

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

SEGUNDO SEMESTRE

PRIMER PARCIAL

PROGRAMACIÓN ORIENTADA A OBJETOS

SISTEMA PARA GESTIONAR UN PARQUEADERO

Link GitHub: <https://github.com/Scarleth20723/Informe-.git>

Alumna:

Loachamin Tipan Nayeli Scarleth

NRC:

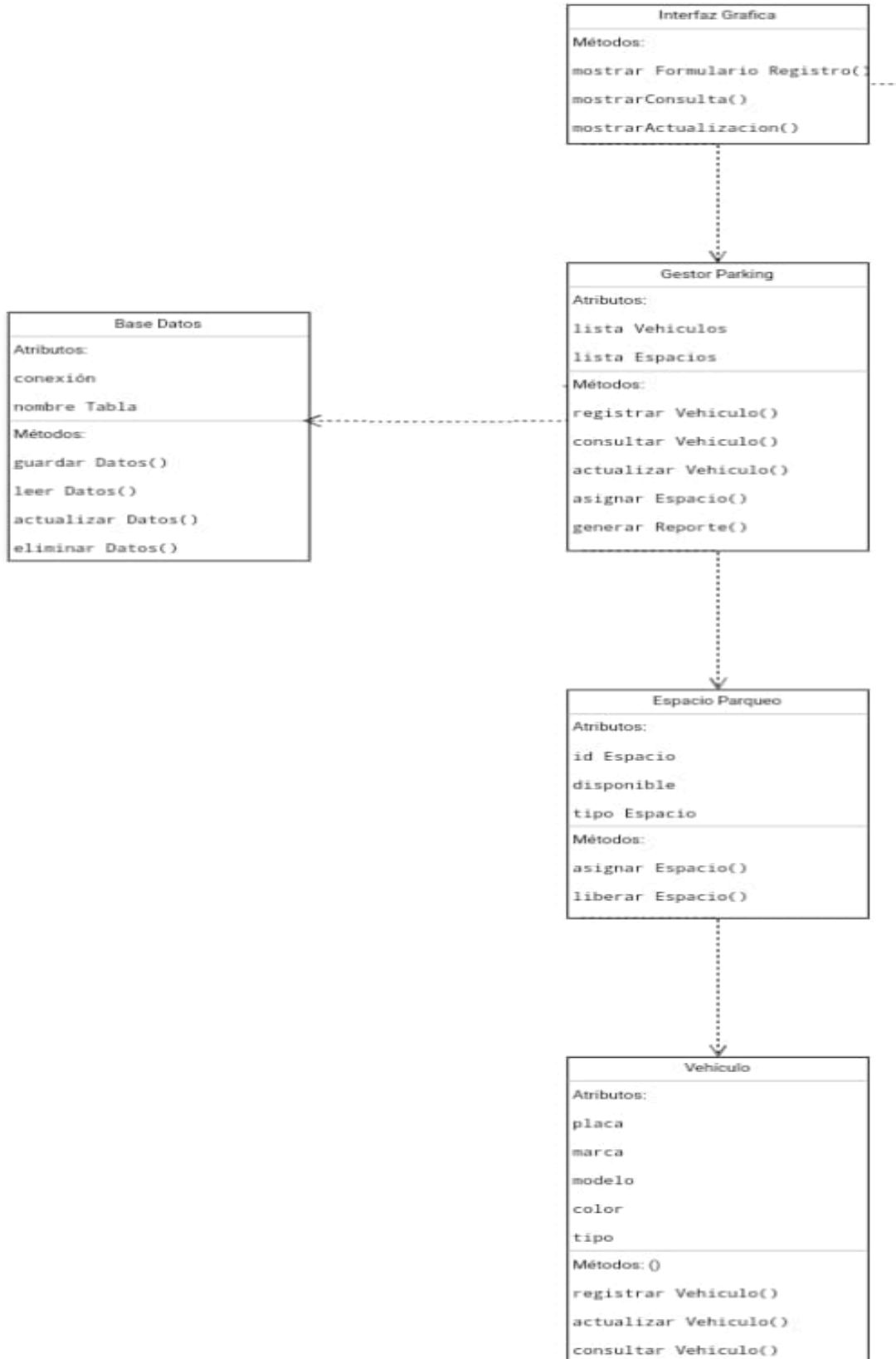
1323

DOCENTE:

