



# Tecnológico de Monterrey

***Programación orientada a objetos***

**Grupo:**

301

**Curso:**

TC1033

**Profesor:**

Roberto Martínez Román

**Equipo #8**

Cesar Ulises Miranda Ortega      A01750481

Propuesta de solución a “Simulador de Automóvil”

Fecha de entrega: 16 junio de 2023

# Tabla de contenidos

Introducción y objetivos.....	3
Diagrama de clases UML .....	4
Capturas de ejecución .....	5
Argumentación de código .....	8
Casos de error .....	12
Conclusiones y comentarios finales.....	12

## Descripción del problema

Se requiere construir un simulador del comportamiento de un Automóvil. El automóvil debe ser capaz de ejecutar tareas básicas como prender, apagar, acelerar, frenar, prender luces, apagar luces. La información del estado del auto se puede representar en un tablero con indicadores de prendido/apagado, velocidad actual, nivel del tanque de gasolina y luces prendidas/apagadas. Cuando se pide al auto que acelere la velocidad aumenta en 15 km/h y la gasolina disminuye 0.005 litros multiplicado por la velocidad actual. Cuando se pide al auto que frene la velocidad disminuye en 25 km/h. La máxima velocidad del auto es 230 km/h. Cuando la velocidad es mayor a 160 km/h se muestra un indicador de peligro. Cuando la gasolina es menor a 15% aparece un recordatorio para llenar el tanque. El tanque de gasolina Tiene un valor constante de 42 litros.

La aplicación debe contener un menú y mostrar, al menos, las siguientes opciones.

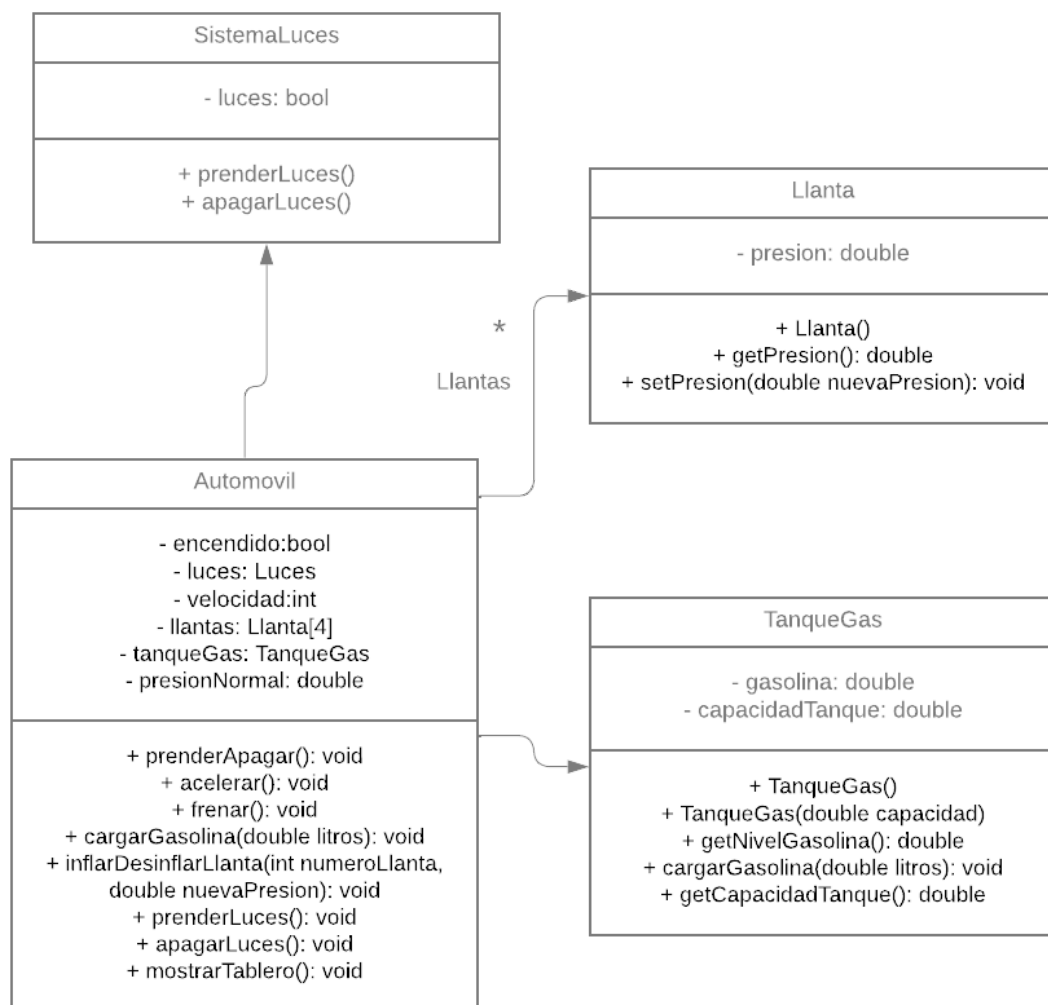
1. Prender/Apagar el auto. Una sola opción, si está apagado lo prende y viceversa.
2. Acelerar.
3. Frenar.
4. Prender luces.
5. Apagar luces.
6. Cargar gasolina. El usuario teclea los litros que va a cargar.
7. Salir.

