

METODOS DE BUSQUEDA Y ORDENAMIENTO

*Departamento de Ingeniería de Sistemas
Universidad de Córdoba, Sahagún*

LOGISTICA AUTOMOTOR

Elaborado por:

Hector Luis Acevedo Contreras

Andrés Eduardo Cárdenas Ramos

Docente:

Ing. Pedro Rafael Guevara Salgado

2025

Contenido

LOGÍSTICA AUTOMOTOR	3
INTRODUCCION	3
OBJETIVOS	3
TECNOLOGIAS USADAS.....	3
PROBLEMA A SOLUCIONAR	3
Requerimientos.....	3
DISEÑO UML	4
Paquete modelo: clase Vehículo y Manejo.....	4
Paquete vistas: Clase Automotor, clase ReporteVehiculos y clase Principal	5
DESCRIPCION DE CLASES Y MÉTODOS.....	7
Clase Vehículo.....	7
Clase Manejo.....	8
Clase Principal.....	10
Clase Automotor (Jframe).....	10
Clase ReporteVehiculos (JFrame).....	11
EJEMPLOS DE USO DEL PROGRAMA	14
Ventana Automotor.....	14
Ventana ReporteVehiculos.	16
CONCLUSIÓN	22

LOGÍSTICA AUTOMOTOR

INTRODUCCIÓN

Este proyecto tiene como objetivo gestionar una lista de vehículos para un parque automotor, permitiendo su almacenamiento, ordenamiento y búsqueda. La aplicación facilita el manejo de información relevante de los vehículos, como su placa, marca, tipo de vehículo y valor comercial.

OBJETIVOS

- Implementar un sistema que almacene y gestione la información de vehículos.
- Facilita la organización de vehículos mediante diferentes métodos de ordenamiento.
- Ordenar la lista de vehículos eficientemente por diferentes criterios.
- Permitir la búsqueda eficiente de vehículos por diferentes criterios.

TECNOLOGÍAS USADAS

- Lenguaje de programación: Java
- Entorno de desarrollo: NetBeans

PROBLEMA A SOLUCIONAR

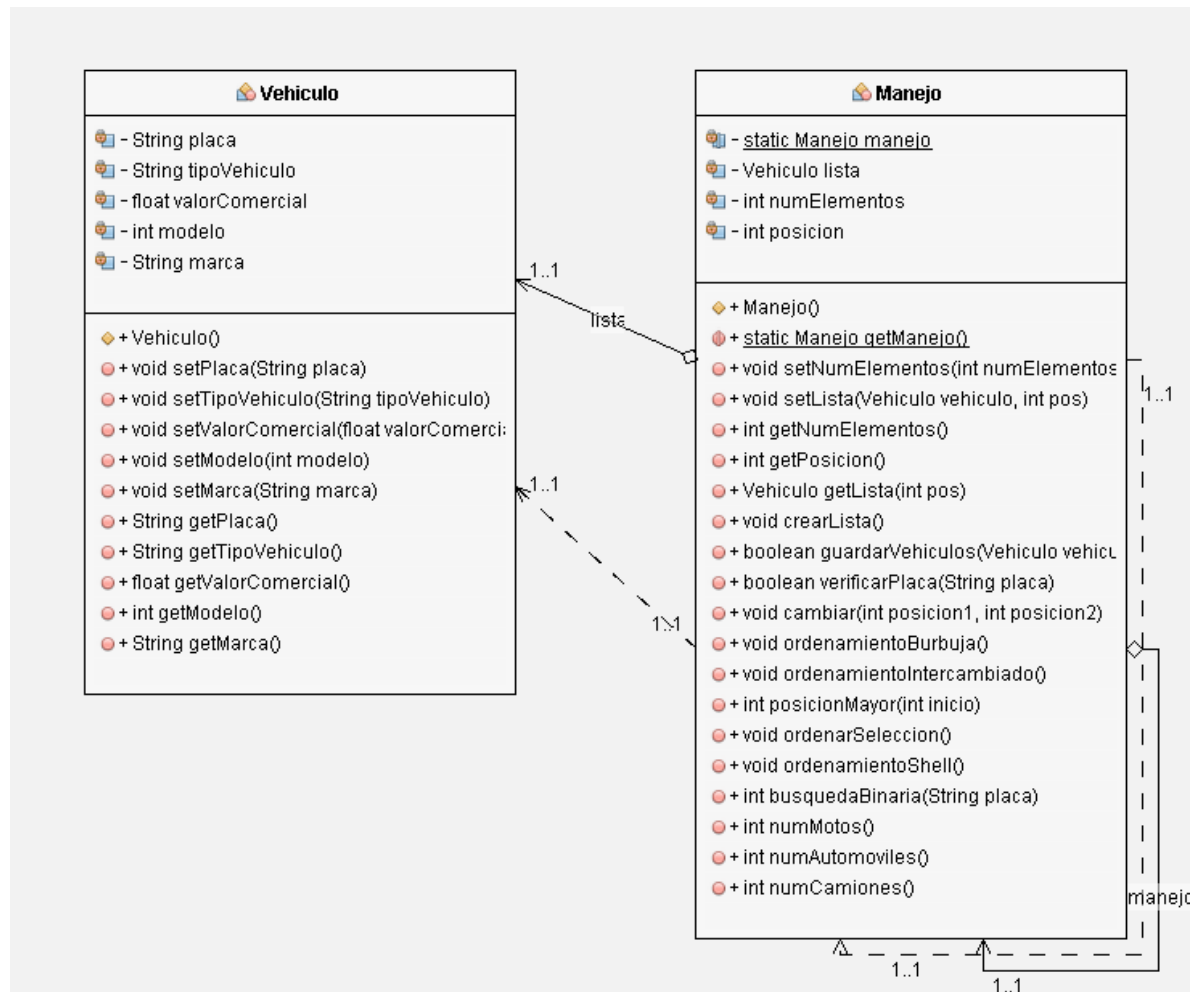
Una empresa dedicada al servicio de logística requiere llevar el inventario del parque automotor con el que prestan servicio. Para el diseño de la aplicación, se requiere registrar la placa de cada vehículo. Dentro de la información relevante de los vehículos, esta el modelo, la marca y el valor comercial del mismo. La empresa cuenta con diferentes tipos de vehículos para realizar sus operaciones; como son: camiones, automóviles y motocicletas.

