

# Fundamentos de POO

Ingeniería en  
Tecnologías de la  
Información

---

KAREN LETICIA GARCÍA VÁSQUEZ

EDER LÓPEZ VILLARREAL

JAZMIN PORTILLO MICHICOL

ERIC JHONATHAN ANAYA MÁRQUEZ

5H T/V

DOCENTE

MTRO. SAUL OLAF LOAIZA

---

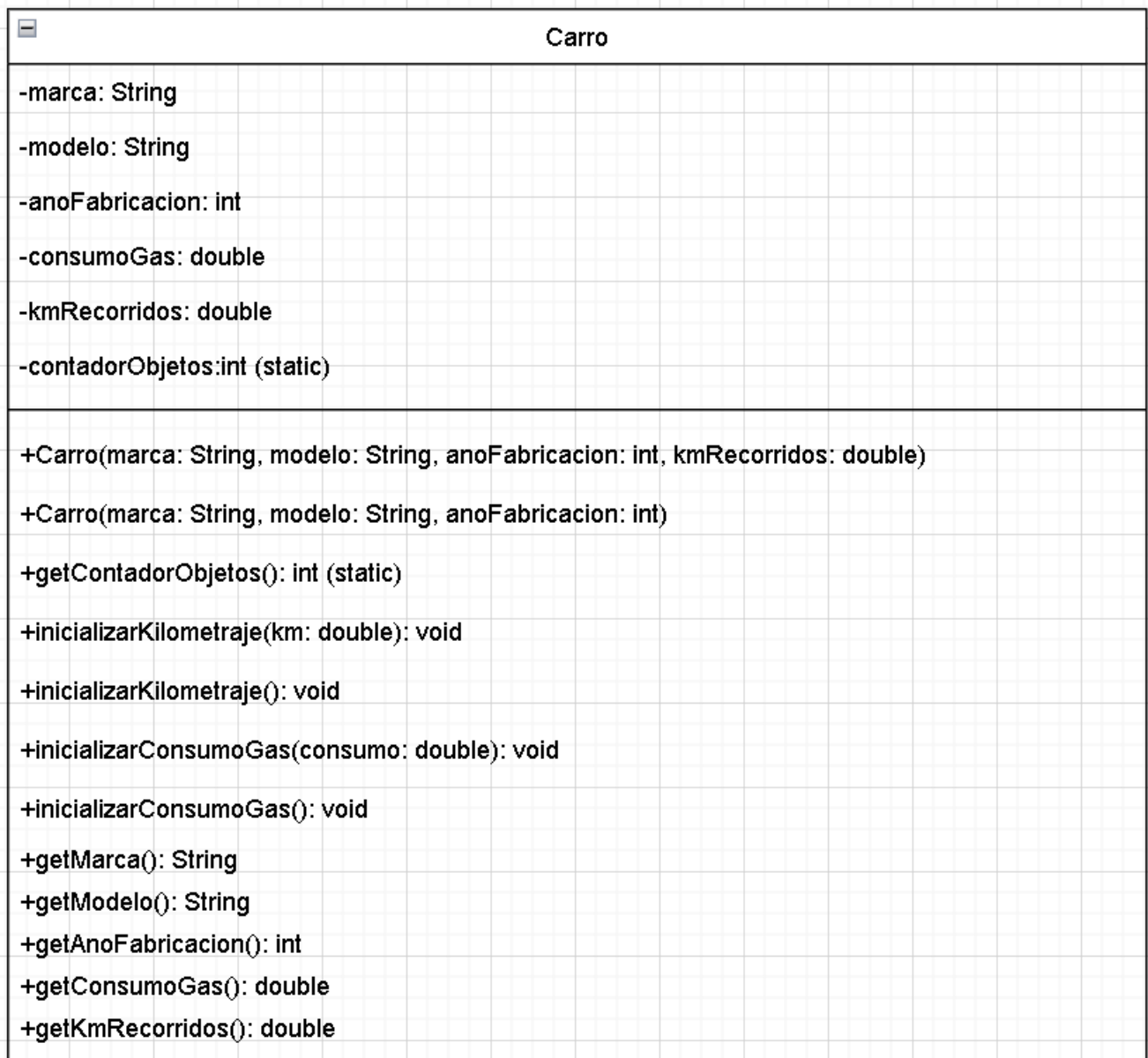
Practica 5

Clase Carro con sobrecarga

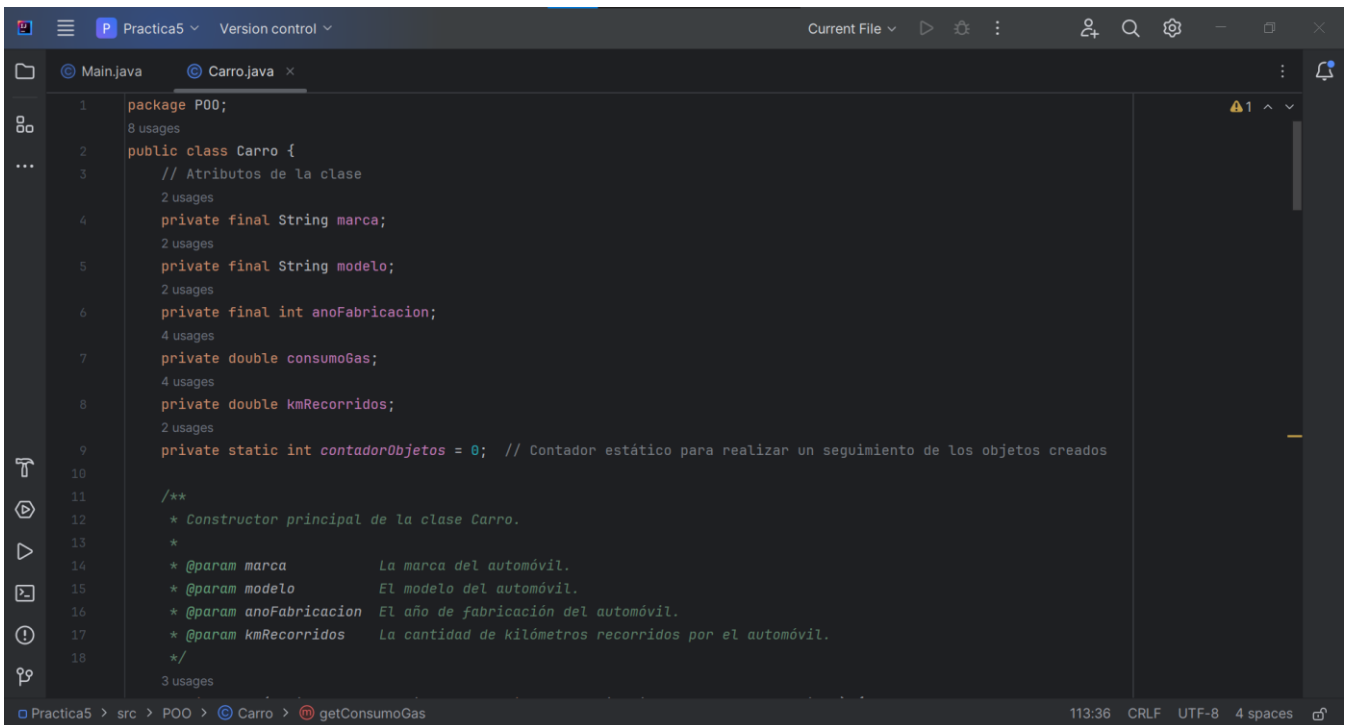
21/01/2024



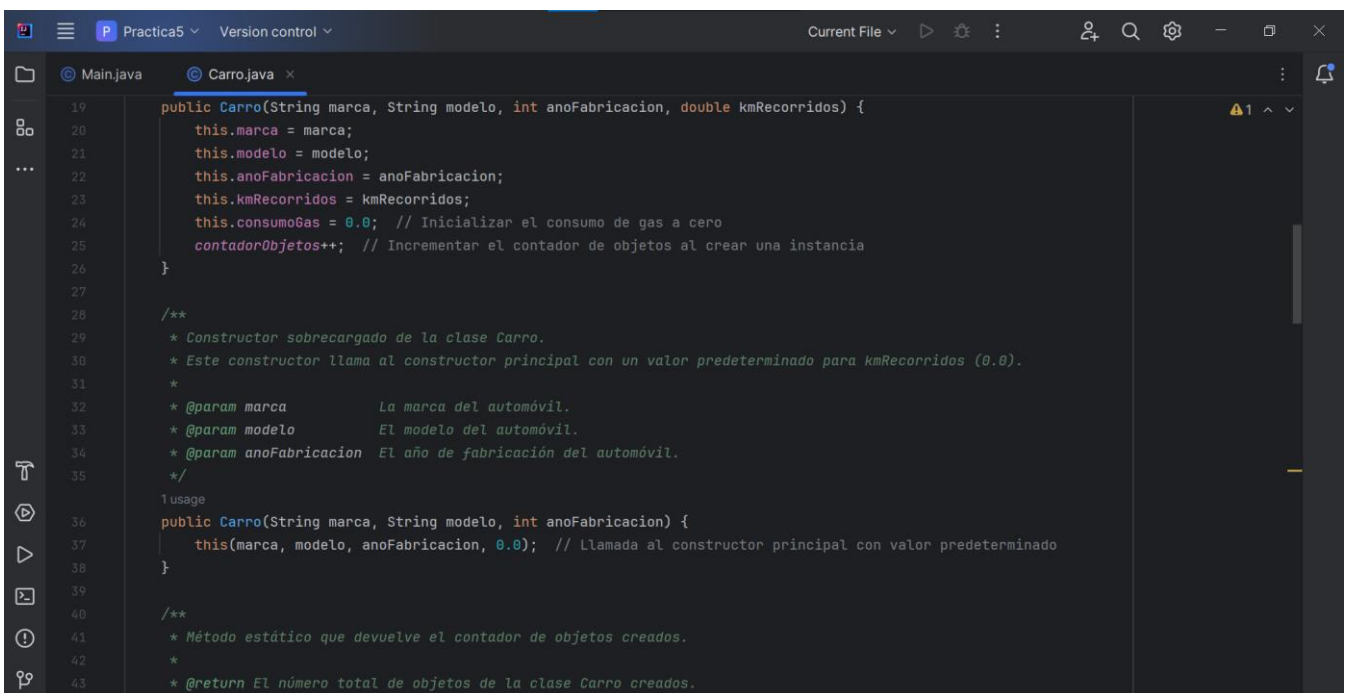
