

Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Estado de México

**Situación Problema**

Unidad de formación: TC1033 Pensamiento computacional orientado a objetos

Grupo: 301

Francisco Tonatihu Castro Flores

Nombre del profesor: Roberto Martínez Román

**Descripción del Problema**

Esta actividad será un programa computacional  que proponga una solución a la situación problema de la UF. Como parte de la actividad, se diseño un diagrama de clases que modela la propuesta de solución que se elaboro y posteriormente, se implemento la solución en C++ utilizando el paradigma orientado a objetos.

Para la implementación del proyecto, se utilizaron los estándares de codificación básicos para C++ que se han revisado en la UF y además, se diseño un plan de pruebas para asegurar la calidad del programa.

Las instrucciones fueron las siguientes:

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 mulJplicado 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 Jene 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 Jene 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.

Los cambios de última hora fueron los siguientes:

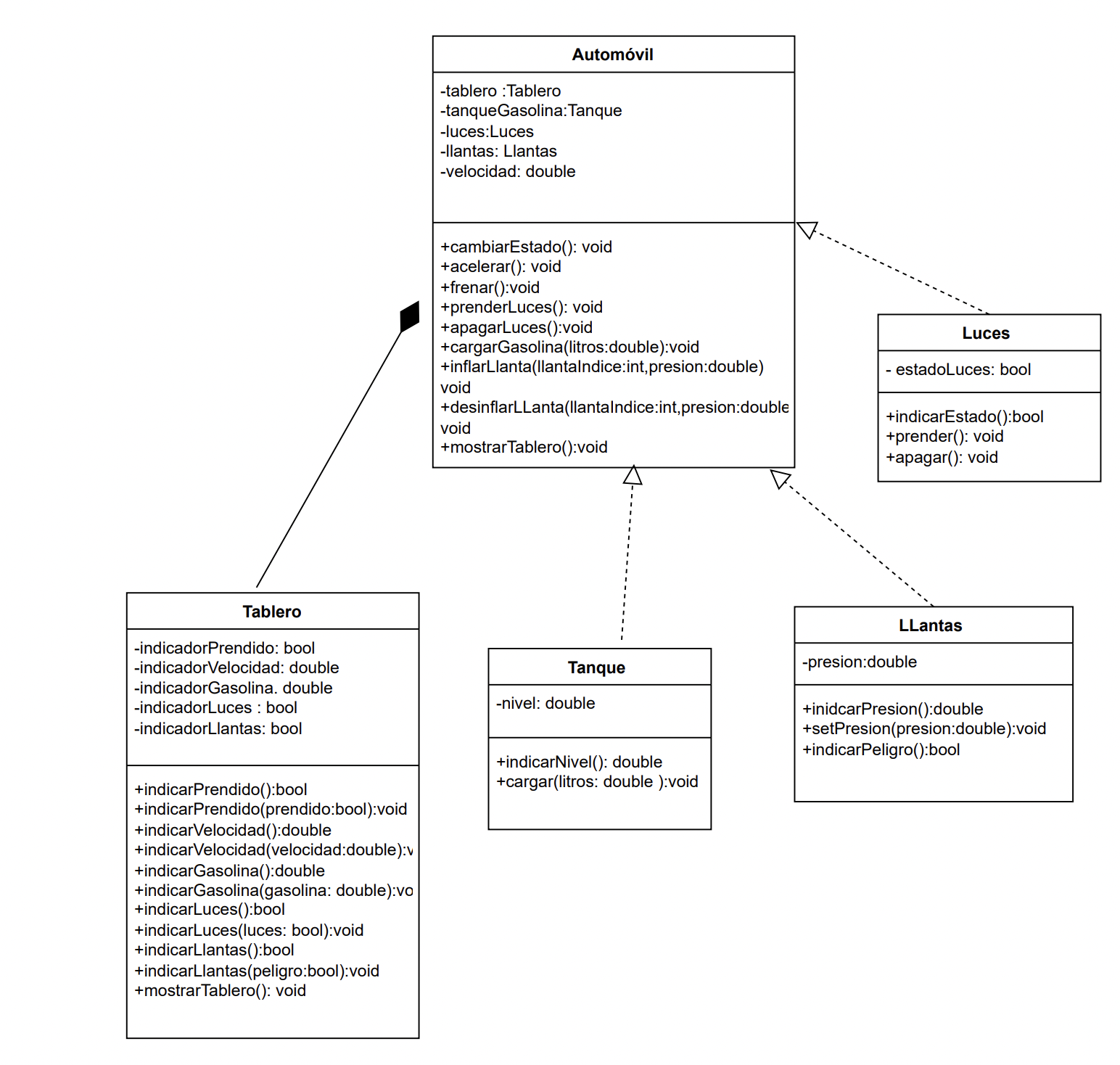
1. Agrega al problema las llantas del auto: una clase Llanta y un arreglo de 4 llantas en el lugar adecuado.