Requerimientos.

- Registrar la información de cada vehículo en un vector de objetos.
- Buscar la información de un vehículo pasando como parámetro de búsqueda la placa.
- Ordenar el arreglo por la marca, placa, modelo y valor comercial.

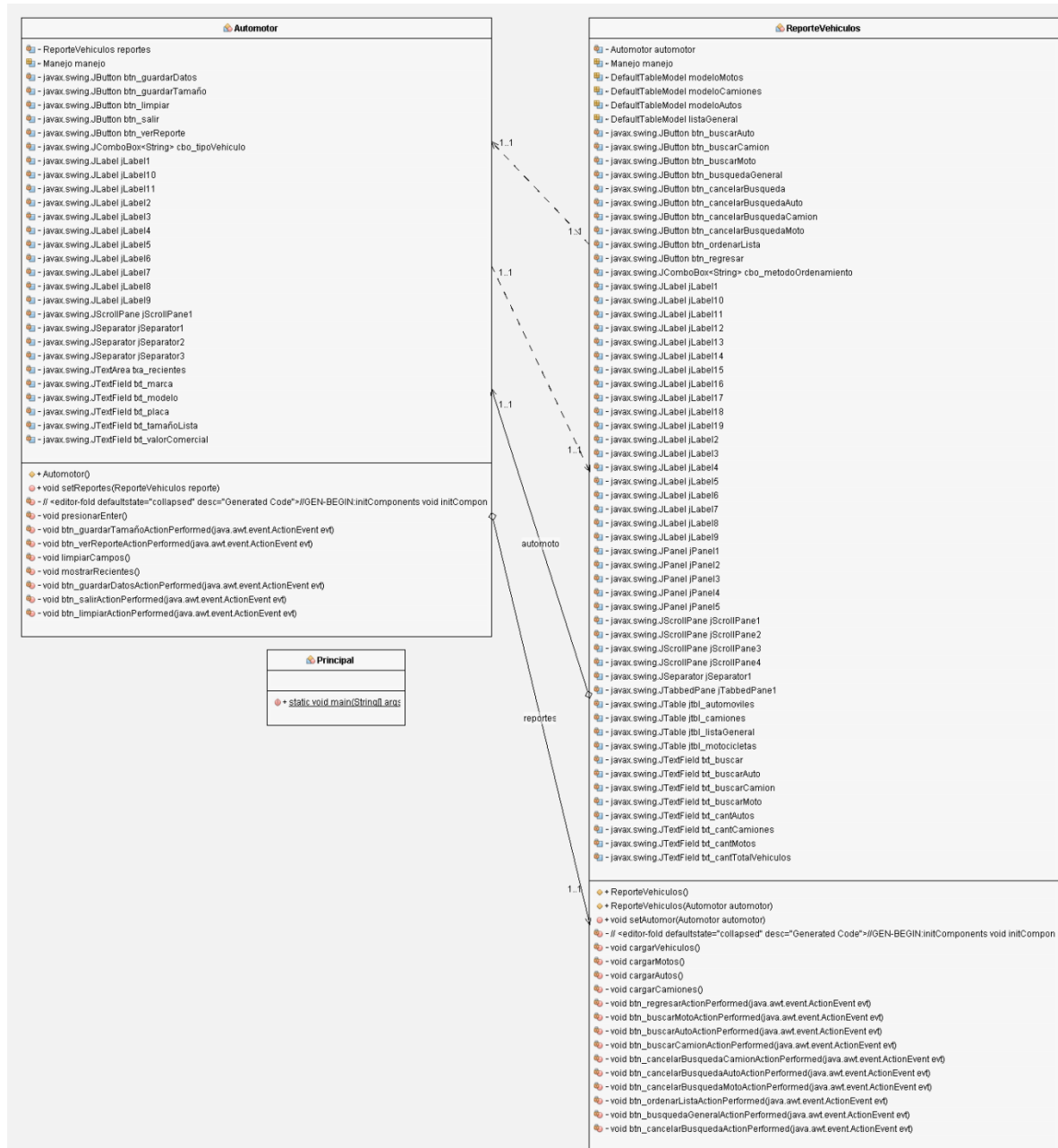
DISEÑO UML

Paquete modelo: clase Vehículo y Manejo



En el paquete modelo se lleva a cabo toda la lógica para el funcionamiento de la aplicación, en este se crea la clase Vehículo, que contiene la información de los vehículos, además se crea la clase Manejo, que es donde se encuentra la lógica para crear el arreglo y el guardado de datos, así como los métodos para ordenar y buscar, además se implementa un patrón singleton con el fin de garantizar una única instancia de la clase Manejo.

Paquete vistas: Clase Automotor, clase ReporteVehiculos y clase Principal .



En el paquete vistas, se lleva a cabo la lógica para manejar las interfaces graficas, con el fin de llevar a cabo el guardado de datos y el manejo de los reportes de los vehículos.

En el proyecto se diseñan las clases en java para implementar el manejo de la logística de un parque automotor, con el fin de organizar la información de la empresa.