Luis Enrique Jaramillo Montaña

SANGOLQUÍ - ECUADOR

## GRÁFICO UML



## ***CÓDIGO INTERFAZ***

```
package com.mycompany.interfazparquadero;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;

public class InterfazParquadero extends JFrame{

    public JTextField placaTextField;

    public JTextField marcaTextField;

    public JTextField modeloTextField;

    public JComboBox<String> tipoVehiculoComboBox;

    public JTextField propietarioTextField;
```

```
public JTextArea resultadosTextArea;
```

```
public InterfazParquero() {
```

```
    // Configuración básica de la ventana
```

```
    setTitle("Sistema de Gestión de Parquero");
```

```
    setSize(600, 500);
```

```
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    setLocationRelativeTo(null);
```

```
    // Panel principal
```

```
    JPanel panelPrincipal = new JPanel(new BorderLayout(10, 10));
```

```
    panelPrincipal.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
```

```
    // Panel de entrada de datos
```

```
    JPanel panelEntrada = new JPanel(new GridLayout(6, 2, 10, 10));
```

```
    panelEntrada.setBorder(BorderFactory.createTitledBorder("Registro de Vehículos"));
```

```
    // Componentes de entrada
```

```
    panelEntrada.add(new JLabel("Placa:"));
```

```
    placaTextField = new JTextField();
```

```
    panelEntrada.add(placaTextField);
```

```
    panelEntrada.add(new JLabel("Marca:"));
```

```
    marcaTextField = new JTextField();
```

```
    panelEntrada.add(marcaTextField);
```

```
    panelEntrada.add(new JLabel("Modelo:"));
```

```
modeloTextField = new JTextField();

panelEntrada.add(modeloTextField);


panelEntrada.add(new JLabel("Tipo de Vehículo:"));

String[] tiposVehiculo = {"Carro", "Moto", "Camioneta"};

tipoVehiculoComboBox = new JComboBox<>(tiposVehiculo);

panelEntrada.add(tipoVehiculoComboBox);


panelEntrada.add(new JLabel("Propietario:"));

propietarioTextField = new JTextField();

panelEntrada.add(propietarioTextField);


// Panel de botones

JPanel panelBotones = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));

JButton registrarButton = new JButton("Registrar");

JButton consultarButton = new JButton("Consultar");

JButton actualizarButton = new JButton("Actualizar");

JButton limpiarButton = new JButton("Limpiar");


panelBotones.add(registrarButton);

panelBotones.add(consultarButton);

panelBotones.add(actualizarButton);

panelBotones.add(limpiarButton);


// Área de resultados

resultadosTextArea = new JTextArea();

resultadosTextArea.setEditable(false);
```

```
JScrollPane scrollPane = new JScrollPane(resultadosTextArea);  
scrollPane.setBorder(BorderFactory.createTitledBorder("Resultados"));
```

```
// Agrego componentes al panel principal
```

```
panelPrincipal.add(panelEntrada, BorderLayout.NORTH);  
panelPrincipal.add(panelBotones, BorderLayout.CENTER);  
panelPrincipal.add(scrollPane, BorderLayout.SOUTH);
```

```
// Agrego listeners de eventos
```

```
registrarButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        registrarVehiculo();  
    }  
});
```

```
consultarButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        consultarVehiculo();  
    }  
});
```

```
actualizarButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        actualizarVehiculo();  
    }  
});
```

```
}  
});
```

```
limpiarButton.addActionListener(new ActionListener() {  
  
    @Override  
  
    public void actionPerformed(ActionEvent e) {  
  
        limpiarCampos();  
  
    }  
  
});
```

```
// Agrego panel principal a la ventana  
  
