

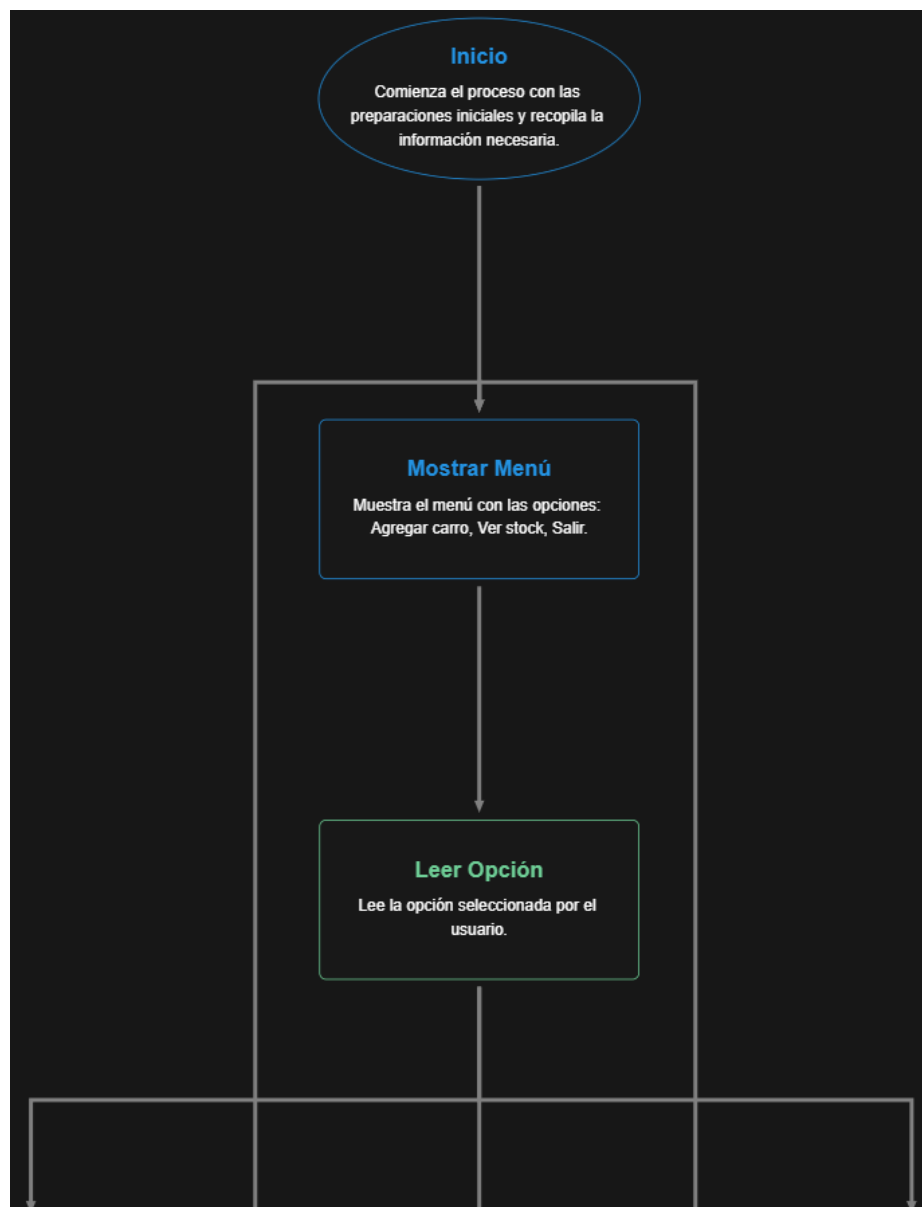
“Actividad 1”

Programación Orientada a Objetos

“Profesor, Carlos Iván Castillo Sepúlveda”