Para este proyecto se realizaron dos **paquetes: modelo y vistas**, esto con el fin de organizar el proyecto.

Dentro del paquete **modelo**, se encuentran las clases para el manejo lógico del proyecto, tales como la clase **Vehículo** y la clase **Manejo**.

Dentro del paquete **vistas**, se encuentran las clases para el manejo grafico del proyecto, tales como la clase **Principal**, donde se encuentra el método main; JFrame **Reporte** que es donde se lleva a cabo el guardado de los datos de los vehículos, y el JFrame **ReporteVehiculos**, que es donde se muestra el listado de los vehículos, discriminado por el tipo de vehículo, además de un listado general, en esta interfaz también se ordenan la listas y se realizan las búsquedas de vehiculos

Inicialmente se implementa una clase **Vehículo**, con los siguientes atributos: **placa, tipoVehiculo, valorComercial, modelo y marca**.

Después de implementar la clase **Vehículo**, se implementa la clase **Manejo**, en donde se declara y se crea un vector, para almacenar objetos de la clase **Vehículo**. En esta clase se implementan los distintos métodos para llevar la logística del parque automotor, como lo son los métodos de guardado, ordenamiento, búsqueda, etc....

También se implementan algunos métodos, como **el patrón singleton**, que se usa para crear una única instancia de la clase **Manejo**.

En la clase **Principal** se maneja la lógica del método **main**, que lo utilizamos para crear las instancias de las vista utilizadas.

Finalmente se crean dos JFrame, en el JFrame **Automotor**, se lleva a cabo el guardado de datos de los vehículos, los cuales se guardan en el vector declarado anteriormente, y se usan los métodos de **llenarLista, ordenamientoShell**, etc.....

En el JFrame **ReporteVehiculos**, se generan los listados de los vehículos registrados anteriormente en el JFrame **Automotor**, aquí se implementan los métodos de **ordenamientoShell, ordenamientoBurbuja, ordenamientoIntercambiado, ordenamientoSeleccion, busquedaBinaria**, etc....., que se crearon previamente en la clase **Manejo**. Esta vista se usa para general los reportes de vehículos registrados y vehículos disponibles, además de la búsqueda de un determinado vehículo.

DESCRIPCION DE CLASES Y MÉTODOS

Clase Vehículo.

Representa un vehículo con los siguientes **atributos**.

- **placa:** Identificador único del vehículo.
- **tipoVehiculo:** El vehículo puede ser de tipo “Motocicleta”, “Automóvil” o “Camion”.
- **valorComercial:** Precio del vehículo en el mercado.
- **modelo:** Año de fabricación del vehículo.
- **marca:** Marca del vehículo.

Constructor:

- **Vehiculo():** Constructor por defecto de la clase **Vehículo**.

Métodos Principales:

- **Getters y setters:** Permiten acceder y modificar los atributos del vehículo.
 - **setPlaca(String placa):** Método para guardar la placa del vehículo.
 - **setTipoVehiculo(String tipoVehiculo):** Método para guardar el tipo de vehículo.
 - **setValorComercial(int valorComercial):** Método para guardar el valor comercial del vehículo.
 - **setModelo(int modelo):** Método para guardar el modelo del vehículo (año de fabricación).
 - **setMarca(String marca):** Metodo para guardar la marca del vehículo.
 - **getPlaca():** Metodo para acceder a la información del atributo **placa**; con este metodo conocemos cual es la placa del vehículo.
 - **getTipoVehiculo():** Metodo para acceder a la información del atributo **tipoVehiculo**; con este método sabemos a que tipo de vehículo pertenece el vehículo guardado.
 - **getValorComercial():** Metodo para acceder a la información del atributo **valorComercial**; con este método conocemos cual es el valor comercial del vehículo guardado.
 - **getModelo():** Metodo para acceder a la información del atributo **modelo**; con este metodo obtenemos el modelo del vehículo.
 - **getMarca():** Metodo para acceder a la información del atributo **marca**; con este metodo podemos obtener la marca del vehículo guardado.

Clase Manejo

Clase encargada de administrar la lista de vehículos mediante un arreglo de objetos **Vehículo**. Implementa el **patrón Singleton** para garantizar una única instancia de esta clase.

Atributos:

- **manejo**: Instancia única de la clase **Manejo**.
- **Lista**: Arreglo de objetos Vehículo.
- **numElementos**: Cantidad total de elementos en la lista.
- **Posición**: Índice actual de la lista, manejo de la posición donde se guardará un vehículo.

Constructor:

- **Manejo()**: Inicializa la lista de **Vehiculos**, **numElementos** y **posición**.

Métodos Principales:

- **getManejo()**: Devuelve la instancia única de **Manejo (Patrón Singleton)**.
- **setNumElementos(int numElementos)**: Establece el numero de elementos que puede tener la lista.
- **setLista(Vehículo vehículo, int pos)**: Reemplaza un vehículo en una posición específica (usado en el metodo **cambiar**).
- **getNumElementos()**: Retorna el tamaño total de la lista. Esta es la capacidad que tiene la lista.
- **getPosicion()**: Devuelve la posición actual en la lista.
- **getList(int posición)**: Devuelve la información de un vehículo en una determinada posición.
- **crearLista()**: Inicializa el arreglo lista dándole como tamaño **numElementos**.
- **guardarVehiculos(Vehículo vehículo)**: Agrega un vehículo a la lista si hay espacio disponible, es decir la posición esta disponible y es menor a **numElementos**.
- **verificarPlaca(String placa)**: Verifica si hay un vehículo con la misma placa ya existente dentro de la lista.
- **cambiar(int posicion1, int posicion2)**: Intercambia dos vehículos en la lista según las posiciones (**usa setLista**); este metodo se usa en los métodos de **ordenamiento**.

