



MANUAL TECNICO

TRAVELMAPGT (JAVA)

Nombre

> Maria Monserrat Gomez Rabatu

Carnet

> 202030849

Curso

> Laboratorio Estructura de Datos

Requisitos del Sistema

Para utilizar el TravelMapGT, tu sistema debe cumplir con los siguientes requisitos:

- **Sistema Operativo:** Windows 7 o superior, macOS, o una distribución de Linux compatible.
- **Entorno Java JDK:** Es necesario tener instalado la última versión de Java. Asegúrate de que el lenguaje esté correctamente configurado en el PATH del sistema para poder ejecutar el archivo ".jar".
- **Espacio en Disco:** Al menos 100 MB de espacio libre en disco para la instalación y almacenamiento de datos.
- **Memoria RAM:** Se recomienda tener al menos 1GB de memoria RAM para un rendimiento óptimo.

◇ Instalación de un Compilador de C++

a. Windows:

1. Descargar Java JDK 21

Ve al sitio web oficial de Oracle para descargar Java JDK 21: [Descargar Java JDK 21](#). (Asegúrate de seleccionar la versión adecuada para tu equipo)

2. Ejecutar el Instalador

Una vez descargado el archivo de instalación, doble clic en él para ejecutarlo. Sigue las instrucciones del asistente de instalación. Acepta los términos y condiciones cuando se te solicite.

3. Configurar Variables de Entorno

Para que puedas compilar y ejecutar programas Java desde cualquier directorio, debes configurar las variables de entorno. Abre el Panel de Control y busca Sistema, haz clic en Configuración avanzada del sistema y en la pestaña Opciones avanzadas, haz clic en Variables de entorno. En la sección Variables del sistema, busca JAVA_HOME y agrega la ruta de la carpeta donde instalaste Java JDK 21. En la misma ventana de Variables de entorno, busca la variable Path en la sección Variables del sistema. Edita la variable y agrega la ruta de la carpeta bin de Java JDK 21.

4. Ejecutar Archivos .jar:

Para ejecutar archivos .jar, simplemente abre una ventana de comandos, navega hasta la ubicación del archivo .jar y ejecuta el comando: `java -jar nombre_del_archivo.jar`

b. MacOS:

1. Descarga el JDK:

Visita el sitio web oficial de Oracle para descargar el JDK: [Descargar Oracle JDK](#). Elige la versión adecuada para macOS.

2. Instala el JDK: Ejecuta el archivo .dmg descargado

Introducción

TravelMapGT es una implementación de un mapa diseñado para ser ejecutado en una interfaz gráfica. Este manual proporciona una visión general de la estructura y el funcionamiento del programa, así como una guía para su uso y desarrollo.

Estructura del Proyecto

El proyecto está organizado en varias clases, cada una con un propósito específico:

- **Archivo travelmapgt:** Contiene las clases necesarias para el arranque de la app,.
- **Archivo grafosN:** Contiene todos los archivos necesarios para la lógica de grafos.
- **Archivo swing:** Contiene los forms y clases necesarias para JavaSwing, incluyendo clases custom.

Clase TravelMapGT

1. Atributos:

- Variable estática `newMain` de tipo `Main`.

2. Método main:

- El método `main` es el punto de entrada de la aplicación. Se ejecuta automáticamente cuando se inicia el programa.
- Instancia de la clase `Main`, se la hace visible en la pantalla.

Clase Main

1. Atributos:

- `mc`: Instancia de la clase `MenuController` para gestionar las acciones del menú.
- `cl`: Instancia de `CardLayout` para administrar los paneles de la interfaz gráfica.

2. Constructor:

- `Main()` inicializa la interfaz gráfica y configura la apariencia y crea una instancia de `MenuController`.

