UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

Tema:

Sistema de Gestión de Parking

Autores:

Anthony Aguilar

Ronny Casco

Alexandra Lalaleo

Ronny Lugmaña

Materia:

Programación Orientada a Objetos

NRC:

5401

TUTOR:

Ing. Luis Enrique Jaramillo Montaño

Sangolquí – Ecuador

Objetivo General

Desarrollar e implementar un sistema de gestión integral de parqueadero utilizando Programación Orientada a Objetos, que permita registrar, consultar, actualizar y administrar de manera eficiente y precisa la información de los vehículos, optimizando el control y la organización del parqueadero.

Objetivos Específicos

- Implementar la clase Vehículo que almacene la información de los vehículos, incluyendo placa, modelo y color.
- Desarrollar la clase Parqueadero que gestione la lista de vehículos y proporcione métodos para registrar, consultar y actualizar vehículos.
- Crear la clase Principal que contenga el método main para probar el funcionamiento del sistema.
- Diseñar un diagrama UML que represente la estructura y las relaciones entre las clases del sistema.

Marco Teórico

La Programación Orientada a Objetos es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

Un programador diseña un programa de software organizando piezas de información y comportamientos relacionados en una plantilla llamada clase. Luego, se crean objetos

individuales a partir de la plantilla de clase. Todo el programa de software se ejecuta haciendo que varios objetos interactúen entre sí para crear un programa más grande.

Los sistemas de gestión de parking son por ello la herramienta que permita monitorizar los vehículos que transitan por diferentes áreas y así mejorar el acceso de vehículos autorizados. De esta manera se puede conseguir un control de todos los usuarios, el cobro por el acceso y la salida eficiente.

Las ventajas de instalar un sistema de Parking

Para evitar este tipo de constantes errores en la gestión del estacionamiento se puede usar un sistema de Parking que tiene las siguientes ventajas:

- Forma de pago
- Opciones de cumplimiento
- Controles de acceso
- Reportes e informes

Un sistema de gestión de aparcamientos suele constar de los siguientes componentes clave:

Control de acceso al aparcamiento: Este componente incluye barreras, puertas y terminales de entrada/salidas equipadas con dispensadores de tickets, lectores de tarjetas o sistemas de reconocimiento de matrículas para controlar el acceso a la zona de aparcamiento.

Seguridad y vigilancia: Este componente incluye dispositivos de control de acceso para garantizar la seguridad de la zona de aparcamiento y disuadir cualquier posible actividad delictiva.

Gestión de datos e informes: Este componente comprende un sistema de software centralizado que recopila y gestiona los datos relacionados con las transacciones de aparcamiento, las tasas de ocupación y otra información relevante.

Sistemas de pago y visualización: Uno de los más comunes en Europa y el Reino Unido. Los usuarios aparcan sus vehículos y compran tickets de aparcamiento en máquinas automáticas. A continuación, los usuarios los muestran en el salpicadero para indicar el pago.

Sistemas de reconocimiento de matrículas: Las cámaras instaladas en los puntos de entrada y salida captan la información de las matrículas de los vehículos. Esto permite la entrada y salida automáticas sin necesidad de tickets físicos.

Sistemas de reserva: Los usuarios pueden reservar plazas de aparcamiento con antelación a través de plataformas en línea o aplicaciones móviles. Esto garantiza una plaza de aparcamiento a la llegada.

Planteamiento del Problema

El parqueadero necesita un sistema eficiente para gestionar la información de los vehículos. Los problemas actuales incluyen la dificultad para registrar, consultar y actualizar los datos de los vehículos de manera rápida y precisa. Un sistema manual es propenso a errores y requiere mucho tiempo. Por lo tanto, se propone desarrollar un sistema automatizado utilizando Programación Orientada a Objetos y utilizando el software Java para mejorar la eficiencia y precisión en la gestión del parqueadero.

Diseño

El diseño del sistema se basa en la creación de clases que representan las dos entidades principales. Cada clase tiene atributos y métodos que permiten gestionar la información y las operaciones relacionadas con esa entidad.