Métodos de Ordenamiento:

Los métodos de ordenamiento tienen un algoritmo específico, pero en todos se hace uso del método **cambiar**, previamente creado.

- **ordenamientoBurbuja()**: Este método ordena la lista según el **modelo** del vehículo usando el **algoritmo burbuja**, ordenándolos de mayor a menor, lo que quiere decir del más reciente al más antiguo (usa el método **cambiar**).
- **ordenamientoIntercambiado()**: Este método ordena los vehículos por su **marca** usando el **algoritmo de intercambio** (usa el método **cambiar**).
- **ordenarSeleccion()**: Este método ordena los vehículos por su valor comercial, del más costoso al más económico usando el **algoritmo de selección**, para esto se usa el método **posicionMayor** y el método **cambiar**.
- **ordenamientoShell()**: Este método ordena la lista de vehículos según su placa, esto usando el **algoritmo Shell** (usa el método **cambiar**).

Método de búsqueda:

- **busquedaBinaria()**: Busca un vehículo por su **placa** mediante **búsqueda binaria**, lo que significa que irá dividiendo el arreglo hasta encontrar la placa buscada (requiere el método **ordenamientoShell**, esto para garantizar que la lista esté organizada según las placas de los vehículos).

Otros métodos:

- **posicionMayor(int inicio)**: Encuentra la posición del vehículo con mayor valor comercial (Se usa en el método **ordenarSeleccion**).
- **numMotos()**: Retorna el número total de vehículos de tipo "Motocicleta" que se encuentran registrados en la lista.
- **numAutomoviles()**: Retorna el número total de vehículos de tipo "Automovil" que se encuentran registrados en la lista.
- **numCamiones()**: Retorna el número total de vehículos de tipo "Camion" que se encuentran registrados en la lista.

Clase Principal

Esta clase contiene el metodo **main** y es el punto de entrada de la aplicación.

Metodo Principal:

- **main(String[]args):**
 - Crea instancias de las vistas **Automotor** y **ReporteVehiculos**.
 - Establece la relación entre ambas clases.
 - Configura y muestra la interfaz grafica.

Clase Automotor (Jframe)

Esta clase representa la interfaz grafica de la aplicación, esta extiende la `javax.swing.JFrame` , permitiendo la interacción con el usuario.

Atributos:

- **reportes:** Instancia de **ReporteVehiculos** para la generación de los reportes.
- **manejo:** Instancia de la clase **Manejo** obtenida mediante **getManejo()** (**Patrón Singleton**).
- Componentes gráficos como **JtextField**, **Jbutton**, **JtextArea**, **JcomboBox**, **Jlabel**, **JscrollPane** y **Jseparator**.

Constructor:

- **Automotor():** Inicializa la interfaz grafica y configura algunos componentes (usa el metodo **initComponents()**).

Métodos Principales:

- **initComponents():** Este metodo es el encargado de iniciar todos los componentes gráficos, darles tamaño y eventos.
- **setReportes(ReporteVehiculos reporte):** Asocia la clase **Automotor** con la clase **ReporteVehiculos**.
- **PresionarEnter():** Permite la navegación entre campos con la tecla Enter.
- **btn_guardarTamañoActionPerformed(ActionEvent evt):** Crea la lista de vehículos y la inicializa con el tamaño establecido (usa el metodo **setNumElementos** y **crearLista** de la clase **Manejo**).
- **btn_verReporteActionPerformed(ActionEvent evt):** Ordena la lista de los vehículos por la placa (usa el metodo **ordemanientoShell()** de la clase **Manejo**), y abre la venta de reportes (**ReporteVehiculos**).
- **btn_guardarDatosActionPerformed(ActionEvent evt):** Guarda los datos ingresados en los campos de texto **JTextFields** en la lista de vehículos (usa el metodo **guardarVehiculos** de la clase **Manejo**).
- **btn_salirActionPerformed(ActionEvent evt):** Cierra la aplicación.

- **btn_limpiarActionPerformed(ActionEvent evt):** Limpia el área de texto de los vehículos recientes.
- **mostrarRecientes():** Muestra los vehículos agregados recientemente a la lista.
- **limpiarCampos():** Limpia los campos de texto de entrada del formulario.

Clase ReporteVehiculos (JFrame)

Esta clase representa la interfaz grafica para los reportes de los vehículos, esta clase extiende de javax.swing.JFrame, permitiendo la interacción con el usuario. En esta clase se muestran los reportes de los vehículos en listados discriminados por el tipo de vehículo, así como un listado general, y la cantidad de vehículos disponibles, según su tipo, además se pueden ordenar las listas, y hacer búsquedas de vehículos.

Atributos:

- **automotor:** Objeto de la clase **Automotor**, utilizado para la referenciar la ventana principal.
- **Manejo:** instancia de la clase **Manejo**, utilizada para manejar la información de los vehículos; se obtiene a través del metodo **getManejo()** (**Patrón Singleton**).
- **modeloMotos:** Establece el modelo de tabla para motocicletas.
- **modeloCamiones:** Establece el modelo de tabla para camiones.
- **modeloAutos:** Establece el modelo de tabla para automóviles.
- **listaGeneral:** Establece el modelo de tabla para la lista general de vehículos.
- Componentes gráficos como **JTextField**, **JButton**, **JTextArea**, **JComboBox**, **JLabel**, **JScrollPane**, **Jseparator**, **JTable** y **JTabbedPane**.

Constructores:

- **ReporteVehiculos():** Inicializa los componentes de la interfaz grafica y configura las tablas con sus respectivas cabeceras.
- **ReporteVehiculos(Automotor automotor):** Variante del constructor que recibe un objeto Automotor para gestionar la ventana principal.

