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:

- Diseñar e implementar un sistema que permita gestionar el ingreso, consulta y actualización de vehículos en un parqueadero.
- Utilizar POO para estructurar el sistema de manera modular y escalable.
- Desarrollar una interfaz gráfica intuitiva que facilite la interacción del usuario.
- 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:

- **Registro de vehículos**: Permitir almacenar información de los vehículos que ingresan, como placa, marca, modelo, color y hora de ingreso.
- **Consulta de vehículos**: Obtener una lista de todos los vehículos registrados en el sistema.
- **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.

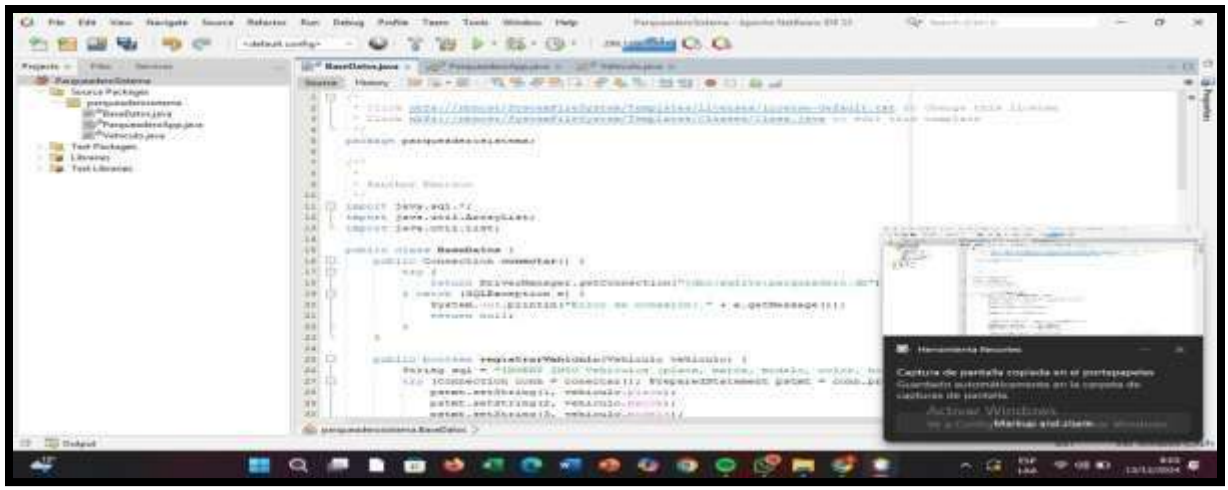
Descripción del Sistema:

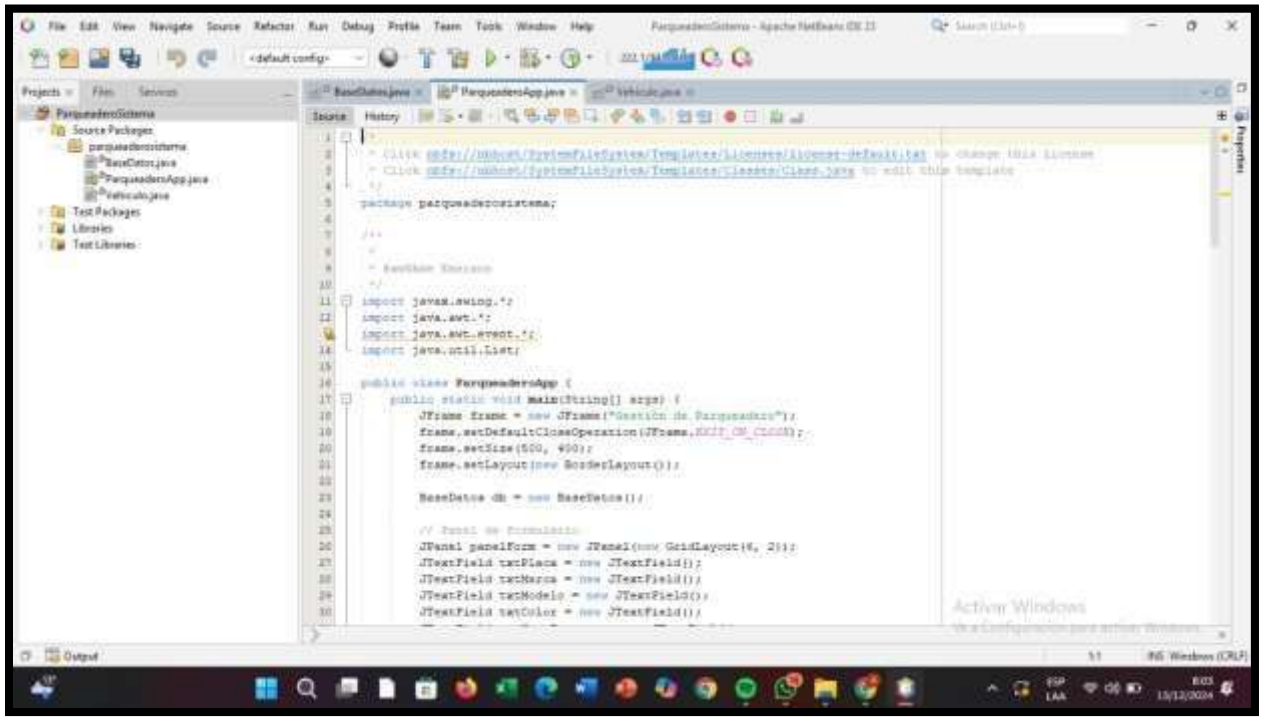
- **Registro de Vehículos:** Permite ingresar información sobre los vehículos que acceden al parqueadero, incluyendo detalles como placa, modelo y hora de ingreso.
- **Consulta de Vehículos:** Facilita la búsqueda de vehículos registrados utilizando diferentes criterios.
- **Actualización de Datos:**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