## Diagrama UML del clasificador de la clase Carro



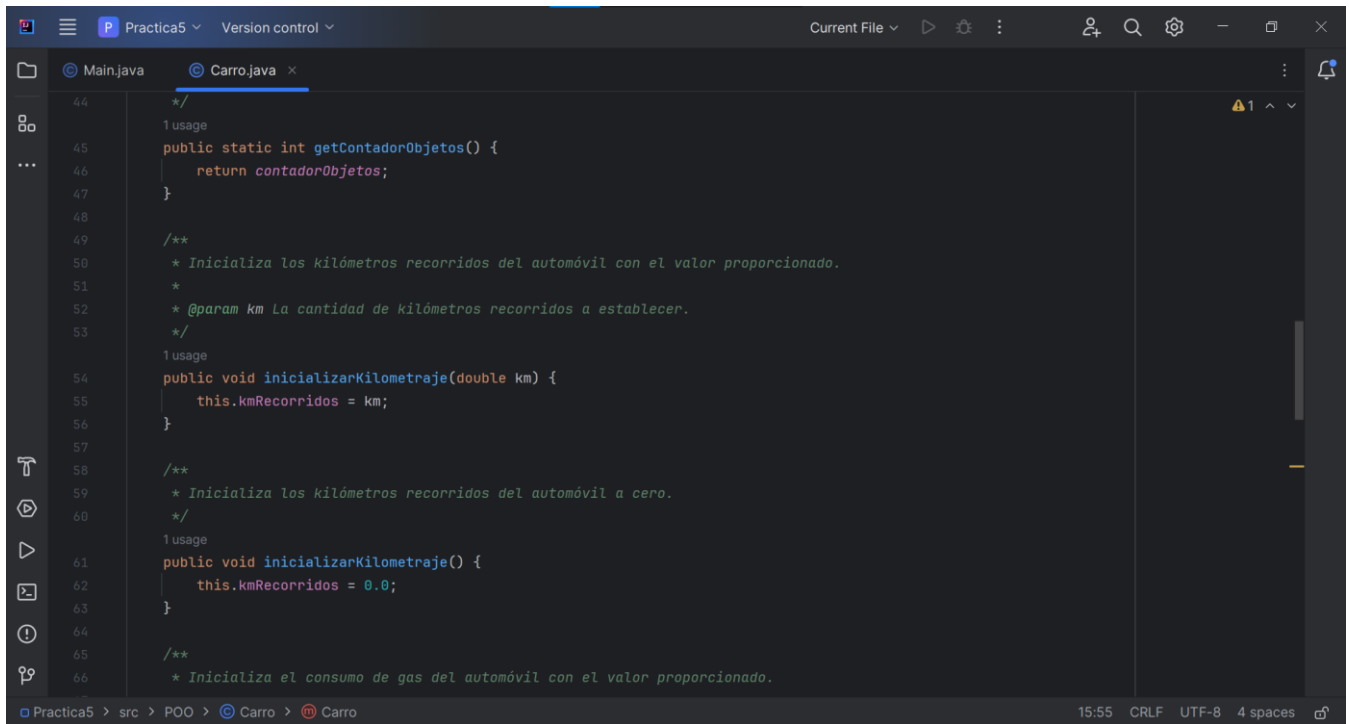
# Código de clase Carro



```
1 package P00;
2 public class Carro {
3     // Atributos de la clase
4     private final String marca;
5     private final String modelo;
6     private final int anoFabricacion;
7     private double consumoGas;
8     private double kmRecorridos;
9     private static int contadorObjetos = 0; // Contador estático para realizar un seguimiento de los objetos creados
10
11     /**
12      * Constructor principal de la clase Carro.
13      *
14      * @param marca La marca del automóvil.
15      * @param modelo El modelo del automóvil.
16      * @param anoFabricacion El año de fabricación del automóvil.
17      * @param kmRecorridos La cantidad de kilómetros recorridos por el automóvil.
18      */
19 }
```

