MANUAL TÉCNICO

TRAVELMAP GT:
PROYECTO FINAL DE ESTRUCTURAS DE DATOS

BREVE DESCRIPCIÓN

El presente proyecto es una aplicación con interfaz gráfica la cual permite la visualización de grafos, en el contexto de mapas con rutas entre municipios y departamentos de Guatemala, tiene funciones de recibir archivos de entrada, cálculo de recorridos entre lugares y visualizaciones de varios tipos. Se utilizan implementaciones de Grafos y Árboles B para guardar estos datos.

HERRAMIENTAS UTILIZADAS

- ## LENGUAJE DE PROGRAMACIÓN: TypeScript

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. Anders Hejlsberg, diseñador de C# y creador de Delphi y Turbo Pascal, ha trabajado en el desarrollo de TypeScript. TypeScript es usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor, o extensiones para programas (Node.js y Deno).

TypeScript extiende la sintaxis de JavaScript, por tanto cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original.

- ## VISUAL STUDIO CODE

Visual Studio Code (también llamado VS Code) es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online).

- ## GITHUB

Es un sistema de control de versiones de código y gestión de proyectos, a su vez también funciona como una plataforma de estilo red social diseñada para desarrolladores para poder compartir código entre más personas y colaborar en el mismo.

- ## MACOS v13 - SISTEMA OPERATIVO

macOS (previamente Mac OS X, luego OS X) es una serie de sistemas operativos gráficos desarrollados y comercializados por Apple desde 2001. Es el sistema operativo principal para la familia de computadoras Mac de Apple. Dentro del mercado de computadoras de escritorio, portátiles, hogareñas y mediante el uso de la web.

macOS se basa en tecnologías desarrolladas entre 1985 y 1997 en NeXT. Se logró la certificación UNIX 03 para la versión Intel de Mac OS X 10.5 Leopard y todos los lanzamientos de Mac OS X 10.6 Snow Leopard hasta la versión actual también tienen la certificación UNIX 03. macOS comparte su núcleo basado en Unix, llamado Darwin, y muchos de sus frameworks con iOS 16, tvOS y watchOS.

- ## FRAMEWORK DE JAVASCRIPT - VUE.JS v3 (FRONTEND)

Vuejs es un framework de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página. Fue creado por Evan You, y es mantenido por él y por el resto de los miembros activos del equipo central que provienen de diversas empresas como Netlify y Netguru.

Vue.js cuenta con una arquitectura de adaptación gradual que se centra en la representación declarativa y la composición de componentes. La biblioteca central se centra sólo en la capa de vista. Las características avanzadas necesarias para aplicaciones complejas como el enrutamiento, la gestión de estados y las herramientas

de construcción se ofrecen a través de librerías y paquetes de apoyo mantenidos oficialmente, con Next.js como una de las soluciones más populares.

Vue.js permite extender el HTML con atributos HTML llamados directivas. Las directivas ofrecen funcionalidad a las aplicaciones HTML, y vienen como directivas incorporadas o definidas por el usuario.

CLASES Y ALGORITMOS

Estructuras implementadas:

Grafo:

Clases Principales:

1. GraphNode:
    - Representa un nodo en el grafo, identificado por su nombre.
    - Cada nodo tiene un nombre único.
    - No tiene métodos adicionales aparte del constructor para inicializar el nombre del nodo.
2. Edge:
    - Representa un borde (o arista) del grafo, que conecta dos nodos.
    - Contiene información sobre el tiempo de viaje en vehículo y caminando, consumo de gasolina, desgaste físico y distancia entre los nodos.
    - También incluye un arreglo de intervalos de tráfico para manejar probabilidades de tráfico en diferentes momentos.
    - No tiene métodos adicionales aparte del constructor para inicializar los atributos del borde.
3. Graph:
    - Esta clase maneja la estructura completa del grafo.
    - Propiedades:
        - `nodes`: Una lista de todos los nodos en el grafo.
        - `edges`: Una lista de todos los bordes en el grafo.

