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



**MANUAL TÉCNICO UDRIVE**

MARCOS DANIEL BONIFASI DE LEON

CARNÉ: 202202410

SECCIÓN: B

GUATEMALA, 1 DE ABRIL DEL 2,024

# ÍNDICE

[**ÍNDICE 1**](#_k1cgegpfdrrl)

[**INTRODUCCIÓN 1**](#_574nc6ehlpl)

[**OBJETIVOS 1**](#_3bsxbi49eln7)

[1. GENERAL 1](#_38w1k6cukb5i)

[2. ESPECÍFICOS 1](#_poh928tpbmga)

[**ALCANCES DEL SISTEMA 1**](#_twvsj8hvwtb9)

[**ESPECIFICACIÓN TÉCNICA 1**](#_hxtoxxv43krt)

[● REQUISITOS DE HARDWARE 1](#_mfriwpcdfh1l)

[● REQUISITOS DE SOFTWARE 1](#_8h6kxoigku56)

[**DESCRIPCIÓN DE LA SOLUCIÓN 2**](#_14axd4p9n3ry)

[**LÓGICA DEL PROGRAMA 2**](#_ng9quldintk8)

[❖ NOMBRE DE LA CLASE  
Captura de las librerías usadas 2](#_xush55kxig5)

[➢ Librerías 2](#_h84kbu3eveip)

[➢ Variables Globales de la clase \_(El nombre de su clase actual) 3](#_jmrvbw5ti5o0)

[➢ Función Main 3](#_lg5om9s2qe42)

[➢ Métodos y Funciones utilizadas 3](#_xbejjcbelcz7)

# 

# INTRODUCCIÓN

El manual técnico busca describir todos los detalles de hardware y software necesarios para ejecutar correctamente la aplicación de escritorio desarrollada en el lenguaje de programación Java. Así mismo, explicar el flujo de la aplicación a nivel de código y la lógica de clases que ésta utilizar para manejar tanto los objetos y sus implementaciones.

# OBJETIVOS

## GENERAL

* 1. Guiar detalladamente al lector sobre todos los detalles técnicos para poder ejecutar correctamente la aplicación de escritorio desarrollada en Java.

## ESPECÍFICOS

* 1. Objetivo 1: Especificar los requisitos técnicos, es decir el software necesario.
  2. Objetivo 2: Detallar el flujo de la aplicación a nivel de código de programación por medio de capturas de pantalla y explicando el hecho por el cual se hizo cada bloque de código.

# ALCANCES DEL SISTEMA

El objetivo de este manual técnico es proporcionar a los usuarios una guía detallada sobre el funcionamiento y la implementación de una aplicación de escritorio desarrollada en Java Swing, que simula el movimiento de vehículos utilizando threads. A través de este manual, los usuarios podrán comprender cómo se ha diseñado y estructurado el software, así como también aprenderán a utilizar eficazmente sus funcionalidades y manipular los hilos de ejecución para simular viajes de carros de manera eficiente. Además, el manual proporciona información sobre las mejores prácticas de programación concurrente en Java y ofrece consejos útiles para la resolución de problemas comunes que puedan surgir durante el desarrollo y la ejecución de la aplicación. Con esta documentación, se pretende capacitar a los usuarios para que puedan sacar el máximo provecho de la aplicación y personalizarla según sus necesidades específicas.

# ESPECIFICACIÓN TÉCNICA

## REQUISITOS DE HARDWARE

* + Memoria RAM igual o mayor a 4GB.

## REQUISITOS DE SOFTWARE

* + Computadora con sistema operativo Windows, MacOS o alguna distribución de Linux.
  + Permisos de administrador
  + Java 11
  + IntelliJ IDEA de JetBrains

# DESCRIPCIÓN DE LA SOLUCIÓN

Para abordar los requerimientos del enunciado, comence con un análisis detallado de las especificaciones proporcionadas. Identificamos claramente la necesidad de desarrollar una aplicación de escritorio en Java Swing que simule viajes de vehículos utilizando threads. Este análisis inicial me permitió comprender la naturaleza del problema y establecer las funcionalidades principales que debían implementarse.

