

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESTRUCTURA DE DATOS

CATEDRÁTICO: ING. HERMAN IGOR VELIZ LINARES

TUTOR ACADÉMICO: ESTEBAN HUMBERTO VALDEZ ENNATI



Ottoniel Fabricio Vásquez Pineda

CARNÉ: 202307820

SECCIÓN: D

GUATEMALA, 29 DE FEBRERO DEL 2,024

# ÍNDICE

<b>ÍNDICE</b>	<b>1</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>OBJETIVOS</b>	<b>1</b>
1. GENERAL	1
2. ESPECÍFICOS	1
<b>ALCANCES DEL SISTEMA</b>	<b>1</b>
<b>ESPECIFICACIÓN TÉCNICA</b>	<b>1</b>
• REQUISITOS DE HARDWARE	1
• REQUISITOS DE SOFTWARE	1
<b>DESCRIPCIÓN DE LA SOLUCIÓN</b>	<b>2</b>
<b>LÓGICA DEL PROGRAMA</b>	<b>2</b>
❖ NOMBRE DE LA CLASE	
Captura de las librerías usadas	2
➤ Librerías	2
➤ Variables Globales de la clase _(El nombre de su clase actual)	3
➤ Función Main	3
➤ Métodos y Funciones utilizadas	3

## **INTRODUCCIÓN**

El presente manual técnico proporciona una detallada descripción de un sistema informático diseñado para gestionar y controlar diversas operaciones relacionadas con viajes, utilizando un enfoque orientado a objetos y técnicas de programación en Java. El sistema está compuesto por varias clases interconectadas que cumplen roles específicos dentro del flujo de trabajo, pero también me gustaría a dar a conocer como fue hecho esta aplicación con Java

## **OBJETIVOS**

### **1. GENERAL**

- 1.1. Demostrar lo que hay detrás de esta aplicación y todo el trabajo que se estuvo haciendo en esta aplicación.

### **ESPECÍFICOS**

- 1.2. Objetivo 1: Demostrar o enseñar lo que es esencial para el código para que el código trabaje correctamente.
- 1.3. Objetivo 2: Enseñar la complejidad que lleva este código y también para que se puedan asociar un poco con la programación de Java.

## **ALCANCES DEL SISTEMA**

Mi meta para este manual técnico es enseñarles y también darles conocimiento de como es el código en esta aplicación y no solo eso sino que también para alentarlos para que se puedan meter a este mundo de programación que es bonito y como alguien me dijo: “Si quieres usar tu imaginación al máximo, únete al mundo de la programación donde lo que imaginas se podrá hacer realidad”.

## **ESPECIFICACIÓN TÉCNICA**

- **REQUISITOS DE HARDWARE**

- No hay requerimientos específicos de hardware más allá de los necesarios para ejecutar una aplicación Java estándar.

- **REQUISITOS DE SOFTWARE**

- JDK (Java Development Kit) instalado para compilar y ejecutar la aplicación.
- Un entorno de desarrollo integrado (IDE) NetBeans, o simplemente el compilador de línea de comandos de Java (javac) y el intérprete (java).
- Sistema operativo compatible con Java (Windows, Linux, macOS).

## **DESCRIPCIÓN DE LA SOLUCIÓN**

- Lo primero que se tubo que ver, es saber que librerías se iban a utilizar, que funcionalidades iban a llevar al principio del programa, en el transcurso y al final del programa, esto se tenía que ver un poco con tan solo verlo y lo demás cuando se comenzó a programar para que nos saliera el programa de manera de que funcione y también para que nos dé una interfaz parecida a la que se ve y también las funcionalidades.

## LÓGICA DEL PROGRAMA

### ❖ Practica2

```
import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.PrintWriter;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
```

#### ➤ Librerías

La librería java io: que más que todo la utilice para almacenar, buscar, leer, enseñar los archivos que anda creando o leyendo.

La librería java util: que más que todo me sirvió para hacer los arraylist y también para las fechas se podría decir.

La librería java text SimpleDateFormat: esta librería me sirvió para darle un formato en específico a mis fechas más que todo.

#### ➤ Variables Globales de la clase \_(El nombre de su clase actual)

```
public class Practica2 {
    public static ArrayList<CSV> ListaCSV = new ArrayList<>();
    public static ArrayList<Viaje> listaViajes = new ArrayList<>();
    public static ArrayList<Vehiculos> Listahistoria = new ArrayList<>();
    private static int idCounter = 1;
```

Estas variables más que todo me ayudaron a hacer algunas listas y también para trabajar con un código para mi tabla Historial.