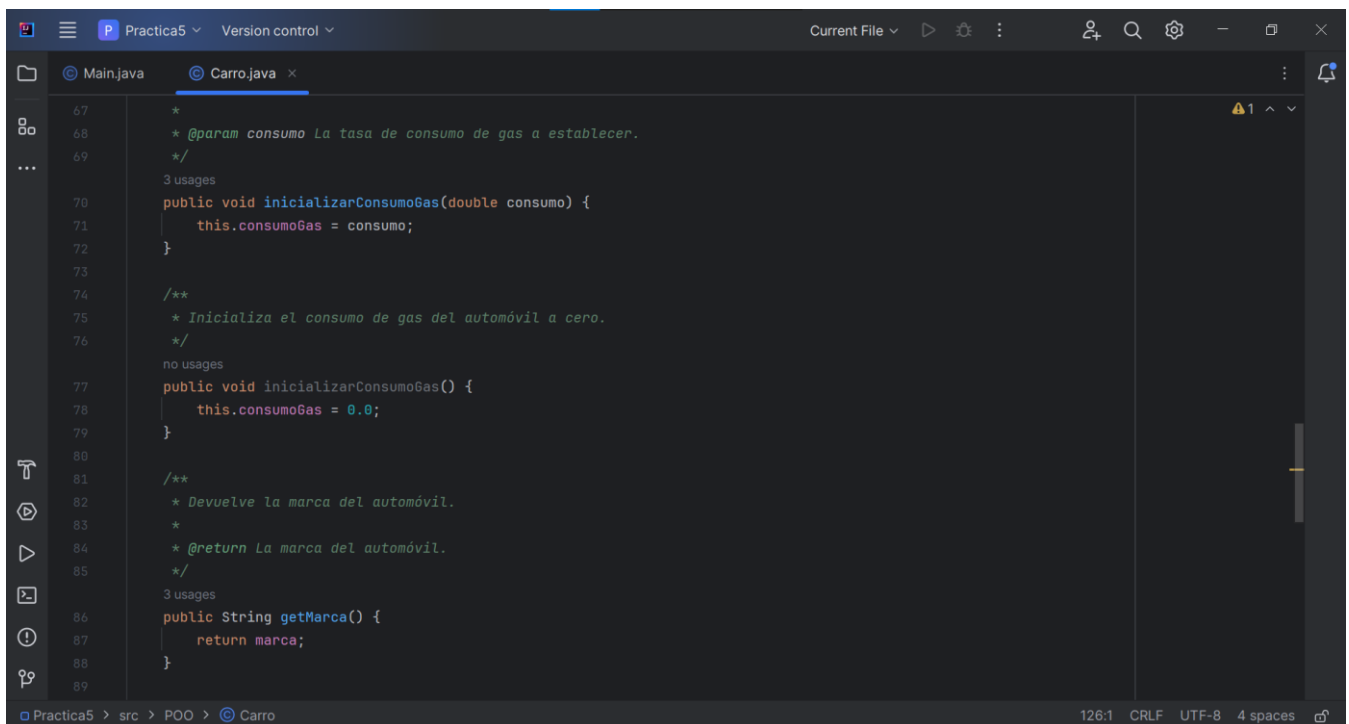


```
19 public Carro(String marca, String modelo, int anoFabricacion, double kmRecorridos) {
20     this.marca = marca;
21     this.modelo = modelo;
22     this.anoFabricacion = anoFabricacion;
23     this.kmRecorridos = kmRecorridos;
24     this.consumoGas = 0.0; // Inicializar el consumo de gas a cero
25     contadorObjetos++; // Incrementar el contador de objetos al crear una instancia
26 }
27
28 /**
29  * Constructor sobrecargado de la clase Carro.
30  * Este constructor llama al constructor principal con un valor predeterminado para kmRecorridos (0.0).
31  *
32  * @param marca La marca del automóvil.
33  * @param modelo El modelo del automóvil.
34  * @param anoFabricacion El año de fabricación del automóvil.
35  */
36 public Carro(String marca, String modelo, int anoFabricacion) {
37     this(marca, modelo, anoFabricacion, 0.0); // Llamada al constructor principal con valor predeterminado
38 }
39
40 /**
41  * Método estático que devuelve el contador de objetos creados.
42  *
43  * @return El número total de objetos de la clase Carro creados.
44  */
45 }
```



```
44  */
45  1 usage
46  public static int getContadorObjetos() {
47      return contadorObjetos;
48  }
49
50  /**
51   * Inicializa los kilómetros recorridos del automóvil con el valor proporcionado.
52   *
53   * @param km La cantidad de kilómetros recorridos a establecer.
54   */
55  1 usage
56  public void inicializarKilometraje(double km) {
57      this.kmRecorridos = km;
58  }
59
60  /**
61   * Inicializa los kilómetros recorridos del automóvil a cero.
62   */
63  1 usage
64  public void inicializarKilometraje() {
65      this.kmRecorridos = 0.0;
66  }
67
68  /**
69   * Inicializa el consumo de gas del automóvil con el valor proporcionado.
```

