

Alumno	Marco Antonio Hernández Velazco
Actividad	1
Carrera	Ingeniería en Desarrollo de Software
Materia	Programación Orientada a Objetos
Nombre del Profesor	Carlos Iván Castillo Sepúlveda

Reporte

En este inicio de semestre llevaremos acabo la materia de Programación orientada a objetos y no hay mejor manera de iniciarla que con una actividad algo desafiante para mí, ya que no estoy familiarizado aún con los lenguajes de programación.

Ahora trabajaremos con Java, antes solo habíamos trabajado con Python, junto con GitHub para crear un repositorio en el cual colocaremos nuestros trabajos de Visual Studio Code.

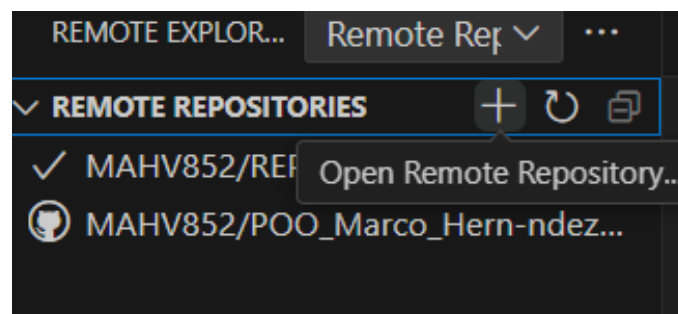
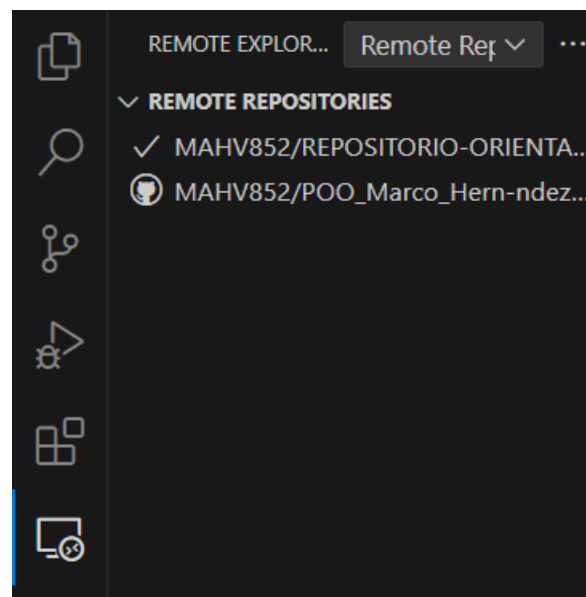
El trabajo de ahora es algo sencillo, un menú donde se subirán algunos datos para posteriormente mostrarlos desde el mismo menú.

Mi equipo escogió hacerlo sobre autos, subir las características de los autos como la marca, modelo, año, etc. La creación del código se me volvió a complicar como en el semestre pasado con Python pero no fue impedimento para crearlo y que cumpla su propósito.

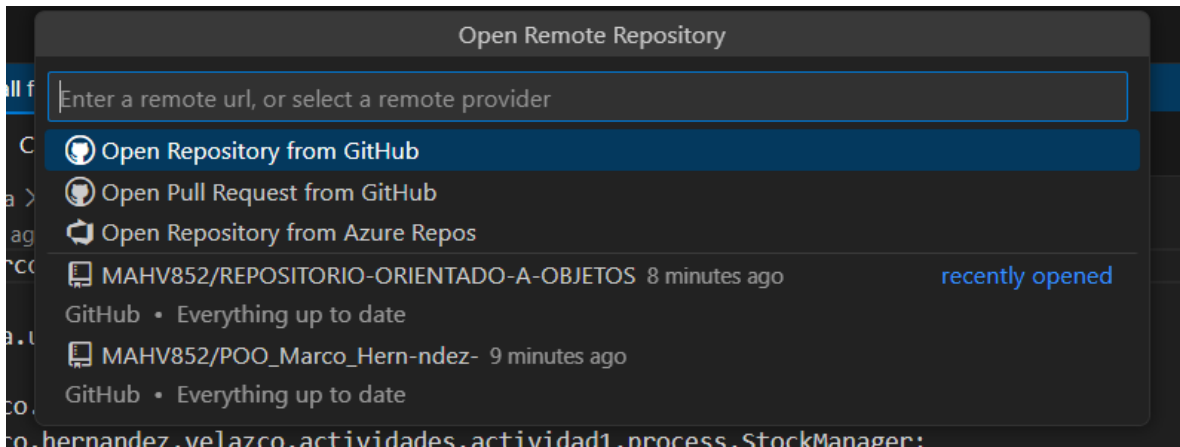
La primera parte era entrar a la página de Github e iniciar o crear un usuario, para poder crear el repositorio donde tendríamos nuestros archivos más adelante.

Descargar las extensiones de Github y java para poder usarlos en Visual Studio Code. Una vez ya descargada las extensiones lo siguiente sería iniciar sesión de github en para poder subir el repositorio a VSC y poder crear los archivos javas a usar posteriormente.