José Antonio Vázquez Mora A107087360

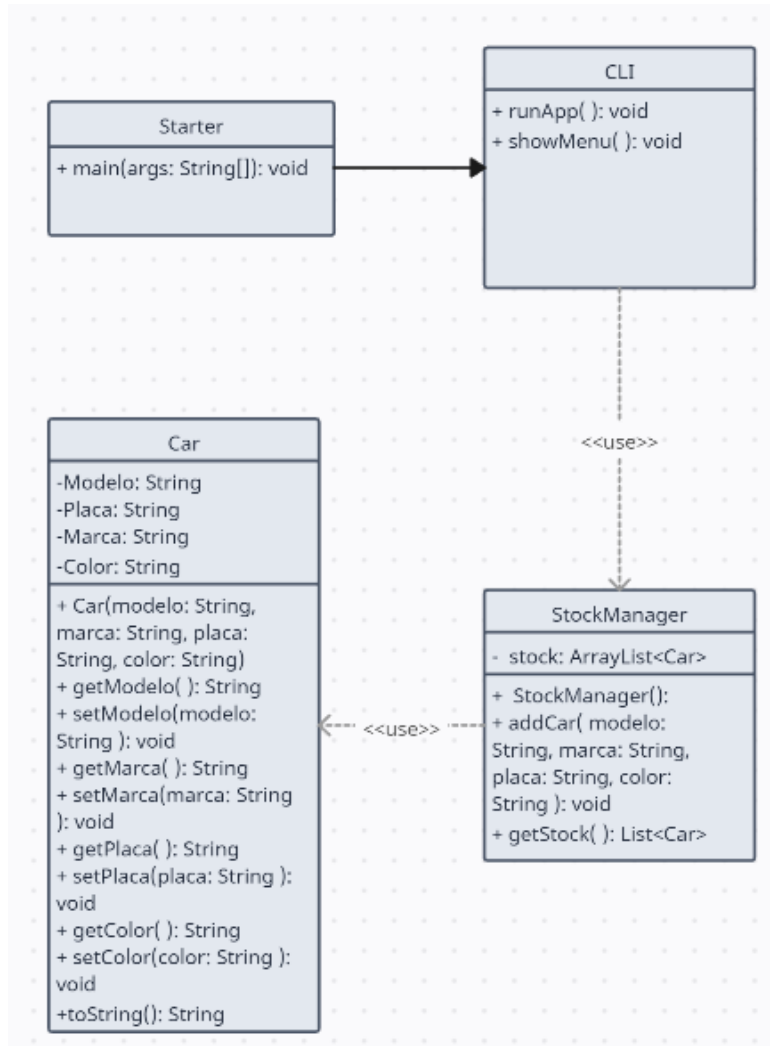
En el presente documento resolvimos un ejercicio presentado por nuestro profesor al momento de querer tener un registro de automóviles y poder consultarlo dentro de un menú interactivo, es importante señalar que utilizamos el paradigma de programación orientada a objetos para crear objetos simples y trabajamos con la línea de comandos de nuestra computadora para tener interacciones con el usuario. Vamos a utilizar diagramas UML para mostrar la lógica que tendrá nuestro programa a continuación:





Aquí presentamos el diagrama de flujo de nuestra clase CLI, como se puede observar todo el programa comienza con la inicialización de la clase, se manda a llamar el método showMenu para mostrar el que será el menú de nuestro programa, podemos observar que se mostrará en la línea de comandos y tendremos 4 posibles casos a partir de este momento, en caso de que el usuario ingrese una opción invalida del menú se mostrará un mensaje de opción inválida y volverá a mostrarte el menú. En caso de elegir la primera opción que es agregar un carro se le solicitará al usuario una serie de parámetros, en este caso se le solicita Modelo, Color, Placa y Marca, al momento de tener todos los datos guardados vamos a llamar el método addCar que se encarga de almacenar la información obtenida dentro de una lista vacía llamada stock. En caso de que se elija la segunda opción y solo quieras ver los registros almacenados se mandará a llamar el método getStock que

mostrará la lista con todos los elementos que tenga almacenados, por último, cuando quieras cerrar el programa solo tendrás que ingresar la opción número 3 que procederá a terminar con la ejecución del código.



Aquí se puede observar el diagrama de clases que tenemos desarrollado en nuestro código, como se puede observar la clase Starter es nuestra clase main, desde ahí vamos a correr la aplicación, como se puede observar depende de la clase CLI que es la encargada de mostrar todo lo que corresponde a diseño para el usuario, así mismo, CLI depende de StockManager ya que utiliza los métodos addCar y getStock para poder mostrarlo en

pantalla, dicho esto, también existe una dependencia entre StockManager y nuestra clase Car, dentro de esta clase es donde tenemos el procesamiento de todos nuestros parámetros, aquí es donde vamos a tener el modelo, la placa, el color y la marca del automóvil.

Como se puede observar en nuestro trabajo utilizamos 4 diferentes clases para realizar nuestra tarea, es importante separar cada clase con los métodos que van a utilizar, esto va a permitir que otros programadores puedan entender a simple vista lo que estamos intentando hacer junto con la documentación necesaria de los métodos que creamos, es importante señalar que nos guiamos bajo las convenciones de nomenclatura propias del lenguaje java para hacer nuestro documento. Creo que en este caso pudimos aprender bastante de como pueden interactuar las clases entre ellas para tener procesos mucho más limpios y sin la necesidad de estar creando una y otra vez los métodos que necesitamos en cada clase, por lo contrario, solo los creamos en uno y los reutilizamos utilizando los paradigmas orientados a objetos.