Métodos Principales:

- **setAutomotor(Automotor automotor):** Asocia la clase **ReporteVehiculos** con la clase **Automotor**.
- **initComponents():** Este metodo es el encargado de iniciar todos los componentes gráficos, darles tamaño y eventos.

Carga de datos:

Estos métodos de cargar datos se utilizan para cargar la información disponible una vez se abra la ventana sin la necesidad de presionar algún botón, una vez que en la ventana **Automotor** se presione el botón “**VER REPORTE COMPLETO**” y se abra esta ventana, los datos ya estarán cargados.

- **cargarVehiculos():** Llena la tabla general con todos los vehículos que contiene la lista de vehículos (**usa el metodo getPosicion y getLista de la clase Manejo**).
- **cargarMotos():** Llena la tabla de motocicletas con todos los vehículos de tipo “Motocicleta” (**usa el metodo getPosicion y getLista de la clase Manejo**).
- **cargarAutos():** Llena la tabla de motocicletas con todos los vehículos de tipo “Automovil” (**usa el metodo getPosicion y getLista de la clase Manejo**).
- **cargarCamiones():** Llena la tabla de camiones con todos los vehículos de tipo “Camion” (**usa el metodo getPosicion y getLista de la clase Manejo**).

Búsqueda de vehículos:

- **btn_buscarMotoActionPerformed(ActionEvent evt):** Busca una motocicleta por su placa y muestra dicha información en la tabla de motos. (usa el metodo **busquedaBinaria** y **getLista** de la clase **Manejo**, además usa una instancia de la clase **Vehículo**).
- **btn_buscarAutoActionPerformed(ActionEvent evt):** Busca un auto por su placa y muestra dicha información en la tabla de autos. (Usa el metodo de **busquedaBinaria** y **getLista** de la clase **Manejo**, además usa una instancia de la clase **Vehículo**).
- **btn_buscarCamionActionPerformed(ActionEvent evt):** Busca un camion por su placa y muestra dicha información en la tabla de camiones. (Usa el metodo de **busquedaBinaria** y **getLista** de la clase **Manejo**, además usa una instancia de la clase **Vehículo**).
- **btn_busquedaGeneralActionPerformed(ActionEvent evt):** Busca un Vehículo por su placa, sin importar el tipo de vehículo, y una vez encontrado muestra esa información en la tabla general. (Usa el metodo **busquedaBinaria**, y **getLista** de la clase **Manejo**, además usa una instancia de la clase **Vehículo**).

Cancelación de búsquedas:

- **btn_cancelarBusquedaMotoActionPerformed(ActionEvent evt):** Restaura a la lista original de motocicletas en la tabla, deja de mostrar la información de la moto buscada y muestra el listado general de motos (usa el metodo **cargarMotos**).
- **Btn_cancelarBusquedaAutoActionPerformed(ActionEvent evt):** Restaura a la lista original de automóviles en la tabla, deja de mostrar la información del auto buscado y muestra el listado general de autos (usa el metodo **cargasAutos**).
- **btn_cancelarBusquedaCamionActionPerformed(ActionEvent evt):** Restaura a la lista original de camiones en la tabla, deja de mostrar la información del camion buscado y muestra el listado general de camiones (usa el metodo **cargarCamiones**).
- **btn_cancelarBusquedaActionPerformed(ActionEvent evt):** Restaura la lista general de los vehículos en la lista general, deja de mostrar la información del vehículo buscado y muestra el listado general de todos los vehículos (usa el metodo **cargarVehiculos**).

Ordenamiento de las listas:

- **btn_ordenarListaActionPerformed(ActionEvent evt):** Ordena la lista de vehículos según el método de ordenamiento seleccionado en el combo box:
 - **Ordenar por placas: OrdenamientoShell.**
 - **Ordenar por el modelo: OrdenamientoBurbuja.**
 - **Ordenar por el valor comercial: OrdenarSeleccion.**
 - **Ordenar alfabéticamente (marca): OrdenamientoIntercambiado.**

Para ordenar la lista se utilizan los métodos de ordenamiento de la clase **Manejo**; además se usan los métodos de cargar datos para cargar nuevamente los datos ya ordenados en las tablas.

Navegación:

- **btn_regresarActionPerformed(ActionEvent evt):** Regresa a la ventana principal donde se ingresan los datos (**Automotor**), y oculta la ventana actual (**ReporteVehiculos**).

EJEMPLOS DE USO DEL PROGRAMA

Ventana Automotor.

The screenshot shows the 'LOGISTICA AUTOMOTOR' window. At the top, it prompts the user to enter the list size for vehicles: 'Por favor ingrese el tamaño de la lista para los vehículos:'. Below this, there is a text input field for 'Tamaño' containing the value '100' and a 'GUARDAR TAMAÑO' button. The main area is divided into two sections. The left section, 'Datos de los vehículos:', contains a sub-prompt '* Por favor ingrese la información del vehículo:' followed by several input fields: 'Tipo de Vehículo' (a dropdown menu currently showing 'Camion'), 'Placa', 'Marca', 'Modelo', and 'Valor Comercial'. A 'GUARDAR VEHICULO' button is at the bottom of this section. The right section, 'Vehiculos agregados recientemente:', features a large empty rectangular box for displaying the list of added vehicles. To the right of this box are three buttons: 'VER REPORTE COMPLETO', 'LIMPIAR LISTA', and 'SALIR'.

Ventana para llevar a cabo el guardado de datos, en esta ventana se crea la lista y se guardan los datos de los vehículos, en el apartado de “vehículos agregados recientemente”, se mostrarán los vehículos que se han agregado anteriormente.