### ➤ Función Main

Lo que hice en mi main fue más que todo escribir 3 cosas, 2 funciones y la tercera que es para me lleve a una ventana, la primera función trata de cargar un id de un archivo que vaya creando a futuro pero este me ayuda a que aunque cierre el programa y vuelva a abrir este id no vuelva a empezar de cero sino que siga con el número con el que termino, lo segundo que es escribirCSV, más que todo este me sirvió para poder trabajar bien con mi aplicación, en la cual hace y crea un archivo CSV y en ese archivo CSV escribe algunos datos.

```
public static void main(String[] args) {  
    cargarIdCounter();  
    EscribirCSV();  
    Ventana ventana = new Ventana();  
}
```

### ➤ Métodos y Funciones utilizadas

A continuación, se dará una explicación general de lo que hace cada función:

- ```
28 public static void EscribirCSV() {
```









Me ayuda a crear un archivo CSV y en ese archivo CSV escribe algunos datos que yo quiero pero desde JAVA, o sea programando

- ```
67 public static void CSV
```

Me ayuda a conseguir los datos adentro del archivo CSV

- ```
72 public static Object[][] convertirDatosCSV_tabla() {
```

Me ayuda a convertir los datos CSV a ponerlos en un arreglo y así ponerlos en una tabla.

- 87  `public static void Vehiculos{`  
Me ayuda a conseguir los datos de los viajes de los vehículos
- 92  `public static Object[][] convertirDatosHistorias_tabla() {`  
Me ayuda a convertir los datos de los viajes de los vehículos a un arreglo y ponerlos en una tabla
- 110  `public static void Viaje(`  
Este me ayuda a obtener datos cuando se vaya generando un viaje
- 115  `public static void serializarDatosMotocicleta(Object vehiculo) {`  
Esta función me ayuda a serializar los datos de todos los vehículos, ignore que le puse Motocicleta pero me serializa todos los datos de los vehículos.
- 221  `private static void cargarIdCounter() {`  
Esta función me ayuda a cargar el dato u obtener el dato guardado que deje en un archivo.bin para usarlos en mi historial.
- 232  `private static void guardarIdCounter() {`  
Este me ayuda a guardar lo IdCounter o sea un contador para mi historial en la cual guarda el ultimo IdCounter que se uso cuando se andaba usando la aplicación.
- 242  `private static Date generarFechaHoraFin(Date fechaHoraInicio) {`  
Este me ayuda a generar las horas o diferencias de horas para mi historial en la cual sean horas randoms pero siempre que la hora de fin sea después de la hora de inicio
- 248  `public static void imprimirDatosSerializados() {`

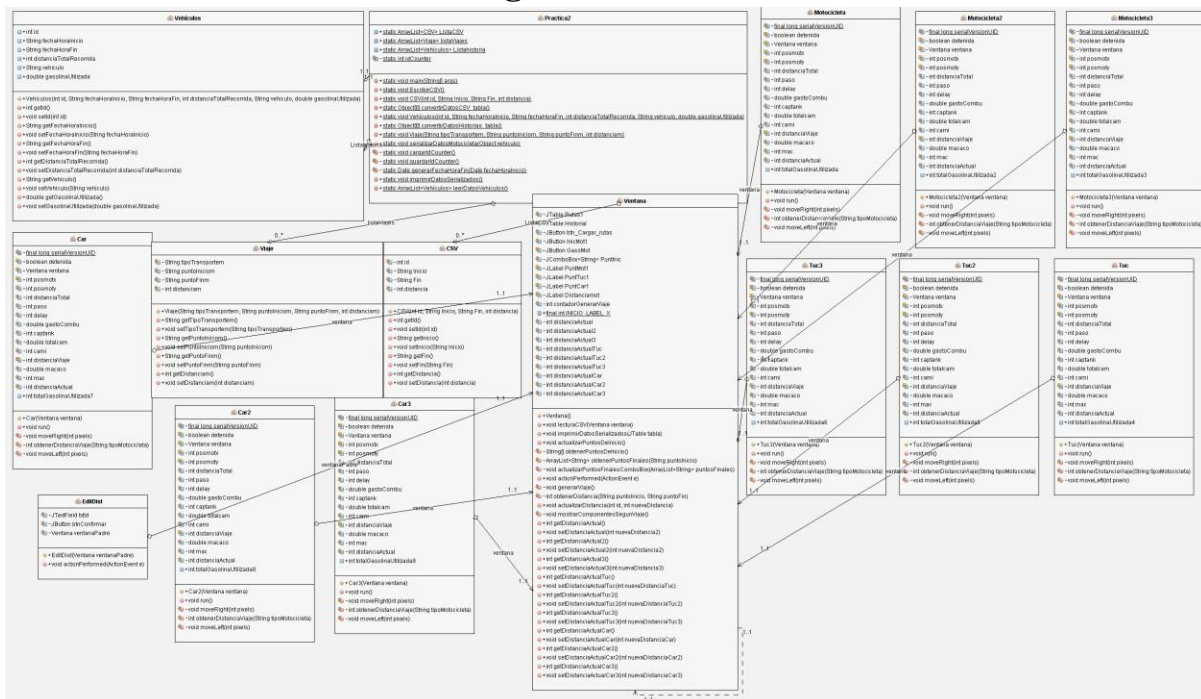


Lo que hago aquí más que todo imprimo los datos serializados para mi historial, para confirmar que si se estén imprimiendo y que también estos se guarden en un arraylist.

275 `public static ArrayList<Vehiculos> leerDatosVehiculos() {`

Esta función me ayuda a guardar los datos de mi archivo bin a un arraylist y también a una clase.

## Diagrama de Clases



## Diagrama de flujo