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**

****

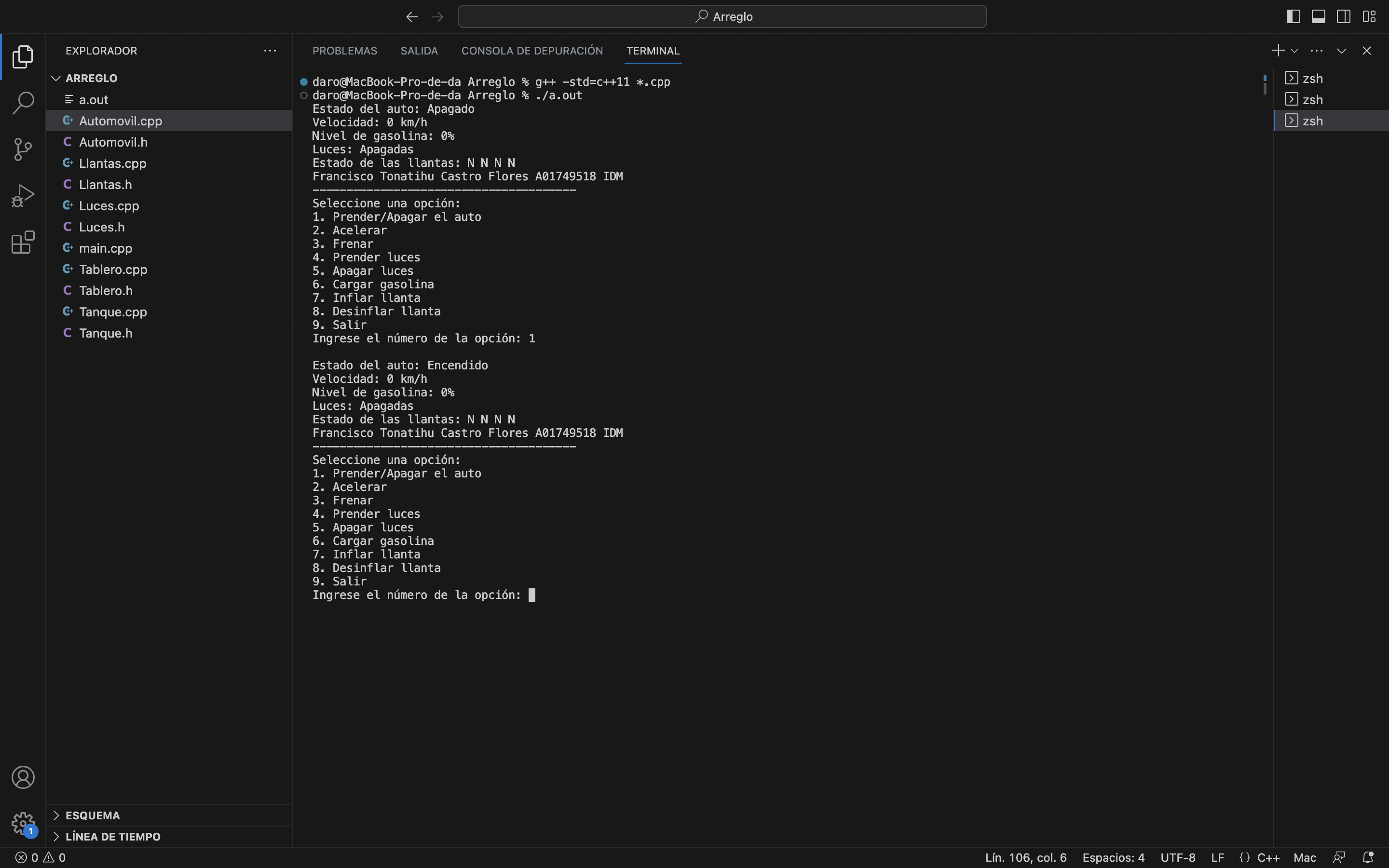
**Plan de pruebas**

**Prueba de encendido**

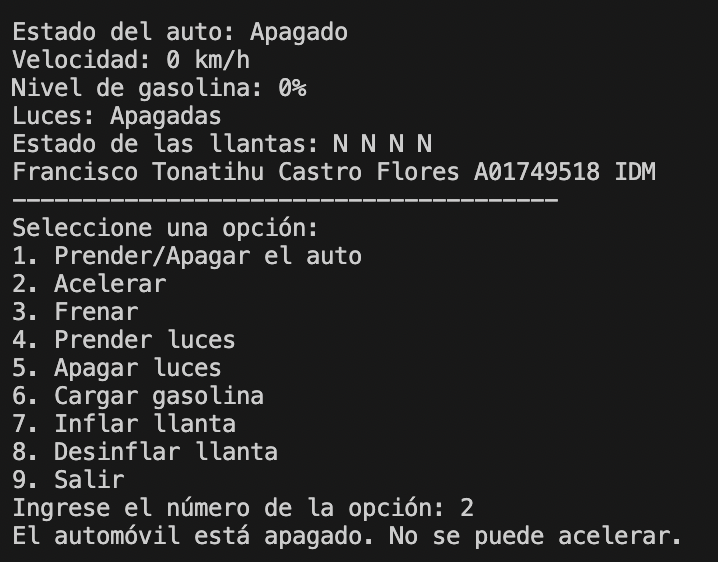
Primero comprobaremos que el auto encienda correctamente el auto solo enciede tecleando 1, caso contrario se mostrara un mensaje sobre que el auto no esta encendido. Si se teclea un numero mayor a 9 o menor a 1 se mostrara un mensaje que diga que es icorrecto.

|  |  |  |
| --- | --- | --- |
| Casos | Datos de entrada | Resultados |
| 1 | 1 | Enciende |
| 2 | 2 | No enciende |
| 3 | 90 | No enciende |

Caso1:



Caso 2:



Caso 3:

