



**UNIVERSIDAD DE LAS FUERZAS ARMADAS**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**

**MATERIA**

**PROGRAMACION ORIENTADA A OBJETOS**

**TEMA**

**SISTEMA DE GESTION DE PARKING TIPO**

**ALUMNO**

- Emerson Coro
- Andrea Castro
- Alexis Murminacho

**NRC 1323**



## **Introducción:**

El presente informe describe el desarrollo de un sistema para la gestión de un parqueadero. Este sistema ha sido diseñado para integrar funcionalidades de registro, consulta y actualización de vehículos, haciendo uso de principios de Programación Orientada a Objetos (POO), una interfaz gráfica amigable y bases de datos robustas. El objetivo principal es ofrecer una solución eficiente y moderna para la administración de parqueaderos.

## **Los objetivos del proyecto son:**

1. Diseñar e implementar un sistema que permita gestionar el ingreso, consulta y actualización de vehículos en un parqueadero.
2. Utilizar POO para estructurar el sistema de manera modular y escalable.
3. Desarrollar una interfaz gráfica intuitiva que facilite la interacción del usuario.
4. Implementar una base de datos que garantice el almacenamiento seguro y eficiente de la información.

## **La metodología de la investigación:**

1. Análisis de Requisitos: Se identificaron las necesidades principales del sistema y se definieron las funcionalidades requeridas.
2. Diseño del Sistema: Se creó un modelo conceptual utilizando diagramas de clases y casos de uso para estructurar el sistema bajo el paradigma de POO.



3. Implementación: Se desarrolló el código fuente del sistema, integrando la interfaz gráfica con el manejo de datos en la base de datos.
4. Pruebas: Se realizaron pruebas funcionales para garantizar la correcta operación de las funcionalidades implementadas.

## Resumen del problema y solución

**Problema:** La actividad plantea la necesidad de desarrollar un sistema para la gestión de un parqueadero. Esto incluye funcionalidades esenciales como:

1. **Registro de vehículos:** Permitir almacenar información de los vehículos que ingresan, como placa, marca, modelo, color y hora de ingreso.
2. **Consulta de vehículos:** Obtener una lista de todos los vehículos registrados en el sistema.
3. **Actualización de vehículos:** Modificar la información de un vehículo existente, como cambios en sus datos o correcciones.

El sistema debe contar con una **interfaz gráfica amigable** para el usuario y **almacenamiento persistente** en una base de datos.

**Solución:** El programa desarrollado aborda estos requerimientos de la siguiente manera:



## 1. Estructura en POO (Programación Orientada a Objetos):

- Se define una clase Vehiculo que representa la entidad principal del sistema, con atributos como placa, marca, modelo, color, horaIngreso y un identificador único (id).
- La clase BaseDatos maneja las interacciones con una base de datos SQLite para registrar, consultar y actualizar vehículos.

## 2. Interfaz gráfica:

- La clase principal, ParqueaderoApp, utiliza Swing para crear una interfaz de usuario con:
  - Campos de texto para ingresar información de los vehículos.
  - Botones para realizar acciones como registrar, consultar y editar vehículos.
  - Un área de texto donde se muestran los resultados de las consultas o mensajes del sistema.

## 3. Conexión con base de datos:

- Se utiliza SQLite como base de datos para almacenar la información de los vehículos. Esto asegura que los datos se mantengan incluso si se cierra el programa.
- Operaciones como **registro**, **consulta** y **edición** se implementan mediante consultas SQL.



## Cómo funciona el sistema:

- El usuario ingresa los datos del vehículo en los campos de texto y presiona el botón "Registrar Vehículo" para guardarlo en la base de datos.
- Al presionar "Consultar Vehículos", se muestra una lista de todos los vehículos almacenados.
- Si se requiere editar un vehículo, el usuario ingresa el ID correspondiente junto con los nuevos datos y presiona "Editar Vehículo". El sistema actualiza los registros en la base de datos.

**Beneficio del programa:** Este sistema proporciona una solución completa y eficiente para gestionar un parqueadero, asegurando que los datos de los vehículos estén organizados, sean accesibles y puedan actualizarse fácilmente. La interfaz gráfica lo hace intuitivo para los usuarios, mientras que la base de datos garantiza la persistencia de los datos.

## Descripción del Sistema:

- **Registro de Vehículos:** Permite ingresar información sobre los vehículos que acceden al parqueadero, incluyendo detalles como placa, modelo y hora de ingreso.
- **Consulta de Vehículos:** Facilita la búsqueda de vehículos registrados utilizando diferentes criterios.
- **Actualización de Datos:** Permite modificar la información de los vehículos o registrar su salida.



El sistema desarrollado cumple con los objetivos establecidos, proporcionando una plataforma funcional y eficiente para la gestión de parqueaderos. Las pruebas realizadas confirmaron la integridad de los datos, la facilidad de uso de la interfaz y el correcto funcionamiento de las funcionalidades implementadas.

**Conclusiones:** El proyecto demostró la importancia de la aplicación de POO y el uso de bases de datos para el desarrollo de sistemas modulares y escalables. Además, el uso de una interfaz gráfica amigable mejora significativamente la experiencia del usuario, haciendo que el sistema sea una herramienta útil para la administración de parqueaderos.