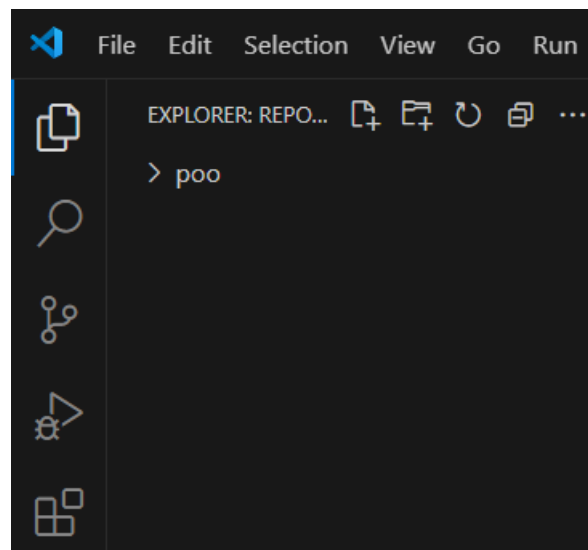
Con la sesión ya iniciada lo siguiente será usar la extensión de remote explorer para agregar el repositorio.



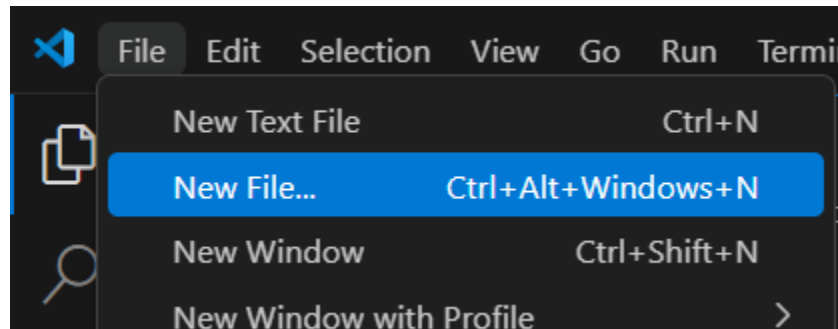
Seleccionar el repositorio que voy a usar que es el Repositorio-Orientado-A-Objetos



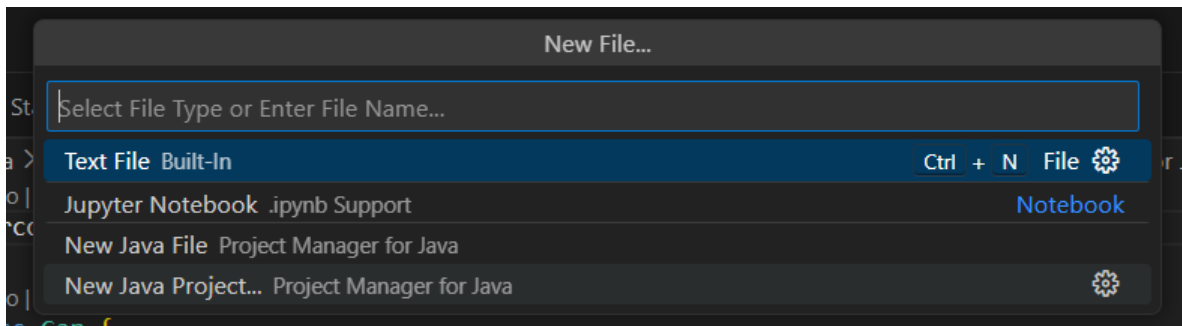
Y se abrirá el repositorio junto con la carpeta /archivos que tengo subido en GitHub