Practica5 > src > POO > Carro > Carro 15:55 CRLF UTF-8 4 spaces



```
67  *
68  * @param consumo La tasa de consumo de gas a establecer.
69  */
70  3 usages
71  public void inicializarConsumoGas(double consumo) {
72      this.consumoGas = consumo;
73  }
74
75  /**
76   * Inicializa el consumo de gas del automóvil a cero.
77   */
78  no usages
79  public void inicializarConsumoGas() {
80      this.consumoGas = 0.0;
81  }
82
83  /**
84   * Devuelve la marca del automóvil.
85   *
86   * @return La marca del automóvil.
87   */
88  3 usages
89  public String getMarca() {
90      return marca;
91  }
92
```

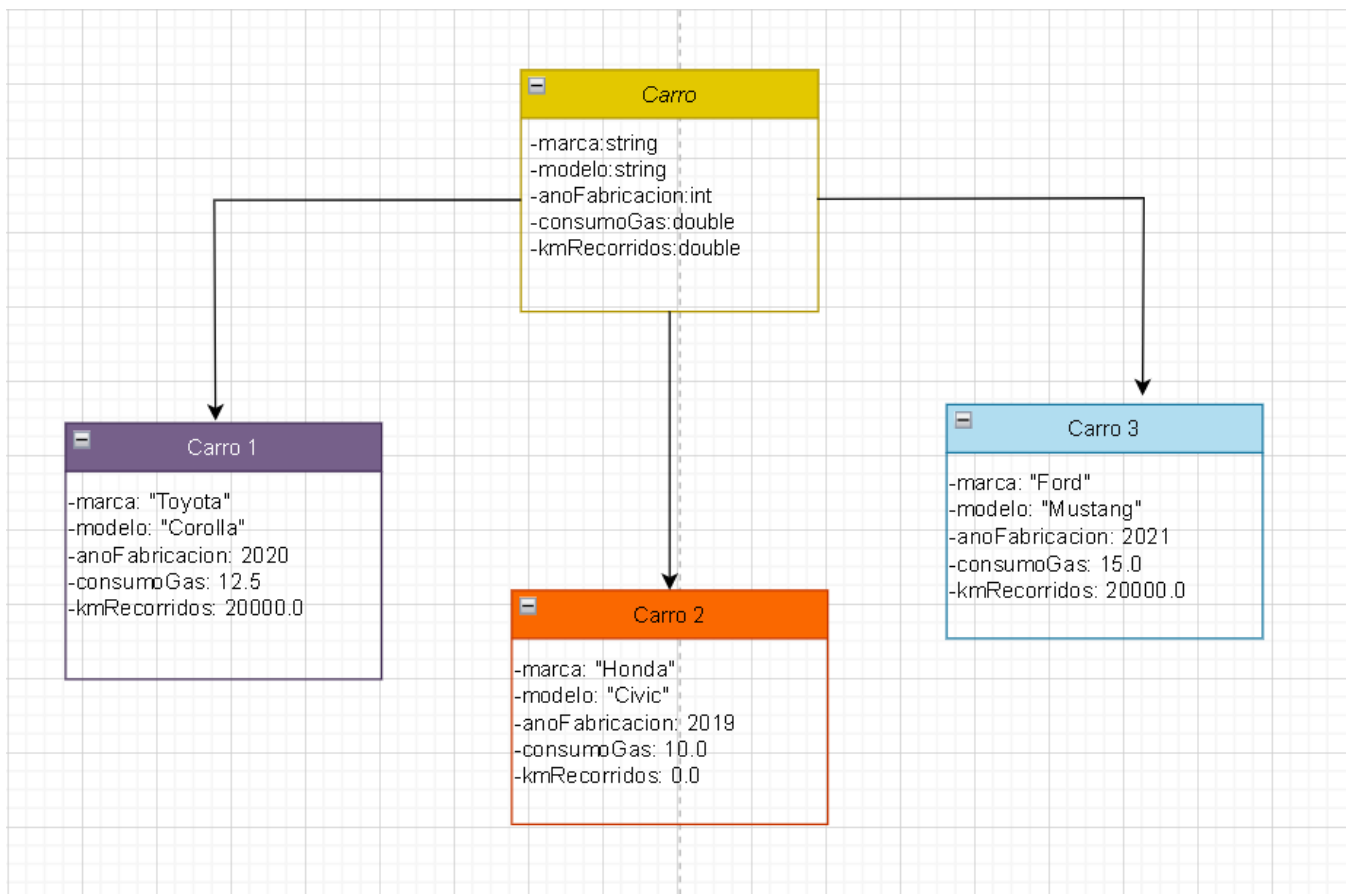
Practica5 > src > POO > Carro 126:1 CRLF UTF-8 4 spaces

```
90  /**
91   * Devuelve el modelo del automóvil.
92   *
93   * @return El modelo del automóvil.
94   */
95  3 usages
96  public String getModelo() {