Menú Principal

```
package PARQUEADERO;

import java.util.Scanner;

public class Principal {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      Parqueadero parqueadero = new Parqueadero();
      int opcion;

      do {
         System.out.println("\n--- MENU ---");
         System.out.println("1. Registrar vehiculo");
         System.out.println("2. Consultar vehiculo");
         System.out.println("3. Actualizar vehiculo");
         System.out.println("4. Eliminar vehiculo");
         System.out.println("5. Mostrar todos los vehiculos");
         System.out.println("6. Salir");
         System.out.print("Elige una opcion: ");
         opcion = sc.nextInt();
         sc.nextLine(); // Consumir salto de linea

         switch (opcion) {
            case 1:
               System.out.print("Ingresa la placa: ");
               String placa = sc.nextLine();
               System.out.print("Ingresa el modelo: ");
               String modelo = sc.nextLine();
               System.out.print("Ingresa el color: ");
               String color = sc.nextLine();
               if (parqueadero.registrarVehiculo(placa, modelo, color)) {
                  System.out.println("Vehiculo registrado correctamente.");
               }
               break;
            case 2:
               System.out.print("Ingresa la placa del vehiculo a consultar: ");
               placa = sc.nextLine();
               Vehiculo v = parqueadero.consultarVehiculo(placa);
               if (v != null) {
```

```
                System.out.println("Vehiculo encontrado: " + v);
            } else {
                System.out.println("No se encontro un vehiculo con esa placa.");
            }
            break;
        case 3:
            System.out.print("Ingresa la placa del vehiculo a actualizar: ");
            placa = sc.nextLine();
            System.out.print("Ingresa el nuevo modelo: ");
            modelo = sc.nextLine();
            System.out.print("Ingresa el nuevo color: ");
            color = sc.nextLine();
            if (parqueadero.actualizarVehiculo(placa, modelo, color)) {
                System.out.println("Vehiculo actualizado correctamente.");
            } else {
                System.out.println("No se encontro un vehiculo con esa placa.");
            }
            break;
        case 4:
            System.out.print("Ingresa la placa del vehiculo a eliminar: ");
            placa = sc.nextLine();
            if (parqueadero.eliminarVehiculo(placa)) {
                System.out.println("Vehiculo eliminado correctamente.");
            } else {
                System.out.println("No se encontro un vehiculo con esa placa.");
            }
            break;
        case 5:
            System.out.println("\n--- Lista de Vehiculos ---");
            parqueadero.mostrarVehiculos();
            break;
        case 6:
            System.out.println("Saliendo del programa.");
            break;
        default:
            System.out.println("Opcion no valida.");
      }
    } while (opcion != 6);

    sc.close();
  }
}
```

Datos del vehículo

```
package PARQUEADERO;

public class Vehiculo {
  private String placa;
  public String modelo;
```

```
    public String color;

    public Vehiculo(String placa, String modelo, String color) {
        this.placa = placa;
        this.modelo = modelo;
        this.color = color;
    }

    public String getPlaca() {
        return placa;
    }

    public void setPlaca(String placa) {
        this.placa = placa;
    }

    public String toString() {
        return "Vehiculo{placa='" + placa + "', modelo='" + modelo + "', color='" + color + "'}";
    }
}
```

Parqueadero

```
package PARQUEADERO;

import java.util.ArrayList;
import java.util.List;

public class Parqueadero {
    public List<Vehiculo> vehiculos;

    public Parqueadero() {
        vehiculos = new ArrayList<>();
    }

    public boolean registrarVehiculo(String placa, String modelo, String color) {
        for (Vehiculo v : vehiculos) {
            if (v.getPlaca().equals(placa)) {
                System.out.println("Ya existe un vehiculo con esta placa.");
                return false;
            }
        }
        vehiculos.add(new Vehiculo(placa, modelo, color));
        return true;
    }

    public Vehiculo consultarVehiculo(String placa) {
        for (Vehiculo v : vehiculos) {
            if (v.getPlaca().equals(placa)) {
```

```
            return v;
        }
    }
    return null;
}

public boolean actualizarVehiculo(String placa, String modelo, String color) {
    Vehiculo v = consultarVehiculo(placa);
    if (v != null) {
        v.modelo = modelo;
        v.color = color;
        return true;
    }
    return false;
}

public boolean eliminarVehiculo(String placa) {
    Vehiculo v = consultarVehiculo(placa);
    if (v != null) {
        vehiculos.remove(v);
        return true;
    }
    return false;
}

public void mostrarVehiculos() {
    if (vehiculos.isEmpty()) {
        System.out.println("No hay vehiculos registrados.");
    } else {
        for (Vehiculo v : vehiculos) {
            System.out.println(v);
        }
    }
}
}
```