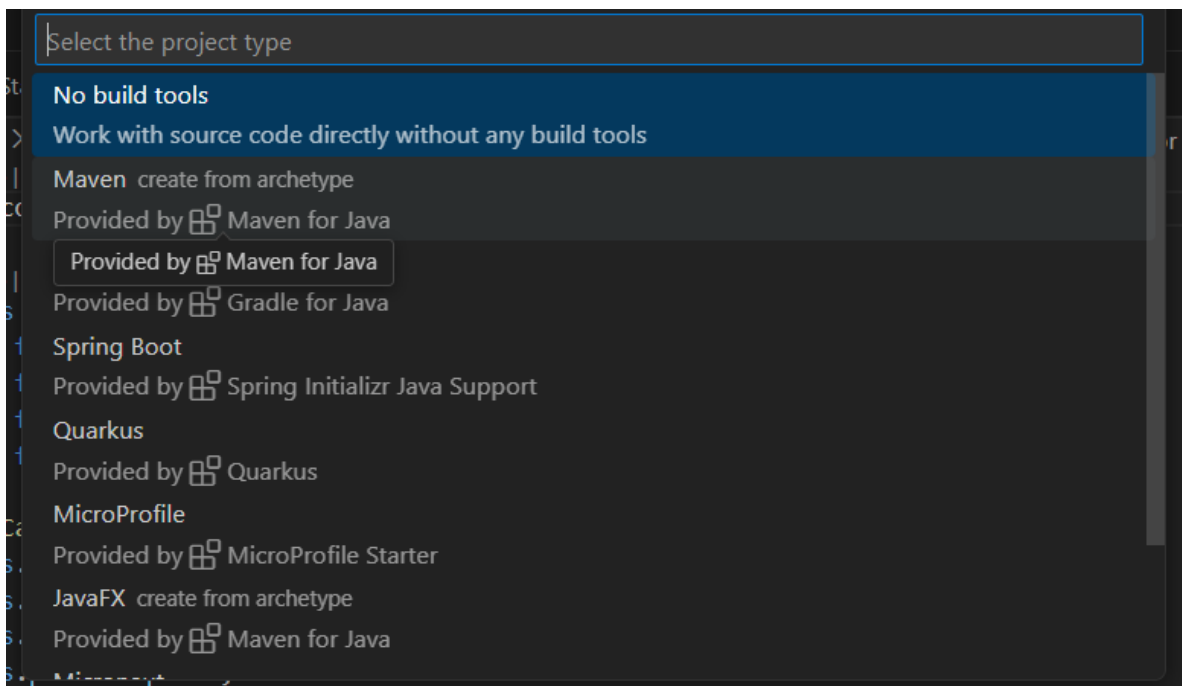
Lo siguiente que hice fue seleccionar un nuevo archivo



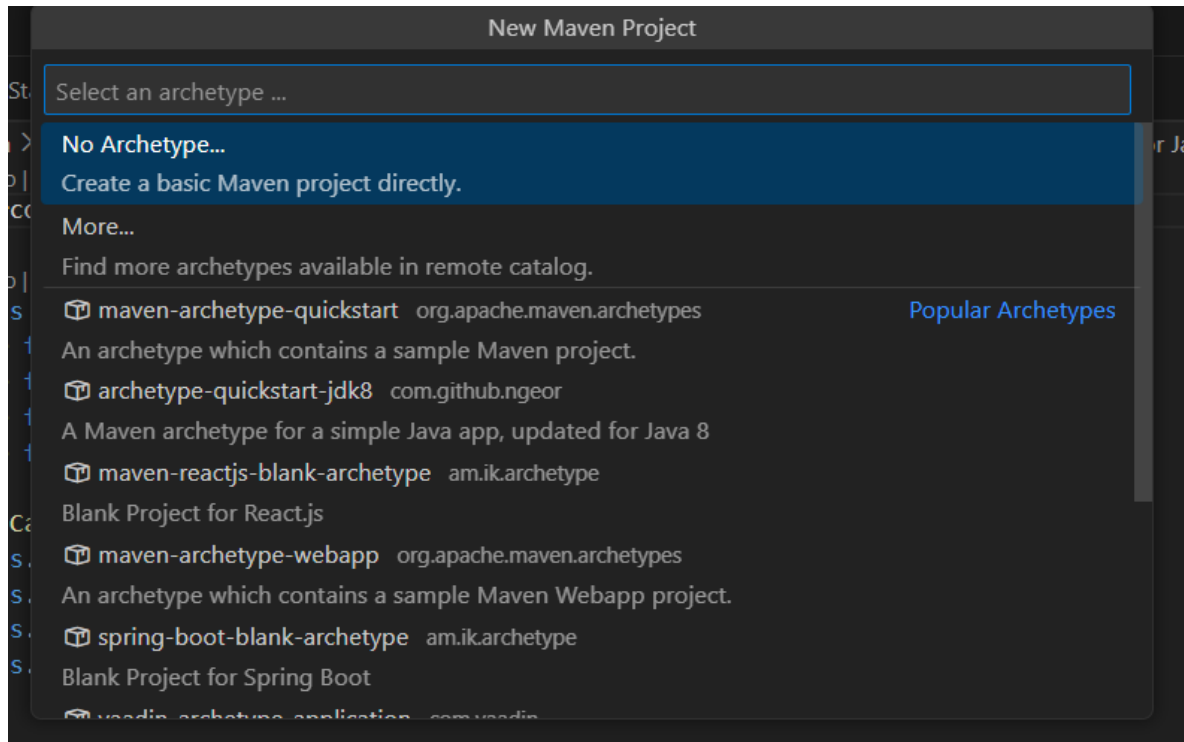
Seleccioné nuevo proyecto en Java



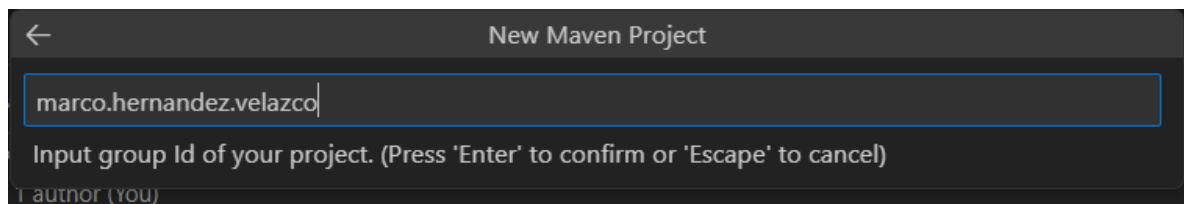
La opción de Maven para gestionar el proyecto



