UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1
CATEDRÁTICO: ING. WILLIAM ESTUARDO ESCOBAR ARGUETA
TUTOR ACADÉMICO: JOSUÉ RODOLFO MORALES CASTILLO



BRAYAN ALEXANDER GUZMAN MARGOS

CARNÉ: 202105658

SECCIÓN: B

GUATEMALA, 1 DE MARZO DEL 2,024

ÍNDICE

ÍNDICE	1
INTRODUCCIÓN	1
OBJETIVOS	1
1. GENERAL	1
2. ESPECÍFICOS	1
ALCANCES DEL SISTEMA	1
ESPECIFICACIÓN TÉCNICA	1
• REQUISITOS DE HARDWARE	1
• REQUISITOS DE SOFTWARE	1
DESCRIPCIÓN DE LA SOLUCIÓN	2
LÓGICA DEL PROGRAMA	2
❖ 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

Bienvenido a nuestro sistema de viajes. Este sistema le permitira crear sus viajes desde su punto de partida hasta su destino final y asi llegar a su destino de una manera segura eligiendo al transporte de preferencia. Aca tendras la guia para que puedas tener la mejor experiencia de nuestra aplicacion.

OBJETIVOS

1. GENERAL

El objetivo del siguiente manual es darle la mejor experiencia a nuestros viajeros al proporcionarles herramientas y sus funcionales útiles, y asi poder llegar a su destino de una manera segura.

2. ESPECÍFICOS

- 2.1. Objetivo 1: El siguiente manual nos llevara a un pequeño rrecorido en lo que es la funcionalidad de nuestro sistema de citas y sus funciones basicas.
- 2.2. Objetivo 2: De igual forma daremos a conocer sus funciones y el por que de cada una de las funciones utilizadas al momento de ejecutar nuestro sistema de viajes y lograr la mejor experiencia.

ALCANCES DEL SISTEMA

Nuestro alacance es mejorar la eficiencia y la calidad en nuestro sistema asi como garantizar un viaje seguro de inicio a fin

y hacerlo en un tiempo mas corto y facil para que cualquiera pueda acceder con facilidad.

Funcionalidades Adicionales: Además de las operaciones básicas de lectura y escritura, el sistema podría incluir otras funcionalidades avanzadas, como la validación de datos, la generación de informes o el análisis estadístico de los datos contenidos en los archivos CSV. Manual Técnico de Hilos en NetBeans: El sistema estará acompañado de un manual técnico que proporciona una guía

detallada sobre cómo trabajar con hilos en Java utilizando NetBeans. Este manual cubrirá los conceptos básicos de hilos, su creación, gestión, sincronización y comunicación, así como su aplicación práctica en el contexto de la manipulación de archivos CSV. En resumen, el alcance del sistema se centra en proporcionar una plataforma para la manipulación eficiente de archivos CSV utilizando hilos en Java, junto con un manual técnico que brinda orientación sobre cómo implementar esta funcionalidad utilizando el entorno de desarrollo NetBeans.

ESPECIFICACIÓN TÉCNICA

• REQUISITOS DE HARDWARE

RAM: 128 MB

• Espacio en disco: 124 MB para JRE; 2 MB para Java Update

• Exploradores compatibles: Internet Explorer 9 y superior, Firefox

• REQUISITOS DE SOFTWARE

- Windows 10 (8u51 y superiores)
- Windows 8.x (escritorio)
- Windows 7 SP1
- Windows Vista SP2
- Windows Server 2008 R2 SP1 (64 bits)
- Windows Server 2012 y 2012 R2 (64 bits)

DESCRIPCIÓN DE LA SOLUCIÓN

Para el siguiente proyecto nos ayudamos de un diagrama para ver por donde podriamos iniciar de igual forma nos ayudamos de herramientas como videos y tutoriales para darle una solucion al proyecto

LÓGICA DEL PROGRAMA

> Librerías

- java.awt.Image:
 - La clase java.awt.Image es una clase abstracta que actúa como la superclase de todas las clases que representan imágenes gráficas en lava
 - Las imágenes deben obtenerse de manera específica según la plataforma.
 - Esta clase se utiliza para manipular y procesar imágenes en aplicaciones gráficas.
 - ➤ Proporciona una base común para trabajar con imágenes, como cargar, mostrar y transformar.
 - ➤ Las clases concretas que heredan
 de Image incluyen BufferedImage y VolatileImage.
 - ➤ Desde: JDK1.0.
- ♦ java.io.BufferedReader:
 - La clase BufferedReader se utiliza para leer texto de un flujo de entrada de caracteres (como un archivo o una cadena).
 - > Proporciona métodos para leer líneas completas o caracteres individuales.
 - ➤ Es útil para leer archivos de texto y procesar su contenido línea por línea.
 - > Se utiliza junto con FileReader para leer archivos de texto.