Al iniciar el programa, se muestra el tablero y después de cada operación se vuelve a dibujar. El auto inicia con el tanque lleno, apagado y con las luces apagadas. Debes validar las entradas de usuario (por ejemplo, los litros a cargar no deben ser negativos o sobrepasar la capacidad del tanque). También el estado del auto no puede pasar de 230 km/h o disminuir por debajo de 0 km/h. Y, por supuesto, no puede acelerar/moverse si no tiene gasolina.

1. Dibuja el diagrama de clases que representa el problema. Aplica el método gramatical.
2. Codifica las clases en C++.
3. Elabora el plan de pruebas para validar el correcto funcionamiento de tu programa.
4. Escribe la función main que realiza la simulación completa.
5. Al inicio del programa imprime tu nombre, matrícula y carrera.

CAMBIOS DE ÚLTIMA HORA. 1. Agrega al problema las llantas del auto: una clase Llanta y un arreglo de 4 llantas en el lugar adecuado. 2. El único atributo de las llantas es la presión en psi. Asigna un valor por default de 33, pero proporciona métodos get/set para modificarla. 3. Agrega al tablero un indicador de la presión de las llantas, tú decides la forma de representar el estado de las 4 llantas, tomando en cuenta que, si están un 20% por arriba o por debajo de la presión normal, significa un estado de peligro. 4. Agrega al menú principal la opción de inflar o desinflar llantas, el programa te pregunta cuál de las 4 llantas y el nuevo valor de presión

Diagrama de clases



Conclusiones:

Al reflexionar sobre mi experiencia previa en la carrera de IID, puedo concluir que haber tomado las materias más avanzadas de Programación Orientada a Objetos (POO) y Estructuras de Datos antes de llegar al nivel inicial de POO ha sido revelador. Ahora me doy cuenta de cuánto me hubiera facilitado las cosas si hubiera inscrito los conceptos fundamentales de POO desde el principio.

Cuando comencé el curso avanzado de POO, me encontré con conceptos más complejos y aplicaciones prácticas que ampliaron mi comprensión de la programación orientada a objetos. Luego, el curso de Estructuras de Datos me brindó un enfoque práctico sobre cómo organizar y manipular información de manera eficiente. Ambas experiencias me han preparado para enfrentar el curso inicial de POO con una perspectiva más completa y una base sólida.

En el nivel inicial de POO, estoy volviendo a revisar los conceptos básicos que quizás pasé por alto en mis cursos anteriores. Ahora puedo apreciar cómo estos conceptos sentaron las bases para todo lo que he aprendido hasta ahora. Me doy cuenta de la importancia de comprender los principios de la encapsulación y la herencia, así como la aplicación de estos conceptos en la resolución de problemas.

Revisar el plan de pruebas para más capturas de pantalla.

```
----- TABLERO -----  
Estado del automovil: Encendido  
Velocidad actual: 5 km/h  
Nivel de gasolina: 41.775 litros  
Las luces estan: Apagadas  
  
Presion de las llantas:  
Llanta 1: 33 psi  
Llanta 2: 33 psi  
Llanta 3: 33 psi  
Llanta 4: 33 psi  
----- MENU -----  
1. Prender/Apagar el auto  
2. Acelerar  
3. Frenar  
4. Cargar gasolina  
5. Inflar/Desinflar llanta  
6. Prender luces  
7. Apagar luces  
8. Salir  
Seleccione una opcion: 8
```

Saliendo del programa

```
----- TABLERO -----  
Estado del automovil: Apagado  
Velocidad actual: 0 km/h  
Nivel de gasolina: 42 litros  
Las luces estan: Apagadas  
  
Presion de las llantas:  
Llanta 1: 33 psi  
Llanta 2: 33 psi  
Llanta 3: 33 psi  
Llanta 4: 33 psi  
-----MENU-----  
1. Prender/Apagar el auto  
2. Acelerar  
3. Frenar  
4. Cargar gasolina  
5. Inflar/Desinflar llanta  
6. Prender luces  
7. Apagar luces  
8. Salir  
Seleccione una opcion: 
```

Entrando al programa