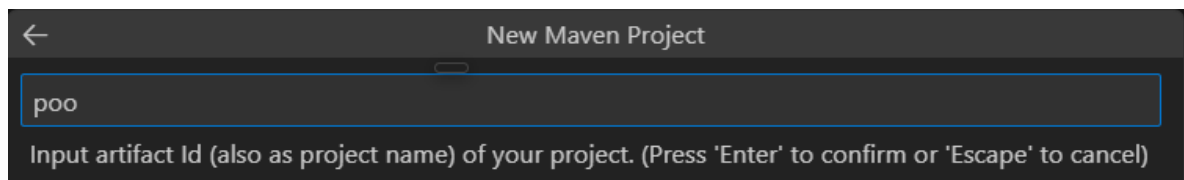
No archetype para un proyecto básico



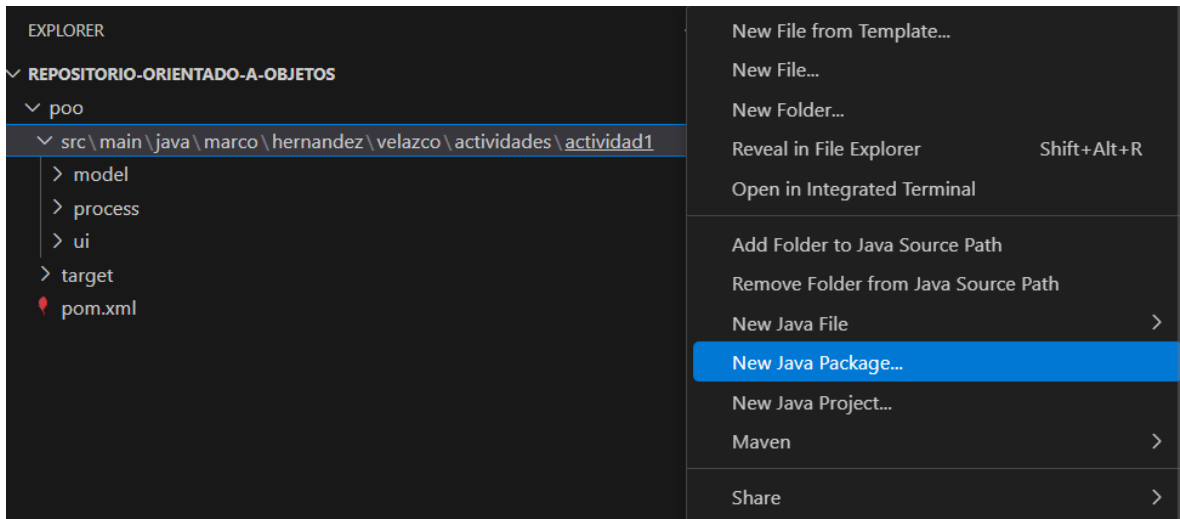
Lo nombré para identificarlo



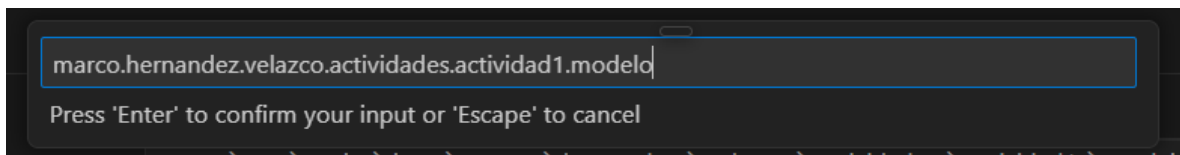
Y el nombre de la materia, también como nombre del proyecto