3. Métodos:

- `accion1()`: Realiza acciones al seleccionar "Crear Nuevo Grafo": Crea un grafo de rutas según las preferencias del usuario (caminando o en vehículo).
- `accion2()`: Realiza acciones al seleccionar "Seleccionar Ruta": Limpia la interfaz y restablece los elementos. Habilita la carga de un nuevo mapa.
- `accion3()`: Realiza acciones al seleccionar "Ver Mapa": Limpia el mapa actual y repinta el grafo.
- `accion4()`: Realiza acciones al buscar rutas: Busca caminos entre el origen y el destino.

Clase MenuController

1. Atributos:

- `aux`: Referencia al panel auxiliar (utilizado para cambiar entre diferentes vistas).
- `actual`: Grafo actual que representa las rutas.
- `actualDoc`: Nombre del archivo actual (sin extensión) utilizado para generar el grafo.

2. Métodos

- `limpiarLabel`: Limpia la representación gráfica del mapa en el panel. Actualiza la interfaz gráfica.
- `crearGrafoNuevo`: Crea un nuevo grafo de rutas a partir de un archivo de entrada. Lee las rutas y sus características (tiempo en vehículo, tiempo a pie, consumo de gasolina, etc.). Genera el grafo y lo dibuja utilizando Graphviz. Muestra un mensaje de éxito y actualiza la representación gráfica del mapa.
- `repintarGrafo`: Actualiza la representación gráfica del grafo actual. Utiliza Graphviz para generar una nueva imagen del grafo.
- `buscarCaminos`: Busca caminos entre el origen y el destino. Crea listas de nodos para las rutas más eficientes y menos eficientes. Actualiza la representación gráfica del mapa.

Clase GrafoN

1. Atributos:

- `primero`: Nodo del grafo que representa al primero de la lista.
- `ultimo`: Nodo del grafo que representa al ultimo de la lista.
- `tipoBusqueda`: Numero que referencia el tipo de busqueda de caminos.

2. Métodos

- `vacio`: Comprueba si el grafo está vacío o no.
- `existeVertice`: Comprueba si ya existe un nodo con el dato que se quiere ingresar
- `newArista`: Crea un nuevo arco si existen los nodos origen y destino, puede o no tener peso.
- `nuevoNodo`: Crea (si no existe) un nuevo nodo.
- `encontrarNodo`: Busca un nodo en el grafo.
- `buscarCamino`: Recorre el grafo desde un nodo inicial "A", buscando caminos que lleven al nodo "B".
- `calcularPeso`: Calcula el peso total de un camino desde un nodo "A" hasta un nodo "B"
- `dibujarGraphviz`: Crea el archivo dot y png basado en la estructura del grafo y sus arcos y vertices.

Clase ListaAdyacencia

1. Atributos:

- primero: Arco del grafo que representa al primero de la lista.
- ultimo: Arco del grafo que que representa al ultimo de la lista

2. Métodos

- vacio: Comprueba si la lista de un nodo está vacia o no.
- adyacente: Comprueba si la relacion entre el nodo "A" entre un nodo "B" ya existe.
- nuevaAdy: Comprueba si ya existe la adyacencia, si no, inserta un nuevo arco

Clase GrafoN

1. Atributos:

- primero: Nodo del grafo que representa al primero de la lista.
- ultimo: Nodo del grafo que representa al ultimo de la lista.
- tipoBusqueda: Numero que referencia el tipo de busqueda de caminos.

2. Métodos

- vacio: Comprueba si el grafo está vacio o no.
- existeVertice: Comprueba si ya existe un nodo con el dato que se quiere ingresar
- newArista: Crea un nuevo arco si existen los nodos origen y destino, puede o no tener peso.
- nuevoNodo: Crea (si no existe) un nuevo nodo.
- encontrarNodo: Busca un nodo en el grafo.
- buscarCamino: Recorre el grafo desde un nodo inicial "A", buscando caminos que lleven al nodo "B".
- calcularPeso: Calcula el peso total de un camino desde un nodo "A" hasta un nodo "B"
- dibujarGraphviz: Crea el archivo dot y png basado en la estructura del grafo y sus arcos y vertices.