

Informe

Nombre: Robinson Josué Romero Nolivos

NRC: 1322

El código está escrito en Java y consiste en un sistema de gestión para un parqueadero. Este sistema permite registrar vehículos, consultar información de vehículos ya registrados, actualizar datos de vehículos y listar todos los vehículos en el parqueadero

Estructura General

El código se compone de tres clases principales:

Main: La clase principal que contiene el método main y gestiona la interacción con el usuario.

Vehículo: Representa un vehículo con sus atributos y métodos de acceso.

Parqueadero: Gestiona una lista de vehículos y proporciona métodos para registrar, consultar, actualizar y listar vehículos.

Main

Contiene el método main, que es el punto de entrada de la aplicación. Posee los siguientes componentes:

Scanner: Se utiliza para capturar la entrada del usuario desde la consola.

Parqueadero: Se instancia un objeto de la clase Parqueadero que se utilizará para gestionar los vehículos.

Bucle do-while: Permite mostrar un menú de opciones al usuario y ejecutar la acción correspondiente basada en la opción seleccionada.

```
import java.util.Scanner;

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

        do {
            System.out.println("1. Registrar vehiculo");
            System.out.println("2. Consultar vehiculo");
            System.out.println("3. Actualizar vehiculo");
            System.out.println("4. Listar vehiculos");
            System.out.println("0. Salir");
            System.out.print("Seleccione una opcion: ");
            opcion = scanner.nextInt();
            scanner.nextLine();
        }
```

Registrar vehículo:

Se solicita al usuario ingresar la placa, modelo y propietario del vehículo.

Se crea un nuevo objeto Vehículo y se registra en el parqueadero.

```
switch (opcion) {
case 1:
    // Registro
    System.out.print("Ingrese la placa: ");
    String placa = scanner.nextLine();
    System.out.print("Ingrese el modelo: ");
    String modelo = scanner.nextLine();
    System.out.print("Ingrese el propietario: ");
    String propietario = scanner.nextLine();
    parqueadero.registrarVehiculo(new Vehiculo(placa, modelo, propietario));
    System.out.println("Vehiculo registrado con exito.");
    break;
```

Consultar vehículo:

Se pide al usuario ingresar la placa del vehículo a consultar.

Se busca el vehículo en la lista de vehículos del parqueadero.

Si se encuentra, se muestra la información del vehículo; de lo contrario, se informa que no se encontró.

```
case 2:
    // Consulta
    System.out.print("Ingrese la placa a consultar: ");
    String placaConsulta = scanner.nextLine();
    Vehiculo vehiculoConsultado = parqueadero.consultarVehiculo(placaConsulta);
    if (vehiculoConsultado != null) {
        System.out.println("Vehiculo encontrado: " + vehiculoConsultado);
    } else {
        System.out.println("Vehiculo no encontrado.");
    }
    break;
```

Actualizar vehículo:

Se solicita al usuario la placa del vehículo a actualizar, así como el nuevo modelo y propietario.

Se busca el vehículo y, si se encuentra, se actualizan sus datos.

```
case 3:
    // ActualizaciC3n
    System.out.print("Ingrese la placa del vehiculo a actualizar: ");
    String placaActualizar = scanner.nextLine();
    System.out.print("Ingrese el nuevo modelo: ");
    String nuevoModelo = scanner.nextLine();
    System.out.print("Ingrese el nuevo propietario: ");
    String nuevoPropietario = scanner.nextLine();
    boolean actualizado = parqueadero.actualizarVehiculo(placaActualizar, nuevoModelo, nuevoPropietario);
    if (actualizado) {
        System.out.println("Vehiculo actualizado con C)xito.");
    } else {
        System.out.println("Vehiculo no encontrado.");
    }
    break;
```

Listar vehículos:

Se imprime una lista de todos los vehículos registrados en el parqueadero.

```

case 4:
    // Listar vehC-culos
    System.out.println("Lista de vehiculos:");
    for (Vehiculo v : parqueadero.listarVehiculos()) {
        System.out.println(v);
    }
    break;

```

Salir:

Termina el programa.

```

case 0:
    System.out.println("Saliendo...");
    break;

default:
    System.out.println("Opcion no valida. Intente de nuevo.");
    break;
}
} while (opcion != 0);

scanner.close();

```

Clase Vehículo

Esta clase representa un vehículo con tres atributos:

placa: Identificación del vehículo.

modelo: Modelo del vehículo.

propietario: Propietario del vehículo.

```

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

```

Métodos de la clase Vehículo

Constructor: Inicializa los atributos con los valores proporcionados.

