|  |  |
| --- | --- |
| **Nombre:** Alejandro Corvera Monroy | **Matrícula:** AL07098211 |
| **Nombre del curso:** Programación orientada a objetos | **Nombre del profesor:** Carlos Iván Castillo Sepúlveda |
| **Fecha:** sábado 18 de enero 2025 | Actividad 1 |

# Actividad 1

## Introducción

En esta actividad introductoria a la materia de la programación orientada a objetos, lo que se busca es mostrar la implementación de un repositorio de GitHub, utilizando un código real para demostrarlo. En este caso, se utiliza el ejemplo de un menú para ver las opciones de un auto, todo esto utilizando los métodos propios de la programación orientada a objetos.

## Elaboración de la práctica

La forma en la que funcionará esta clase durante todo este ciclo es teniendo un repositorio de nuestros trabajos en GitHub, ya que esto permite que tengamos múltiples copias de cualquier código que vayamos a estar trabajando. La forma en la que funciona es que se crea un código principal (código Master o código Main), y cuando se quieran realizar cambios, estos tienen que ser hechos en un subcódigo (Branch). Una vez que se terminan estos cambios, se tienen que subir, y cada vez que haya cambios, estos no reemplazan al código Main.

Como se mencionó en la introducción del trabajo, el código busca crear un menú, en el que un usuario puede interactuar con el programa y elegir qué tipo de auto quiere tener. La forma en la que esto funciona es utilizando el lenguaje de programación Java. Este lenguaje está enfocado en clases y en objetos.

Una vez estando en el código, se tiene que crear una carpeta en la que irá esta actividad. Habiendo hecho esto, dentro de esa carpeta, lo siguiente es crear cuatro carpetas, que serán el lugar en donde irán todas las partes del código. Estas tomarán los nombres de “classes”, “models”, “process” y “ui”. Cada una de estas es una parte del código, y terminarán trabajando entre sí para que todo funcione correctamente. En cada una de estas carpetas se tiene que crear algo que sea un ejemplo de esta, como en el caso de la “ui”, es el apartado de la interfaz del usuario, por lo que ahí es donde iría el apartado de la CLI (Command Line Interface / Interfaz de líneas de comandos).

Lo primero es que cada uno de estos apartados tiene una ruta, con la cual se identificarán. En el caso del código de la CLI, se vería como: *package edu.alejandro.corvera.actividades.actividad1.ui; .* La última parte es la que está indicando que viene de la interfaz del usuario. Y para hacer que se comuniquen entre sí, se hace uso de la función de importar. Con ella, se pueden mandar a llamar a los demás apartados para utilizar sus funciones.

Cuando se completa el código, lo que sigue es enviar la copia en la que hemos estado trabajando (la branch), ya se puede mandar de nuevo al GitHub.