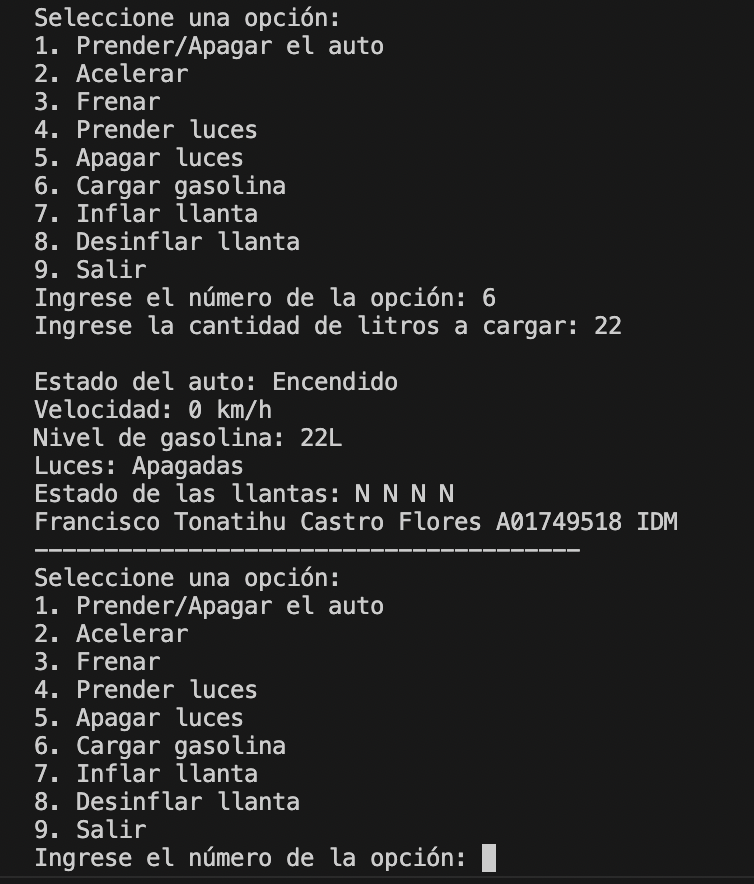


**Prueba de carga de gasolina**

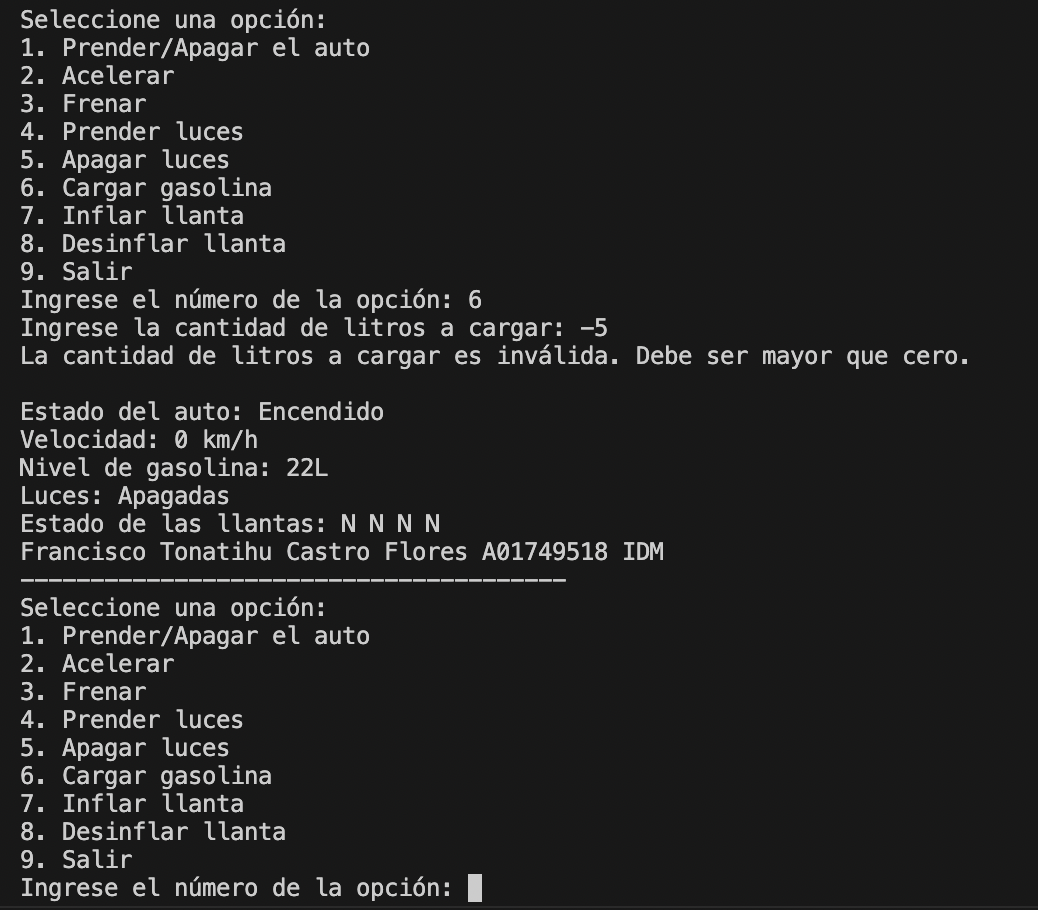
Una vez encendido el auto veremos que el tanque esta sin gasolina lo que haremos será seleccionar la opción “Cargar gasolina” y seleccionaremos una cantidad valida por ejemplo 22 y verificaremos que llene el tanque y muestre la cantidad elegida. Después seleccionamos nuevamente la opción y, pero esta vez llenamos con un numero como -5, y por último un numero como 43 para comprobar que no llene más de la capacidad del tanque.

|  |  |  |
| --- | --- | --- |
| Casos | Datos de entrada | Resultados |
| 1 | 22 | 22 |
| 2 | -5 | Invalido |
| 3 | 43 | Invalido |

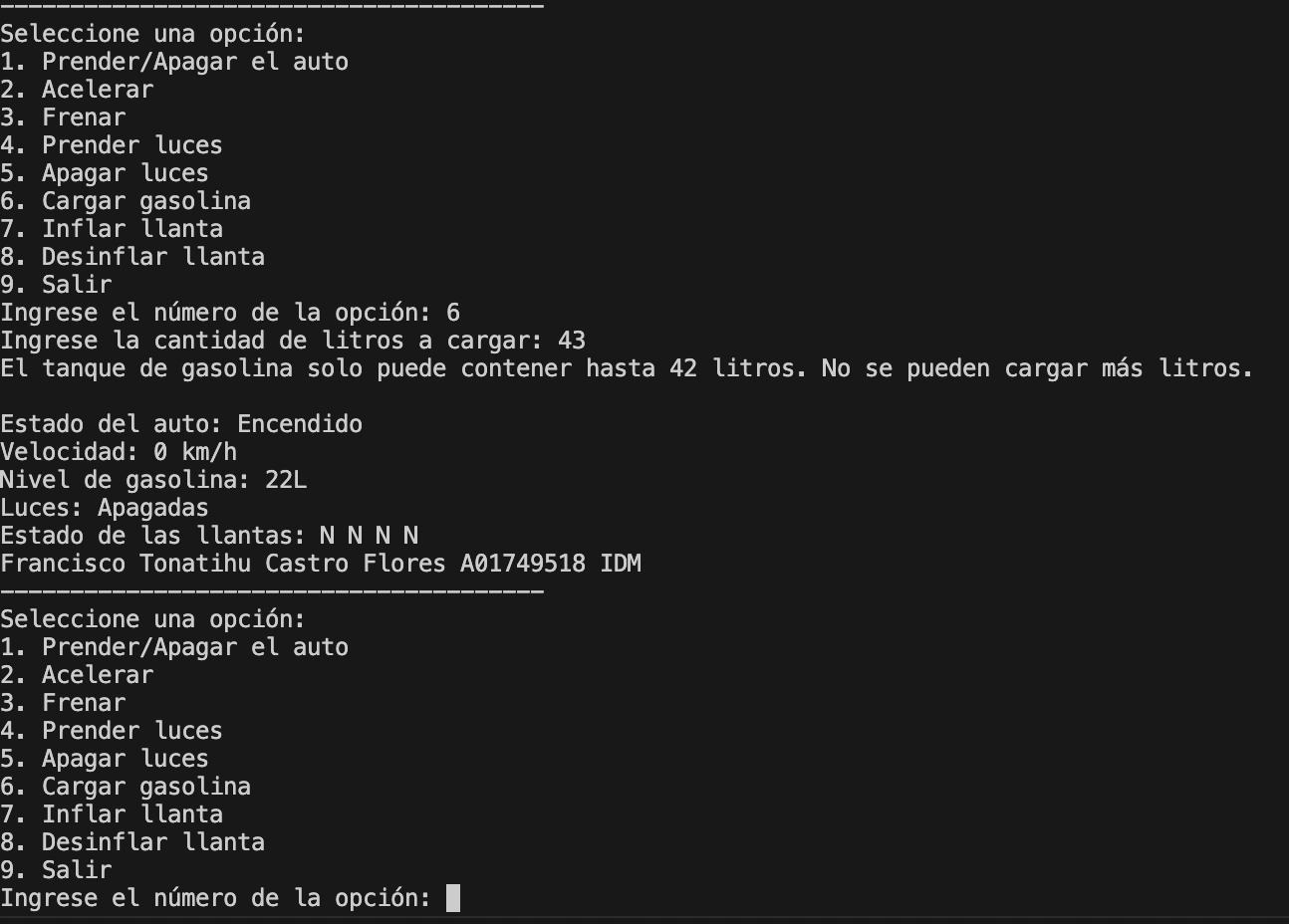
Caso 1:



Caso 2:



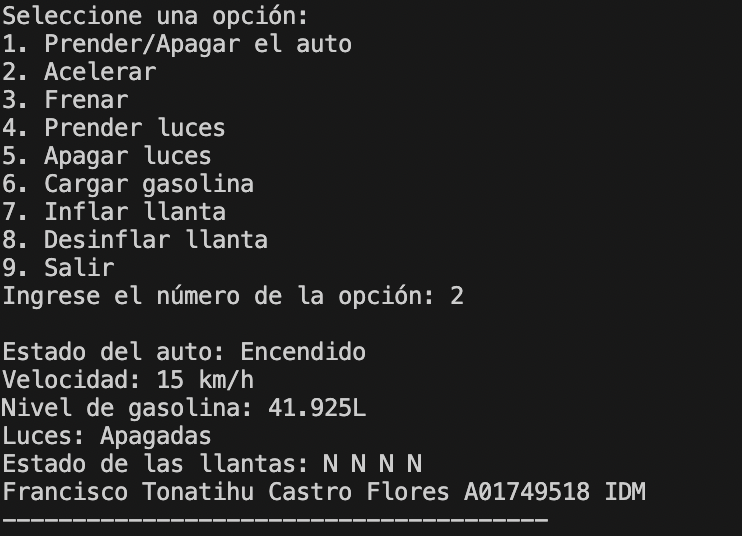
Caso 3:



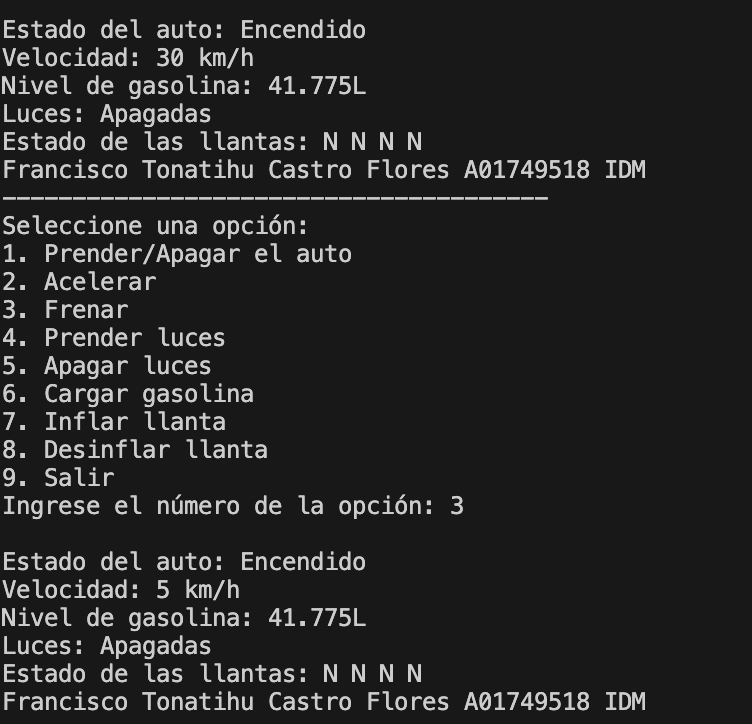
**Prueba de Acelerado y frenado:**Ahora probaremos la opción acelerar, primero se debe realizar todo lo anterior desde iniciar el auto y llenar el tanque después seleccionamos la opción acelerar y comprobamos que el auto aumente la velocidad. Seleccionamos frenar y vemos que la aceleración disminuya.

|  |  |  |
| --- | --- | --- |
| Casos | Datos de entrada | Resultados |
| 1 | 2 | +15 km/h |
| 2 | 3 | -25 km/h |

Caso1:



Caso2:

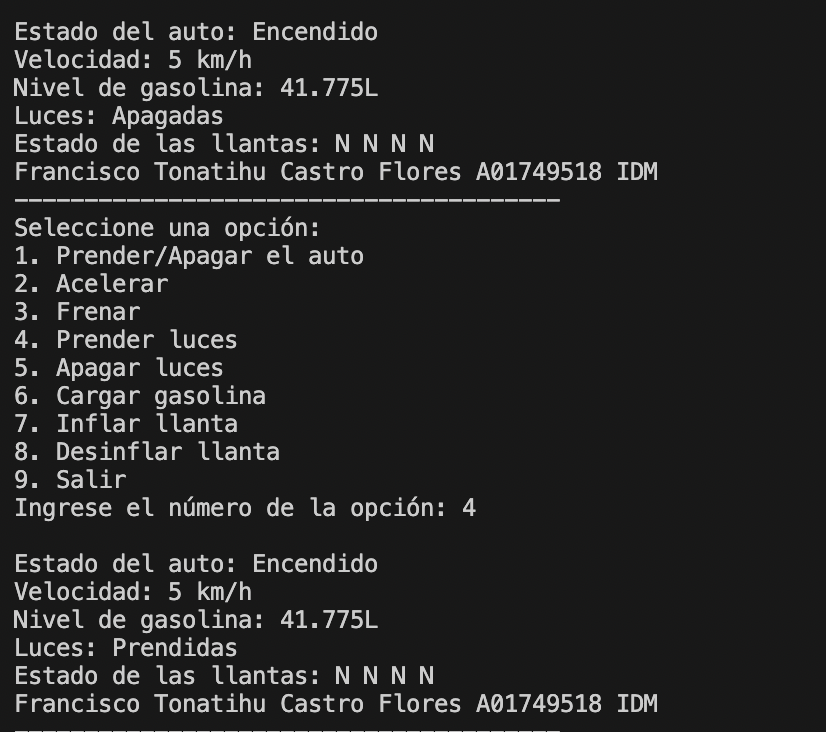


**Prueba de encendido y apagado de luces**

Con el auto encendido comprobamos que las luces funcionen.