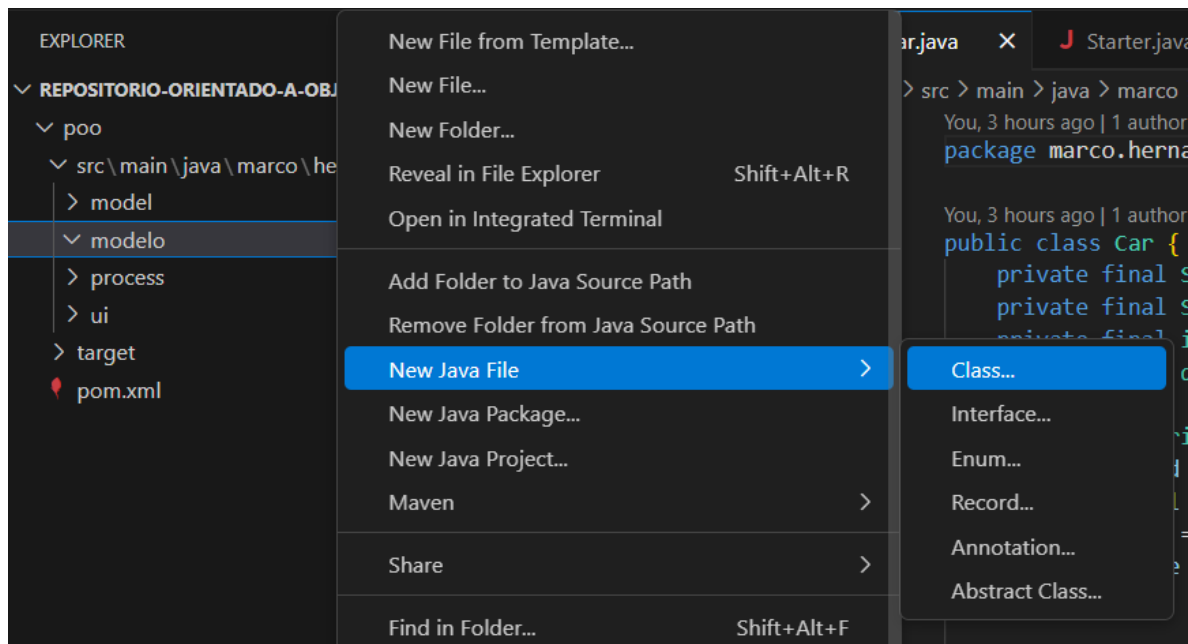
Una vez creado, click derecho, seleccionar “New java Package”



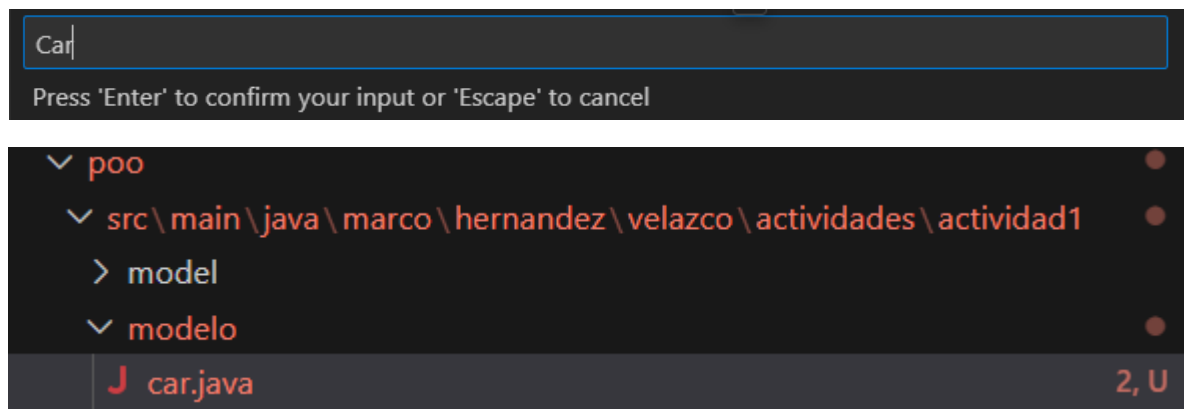
Nombramos la parte del código en la que vamos a trabajar, según tengo entendido es como una carpeta o al menos así la relacionó



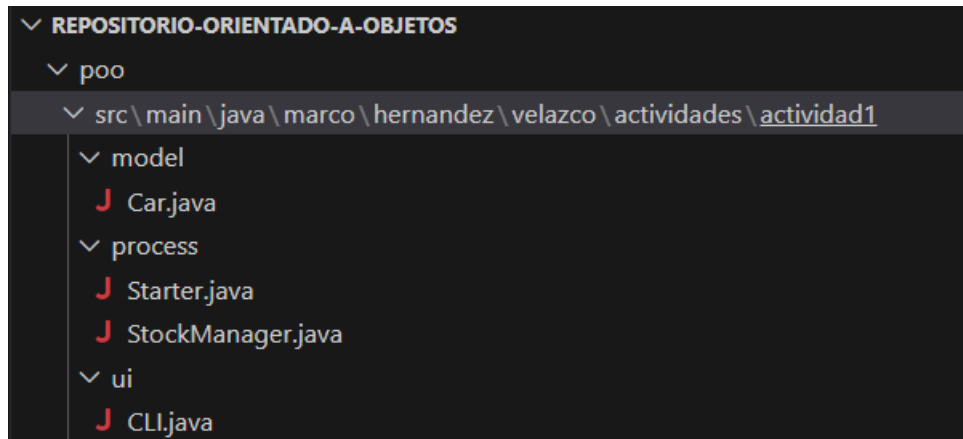
Después crear un nuevo archivo java tipo class