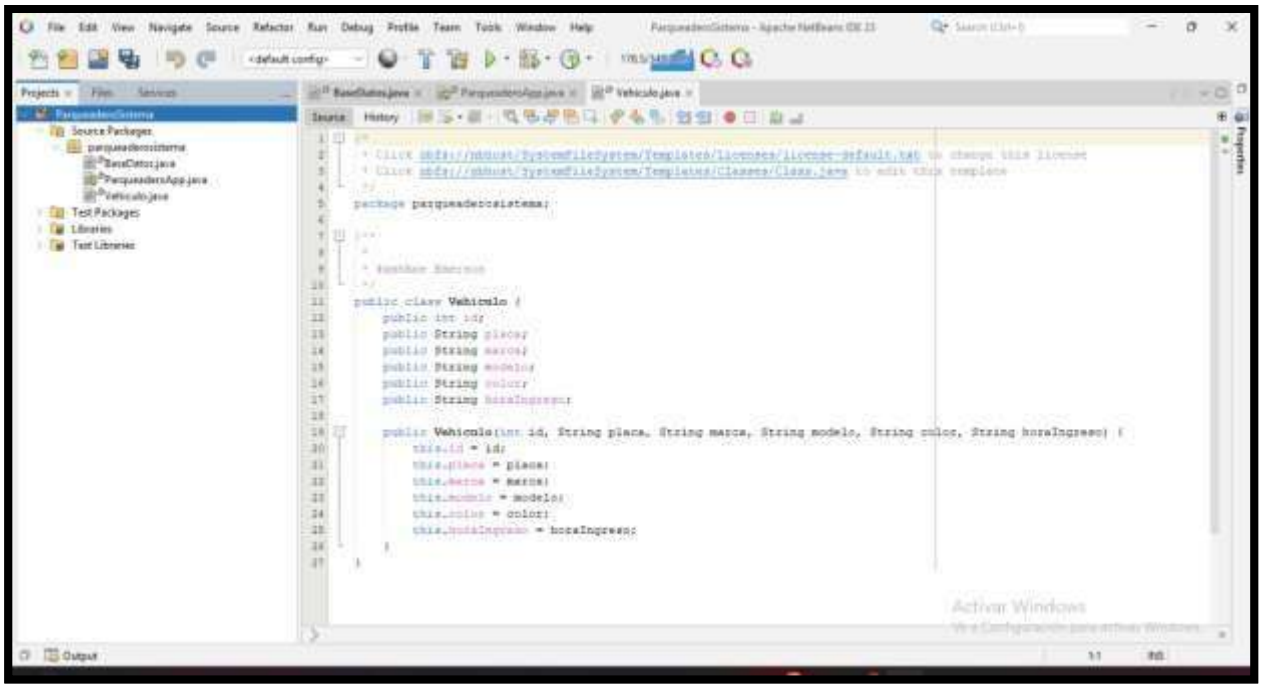
El sistema desarrollado cumple con los objetivos establecidos, proporcionando una plataforma funcional y eficiente para la gestión de parqueaderos. Las pruebas realizadas confirmaron la integridad de los datos, la facilidad de uso de la interfaz y el correcto funcionamiento de las funcionalidades implementadas.