add(panelPrincipal);  
  
}
```

```
public void registrarVehiculo() {  
  
    // Método para registrar vehículo  
  
    String mensaje = "Registrando vehículo:\n" +  
  
        "Placa: " + placaTextField.getText() + "\n" +  
  
        "Marca: " + marcaTextField.getText() + "\n" +  
  
        "Modelo: " + modeloTextField.getText() + "\n" +  
  
        "Tipo: " + tipoVehiculoComboBox.getSelectedItem() + "\n" +  
  
        "Propietario: " + propietarioTextField.getText();  
  
  
    resultadosTextArea.setText(mensaje);  
  
    JOptionPane.showMessageDialog(this, "Vehículo registrado exitosamente",  
  
        "Registro Exitoso", JOptionPane.INFORMATION_MESSAGE);  
  
}
```

```
public void consultarVehiculo() {  
    // Método para consultar vehículo  
    String placa = placaTextField.getText();  
    if (!placa.isEmpty()) {  
        resultadosTextArea.setText("Consultando vehículo con placa: " + placa);  
    } else {  
        JOptionPane.showMessageDialog(this, "Ingrese una placa para consultar",  
            "Error", JOptionPane.ERROR_MESSAGE);  
    }  
}  
  
public void actualizarVehiculo() {  
    // Método para actualizar vehículo  
    String mensaje = "Actualizando información del vehículo:\n" +  
        "Placa: " + placaTextField.getText();  
  
    resultadosTextArea.setText(mensaje);  
    JOptionPane.showMessageDialog(this, "Información actualizada",  
        "Actualización", JOptionPane.INFORMATION_MESSAGE);  
}  
  
public void limpiarCampos() {  
    placaTextField.setText("");  
    marcaTextField.setText("");  
    modeloTextField.setText("");  
    tipoVehiculoComboBox.setSelectedIndex(0);
```



```
    propietarioTextField.setText("");  
    resultadosTextArea.setText("");  
}  
  
public static void main(String[] args) {  
    // Uso Event Dispatch Thread para GUI  
    SwingUtilities.invokeLater(new Runnable() {  
        @Override  
        public void run() {  
            new InterfazParquadero().setVisible(true);  
        }  
    });  
}  
}
```

```

File Edit Selection View Go Run ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

J InterfazParqueadero.java X
Users > darwi > AppData > Local > Temp > 123943e6-9800-40fa-9979-0c8e6da64447_InterfazParqueadero[1].zip.447 > InterfazParqueadero > src > main > java > com > mycompany > interf

21 public class InterfazParqueadero extends JFrame{
30 public InterfazParqueadero() {
60 tipoVehiculoComboBox = new JComboBox<>(tiposVehiculo);
61 panelEntrada.add(tipoVehiculoComboBox);
62
63 panelEntrada.add(new JLabel("Propietario:"));
64 propietarioTextField = new JTextField();
65 panelEntrada.add(proprietarioTextField);
66
67 // Panel de botones
68 JPanel panelBotones = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));
69 JButton registrarButton = new JButton("Registrar");
70 JButton consultarButton = new JButton("Consultar");
71 JButton actualizarButton = new JButton("Actualizar");
72 JButton limpiarButton = new JButton("Limpiar");
73
74 panelBotones.add(registrarButton);
75 panelBotones.add(consultarButton);
76 panelBotones.add(actualizarButton);
77 panelBotones.add(limpiarButton);
78
79 // Área de resultados
80 resultadosTextArea = new JTextArea();
81 resultadosTextArea.setEditable(false);
82 JScrollPane scrollPane = new JScrollPane(resultadosTextArea);
83 scrollPane.setBorder(BorderFactory.createTitledBorder("Resultados"));
84
85 // Agrego componentes al panel principal
86 panelPrincipal.add(panelEntrada, BorderLayout.NORTH);
87 panelPrincipal.add(panelBotones, BorderLayout.CENTER);