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

Getters: Métodos para acceder a los atributos placa, modelo y propietario.

```
public String getPlaca() {  
    return placa;  
}  
  
public String getModelo() {  
    return modelo;  
}  
  
public String getPropietario() {  
    return propietario;  
}
```

Setters: Métodos para actualizar el modelo y propietario.

```
public void setModelo(String modelo) {  
    this.modelo = modelo;  
}  
  
public void setPropietario(String propietario) {  
    this.propietario = propietario;  
}
```

toString(): Método que devuelve una representación en cadena del objeto Vehículo, útil para imprimir la información del vehículo de forma legible.

```

@Override
public String toString() {
    return "Vehiculo{" +
        "placa='" + placa + '\'' +
        ", modelo='" + modelo + '\'' +
        ", propietario='" + propietario + '\'' +
        '}';
}

```

Clase Parqueadero

Esta clase es responsable de gestionar la colección de vehículos.

Atributos

vehiculos: Una lista (ArrayList) que almacena los objetos Vehiculo.

```

import java.util.ArrayList;

public class Parqueadero {
    private ArrayList<Vehiculo> vehiculos;
}

```

Métodos de la clase Parqueadero

Constructor: Inicializa la lista de vehículos.

```

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

```

registrarVehiculo(Vehiculo vehiculo): Agrega un nuevo vehículo a la lista.

```

public void registrarVehiculo(Vehiculo vehiculo) {
    vehiculos.add(vehiculo);
}

```

consultarVehiculo(String placa): Busca un vehículo por su placa y devuelve el objeto si se encuentra, o null si no.

```
public Vehiculo consultarVehiculo(String placa) {  
    for (Vehiculo v : vehiculos) {  
        if (v.getPlaca().equalsIgnoreCase(placa)) {  
            return v;  
        }  
    }  
    return null;  
}
```

actualizarVehiculo(String placa, String nuevoModelo, String nuevoPropietario): Busca un vehículo por su placa y actualiza su modelo y propietario si se encuentra, retornando un booleano que indica si la operación fue exitosa.

```
public boolean actualizarVehiculo(String placa, String nuevoModelo, String nuevoPropietario) {  
    Vehiculo vehiculo = consultarVehiculo(placa);  
    if (vehiculo != null) {  
        vehiculo.setModelo(nuevoModelo);  
        vehiculo.setPropietario(nuevoPropietario);  
        return true;  
    }  
    return false;  
}
```

listarVehiculos(): Devuelve la lista de vehículos registrados.

```
public ArrayList<Vehiculo> listarVehiculos() {  
    return vehiculos;  
}
```

Ejecución del programa:

```
1. Registrar vehiculo
2. Consultar vehiculo
3. Actualizar vehiculo
4. Listar vehiculos
0. Salir
Seleccione una opcion: 1
Ingrese la placa: POU-0721
Ingrese el modelo: Chevrolet
Ingrese el propietario: Propietario
Vehiculo registrado con exito.
1. Registrar vehiculo
2. Consultar vehiculo
3. Actualizar vehiculo
4. Listar vehiculos
0. Salir
Seleccione una opcion: 2
Ingrese la placa a consultar: POU-0721
Vehiculo encontrado: Vehiculo{placa='POU-0721', modelo='Chevrolet', propietario='Propietario'}
1. Registrar vehiculo
2. Consultar vehiculo
3. Actualizar vehiculo
4. Listar vehiculos
0. Salir
Seleccione una opcion: 4
Lista de vehiculos:
Vehiculo{placa='POU-0721', modelo='Chevrolet', propietario='Propietario'}
1. Registrar vehiculo
2. Consultar vehiculo
3. Actualizar vehiculo
4. Listar vehiculos
0. Salir
```