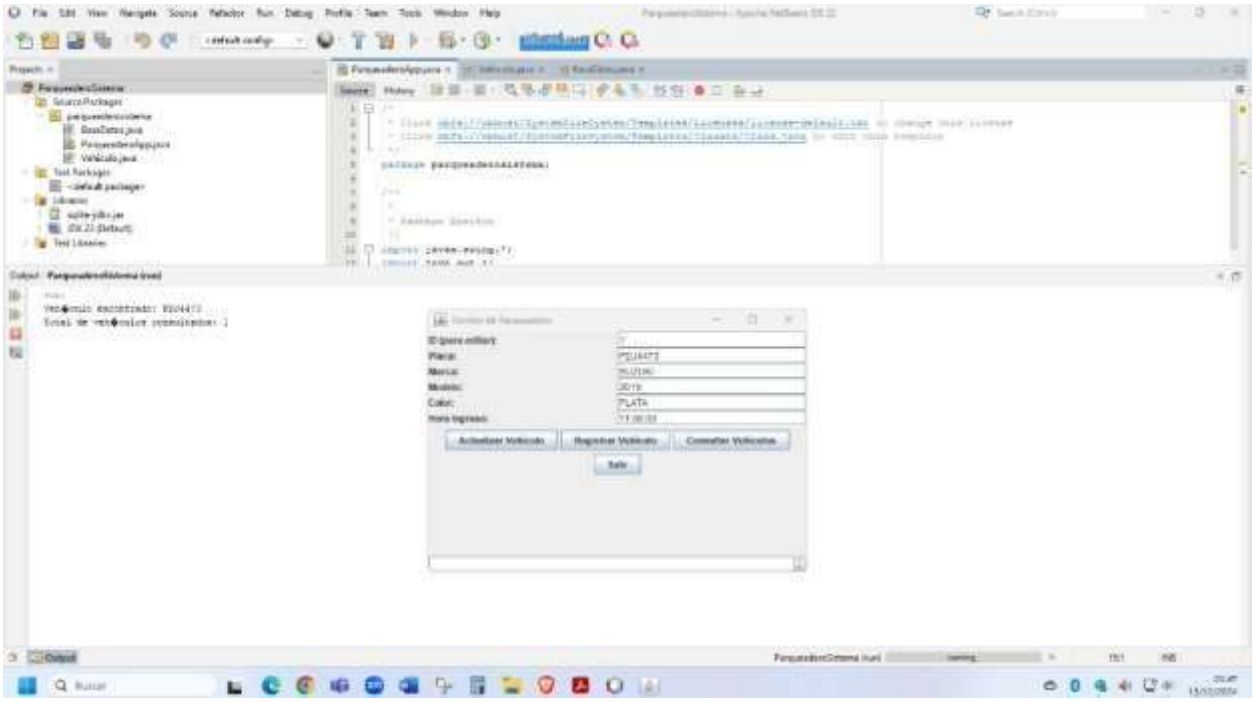
Conclusiones

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.






File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
ParqueaderoSistema
Source Packages
parqueaderosistema
BaseDatos.java
ParqueaderoApp.java
Vehiculo.java
Test Packages
Libraries
Test Libraries
Source
History
package parqueaderosistema;
public class Vehiculo {
public int id;
public String placa;
public String marca;
public String modelo;
public String color;
public String horaIngreso;
public Vehiculo(int id, String placa, String marca, String modelo, String color, String horaIngreso) {
this.id = id;
this.placa = placa;
this.marca = marca;
this.modelo = modelo;
this.color = color;
this.horaIngreso = horaIngreso;
}
}
Activar Windows
Output


ID (para editar):
Placa:
Marca:
Modelo:
Color:
PLATA
Hora Ingreso:
Actualizar Vehículo
Registrar Vehículo
Consultar Vehículos
Salir

BIBLIOGRAFIA

GeeksforGeeks. (2024, 28 noviembre). *Java tutorial*. GeeksforGeeks. https://www.geeksforgeeks.org/java/?form=MG0AV3

Swing Introduction - javatpoint. (s. f.). www.javatpoint.com. https://www.javatpoint.com/java-swing

Anónimo, U. (2024, 6 noviembre). *Tutorial básico de bases de datos en Java mediante JDBC - Adictos al trabajo*. Adictos Al Trabajo. https://adictosaltrabajo.com/2011/02/25/tutorial-basico-jdbc/