apagar
- Acelerar
- Frenar
- Frenar bruscamente

Las siguientes serán condiciones para la conducción del vehículo:

- Un vehículo encendido no se puede encender de nuevo.
- Un vehículo apagado no se puede apagar de nuevo.
- Un vehículo apagado no se puede acelerar o frenar.
- Un vehículo detenido no se puede frenar, su acción no surte efecto.
- Si se frena bruscamente a un vehículo cuya velocidad supere el límite permitido por sus llantas, el vehículo patinará.
- Si se frena a un vehículo con una intensidad mayor a la de su velocidad actual, el vehículo patinará.
- Si se apaga el vehículo y su velocidad es superior a los 60 Km/h, este se accidenta.
- Si se acelera el vehículo más allá de la capacidad de su motor, este se accidenta.
- Si un vehículo patina, este pierde el control y sólo se recupera cuando se ha detenido por completo.
- Si un vehículo se accidenta, este no sólo se detiene sino que se apaga.

Las acciones de frenado o aceleración se expresarán en la magnitud de los km/h en que afectarán la velocidad actual del vehículo. Una aceleración o frenado con una magnitud mayor a 30 km/h será considerada brusca.

Las **llantas** y el **motor** del vehículo podrán ser configuradas en el taller mediante la especificación de un archivo de texto que será leído por la aplicación de simulación al inicio de su ejecución. Cada característica será especificada en una línea independiente como se muestra en el siguiente formato del archivo de configuración.

llantas TIPO

motor CILINDRAJE

Los posibles *tipos de llantas y la velocidad máxima permitida para estas (más allá de la cual pueden patinar)* se muestran a continuación.

Nombre	Límite Permitido
--------	------------------

Buenas	110 Km/h
--------	----------

Bonitas	70 Km/h
---------	---------

Baratas	50 Km/h
---------	---------

Los posibles *cilindrajes del motor y sus velocidades máximas* se muestran a continuación.

Nombre	Velocidad Máxima
--------	------------------

1000 cc	100 Km/h
---------	----------

2000 cc	160 Km/h
---------	----------

3000 cc	220 Km/h
---------	----------

De esta manera, el siguiente deberá ser el contenido del archivo de configuración para construir un vehículo con llantas bonitas y motor de 3000 cc.

llantas Bonitas

motor 3000

## Observaciones

1. La aplicación deberá mantener informado al usuario acerca de las variables del vehículo y de los sucesos que han acontecido durante su conducción.
2. Los mensajes presentados al usuario deberán ser completamente informativos indicando lo que ha sucedido.
3. Para la implementación técnica de los sucesos se deberá utilizar excepciones propias desarrolladas por los estudiantes.
4. Para leer el archivo de especificación del vehículo, utilice las clases para lectura y escritura de archivos de texto vistas en clase o utilice la clase RandomAccessFile.

## **Entregables**

Enlace de repositorio remoto Github publicado en UAMVirtual con el siguiente contenido:

1. El modelo de clases incluyendo: clases (con atributos y métodos), relaciones y tipos de clases.
2. Listado de excepciones utilizadas con una descripción de su significado.
3. El proyecto de Netbeans con la solución computacional.

## 1. Identificación de clases

- Vehiculo:

### Atributos

☐ velocidadActual: int

### Métodos

- ☐ encenderAuto(): void
- ☐ apagarAuto(): void
- ☐ acelerarAutor(): void
- ☐ frenarAutor():void
- ☐ frenarAutoBruscamente(): void
- ☐ mostrarEstado(): void
- ☐ estaConfigurado() : boolean

- Llanta

### Atributos

tipoLlanta: String

velocidadMaxima : int

### Métodos

- ☐ verificarVelocidadMaxima(cantidad : int) : void
- ☐ verificarTipoLlanta() : void

- Motor

### Atributos

- ☐ tipoMotor: String
- ☐ velocidadMaxima: int
- ☐ estaEncendido : boolean

**Métodos**

- ☐ encender() : String
- ☐ apagar() : String
- ☐ mostrarEstado() : String
- ☐ estaEncendido() : boolean

- Simulador  
a través del simulador debe gestionar el taller

**Atributos****Métodos**

- ☐ iniciar(ruttaArchivo : String) : void

- Taller

**Atributos****Métodos**

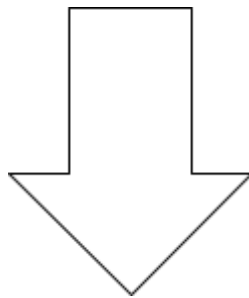
- ☐ configurarVehiculo(rutaArchivo : String) : void
- ☐ guardarConfiguracion() : void
- ☐ crearLlantaPorTipo(tipo : String) : Llanta
- ☐ crearMotorPorTipo(tipo : String) : Motor

- Lector  
configurar vehículo a partir de lector
- Escritor  
si el usuario modifica el carro llama a escritor

**2. Lista excepciones**

- **IOException:** Ocurre cuando no se puede escribir o leer un archivo de manera correcta.
- **NullPointerException:** Ocurre cuando se intenta acceder a un objeto que es null o tiene ese valor.
- **Exception:** Manejar errores que puedan ocurrir en tiempo de ejecución sin que el programa se caiga.
- **VehiculoApagadoException:** Intentar realizar acciones en un vehículo apagado.