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
```

```
/* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package parqueaderosistema;

/**
 *
 * @author Emerson
 */
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class BaseDatos {

    public Connection conectar() {
        try {
            return DriverManager.getConnection("jdbc:sqlite:parqueadero.db");
        } catch (SQLException e) {
            System.out.println("Error de conexión: " + e.getMessage());
            return null;
        }
    }

    public boolean registrarVehiculo(Vehiculo vehiculo) {
        String sql = "INSERT INTO Vehiculos (placa, marca, modelo, color, ho";
        try (Connection conn = conectar(); PreparedStatement pstmt = conn.pr
            pstmt.setString(1, vehiculo.placa);
            pstmt.setString(2, vehiculo.marca);
            pstmt.setString(3, vehiculo.modelo);
        }
    }
}
```

parqueaderosistema.BaseDatos

Herramienta Recortes  
Captura de pantalla copiada en el portapapeles  
Guardado automáticamente en la carpeta de  
capturas de pantalla.  
Activar Windows  
Ve a Config Markup and share var Windows.

621 INS Windows (CTRL)

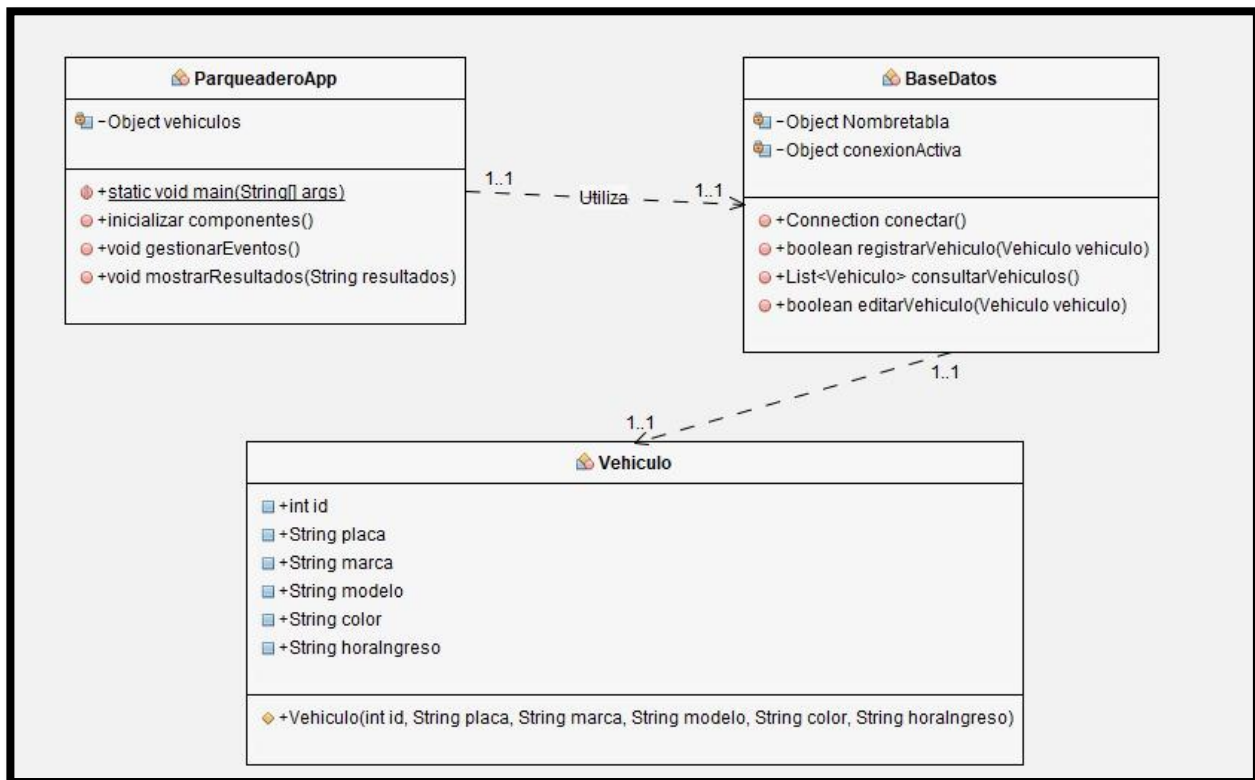


# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package parqueaderosistema;
6
7  /**
8   *
9   * @author Emerson
10  */
11  import javax.swing.*;
12  import java.awt.*;
13  import java.awt.event.*;
14  import java.util.List;
15
16  public class ParqueaderoApp {
17      public static void main(String[] args) {
18          JFrame frame = new JFrame("Gestión de Parqueadero");
19          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20          frame.setSize(500, 400);
21          frame.setLayout(new BorderLayout());
22
23          BaseDatos db = new BaseDatos();
24
25          // Panel de formulario
26          JPanel panelForm = new JPanel(new GridLayout(6, 2));
27          JTextField txtPlaca = new JTextField();
28          JTextField txtMarca = new JTextField();
29          JTextField txtModelo = new JTextField();
30          JTextField txtColor = new JTextField();
```

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package parqueaderosistema;
6
7  /**
8   *
9   * @author Emerson
10  */
11  public class Vehiculo {
12      public int id;
13      public String placa;
14      public String marca;
15      public String modelo;
16      public String color;
17      public String horaIngreso;
18
19      public Vehiculo(int id, String placa, String marca, String modelo, String color, String horaIngreso) {
20          this.id = id;
21          this.placa = placa;
22          this.marca = marca;
23          this.modelo = modelo;
24          this.color = color;
25          this.horaIngreso = horaIngreso;
26      }
27  }
```



LINK REPOSITORIO:

<https://github.com/Andre17C/DeberesAndrea2.git>