    - `adjacencyList`: Un mapa que almacena la lista de adyacencia del grafo, que mapea cada nodo a sus nodos adyacentes.
    - `isDirected`: Una bandera que indica si el grafo es dirigido o no.
- Métodos:
    - `getNodes()`: Retorna una lista de nodos.
    - `isEmpty()`: Verifica si el grafo está vacío.
    - `addNode(node: GraphNode)`: Agrega un nodo al grafo si no existe.
    - `addEdge(edge: Edge)`: Agrega un borde al grafo y actualiza la lista de adyacencia.
    - `buildGraph(routeData: string[])`: Construye el grafo a partir de datos de rutas proporcionados.
    - `findAllRoutes(startName: string, endName: string)`: Encuentra todas las rutas posibles entre dos nodos utilizando búsqueda en profundidad (DFS).
    - `showAvailableOptions(startName: string, endName: string)`: Muestra todas las opciones de ruta entre dos nodos.
    - `showBestRoute(startName: string, endName: string)`: Encuentra y muestra la mejor ruta entre dos nodos según la distancia.
    - `showWorstRoute(startName: string, endName: string)`: Encuentra y muestra la peor ruta entre dos nodos según la distancia.
    - `findBestAndWorstRoute(transportMode: TransportationMode, startName: string, endName: string, criteria1: OptimizationCriteria, criteria2?: OptimizationCriteria)`: Encuentra la mejor y peor ruta entre dos nodos considerando criterios de optimización.
    - `convertToUndirectedGraph()`: Convierte el grafo dirigido en no dirigido.
    - `updateTrafficData(trafficData: string[])`: Actualiza los datos de tráfico en los bordes del grafo.
    - `generateDotGraph(edgesToHighlight: Edge[] = [])`: Genera una representación del grafo en formato DOT para su visualización.

Uso y Funcionalidades:

- La clase `Graph` proporciona herramientas para construir, manipular y analizar grafos de rutas de transporte.
- Permite encontrar las rutas más cortas y largas, considerando diferentes criterios como distancia, consumo de combustible y desgaste físico.
- Puede generar visualizaciones del grafo para ayudar en la comprensión y la toma de decisiones.
- La clase es altamente modular, lo que facilita la incorporación de nuevas funcionalidades o extensiones en el futuro.

Arbol B:

Clases Principales:

1. TreeElement<T>:
    - Interfaz que define la estructura de los elementos del árbol.
    - Cada elemento tiene un identificador único (`id`) y un valor (`value`).
2. TreeNode<T>:
    - Representa un nodo en el árbol B.
    - Tiene una lista de claves (`keys`) que contiene los elementos del nodo.
    - También tiene una lista de hijos (`children`) que son los nodos descendientes.
3. BTree<T>:
    - Esta clase implementa un árbol B.
    - Propiedades:
        - `root`: El nodo raíz del árbol.
        - `degree`: El grado del árbol, que determina el número máximo de claves por nodo.

        - `dotGraph`: Una cadena que representa el gráfico en formato DOT.
        - `idCounter`: Un contador utilizado para asignar identificadores únicos a los elementos del árbol.
    - Métodos:
        - `search(id: number): boolean`: Busca un elemento en el árbol dado su identificador.
        - `insert(value: T): void`: Inserta un nuevo elemento en el árbol.
        - `splitChild(parent: TreeNode<T>, index: number): void`: Divide un nodo hijo de un nodo padre en caso de que esté lleno.
        - `insertNonFull(node: TreeNode<T>, element: TreeElement<T>): void`: Inserta un elemento en un nodo no lleno del árbol.
        - `generateDOT(node: TreeNode<T>): void`: Genera la representación del árbol en formato DOT.
        - `generateGraph(): string`: Genera el gráfico completo del árbol en formato DOT.

Funcionalidades y Uso:

- La clase `BTree` permite buscar elementos y agregar nuevos elementos al árbol.
- Implementa la inserción siguiendo las reglas de un árbol B, dividiendo nodos cuando sea necesario.

- Proporciona un método para generar una representación visual del árbol en formato DOT, que puede ser utilizado para visualizar el árbol utilizando herramientas como Graphviz.
- Especifica un grado para el árbol, lo que determina la cantidad máxima de claves por nodo y la estructura general del árbol.

Otras clases y utilidades:

También se encuentran otros archivos, que simplemente se utilizan para el manejo de la sesión y todas las estructuras que una de estas puede llegar a poseer mientras se ejecuta el programa, por ejemplo el *store* currentSession se encarga de manejar una sola misma estructura tanto de árbol como de grafo durante una misma sesión y preservar los datos entre los cambios

INSTRUCCIONES PARA EJECUCIÓN

Para ejecutar este programa es necesario tener instalado el programa Nodejs en donde se desee ejecutar.

Pasos para la ejecución:

- Descargar el código fuente del repositorio
- Utilizar una terminal y navegar hasta el codigo descargado
- Correr el comando npm install de node para instalar todo lo requerido para el proyecto
- Correr el comando npm run dev para poner en marcha el programa
- Con la dirección proveída por el programa de terminal, abrirlo en un navegador web moderno