Lo nombramos



Repetí hasta tener las tres carpetas con los archivos java

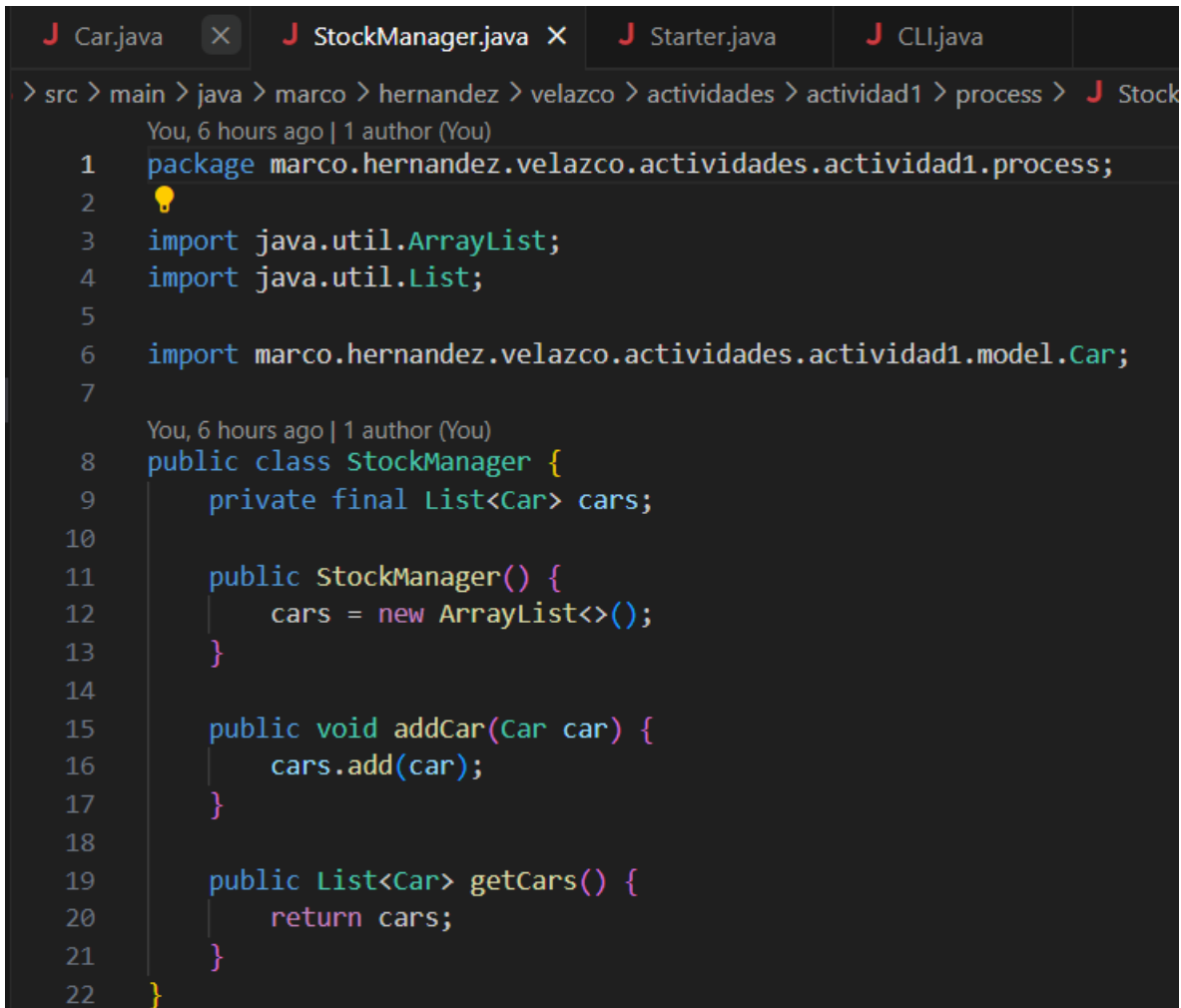


Aquí fue donde se me complicó bastante y es la parte de programar usando Java, nunca había usado java así que tuve que hacer una investigación y no me ha quedado del todo claro el como usarlo puesto que fue una reseña algo rápida pero no fue impedimento para realizar el código que desplegar el menú solicitado.

Lo primero que creé fue “Car” ya que este no depende de las otras clases (New Java File – Class). Solo define la estructura de los objetos (autos).

```
J Car.java X J Starter.java J StockManager.java J CLI.java
poo > src > main > java > marco > hernandez > velazco > actividades > actividad1 > model > J Car.java
100, 0 hours ago | 1 author (you)
• 3 public class Car {
4     private final String brand;
5     private final String model;
6     private final int year;
7     private final double price;
8
9     public Car(String brand, String model, int year, double price) {
10         this.brand = brand;
11         this.model = model;
12         this.year = year;
13         this.price = price;
14     }
15
16     public String getBrand() {
17         return brand;
18     }
19
20     public String getModel() {
21         return model;
22     }
23
24     public int getYear() {
25         return year;
26     }
27
28     public double getPrice() {
29         return price;
30     }
31     @Override
32     public String toString() {
33         return "Marca: " + brand + "\n" +
34             "Modelo: " + model + "\n" +
35             "Año: " + year + "\n" +
36             "Precio: $" + String.format("%.2f", price);
37     }
38 }
```