97      return modelo;
98  }
99
100  /**
101   * Devuelve el año de fabricación del automóvil.
102   *
103   * @return El año de fabricación del automóvil.
104   */
105  3 usages
106  public int getAnoFabricacion() {
107      return anoFabricacion;
108  }
109
110  /**
111   * Devuelve el consumo de gas del automóvil.
112   *
113   * @return La tasa de consumo de gas del automóvil.
114   */
115  3 usages
```

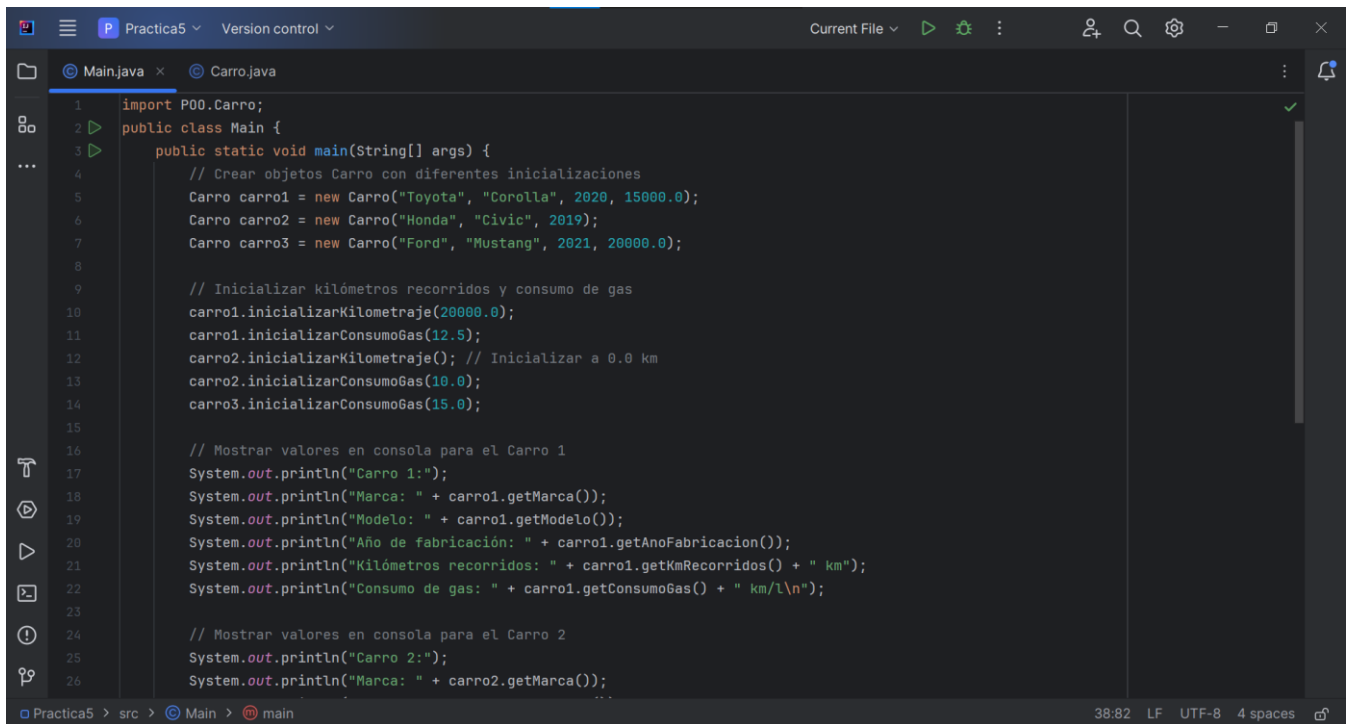
Practica5 > src > POO > Carro 126:1 CRLF UTF-8 4 spaces

```
113  public double getConsumoGas() {
114      return consumoGas;
115  }
116
117  /**
118   * Devuelve la cantidad de kilómetros recorridos por el automóvil.
119   *
120   * @return La cantidad de kilómetros recorridos por el automóvil.
121   */
122  3 usages
123  public double getKmRecorridos() {
124      return kmRecorridos;
125  }
```