```

```

File Edit Selection View Go Run ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

J InterfazParqueadero.java X
Users > darwi > AppData > Local > Temp > 123943e6-9800-40fa-9979-0c8e6da64447_InterfazParqueadero[1].zip.447 > InterfazParqueadero > src > main > java > com > mycompany > interfazparqueadero

21 public class InterfazParqueadero extends JFrame{
30 public InterfazParqueadero() {
88 panelPrincipal.add(scrollPane, BorderLayout.SOUTH);
89
90 // Agrego listeners de eventos
91 registrarButton.addActionListener(new ActionListener() {
92     @Override
93     public void actionPerformed(ActionEvent e) {
94         registrarVehiculo();
95     }
96 });
97
98 consultarButton.addActionListener(new ActionListener() {
99     @Override
100     public void actionPerformed(ActionEvent e) {
101         consultarVehiculo();
102     }
103 });
104
105 actualizarButton.addActionListener(new ActionListener() {
106     @Override
107     public void actionPerformed(ActionEvent e) {
108         actualizarVehiculo();
109     }
110 });
111
112 limpiarButton.addActionListener(new ActionListener() {
113     @Override
114     public void actionPerformed(ActionEvent e) {
115         limpiarCampos();

```

```

File Edit Selection View Go Run ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

J InterfazParqueadero.java X
Users > darwi > AppData > Local > Temp > 123943e6-9800-40fa-9979-0c8e6da64447_InterfazParqueadero[1].zip.447 > InterfazParqueadero > src > main > java > com > mycompa

21 public class InterfazParqueadero extends JFrame{
30     public InterfazParqueadero() {
112         // Agrego panel principal a la ventana
117         add(panelPrincipal);
118     }
119
120     public void registrarVehiculo() {
121         // Metodo para registrar vehiculo
122         String mensaje = "Registrando vehiculo:\n" +
123             "Placa: " + placaTextField.getText() + "\n" +
124             "Marca: " + marcaTextField.getText() + "\n" +
125             "Modelo: " + modeloTextField.getText() + "\n" +
126             "Tipo: " + tipoVehiculoComboBox.getSelectedItem() + "\n" +
127             "Propietario: " + propietarioTextField.getText();
128
129         resultadosTextArea.setText(mensaje);
130         JOptionPane.showMessageDialog(this, "Vehiculo registrado exitosamente",
131             "Registro Exitoso", JOptionPane.INFORMATION_MESSAGE);
132     }
133
134     public void consultarVehiculo() {
135         // Metodo para consultar vehiculo
136         String placa = placaTextField.getText();
137         if (!placa.isEmpty()) {
138             resultadosTextArea.setText("Consultando vehiculo con placa: " + placa);
139         } else {
140             JOptionPane.showMessageDialog(this, "Ingrese una placa para consultar",
141                 "Error", JOptionPane.ERROR_MESSAGE);
142         }
143     }
144 }

```

```

File Edit Selection View Go Run ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

J InterfazParqueadero.java X
Users > darwi > AppData > Local > Temp > 123943e6-9800-40fa-9979-0c8e6da64447_InterfazParqueadero[1].zip.447 > InterfazParqueadero > src > main > java

21 public class InterfazParqueadero extends JFrame{
137     public void consultarVehiculo() {
145     }
146 }
147
148     public void actualizarVehiculo() {
149         // Metodo para actualizar vehiculo
150         String mensaje = "Actualizando informacion del vehiculo:\n" +
151             "Placa: " + placaTextField.getText();
152
153         resultadosTextArea.setText(mensaje);
154         JOptionPane.showMessageDialog(this, "Informacion actualizada",
155             "Actualizacion", JOptionPane.INFORMATION_MESSAGE);
156     }
157
158     public void limpiarCampos() {
159         placaTextField.setText("");
160         marcaTextField.setText("");
161         modeloTextField.setText("");
162         tipoVehiculoComboBox.setSelectedIndex(0);
163         propietarioTextField.setText("");
164         resultadosTextArea.setText("");
165     }
166
167     public static void main(String[] args) {
168         // Uso Event Dispatch Thread para GUI
169         SwingUtilities.invokeLater(new Runnable() {
170             @Override
171             public void run() {
172                 new InterfazParqueadero().setVisible(true);
173             }
174         });
175     }