Guardar Tamaño y crear lista:

This screenshot shows the same 'LOGISTICA AUTOMOTOR' window as before, but with a small dialog box overlaid in the center. The dialog box has a title bar with a close button and contains an information icon, the text 'Lista Creada.', and an 'OK' button. In the background window, the 'GUARDAR TAMAÑO' button is circled in red, indicating the action just completed. The 'Tamaño' input field now shows '100'. The rest of the window's layout and controls remain the same.

Para guardar el tamaño se debe ingresar en el campo del tamaño y se presiona el botón “GUARDAR TAMAÑO”, al presionar el botón se guarda el tamaño y se crea la lista.

Guardado de Datos:

The screenshot shows the 'LOGISTICA AUTOMOTOR' application window. At the top, it says 'Por favor ingrese el tamaño de la lista para los vehículos:'. Below this is a 'Tamaño:' input field and a 'GUARDAR TAMAÑO' button. The main section is 'Datos de los vehículos:' with a sub-instruction '* Por favor ingrese la información del vehículo:'. It contains several input fields: 'Tipo de Vehículo:' (a dropdown menu set to 'Automovil'), 'Placa:' (text 'HGT479'), 'Marca:' (text 'Lamborghini'), 'Modelo:' (text '2025'), and 'Valor Comercial:' (text '20000000'). A red circle highlights the 'GUARDAR VEHICULO' button at the bottom of this section. A modal dialog box titled 'Datos Guardados.' with an 'OK' button is overlaid on the form. To the right of the form are three buttons: 'VER REPORTE COMPLETO', 'LIMPIAR LISTA', and 'SALIR'.

Para guardar los datos de un vehículo, se deben ingresar los datos en los respectivos campos de texto, una vez que todos estén llenos, se debe presionar el botón “GUARDAR VEHICULO”, una vez presionado se mostrara un mensaje confirmando que se ha guardado el dato, en caso de que no se puedan agregar datos se mostrara un mensaje indicándolo.

This screenshot shows the same 'LOGISTICA AUTOMOTOR' application window. The 'Tipo de Vehículo:' dropdown is now set to 'Motocicleta'. The 'Placa:', 'Marca:', and 'Modelo:' fields are empty. The 'Valor Comercial:' field still contains '20000000'. The 'GUARDAR VEHICULO' button remains highlighted with a red circle. A new section titled 'Vehiculos agregados recientemente:' has appeared on the right side of the form, containing a list of 16 vehicles with their type and license plate. The buttons 'VER REPORTE COMPLETO', 'LIMPIAR LISTA', and 'SALIR' are still present on the far right.

Vehiculos agregados recientemente:	
1 Tipo :	Automovil placa: YTS294
2 Tipo :	Automovil placa: NVS295
3 Tipo :	Camion placa: BRW528
4 Tipo :	Camion placa: AKE268
5 Tipo :	Camion placa: BST735
6 Tipo :	Automovil placa: HTS837
7 Tipo :	Motocicleta placa: MDT948
8 Tipo :	Motocicleta placa: GRU935
9 Tipo :	Motocicleta placa: HTU893
10 Tipo :	Motocicleta placa: FRO893
11 Tipo :	Motocicleta placa: OST839
12 Tipo :	Automovil placa: NGW295
13 Tipo :	Automovil placa: JSO893
14 Tipo :	Camion placa: VKF473
15 Tipo :	Camion placa: FYS289
16 Tipo :	Motocicleta placa: KSU792

Como se puede ver en la imagen anterior, se muestra un resumen de los vehículos agregados recientemente.

Ver reporte completo de los vehículos.

LOGISTICA AUTOMOTOR

Por favor ingrese el tamaño de la lista para los vehículos:

Tamaño:

Datos de los vehículos:

* Por favor ingrese la informacion del vehiculo:

Tipo de Vehiculo:

Placa:

Marca:

Modelo:

Valor Comercial:

Vehiculos agregados recientemente:

- 0 Tipo : Automovil placa: HGT479
- 1 Tipo : Automovil placa: YTS294
- 2 Tipo : Automovil placa: NVS295
- 3 Tipo : Camion placa: BRW528
- 4 Tipo : Camion placa: AKE268
- 5 Tipo : Camion placa: BST735
- 6 Tipo : Automovil placa: HTS837
- 7 Tipo : Motocicleta placa: MDT948
- 8 Tipo : Motocicleta placa: GRU935
- 9 Tipo : Motocicleta placa: HTU893
- 10 Tipo : Motocicleta placa: FRO893
- 11 Tipo : Motocicleta placa: OST839
- 12 Tipo : Automovil placa: NGW295
- 13 Tipo : Automovil placa: JSO893
- 14 Tipo : Camion placa: VKF473
- 15 Tipo : Camion placa: FYS289
- 16 Tipo : Motocicleta placa: KSU792

Para ver el reporte completo de los vehículos se debe presionar el botón “VER REPORTE COMPLETO”, al presionar el botón nos dirigirá a la ventana con toda la información de los vehículos.

Ventana ReporteVehiculos.

Esta ventana se compone de tres vistas más, donde en cada uno de los apartados se mostrara una información diferente, en el apartado “REPORTE MOTOCICLETAS”, se muestra toda la información de las motocicletas registradas; en el apartado “REPORTE AUTOMOVILES” se muestra toda la información de los automóviles registrados; en el apartado “REPORTE CAMIONES”, se muestra la información de los camiones registrados, y en el apartado de “REPORTE GENERAL” se muestran todos los vehículos, sin importar su tipo. Además de esto, en cada apartado se puede observar la cantidad disponible de cada tipo de vehículos, también se podrán buscar un vehículo por su placa, como se ve a continuación. Es importante recordar que en todos los apartados se puede ordenar la lista, según el modelo, placa, valor y marca.