Diagrama UML

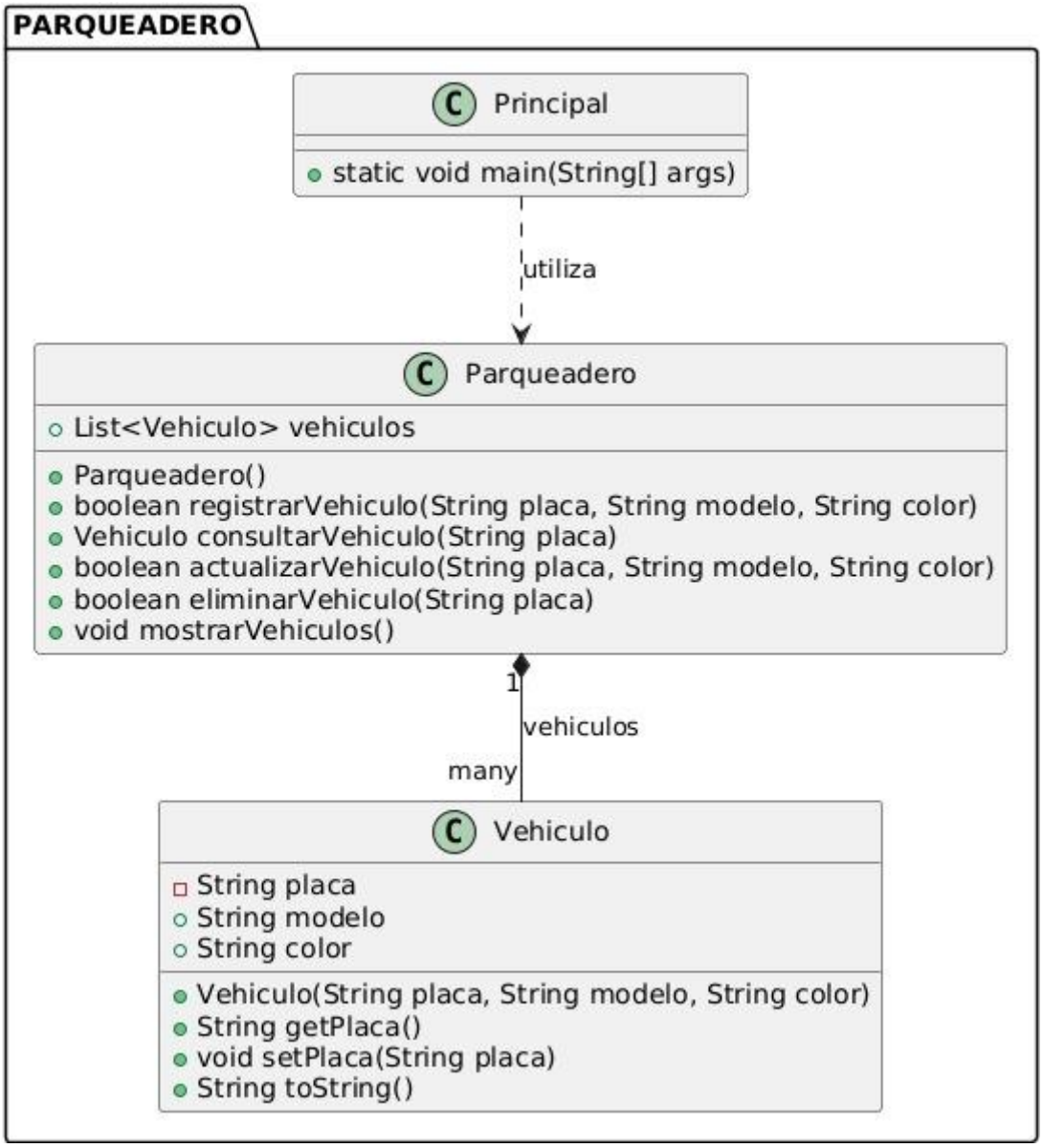

Figura 1: Diagrama UML

Implementación del Programa

El programa principal proporciona una interfaz interactiva para el usuario mediante un menú de opciones que permite registrar vehículo, consultar vehículo, actualizar vehículo, eliminar vehículo, mostrar todos los vehículos y salida del menú.

```
--- MENU ---
1. Registrar vehiculo
2. Consultar vehiculo
3. Actualizar vehiculo
4. Eliminar vehiculo
5. Mostrar todos los vehiculos
6. Salir
Elige una opcion: 5

--- Lista de Vehiculos ---
Vehiculo{placa='PCI 1786', modelo='CHEVROLET', color='NEGRO '}
```

Settings | Vehiculo.java | Parqueadero.j

PARQUEADERO > Principal.java > ...

```
5   public class Principal {
6       public static void main(String[] args) {
60                  placa = sc.nextLine();
61                  if (parqueadero.eliminarVehiculo(placa)) {
62                      System.out.println(x:"Vehiculo eliminado correctamente.");
63                  } else {
64                      System.out.println(x:"No se encontro un vehiculo con esa placa.");
65                  }
66                  break;
67              case 5:
68                  System.out.println(x:"\n--- Lista de Vehiculos ---");
69                  parqueadero.mostrarVehiculos();
70                  break;
71              case 6:
72                  System.out.println(x:"Saliendo del programa.");
73                  break;
74              default:
75                  System.out.println(x:"Opcion no valida.");
76          }
```

PROBLEMS 3 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
PS C:\Users\Usuario\Documents\programaciones>  & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.5.11-hotspot\bin\j
ionMessages' '-cp' 'C:\Users\Usuario\AppData\Roaming\Code\User\workspaceStorage\491f971ef0bc57ea92da0d87b967010
78cd07\bin' 'PARQUEADERO.Principal'