```

## ***CÓDIGO DE BASE DE DATOS***

```
import java.sql.*;

class ConexionBD {

    static final String URL = "jdbc:mysql://localhost:3306/gestion_parquadero";

    static final String USUARIO = "root"; // Cambiar según configuración

    static final String CONTRASENA = ""; // Cambiar según configuración


    static Connection obtenerConexion() throws SQLException {

        return DriverManager.getConnection(URL, USUARIO, CONTRASENA);

    }

}

class VehiculoDAO {

    static void registrarVehiculo(String placa, String tipo, String marca, String modelo) {

        String sql = "INSERT INTO Vehiculos (placa, tipo, marca, modelo) VALUES (?, ?, ?, ?)";

        try (Connection conn = ConexionBD.obtenerConexion();

            PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, placa);

            stmt.setString(2, tipo);

            stmt.setString(3, marca);

            stmt.setString(4, modelo);

            stmt.executeUpdate();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}
```

```
static void consultarVehiculos() {

    String sql = "SELECT * FROM Vehiculos";

    try (Connection conn = ConexionBD.obtenerConexion();

        Statement stmt = conn.createStatement();

        ResultSet rs = stmt.executeQuery(sql)) {

        while (rs.next()) {

            System.out.println("Placa: " + rs.getString("placa") +

                                ", Tipo: " + rs.getString("tipo") +

                                ", Marca: " + rs.getString("marca") +

                                ", Modelo: " + rs.getString("modelo"));

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

}

class ParqueaderoApp {

    static void main(String[] args) {

        VehiculoDAO.registrarVehiculo("ABC123", "Automovil", "Toyota", "Corolla");

        VehiculoDAO.registrarVehiculo("XYZ789", "Moto", "Honda", "CBR500R");

        System.out.println("Listado de vehiculos registrados:");

        VehiculoDAO.consultarVehiculos();

    }

}
```

```
File Edit Selection View Go Run ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
base_de_datos[1].java X
C:\Users\darwi> AppData\Local\Microsoft\Windows\INetCache\IE\ZZCMXCKV\base_de_datos[1].java
1 import java.sql.*;
2
3 class ConexionBD {
4     static final String URL = "jdbc:mysql://localhost:3306/gestion_parqueadero";
5     static final String USUARIO = "root"; // Cambiar según configuración
6     static final String CONTRASEÑA = ""; // Cambiar según configuración
7
8     static Connection obtenerConexion() throws SQLException {
9         return DriverManager.getConnection(URL, USUARIO, CONTRASEÑA);
10    }
11 }
12
13 class VehiculoDAO {
14     static void registrarVehiculo(String placa, String tipo, String marca, String modelo) {
15         String sql = "INSERT INTO Vehiculos (placa, tipo, marca, modelo) VALUES (?, ?, ?, ?)";
16         try (Connection conn = ConexionBD.obtenerConexion();
17             PreparedStatement stmt = conn.prepareStatement(sql)) {
18             stmt.setString(1, placa);
19             stmt.setString(2, tipo);
20             stmt.setString(3, marca);
21             stmt.setString(4, modelo);
22             stmt.executeUpdate();
23         } catch (SQLException e) {
24             e.printStackTrace();
25         }
26     }
27
28     static void consultarVehiculos() {
29         String sql = "SELECT * FROM Vehiculos";
30         try (Connection conn = ConexionBD.obtenerConexion();
```

```
File Edit Selection View Go Run ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
base_de_datos[1].java X
C:\Users\darwi> AppData\Local\Microsoft\Windows\INetCache\IE\ZZCMXCKV\base_de_datos[1].java
13 class VehiculoDAO {
28     static void consultarVehiculos() {
30         try (Connection conn = ConexionBD.obtenerConexion();
31             Statement stmt = conn.createStatement();
32             ResultSet rs = stmt.executeQuery(sql)) {
33             while (rs.next()) {
34                 System.out.println("Placa: " + rs.getString("placa") +
35                                     " Tipo: " + rs.getString("tipo") +
36                                     " Marca: " + rs.getString("marca") +
37                                     " Modelo: " + rs.getString("modelo"));
38             }
39         } catch (SQLException e) {
40             e.printStackTrace();
41         }
42     }
43 }
44
45 class ParqueaderoApp {
46     static void main(String[] args) {
47         VehiculoDAO.registrarVehiculo("ABC123", "Automovil", "Toyota", "Corolla");
48         VehiculoDAO.registrarVehiculo("XYZ789", "Moto", "Honda", "CBR500R");
49
50         System.out.println("Listado de veh[ic]ulos registrados:");
51         VehiculoDAO.consultarVehiculos();
52     }
53 }
54 }
```