- > Puede manejar la lectura de archivos de manera eficiente al almacenar en búfer los datos leídos.
- ➤ Es parte del paquete java.io y se utiliza para operaciones de entrada/salida.
- ➤ Desde: JDK1.1
- java.io.File:
 - ➤ La clase File representa una ruta de archivo o directorio en el sistema de archivos.
 - > Se utiliza para crear, leer, escribir o manipular archivos y directorios.
 - > Proporciona métodos para verificar la existencia de archivos, obtener información sobre ellos y realizar operaciones de archivo.
 - No representa el contenido real del archivo, sino solo su ubicación y metadatos.
 - ➤ Es parte del paquete java.io.
 - ➤ Desde: JDK1.0
- * javax.swing.Icon:
 - ➤ La interfaz Icon define un objeto que representa una imagen o icono pequeño.
 - ➤ Se utiliza principalmente en componentes de interfaz gráfica de usuario (GUI) para mostrar iconos en botones, etiquetas, etc.
 - ➤ Las clases que implementan Icon proporcionan métodos para obtener el ancho, alto y dibujar el icono.
 - ➤ Es parte del paquete javax.swing.
 - ➤ Desde: JDK1.2
 - > javax.swing.ImageIcon:
 - La clase ImageIcon implementa la interfaz Icon y representa un icono basado en una imagen.
 - > Se utiliza para cargar imágenes desde archivos o recursos y convertirlas en iconos.
 - ➤ Es comúnmente utilizada en componentes de GUI como botones y etiquetas.
 - > Proporciona métodos para obtener la imagen subyacente y dimensiones del icono.
 - ➤ Es parte del paquete javax.swing.
 - ➤ Desde: JDK1.2

```
import java.awt.Image;
import java.io.BufferedReader;
import java.io.File;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.IOException;
import java.swing.DefaultComboBoxModel;
import javax.swing.ImageIcon;
import javax.swing.TileChooser;
import javax.swing.Jabel;
import javax.swing.Jabel;
import javax.swing.JoptionFane;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
```

> Modulo

En el modulo hicimos toda la parte de interfaz grafica asi como arrancar hilos

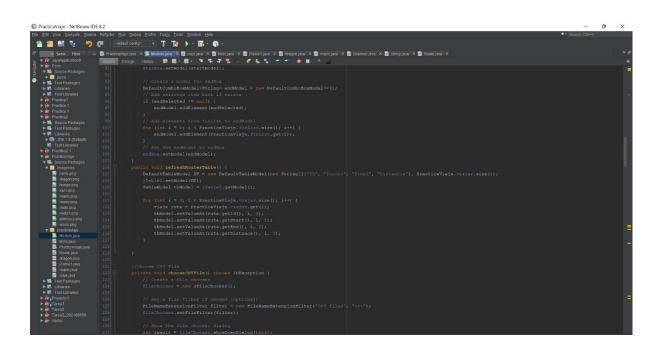
```
// Acc finalize

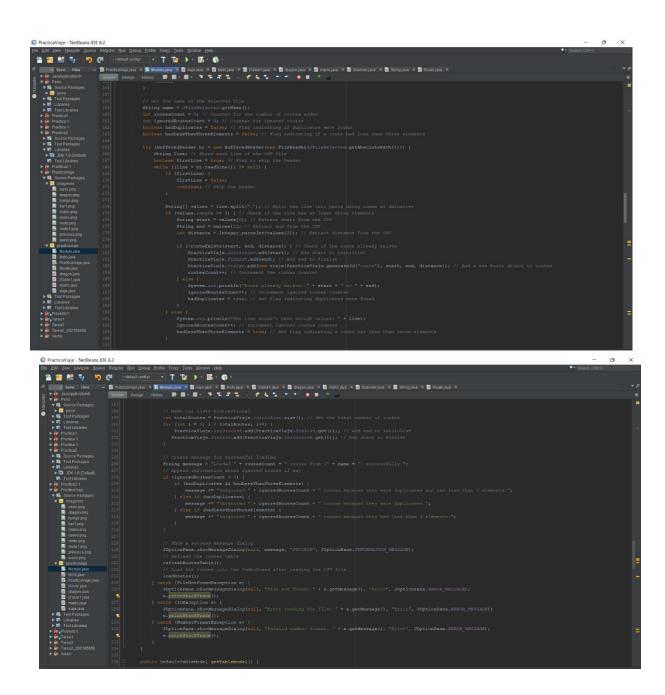
// Acc finalize

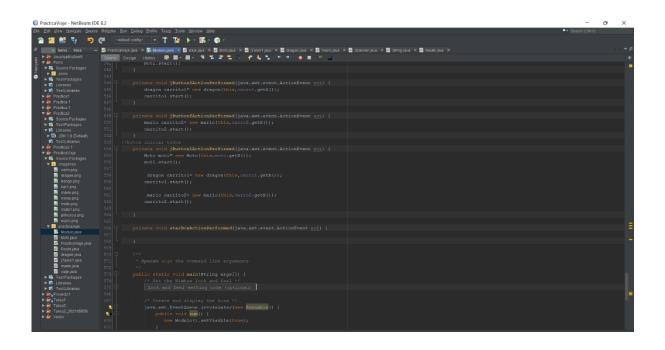
public Module() (
inticomponents();
setLocationRelativeTh(mull);

moto.setBounds(5, 120, 100, 60);

ImageIcon mot * new ImageIcon(quitampe().getResource(*../imagenes/motol.pmg*));
Icon motol * new ImageIcon(motol.getResource(*../imagenes/motol.pmg*));
Icon motol * new ImageIcon(quitampe().getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../imagenes/motol.getResource(*../i
```







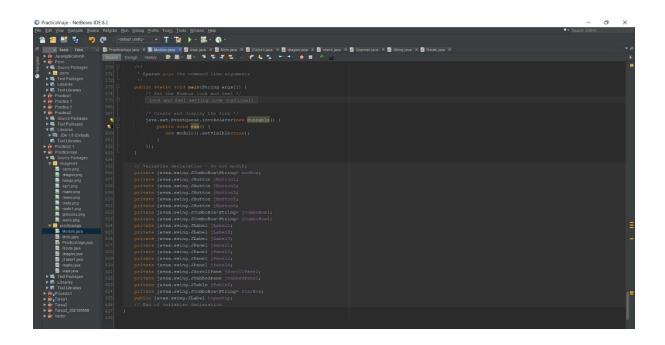


DIAGRAMA DE CLASE