--- MENU ---
1. Registrar vehiculo
2. Consultar vehiculo
3. Actualizar vehiculo
4. Eliminar vehiculo
5. Mostrar todos los vehiculos
6. Salir
Elige una opcion: 1
Ingresa la placa: PCI 1786
Ingresa el modelo: CHEVROLET
Ingresa el color: NEGRO
Vehiculo registrado correctamente.
```

Figura 2: Programa ejecutado.

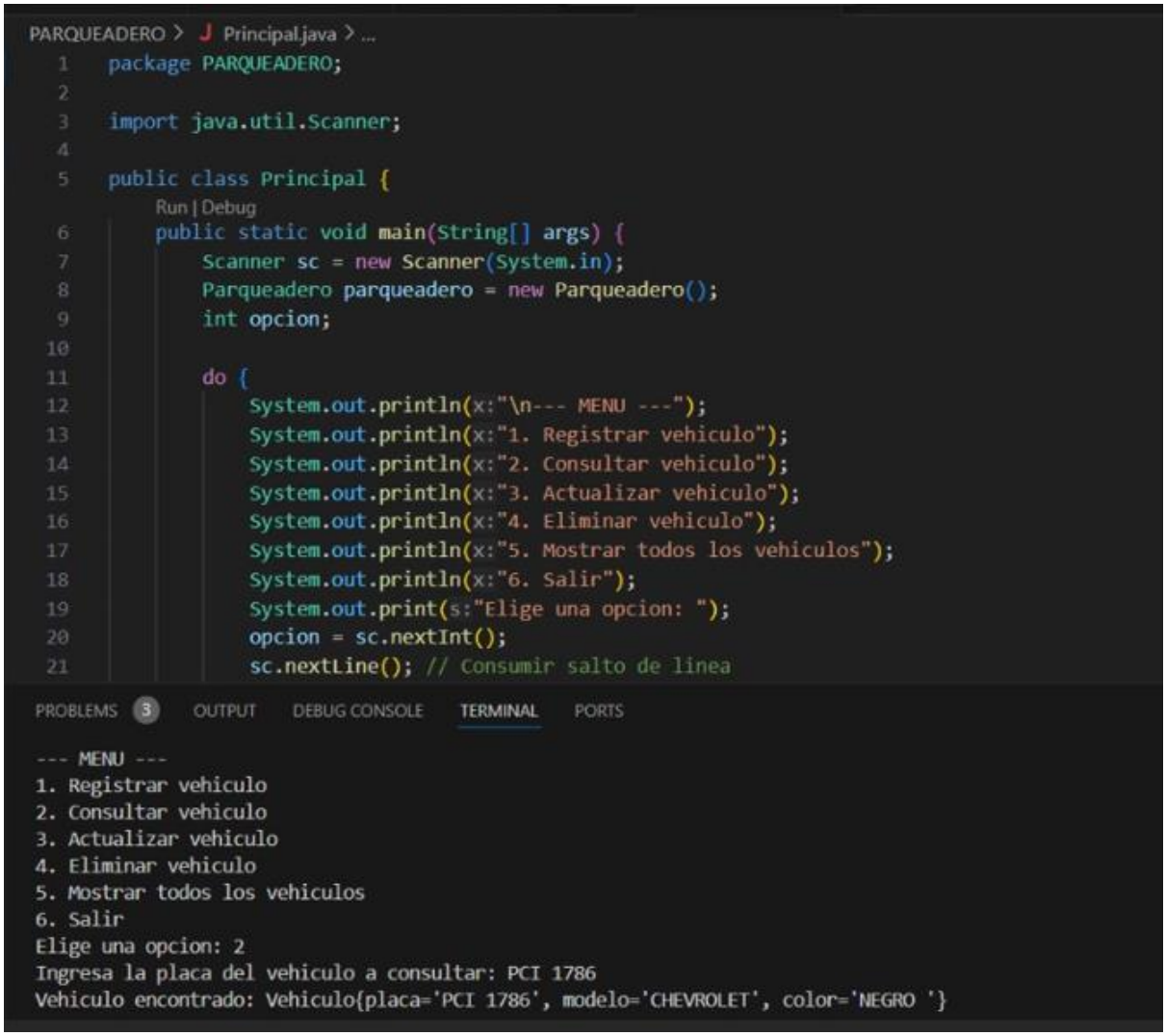

Figura 3: consulta de un vehículo.

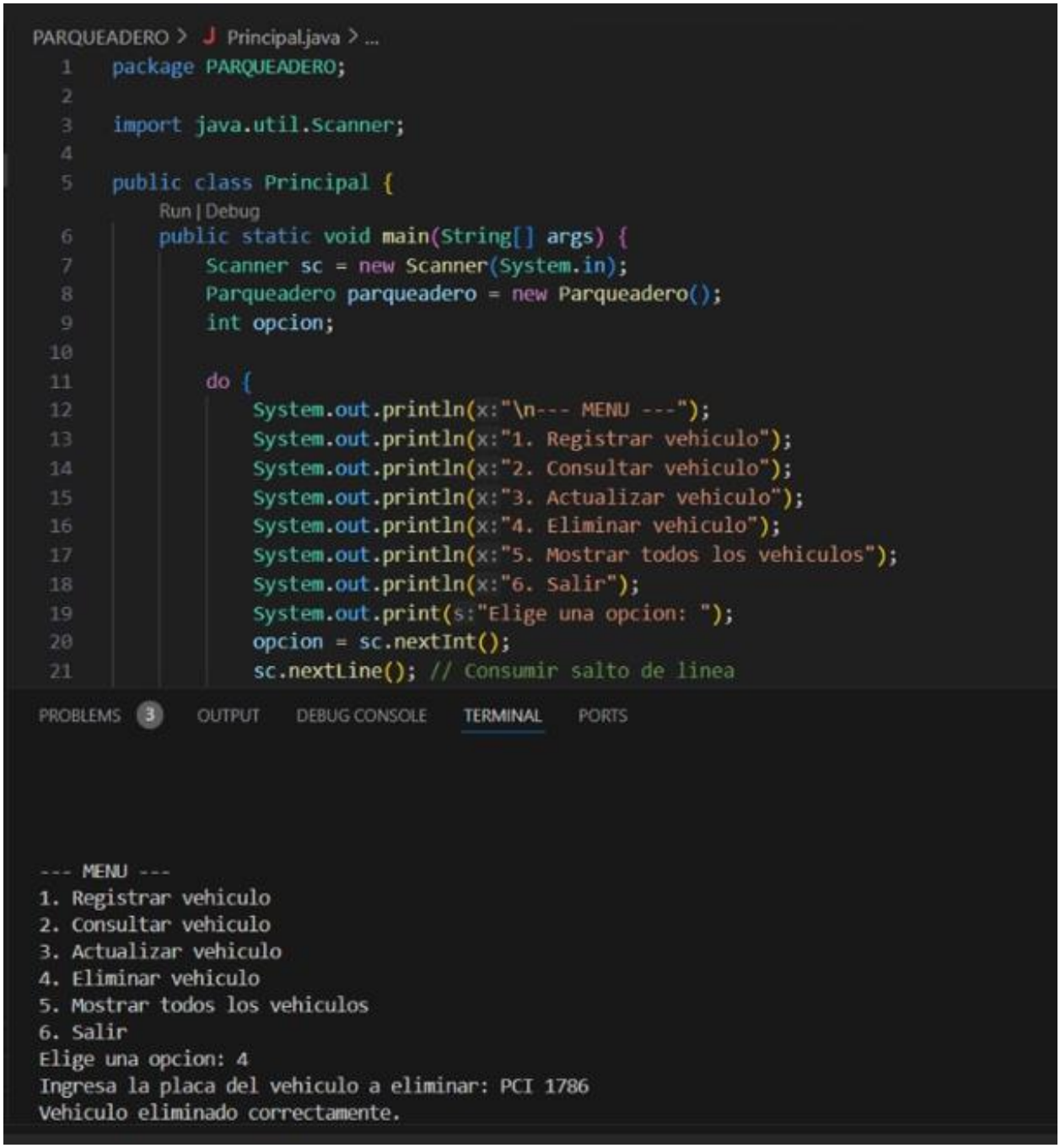

Figura 4: Eliminación de un vehículo.

Figura 5: Lista de vehículos.

Conclusiones

El sistema desarrollado permite gestionar de manera efectiva los vehículos en un parqueadero utilizando Programación Orientada a Objetos y utilizando el software Java. Se implementaron las funcionalidades básicas de registro, consulta y actualización de vehículos. Las pruebas realizadas en la clase Principal demuestran que el sistema funciona correctamente y cumple con los objetivos planteados.

Recomendaciones

- Implementar validaciones para asegurar que los datos ingresados por los usuarios sean correctos y completos.
- Implementar una forma de almacenar y recuperar los datos de los vehículos para que no se pierdan al cerrar la aplicación.
- Añadir comentarios y documentación técnica para facilitar el mantenimiento y la ampliación del sistema por futuros desarrolladores.

Bibliografía

Byrne, L. (2023, June 14). Guía completa del sistema de gestión de aparcamientos. Workero - Integrated Workplace Solutions to Optimise Hybrid Work. https://www.workero.com/es/guia-del-sistema-de-gestion-de-aparcamientos/

Canelo, M. M. (2020, November 2). ¿Qué es la Programación Orientada a Objetos? Profile Software Services. https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/

¿En qué consiste un sistema de gestión de parking? (2021, August 5). Came Soluciones. https://camesoluciones.com/blog/en-que-consiste-un-sistema-de-gestion-de-parking/