Lo siguiente fue el StockManager ya que esté gestiona los objetos creados, en este caso los autos, en la clase Car.



```
1 package marco.hernandez.velazco.actividades.actividad1.process;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import marco.hernandez.velazco.actividades.actividad1.model.Car;
7
8 public class StockManager {
9     private final List<Car> cars;
10
11     public StockManager() {
12         cars = new ArrayList<>();
13     }
14
15     public void addCar(Car car) {
16         cars.add(car);
17     }
18
19     public List<Car> getCars() {
20         return cars;
21     }
22 }
```

Después fue el CLI que lo que tengo entendido se encarga de la interacción con el usuario a través de la interfaz o en este caso el menú basado en texto. Este si es bastante extenso a diferencia de las otras dos clases, pero tiene sentido teniendo en cuenta que se

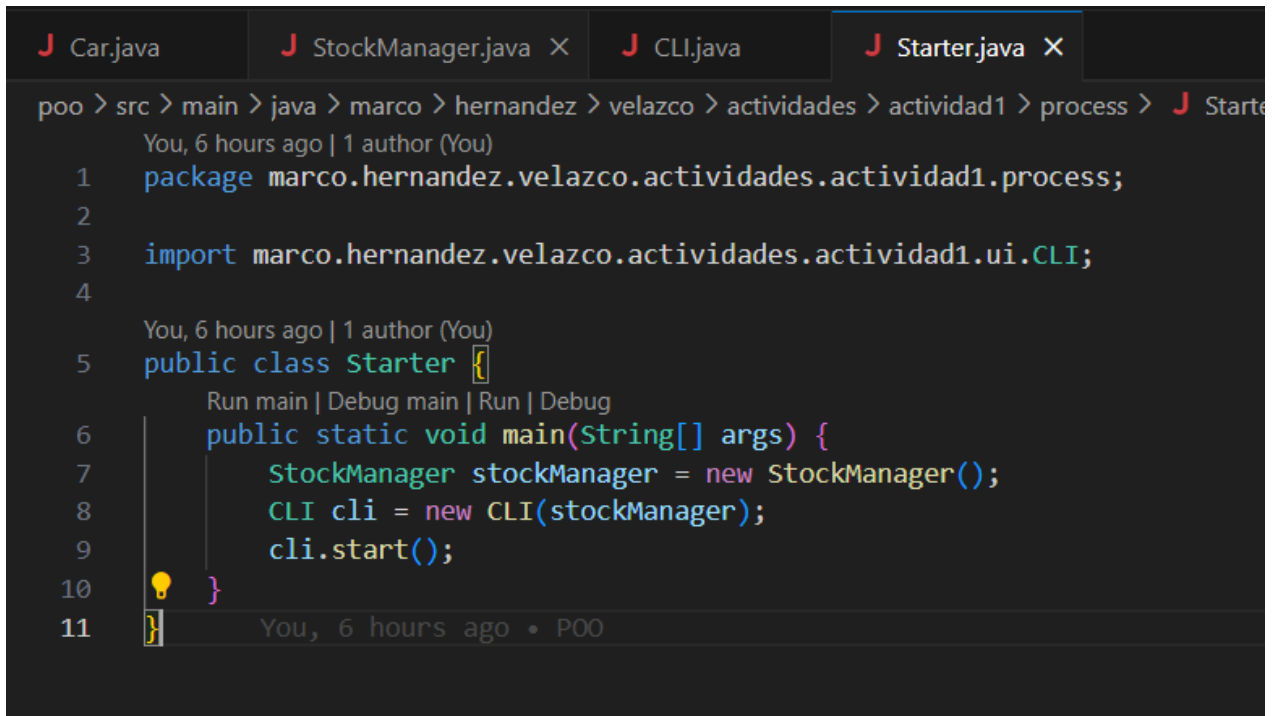
encarga de la interfaz para poder subir, mostrar el catalogo y salir del programa.

```
J Car.java  X  J StockManager.java  J CLI.java  X  J Starter.java
poo > src > main > java > marco > hernandez > velazco > actividades > actividad1 > ui > J CLI.java > ...
You, 6 hours ago | 1 author (You)
1 package marco.hernandez.velazco.actividades.actividad1.ui;
2   You, 6 hours ago • P00
3 import java.util.Scanner;
4
5 import marco.hernandez.velazco.actividades.actividad1.model.Car;
6 import marco.hernandez.velazco.actividades.actividad1.process.StockManager;
7
8 You, 6 hours ago | 1 author (You)
9 public class CLI {
10     private final StockManager stockManager;
11
12     public CLI(StockManager stockManager) {
13         this.stockManager = stockManager;
14     }
15
16     public void start() {
17         try (Scanner scanner = new Scanner(System.in)) {
18             int choice;
19
20             do {
21                 System.out.println("\n=== MENÚ ===");
22                 System.out.println("1. Subir un auto");
23                 System.out.println("2. Mostrar catálogo");
24                 System.out.println("3. Salir");
25                 System.out.print("\nElige una opción: ");
26                 choice = scanner.nextInt();
27                 scanner.nextLine(); // Limpiar el buffer
28
29                 switch (choice) {
30                     case 1 -> addCar(scanner);
31                     case 2 -> showCatalog();
32                     case 3 -> System.out.println("\nSaliendo del programa...");
33                     default -> System.out.println("\nOpción no válida. Inténtalo de nuevo.");
34                 }
35             } while (choice != 3);
36         }
37     }
38 }
```

```
J Car.java      J StockManager.java ×  J CLI.java ×  J Starter.java
poo > src > main > java > marco > hernandez > velazco > actividades > actividad1 > ui > J CLI.java > ...
8  public class CLI {

37
38      private void addCar(Scanner scanner) {
39          System.out.println("\n--- Subir un Auto ---");
40          System.out.print("Ingrese la marca del auto: ");
41          String brand = scanner.nextLine();
42
43          System.out.print("Ingrese el modelo del auto: ");
44          String model = scanner.nextLine();
45
46          System.out.print("Ingrese el año del auto: ");
47          int year = scanner.nextInt();
48
49          System.out.print("Ingrese el precio del auto: ");
50          double price = scanner.nextDouble();
51          scanner.nextLine(); // Limpiar el buffer
52
53          Car car = new Car(brand, model, year, price);
54          stockManager.addCar(car);
55          System.out.println("\nAuto agregado al catálogo con éxito.");
56      }
57
58      private void showCatalog() {
59          System.out.println("\n=== Catálogo de Autos ===");
60
61          if (stockManager.getCars().isEmpty()) {
62              System.out.println("No hay autos en el catálogo.");
63          } else {
64              for (Car car : stockManager.getCars()) {
65                  System.out.println(car);
66                  System.out.println("-----");
67              }
68          }
69      }
70  }
```