# Diagrama UML de los objetos para verificar la prueba



# Clase principal

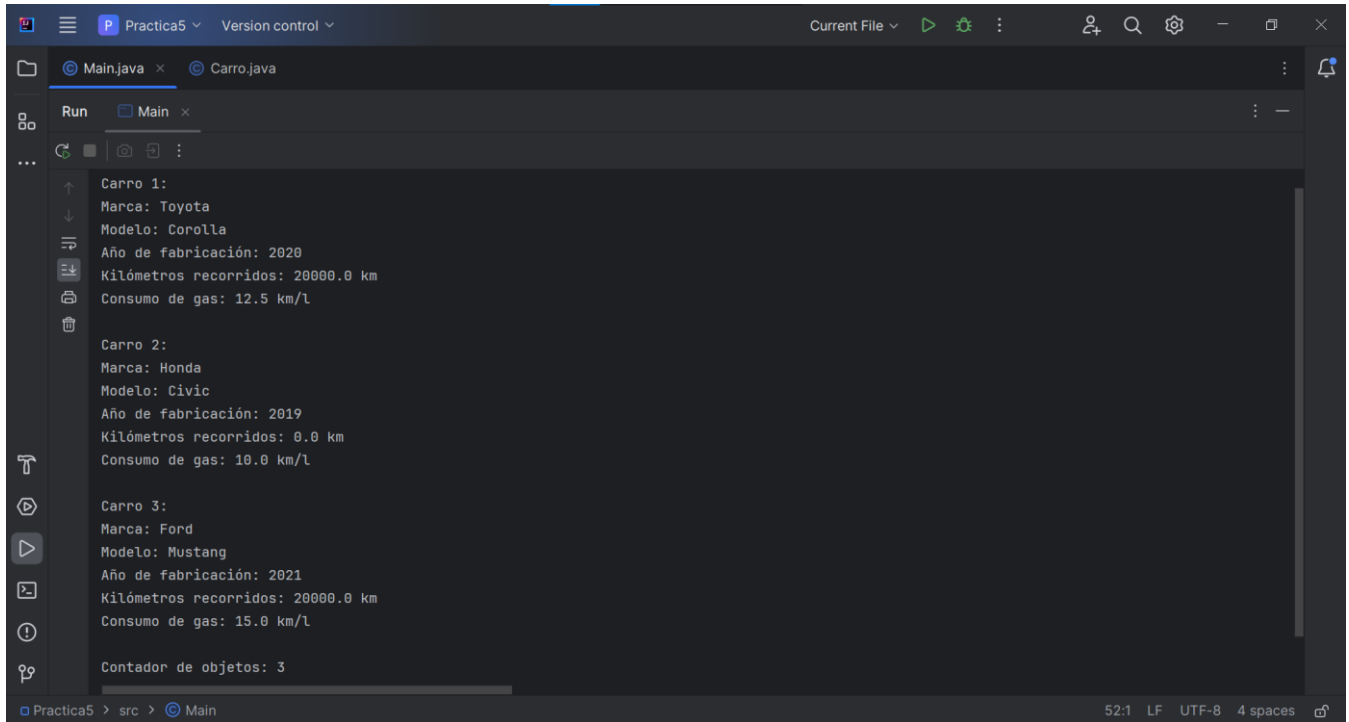


```
1 import P00.Carro;
2 public class Main {
3     public static void main(String[] args) {
4         // Crear objetos Carro con diferentes inicializaciones
5         Carro carro1 = new Carro("Toyota", "Corolla", 2020, 15000.0);
6         Carro carro2 = new Carro("Honda", "Civic", 2019);
7         Carro carro3 = new Carro("Ford", "Mustang", 2021, 20000.0);
8
9         // Inicializar kilómetros recorridos y consumo de gas
10        carro1.inicializarKilometraje(20000.0);
11        carro1.inicializarConsumoGas(12.5);
12        carro2.inicializarKilometraje(); // Inicializar a 0.0 km
13        carro2.inicializarConsumoGas(10.0);
14        carro3.inicializarConsumoGas(15.0);
15
16        // Mostrar valores en consola para el Carro 1
17        System.out.println("Carro 1:");
18        System.out.println("Marca: " + carro1.getMarca());
19        System.out.println("Modelo: " + carro1.getModelo());
20        System.out.println("Año de fabricación: " + carro1.getAnoFabricacion());
21        System.out.println("Kilómetros recorridos: " + carro1.getKmRecorridos() + " km");
22        System.out.println("Consumo de gas: " + carro1.getConsumoGas() + " km/l\n");
23
24        // Mostrar valores en consola para el Carro 2
25        System.out.println("Carro 2:");
26        System.out.println("Marca: " + carro2.getMarca());
```



```
27        System.out.println("Modelo: " + carro2.getModelo());
28        System.out.println("Año de fabricación: " + carro2.getAnoFabricacion());
29        System.out.println("Kilómetros recorridos: " + carro2.getKmRecorridos() + " km");
30        System.out.println("Consumo de gas: " + carro2.getConsumoGas() + " km/l\n");
31
32        // Mostrar valores en consola para el Carro 3
33        System.out.println("Carro 3:");
34        System.out.println("Marca: " + carro3.getMarca());