APARTADO REPORTE GENERAL:

REPORTE GENERAL DE VEHICULOS

REPORTE MOTOCICLETAS

REPORTE AUTOMOVILES

REPORTE CAMIONES

REPORTE GENERAL

Reporte General

REPORTE GENERAL DE VEHICULOS

TIPO VEHICULO	PLACA	MARCA	MODELO	VALOR COMERCIAL
Motocicleta	GRU935	Suzuki	2000	8000000.0
Motocicleta	FRO893	Kawasaki	2010	9000000.0
Motocicleta	OST839	Honda	2011	5000000.0
Camion	FYS289	Renault	2014	1.4E8
Automovil	NGW295	Audi	2014	1.2E8
Camion	BS1735	Iveco	2015	1.3E10
Automovil	JSO893	Renault	2016	2.3E8
Camion	BRW528	Scania	2017	1.0E8
Automovil	YTS294	Ferrari	2019	2.0E8
Automovil	NVS295	Nissan	2020	3.0E7
Motocicleta	HTU893	Yamaha	2020	2.0E7

Cantidad disponible: 17

Buscar por placa:

BUSCAR

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por: Ordenar por modelo

ORDENAR LISTA

REGRESAR

APARTADO REPORTE MOTOCICLETAS:

REPORTE GENERAL DE VEHICULOS

REPORTE MOTOCICLETAS

REPORTE AUTOMOVILES

REPORTE CAMIONES

REPORTE GENERAL

Reporte de las Motocicletas

MOTOCICLETAS

PLACA	MARCA	MODELO	VALOR COMERCIAL
FRO893	Kawasaki	2010	9000000.0
GRU935	Suzuki	2000	8000000.0
HTU893	Yamaha	2020	2.0E7
KSU792	Yamaha	2025	2.0E7
MDT948	BMW	2020	1.2E7
OST839	Honda	2011	5000000.0

Cantidad Disponible: 6

Buscar por placa:

BUSCAR

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por: Ordenar por placa

ORDENAR LISTA

REGRESAR

APARTADO REPORTE AUTOMOVILES:

REPORTES

REPORTES DE VEHICULOS

REPORTE MOTOCICLETAS

REPORTE AUTOMOVILES

REPORTE CAMIONES

REPORTE GENERAL

Reporte Automoviles

AUTOMOVILES

PLACA	MARCA	MODELO	VALOR COMERCIAL
HGT479	Lamborghini	2025	2.0E7
HTS837	Mercedes Benz	2025	2.0E9
JSO893	Renault	2016	2.3E8
NGW295	Audi	2014	1.2E8
NVS295	Nissan	2020	3.0E7
YTS294	Ferrari	2019	2.0E8

Cantidad Disponible: 6

Buscar por placa:

BUSCAR

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por: Ordenar por placa

ORDENAR LISTA

REGRESAR

APARTADO REPORTE CAMIONES:

REPORTES

REPORTES DE VEHICULOS

REPORTE MOTOCICLETAS

REPORTE AUTOMOVILES

REPORTE CAMIONES

REPORTE GENERAL

Reporte Camiones

CAMIONES

PLACA	MARCA	MODELO	VALOR COMERCIAL
AKE268	Volvo	2023	4.0E9
BRW528	Scania	2017	1.0E8
BST735	Iveco	2015	1.3E10
FYS289	Renault	2014	1.4E8
VKF473	Scania	2021	2.3E8

Cantidad Disponible:

5

Buscar por placa:

BUSCAR

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por:

Ordenar por placa

ORDENAR LISTA

REGRESAR

En estos apartados se muestra la información de los vehículos, así como su cantidad, y se realizan las respectivas búsquedas.

REPORTE GENERAL DE VEHICULOS

REPORTE MOTOCICLETAS

REPORTE AUTOMOVILES

REPORTE CAMIONES

REPORTE GENERAL

Reporte General

REPORTE GENERAL DE VEHICULOS

TIPO VEHICULO	PLACA	MARCA	MODELO	VALOR COMERCIAL
Automovil	NGW295	Audi	2014	1.2E8
Motocicleta	MDT948	BMW	2020	1.2E7
Automovil	YTS294	Ferrari	2019	2.0E8
Motocicleta	OST839	Honda	2011	5000000.0
Camion	BST735	Iveco	2015	1.3E10
Motocicleta	FRO893	Kawasaki	2010	9000000.0
Automovil	HGT479	Lamborghini	2025	2.0E7
Automovil	HTS837	Mercedes Benz	2025	2.0E9
Automovil	NVS295	Nissan	2020	3.0E7
Automovil	JSO893	Renault	2016	2.3E8
Camion	FYS289	Renault	2014	1.4E8

Cantidad disponible: 17

Buscar por placa: HTS837

BUSCAR

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por: Ordenar alfabeticamente

ORDENAR LISTA

REGRESAR

Al ingresar a la ventana ReporteVehiculos, se mostrará la cantidad disponible de vehículos, como se ve en la imagen anterior.

Buscar un vehículo por su placa:

REPORTE GENERAL DE VEHICULOS

REPORTE MOTOCICLETAS

REPORTE AUTOMOVILES

REPORTE CAMIONES

REPORTE GENERAL

Reporte General

REPORTE GENERAL DE VEHICULOS

TIPO VEHICULO	PLACA	MARCA	MODELO	VALOR COMERCIAL
Automovil	HTS837	Mercedes Benz	2025	2.0E9

Cantidad disponible: 17

Buscar por placa: HTS837

BUSCAR

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por: Ordenar por placa

ORDENAR LISTA

REGRESAR

Para realizar una búsqueda, se debe ingresar la placa del vehículo, como se indica, y luego presionar el botón “BUSCAR”, una vez presionado el botón, la información de ese vehículo se mostrará en la tabla como se ve en la imagen anterior.