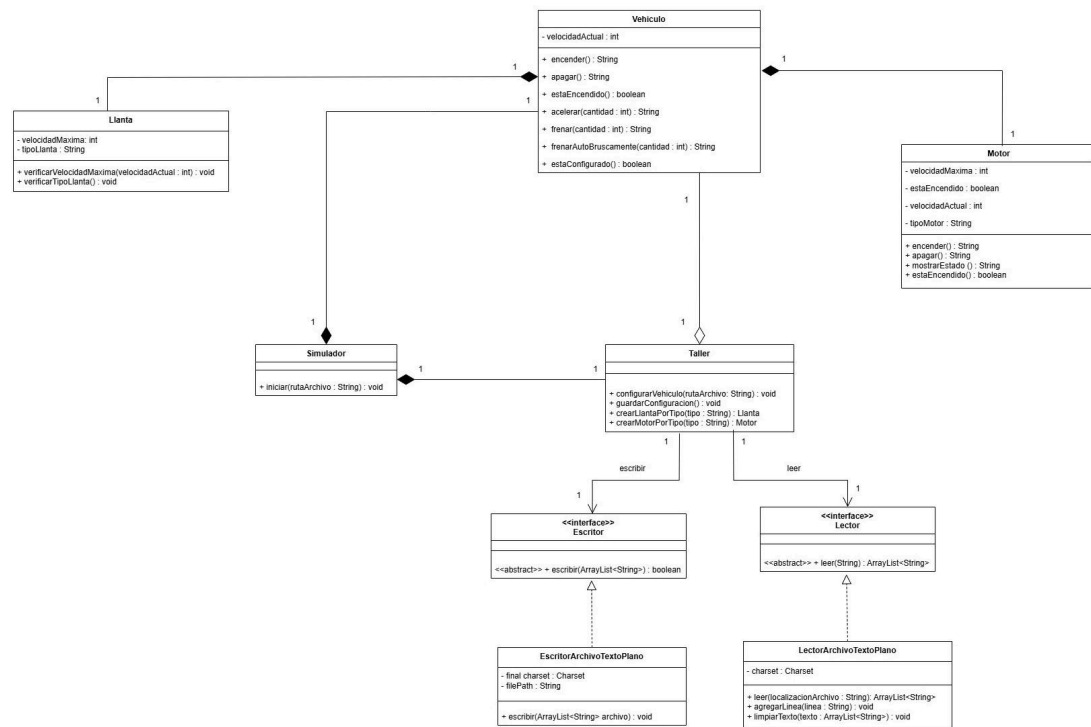
- **VehiculoEncendidoException:** Intentar encender un vehículo que ya está encendido.
- **VehiculoDetenidoException:** Intentar frenar un vehículo que ya está detenido.
- **LimiteVelocidadExcedidoException:** Intentar superar el límite de velocidad del vehículo o sus llantas.
- **AccidentePorApagadoException:** Intentar apagar el vehículo con velocidad superior a 60 km/h.
- **AccidentePorAceleracionException:** Intentar acelerar el vehículo más allá de su capacidad máxima.
- **PatinajeException:** El vehículo patina debido a un frenado brusco o exceso de velocidad.
- **KilometrajeInsuficienteException:** Intentar calcular el desgaste sin haber recorrido suficientes kilómetros.
- **TerrenoIrregularException:** Intentar recorrer un terreno irregular con llantas inapropiadas.
- **ErrorEnArchivoConfiguracionException:** Error al leer o interpretar el archivo de configuración.
- **VehiculoNoConfiguradoException:** Cuando se intenta realizar cualquier funcionalidad del simulador sin configurar el vehículo antes.
- **IllegalArgumentException:** Cuando un argumento no es válido, como un número negativo.







### 3. Diagrama de clases



4. Enlace GitHub: <https://github.com/heily0118/SimuladorAutomovilApp.git>

5. Enlace del video explicativo:

[https://drive.google.com/file/d/1JcU9lA0fO5l9XSjRmSC6DsDInug-G4aj/view?usp=drive\\_sdk](https://drive.google.com/file/d/1JcU9lA0fO5l9XSjRmSC6DsDInug-G4aj/view?usp=drive_sdk)