35        System.out.println("Modelo: " + carro3.getModelo());
36        System.out.println("Año de fabricación: " + carro3.getAnoFabricacion());
37        System.out.println("Kilómetros recorridos: " + carro3.getKmRecorridos() + " km");
38        System.out.println("Consumo de gas: " + carro3.getConsumoGas() + " km/l\n");
39
40        // Mostrar el contador de objetos creados
41        System.out.println("Contador de objetos: " + Carro.getContadorObjetos());
42    }
43 }
```

# Prueba



The screenshot shows an IDE window with a dark theme. The top bar includes a menu icon, a tab labeled 'Practica5', and a 'Version control' dropdown. The right side of the top bar has icons for 'Current File', a play button, a bug icon, a settings gear, and window management icons. Below the top bar, there are two tabs: 'Main.java' (active) and 'Carro.java'. The main editor area displays the output of a Java program. The output is organized into sections for three cars and a final counter. The first car is a Toyota Corolla from 2020 with 20000.0 km traveled and 12.5 km/L fuel consumption. The second car is a Honda Civic from 2019 with 0.0 km traveled and 10.0 km/L fuel consumption. The third car is a Ford Mustang from 2021 with 20000.0 km traveled and 15.0 km/L fuel consumption. The final line shows 'Contador de objetos: 3'. The bottom status bar indicates the file path 'Practica5 > src > Main', the line number '52:1', and the encoding 'LF UTF-8 4 spaces'.

```
Carro 1:  
Marca: Toyota  
Modelo: Corolla  
Año de fabricación: 2020  
Kilómetros recorridos: 20000.0 km  
Consumo de gas: 12.5 km/L  
  
Carro 2:  
Marca: Honda  
Modelo: Civic  
Año de fabricación: 2019  
Kilómetros recorridos: 0.0 km  
Consumo de gas: 10.0 km/L  
  
Carro 3:  
Marca: Ford  
Modelo: Mustang  
Año de fabricación: 2021  
Kilómetros recorridos: 20000.0 km  
Consumo de gas: 15.0 km/L  
  
Contador de objetos: 3
```