Luego, llevé a cabo un proceso de diseño en el que definimos la arquitectura general de la aplicación. Esto incluyó la estructura de clases y la interacción entre los diferentes componentes. Consideré cómo gestionaríamos los hilos de ejecución para simular el movimiento de los carros de manera eficiente y sincronizada.

Además, evalué las mejores prácticas de programación concurrente en Java para garantizar un diseño robusto y seguro. Esto implicó la identificación y la mitigación de posibles problemas de concurrencia, como las condiciones de carrera y el acceso concurrente a recursos compartidos.

En resumen, el enfoque se centró en comprender completamente los requerimientos del enunciado, diseñar una solución efectiva y segura, y aplicar las mejores prácticas de desarrollo de software en el contexto de la programación concurrente en Java. Este proceso me permitió crear una aplicación que cumple con los objetivos planteados de manera eficiente y confiable.

# LÓGICA DEL PROGRAMA

## Main

A screen shot of a computer program

Description automatically generated

### Librerías

Javax.swing: esta libería implementa todo el paquete para el desarrollo de aplicaciones de escritorio en Java, GUI.

Javax.awt: es un kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java.

Java.util: utilidades generales para el desarrollo de aplicaciones en Java.

Java.time: para trabajar con fechas en Java.

Java.io: se utilizó para la serealización de datos.

### Variables Globales de la clase Main

A screenshot of a computer program

Description automatically generated

Routes: para persistir los datos de las rutas.

Trips: para persistir los datos de los viajes.

vehiclesAvailable: para persistir los vehiculos disponibles.

onGoingTrips: viajes actuales.

counterRoutes: contador para generar los ids de las rutas.

counterTrips: contador para generar los ids de los viajes.

### Función Main

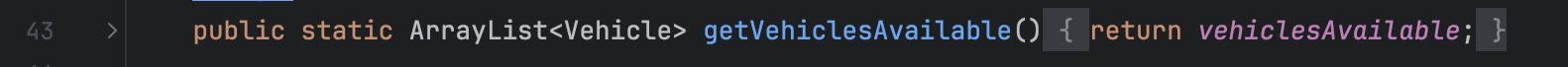
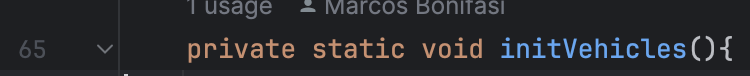
Sirvió para le inicializar la aplicación e inicializar valores por defecto, así como los vehículos. También para la lectura del archivo binario.

A screen shot of a computer

Description automatically generated

### Métodos y Funciones utilizadas

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

*   
  Retorna las rutas.
*   
  Retorna los vehículos disponibles.
* 

Inicializa los vehículos por defecto (9).

* A black background with text

  Description automatically generated

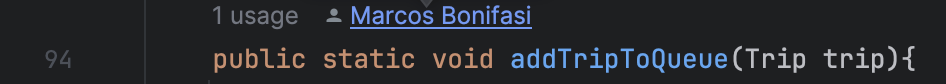
Genera un ID dependiendo el parámetro dado.

* 

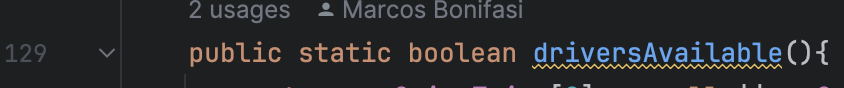
Agrega una ruta al arraylist de rutas.

* 

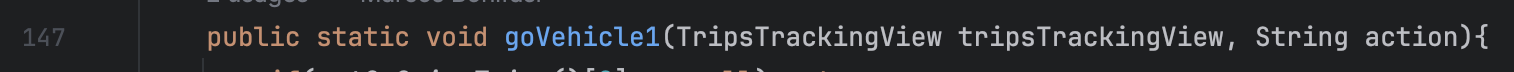
Agrega un viaje al arraylist de viajes.

* 

Agrega un viaje a la cola de viajes.

* 

Bandera para saber si hay conductores disponibles.

* 

Corre el hilo para el vehículo 1.