MANUAL TECNICO:  
La empresa Code&Bugs ha analizado su dos prácticas anteriores, dadas sus habilidades ha decidido confiar en usted, por lo que se le contrata para el proyecto TravelMapGT, el cual es una aplicación basada en algunas características de Google Maps

Requerimientos mínimos para el entorno

de desarrollo:

Para el desarrollo de este programa e necesito con las siguiente características:

ProcesadorIntel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz

RAM instalada16.0 GB (15.8 GB utilizable)

Tipo de sistema Sistema operativo de 64 bits, procesador x64

Windows 11 última versión.

## Diccionario de Clases y Métodos

* **Métodos del evento ActionPerformed:**
  + AplicarActionPerformed: Aplica los valores ingresados en algún componente (reloj probablemente).
  + ControlRelojActionPerformed: Similar al anterior, parece relacionado con el control del reloj.
  + ReanudarActionPerformed: Reanuda alguna acción pausada (reloj tal vez).
  + PausaActionPerformed: Pausa alguna acción en curso (reloj tal vez).
  + ArchivoActionPerformed: Abre un archivo de texto con información de lugares y distancias.
  + Archivo2ActionPerformed: Abre un archivo de texto con información sobre el tráfico.
  + OrigenActionPerformed: Actualiza la gráfica en base al origen seleccionado.
  + DestinoActionPerformed: Actualiza la gráfica en base al destino seleccionado.
  + OpcionesActionPerformed: Muestra información relacionada a la ruta seleccionada (consumo de gas, desgaste, etc.).
  + GasActionPerformed, FisicoActionPerformed, DistanciaActionPerformed: No tienen implementación específica en el código provisto.
  + ArbolActionPerformed: Calcula rutas posibles y las grafica en un Árbol B.
* **Métodos relacionados con la lectura de archivos:**
  + actualizarControlReloj: Actualiza un control (reloj posiblemente) con la hora actual.
* **Métodos relacionados con la gráfica:**
  + graficar: Genera la imagen de la gráfica en formato DOT y la muestra en la interfaz.
  + cargarImagenGrafo: Carga la imagen de la gráfica en un componente de la interfaz.
  + mostrarImagen: Muestra una imagen en un componente de la interfaz (posiblemente para mostrar imágenes de grafos o del Árbol B).
  + generarImagenDesdeDot: Genera una imagen en formato PNG utilizando el comando dot del sistema operativo.
* **Métodos del algoritmo de Dijkstra:**
  + calcularRutaMasCorta: Calcula la ruta más corta entre dos nodos utilizando el algoritmo de Dijkstra.
  + obtenerNodoMasCercano: Obtiene el nodo no visitado más cercano según las distancias calculadas.
* **Método relacionado con niveles (sin uso aparente):**
  + calcularNivel: Calcula el nivel de un nodo en relación a un nodo origen (parece estar corregido en el código).
* **Método relacionado con el Árbol B:**
  + graficarArbolB: Genera la imagen del Árbol B utilizando Graphviz.

**Clase:**

* ArbolB: Esta clase representa un árbol B.
  + **Atributos:**
    - raiz: Puntero al nodo raíz del árbol.
    - orden: Orden del árbol B (número máximo de claves por nodo).
  + **Métodos:**
    - constructor(int ordenArbol): Crea un árbol B vacío con el orden especificado.
    - buscar(int clave): Busca una clave en el árbol y devuelve una lista con los nodos recorridos en la búsqueda.
    - buscarRecursivo(int clave, NodoArbolB nodo): Función recursiva auxiliar para la búsqueda.
    - agregarNodosRuta(NodoArbolB nodo, List<String> ruta): Agrega los nodos de una ruta a una lista.
    - insertar(int idRuta, LinkedList<String> ruta): Inserta una ruta en el árbol.
    - insertarNoLleno(NodoArbolB nodo, int idRuta, LinkedList<String> ruta): Inserta una clave en un nodo no lleno.
    - dividirHijo(NodoArbolB padre, int indice, NodoArbolB hijo): Divide un nodo hijo en dos nodos.
    - generarCodigoDot(): Genera el código DOT del árbol B.
    - generarCodigoDotRecursivo(NodoArbolB nodo, StringBuilder dotBuilder): Función recursiva auxiliar para generar el código DOT.

**Clase interna:**

* NodoArbolB: Representa un nodo del árbol B.
  + **Atributos:**
    - claves: Arreglo que almacena las claves del nodo.
    - numClaves: Número de claves almacenadas en el nodo.
    - hijos: Arreglo de punteros a nodos hijos.
    - hojaNodo: Indica si el nodo es una hoja (no tiene hijos).
* actualizarControlReloj: Actualiza un control (reloj posiblemente) con la hora actual.
* graficar: Genera la imagen de la gráfica en formato DOT y la muestra en la interfaz.
* cargarImagenGrafo: Carga la imagen de la gráfica en un componente de la interfaz.
* mostrarImagen: Muestra una imagen en un componente de la interfaz (posiblemente para mostrar imágenes de grafos o del Árbol B).
* generarImagenDesdeDot: Genera una imagen en formato PNG utilizando el comando dot del sistema operativo.
* calcularRutaMasCorta: Calcula la ruta más corta entre dos nodos utilizando el algoritmo de Dijkstra.
* obtenerNodoMasCercano: Obtiene el nodo no visitado más cercano según las distancias calculadas.
* calcularNivel: Calcula el nivel de un nodo en relación a un nodo origen (parece estar corregido en el código).
* graficarArbolB: Genera la imagen del Árbol B utilizando Graphviz.