Por último, fue la Clase Starter que es el punto de entrada del programa y necesita que las anteriores 3 clases estén implementadas. Hace llamado a la función basada a las anteriores 3 clases para mostrarme el menú y solicitar los datos que requiero.



```
Car.java StockManager.java X CLI.java Starter.java X
poo > src > main > java > marco > hernandez > velazco > actividades > actividad1 > process > Starter.java
You, 6 hours ago | 1 author (You)
1 package marco.hernandez.velazco.actividades.actividad1.process;
2
3 import marco.hernandez.velazco.actividades.actividad1.ui.CLI;
4
5 You, 6 hours ago | 1 author (You)
6 public class Starter {
7     Run main | Debug main | Run | Debug
8     public static void main(String[] args) {
9         StockManager stockManager = new StockManager();
10        CLI cli = new CLI(stockManager);
11        cli.start();
12    }
13 }
```

Dando este pequeño y simple menú como resultado final:

```
J Car.java J StockManager.java >
poo > src > main > java > marco > hernande
You. 6 hours ago | 1 author (You)
PROBLEMS OUTPUT DEBUG CONSOLE
=== MENÚ ===
1. Subir un auto
2. Mostrar catálogo
3. Salir

Elige una opción: 1

--- Subir un Auto ---
Ingrese la marca del auto: Mazda
Ingrese el modelo del auto: Mazda 3
Ingrese el año del auto: 2025
Ingrese el precio del auto: 400,000

Auto agregado al catálogo con éxito

=== MENÚ ===
1. Subir un auto
2. Mostrar catálogo
3. Salir

Elige una opción: 2

=== Catálogo de Autos ===
Marca: Mazda
Modelo: Mazda 3
Año: 2025
Precio: $400000.00
-----

=== MENÚ ===
1. Subir un auto
2. Mostrar catálogo
3. Salir

Elige una opción: 3

Saliendo del programa...
```

Reflexión:

Primero me disculpo por entregar este reporte tarde, tuve varias complicaciones con la luz entre otros temas personales. Regresando a la actividad siendo muy honesto se me complicó no poder entender Java, al momento de escribir el código tuve que ir pidiendo ayuda a amigos, videos y páginas que explicaran sobre Java pero aún así no lograba comprender bien sobre cómo hacer el código o por donde iniciar, hasta que de una u otra forma logré ir creando las Clases y conforme terminaba el código de cada una, pasaba con la siguiente hasta haber terminado el código.

No funcionaba al inicio, todo el día fueron de solo intentos fallidos, me desesperaba que no funcionaba incluso me había rendido sin embargo con ayuda de amigos lograba avanzar poco a poco hasta lograr crear un código que finalmente me desplegara la interfaz o el menú que me permitiera subir un auto al catálogo para posteriormente mostrármelo.

Fue un dolor de cabeza y estrés, pero al momento de terminar el código fue muy satisfactorio lograr ver el menú desplegado. Al final valió la pena el hacer este trabajo pues he aprendido un poco sobre el lenguaje de programación JAVA.