|  |  |  |
| --- | --- | --- |
| Casos | Datos de entrada | Resultados |
| 1 | 4 | Luces Prendidas |
| 2 | 5 | Luces Apagadas |

Caso 1:



Caso 2:



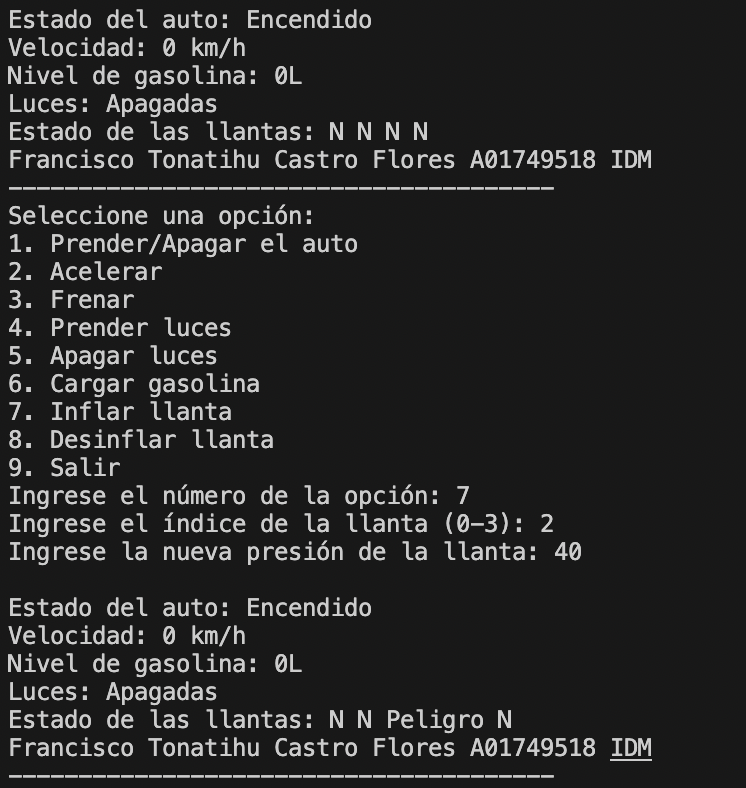
**Prueba de inflado y desinflado de llantas**

Ahora probaremos las llantas. Primero con el auto encendido elegiremos la opción “Inflar llanta” (7), elegimos una llanta al azar, ya que todas las llantas funcionan de la misma manera, e ingresamos la presión que deseemos a la llanta en este caso será el 40.

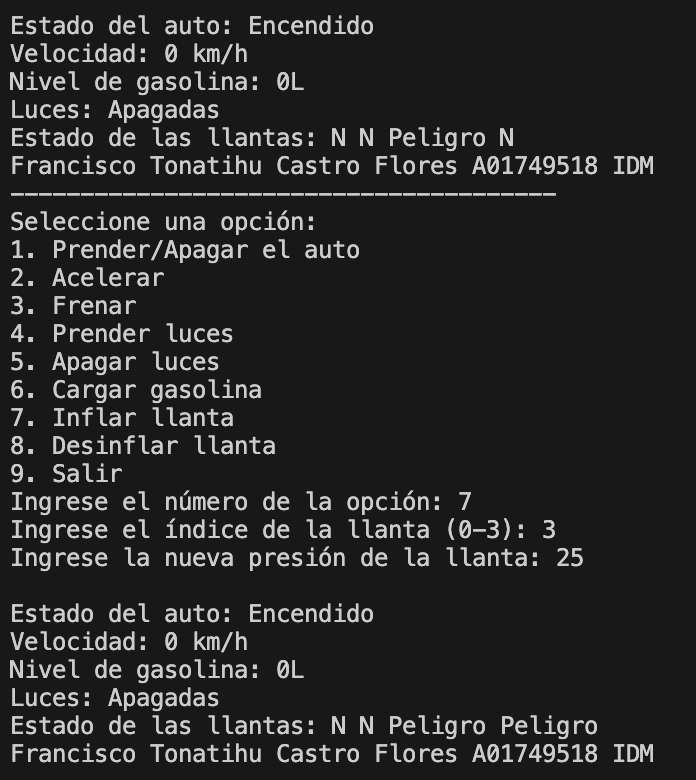
Repetimos el proceso, pero esta vez con otra llanta, tecleamos el 25, y por último el 38.

|  |  |  |
| --- | --- | --- |
| Casos | Datos de entrada | Resultados |
| 1 | 40 | Peligro |
| 2 | 25 | Peligro |
| 3 | 38 | N |

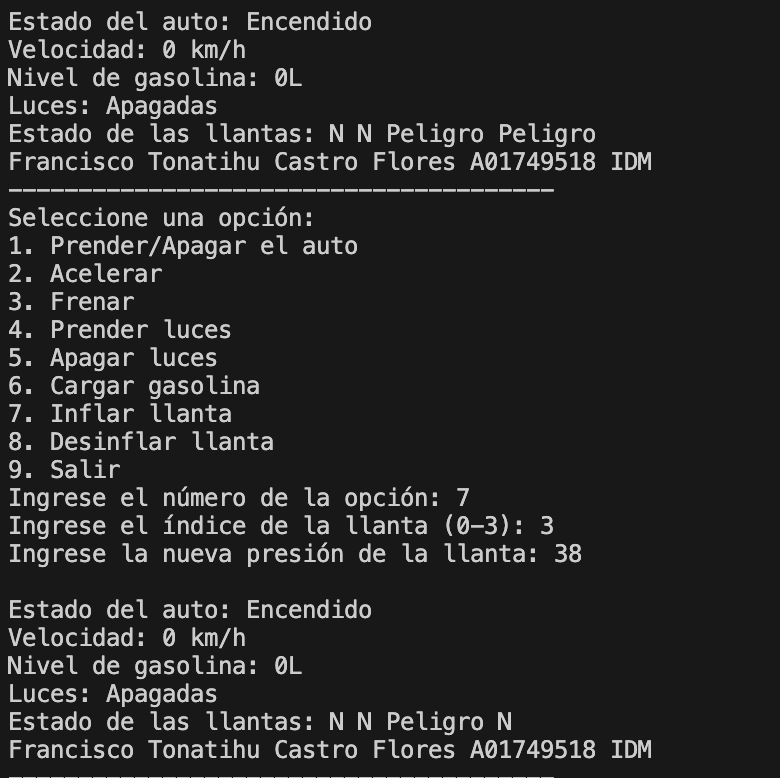
Caso 1:



Caso 2:



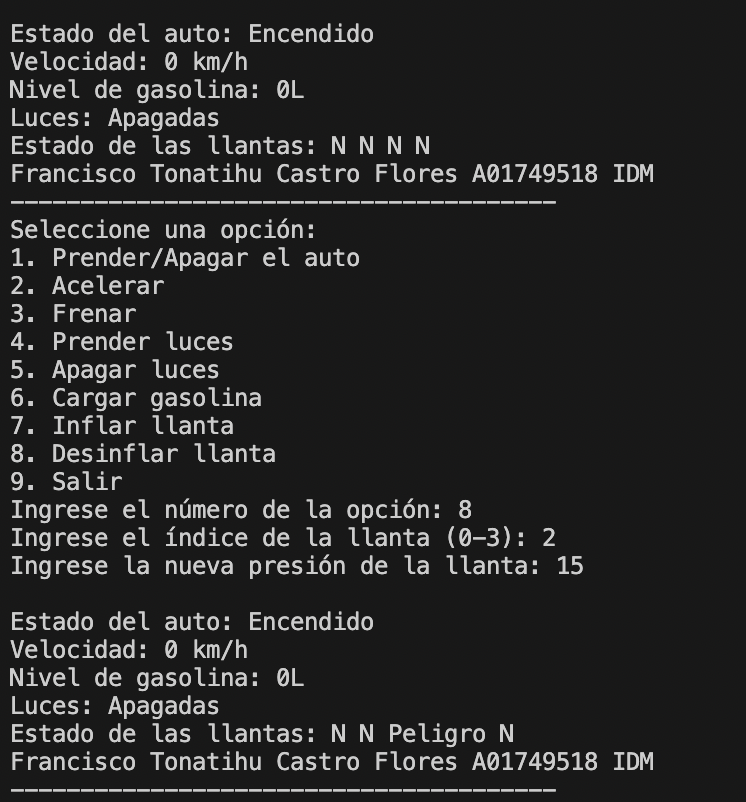
Caso 3:



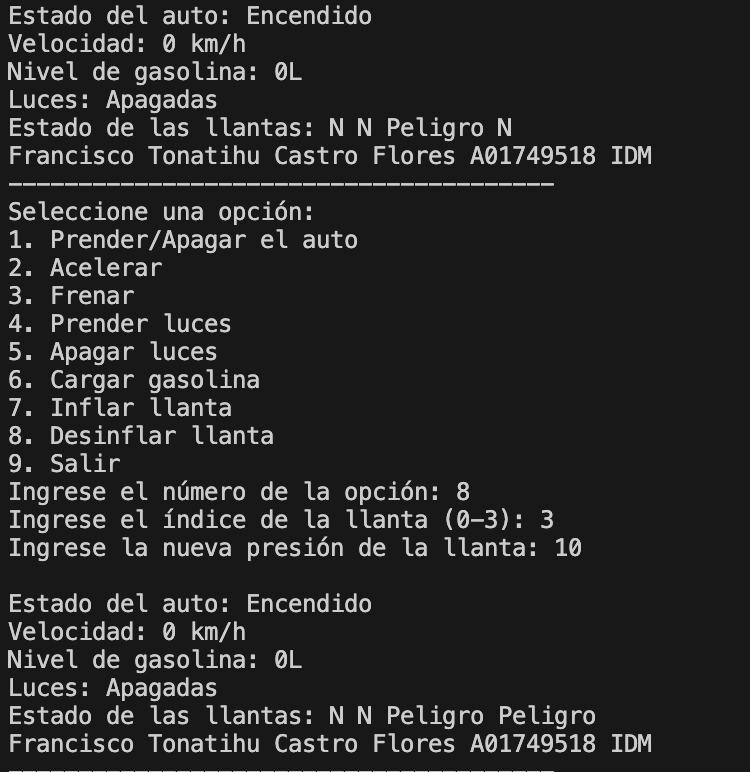
Ahora probaremos desinflar las llantas, para esto seleccionamos la opción “Desinflar llanta” (8) seleccionamos una llanta y le introducimos un valor en este caso será el 15, luego con otra llanta introducimos el 10 y con otra el 35.

|  |  |  |
| --- | --- | --- |
| Casos | Datos de entrada | Resultados |
| 1 | 15 | Peligro |
| 2 | 10 | Peligro |
| 3 | 35 | N |

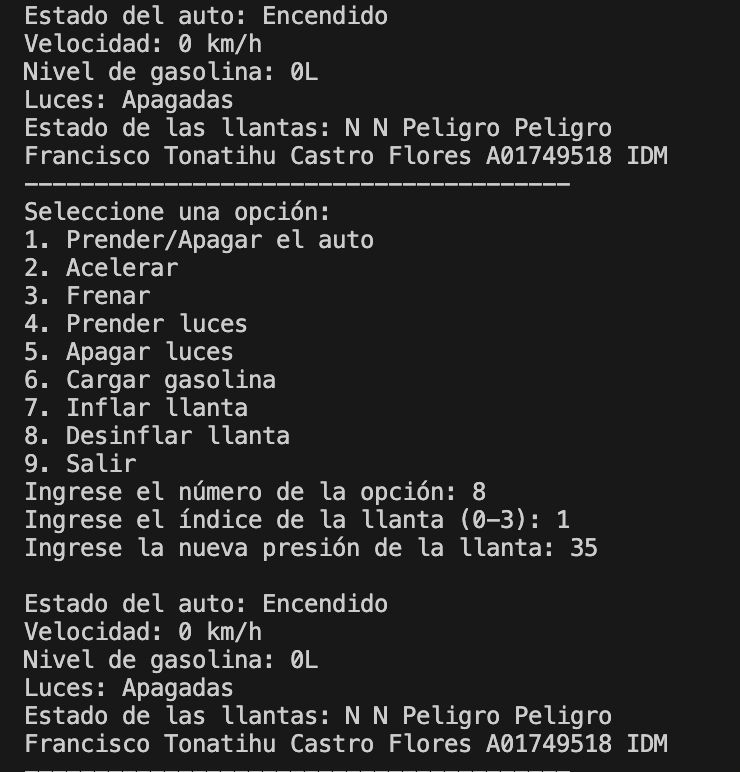
Caso 1:



Caso 2:



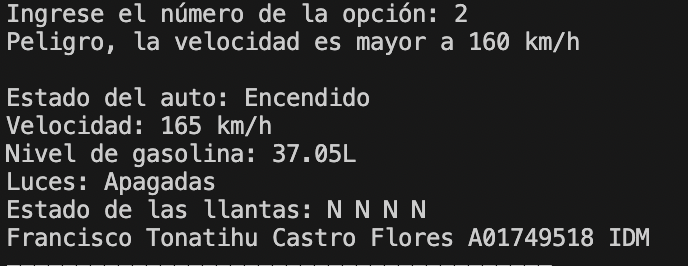
Caso 3:



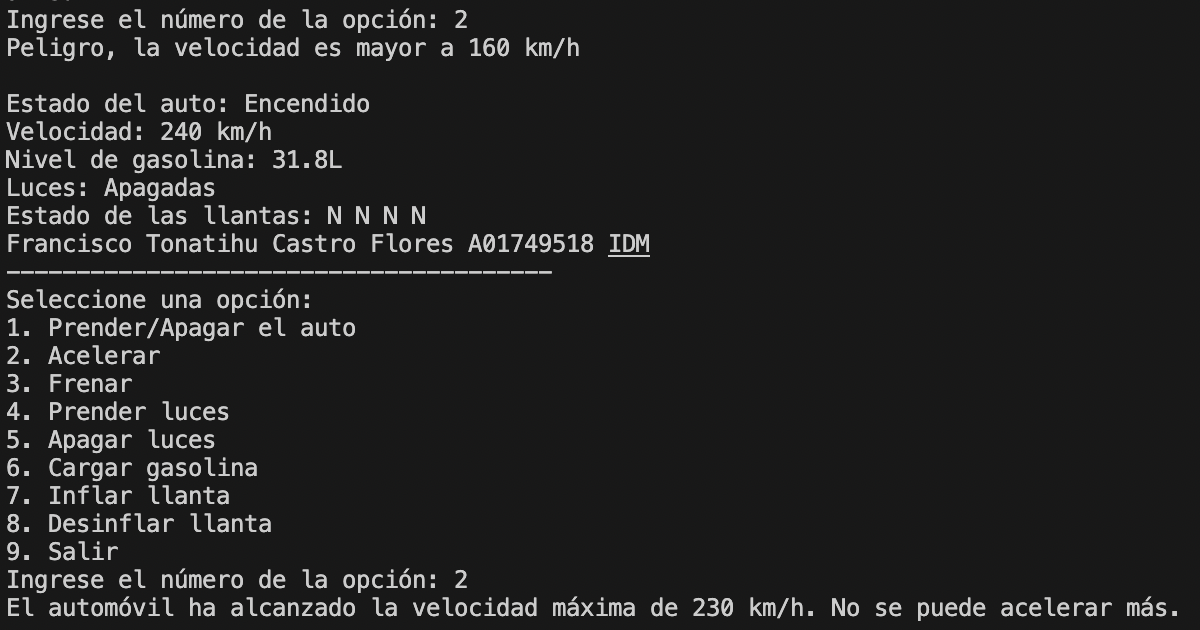
**Prueba de límites de velocidad**

Para probar los limites de velocidad se selecciona acelerar hasta que se llegue al límite de velocidad, antes de eso en velocidades mayores a 160 aparecerá un indicador de peligro y al llegar a 230 se alcanzara el límite de velocidad.

Advertencia de peligro:



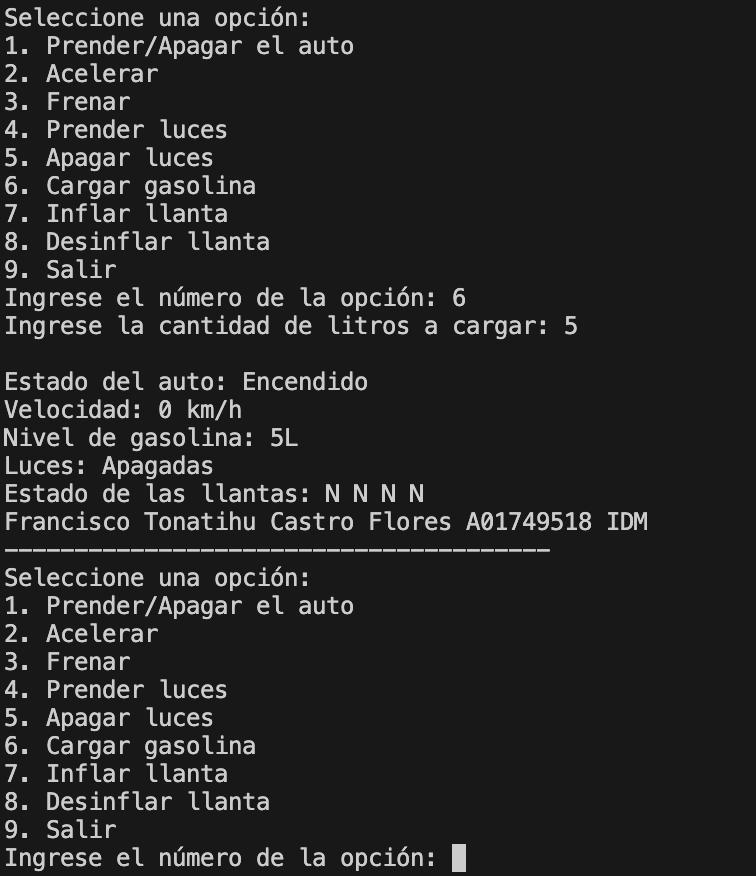
Límite de velocidad:



**Prueba de nivel de gasolina**

Para probar que el nivel de gasolina funcione se reducirá la gasolina hasta cero acelerando y frenando cuando se haya alcanzado el límite o bien se puede pedir iniciar de nuevo el programa, ya que se empieza con 0L de gasolina. Una vez que la gasolina este en cero se recarga el tanque con un nivel meno al 15% de su capacidad

Prueba:



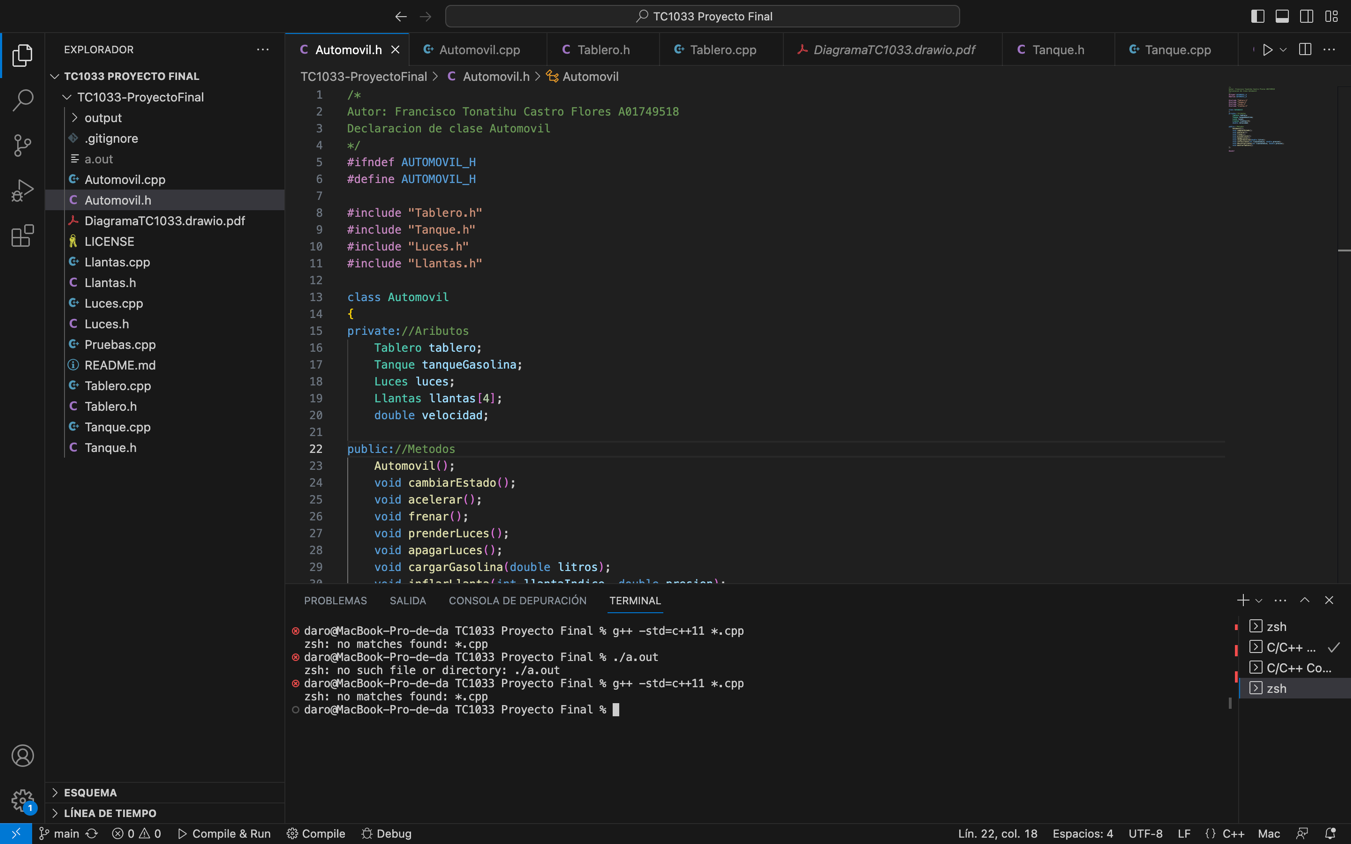
Como vemos el programa tiene un problema para identificar que la gasolina es menor al 15%, esto debido a que no se logró implementarle esta función de manera correcta.

Conclusiones:

Realizar este proyecto fue un arduo recorrido que me dejo muchas enseñanzas y me permitió aplicar lo aprendido durante el curso, así como conocimientos adquiridos de manera externa. Hubo muchas partes difíciles de implementar las cuales fueron, principalmente, el cpp de automóvil y de tablero los cuales fueron los que requirieron más tiempo e incluso no se pudieron lograr cumplir todos los objetivos como lo son el nivel de gasolina y que el auto no diferencie bien entre inflar y desinflar una llanta ya que no pude implementar los métodos set y get correctamente, por lo que opte por otras alternativas. Gracias al diagrama de clases se pudo tener una guía que ayudara diseñar el código. En cuanto a la codificación y mi aprendizaje puedo decir que fue un problema complejo en el que adquirí conocimiento y experiencia y me creo nuevas dudas e incertidumbres, principalmente sobre lo que no pude solucionar en mi código a pesar de intentar varias soluciones y métodos, sin embargo, gracias a esto aprendí la importancia de tener un orden a la hora de programar, un plan de pruebas y recursos que puedas consultar cuando no puedas avanzar más.

**NOTA IMPORTANTE.**

El código en no se puede compilar cuando lo haces desde el proyecto “TC1033 Proyecto Final”. Desconozco porque pasa esto y ocurrió desde que borre archivos debido a los errores que marcaba, a partir de ahí no pudo compilar en ese proyecto por lo que opte trabajar fuera de este y en otro si pudo compilar, anexo en la entrega el archivo.zip de donde si compilo el cual contiene exactamente el mismo código que la carpeta original y que se encuentra en Github, también anexo las capturas de lo que me muestra en mi caso al intentar compilar.



Referencias:

Microsoft. (2023). Operadores integrados de C++, precedencia y asociatividad. Recuperado de: <https://learn.microsoft.com/es-es/cpp/cpp/cpp-built-in-operators-precedence-and-associativity?view=msvc-170>

IBM. (2023). Definicon de simbolos en C y C++. Recuperado de: <https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/design-rhapsody/9.0.1?topic=symbols-defining-in-c-c>