Ordenar la lista de vehículos:

Reporte General de Vehículos

Reporte Motocicletas Reporte Automoviles Reporte Camiones **Reporte General**

Reporte General

Reporte General de Vehículos

TIPO VEHICULO	PLACA	MARCA	MODELO	VALOR COMERCIAL
Camion	AKE268	Volvo	2023	4.0E9
Camion	BRW528	Scania	2017	1.0E8
Camion	BST735	Iveco	2015	1.3E10
Motocicleta	FRO893	Kawasaki	2010	9000000.0
Camion	FYS289	Renault	2014	1.4E8
Motocicleta	GRU935	Suzuki	2000	8000000.0
Automovil	HGT479	Lamborghini	2025	2.0E7
Automovil	HTS837	Mercedes Benz	2025	2.0E9
Motocicleta	HTU893	Yamaha	2020	2.0E7
Automovil	JSO893	Renault	2016	2.3E8
Motocicleta	KSU792	Yamaha	2025	2.0E7

Cantidad disponible: 17

Buscar por placa: HTS837 **BUSCAR**

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por: Ordenar por placa **ORDENAR LISTA**

Ordenar por placa
Ordenar alfabeticamente
Ordenar por modelo
Ordenar por valor comercial

REGRESAR

Para ordenar la lista, primero se debe seleccionar la forma en la que se quiere ordenar, ya que se puede ordenar por placa, marca(alfabéticamente), modelo y valor comercial; una vez seleccionado el modo de ordenamiento, se debe presionar el botón “ORDENAR LISTA”, esto ordenara la lista y actualizara la tabla mostrando la lista ordenada, esto en todos los apartados, sin importar en cual se haga.

Regresar a la ventana Principal (Automotor).

REPORTE GENERAL DE VEHICULOS

REPORTE MOTOCICLETAS REPORTE AUTOMOVILES REPORTE CAMIONES **REPORTE GENERAL**

Reporte General

REPORTE GENERAL DE VEHICULOS

TIPO VEHICULO	PLACA	MARCA	MODELO	VALOR COMERCIAL
Automovil	NGW295	Audi	2014	1.2E8
Motocicleta	MDT948	BMW	2020	1.2E7
Automovil	YTS294	Ferrari	2019	2.0E8
Motocicleta	OST839	Honda	2011	5000000.0
Camion	BST735	Iveco	2015	1.3E10
Motocicleta	FRO893	Kawasaki	2010	9000000.0
Automovil	HGT479	Lamborghini	2025	2.0E7
Automovil	HTS837	Mercedes Benz	2025	2.0E9
Automovil	NVS295	Nissan	2020	3.0E7
Automovil	JSO893	Renault	2016	2.3E8
Camion	FYS289	Renault	2014	1.4E8

Cantidad disponible: 17 Buscar por placa: **BUSCAR**

CANCELAR BUSQUEDA

Ordenar la lista de vehiculos

Ordenar por: **Ordenar alfabeticamente** **ORDENAR LISTA**

REGRESAR



LOGISTICA AUTOMOTOR

Por favor ingrese el tamaño de la lista para los vehiculos:

Tamaño: **GUARDAR TAMAÑO**

Datos de los vehiculos:

* Por favor ingrese la informacion del vehiculo:

Tipo de Vehiculo: **Motocicleta**

Placa:

Marca:

Modelo:

Valor Comercial:

GUARDAR VEHICULO

Vehiculos agregados recientemente:

- 0 Tipo : Automovil placa: HGT479
- 1 Tipo : Automovil placa: YTS294
- 2 Tipo : Automovil placa: NVS295
- 3 Tipo : Camion placa: BRW528
- 4 Tipo : Camion placa: AKE268
- 5 Tipo : Camion placa: BST735
- 6 Tipo : Automovil placa: HTS837
- 7 Tipo : Motocicleta placa: MDT948
- 8 Tipo : Motocicleta placa: GRU935
- 9 Tipo : Motocicleta placa: HTU893
- 10 Tipo : Motocicleta placa: FRO893
- 11 Tipo : Motocicleta placa: OST839
- 12 Tipo : Automovil placa: NGW295
- 13 Tipo : Automovil placa: JSO893
- 14 Tipo : Camion placa: VKF473
- 15 Tipo : Camion placa: FYS289
- 16 Tipo : Motocicleta placa: KSU792

VER REPORTE COMPLETO

LIMPIAR LISTA

SALIR

Al presionar el botón “REGRESAR” de la vista ReporteAutomoviculos, no redirigirá a la ventana principal (Automotor), mostrándonos la información que se había agregado anteriormente.

CONCLUSIÓN

Este proyecto ha permitido desarrollar una solución eficiente para la gestión del inventario del parque automotor de la empresa de logística. Mediante la implementación de métodos de búsqueda y ordenamiento, se optimizó el almacenamiento y recuperación de información sobre los vehículos, garantizando un acceso rápido y preciso a datos como la placa, el modelo, la marca y el valor comercial del vehículo.

El uso de técnicas de ordenamiento como el “ordenamiento Shell”, permitió organizar los registros de manera estructurada, facilitando la consulta y actualización de la información. Así mismo, los algoritmos de búsqueda empleados agilizaron la localización de vehículos dentro del sistema, mejorando la eficiencia operativa y reduciendo tiempos de procesamiento.

Como resultado, la solución desarrollada no solo cumple con los requerimientos de la empresa, sino que también sienta las bases para futuras mejoras y expansiones del sistema. La correcta implementación de estos métodos ha demostrado su impacto positivo en la gestión del parque automotor, ofreciendo una herramienta funcional, escalable y adaptable a las necesidades de la empresa.