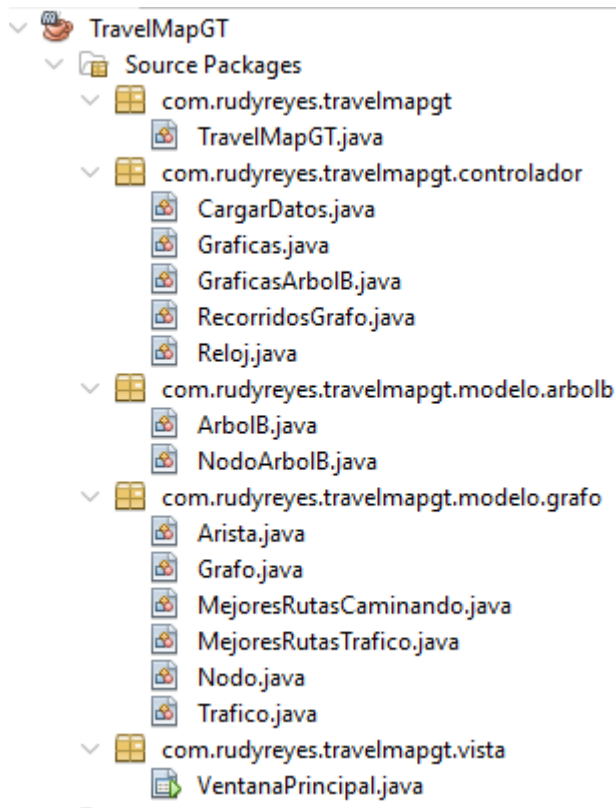


MANUAL TÉCNICO

1. Introducción y Propósito:

El siguiente proyecto contiene un software desarrollado en java, cuya función permite a las personas de Guatemala organizar sus viajes desde un punto A a un punto B por medio de las siguientes opciones: Vehículo y Caminando.

Arquitectura y Diseño:



Esta es la estructura del proyecto que esta basada en el modelo arquitectónico MVC (Modelo-Vista-Controlador).

Explicación de los paquetes:

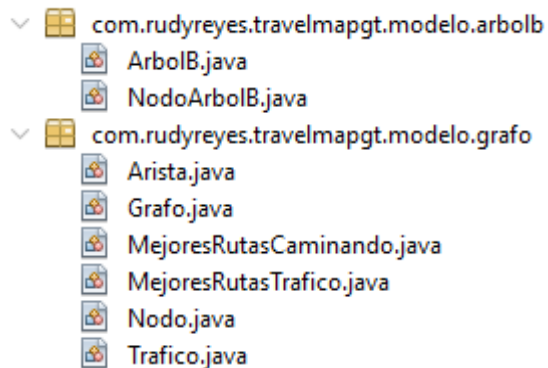
controlador: Aqui se encuentran todos los paquetes que nos permitirá obtener las acciones del usuario en la Vista y procesarla aquí haciendo uso del modelo.

modelo: Son las clases necesarias para el funcionamiento del sistema, y para guardar los datos de este.

vista: Contiene la ventana principal, que será mostrada al usuario

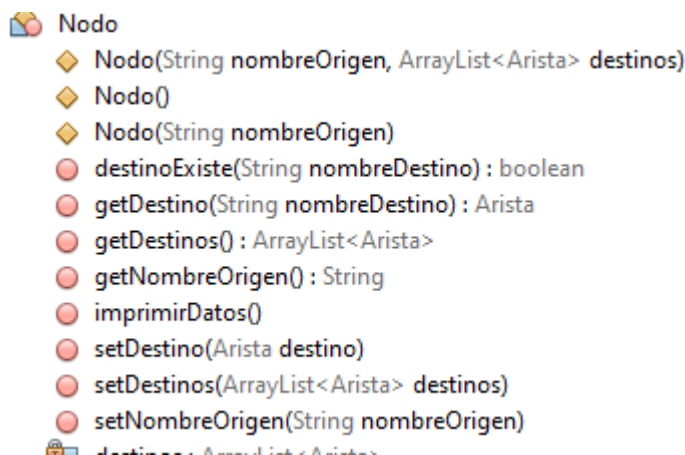
Detalle de las Clases:

Modelo:



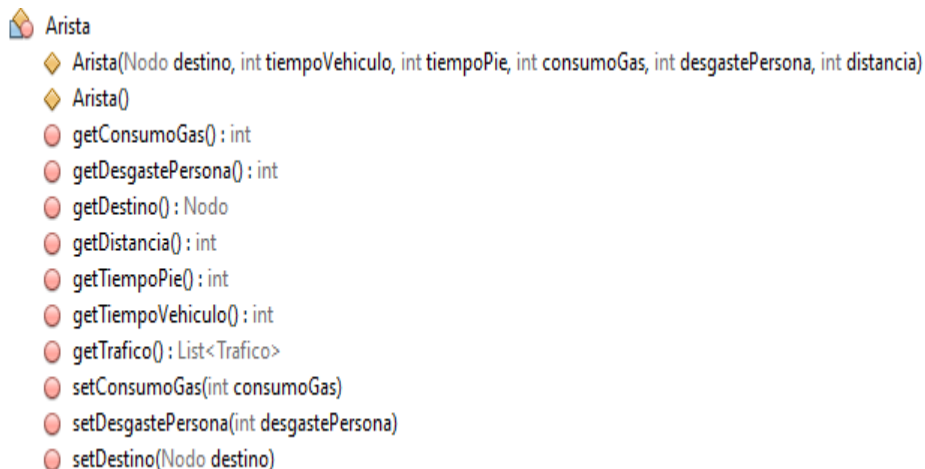
Como ya se había mencionado anteriormente estas clases nos permiten el funcionamiento del software, estas son necesarias para definir los objetos del sistema, como lo son los árboles, grafos y nodos.

Clase Nodo:



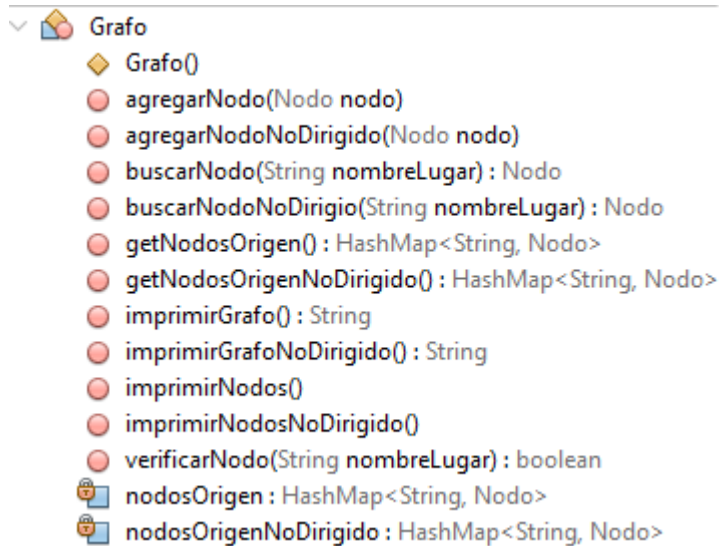
Esta clase contiene los atributos de un nombreOrigen y un arreglo de destinos, este arreglo contiene objetos del tipo Arista, más adelante se explicará que elementos contiene esta arista, los demás métodos conformado por esta clase, son métodos getters y setters.

Clase Arista:



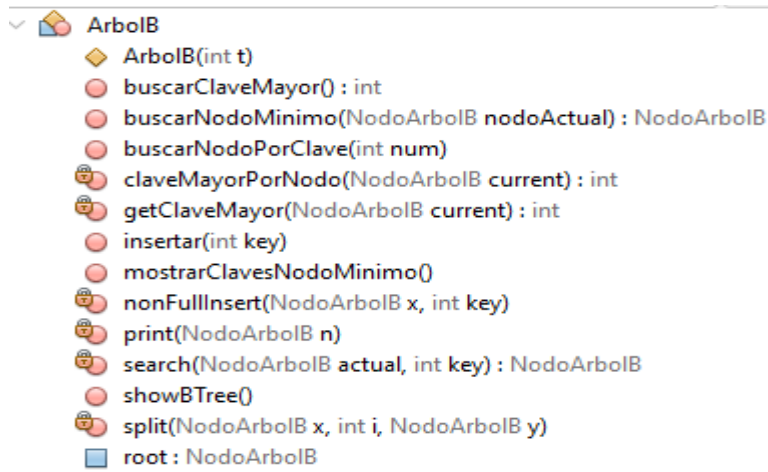
Los atributos que contiene esta clase son un nodo Destino y atributos del tipo entero como lo son el tiempo,, el desgaste y etc. métodos getters y setters.

Clase Grafo:



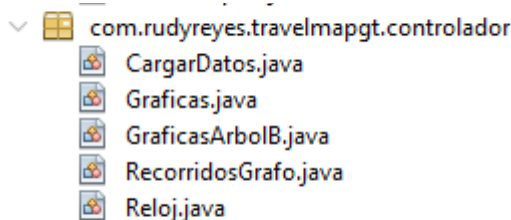
Los atributos de esta clase prácticamente son dos, y mapa de NodosOrigen y otro mapa de NodosOrigenNoDirigido, el primero tiene la función de mapear un grafo dirigido y el segundo un grafo no dirigido, los demás son getters y setters además clases para imprimir los datos.

Clase ArbolB:



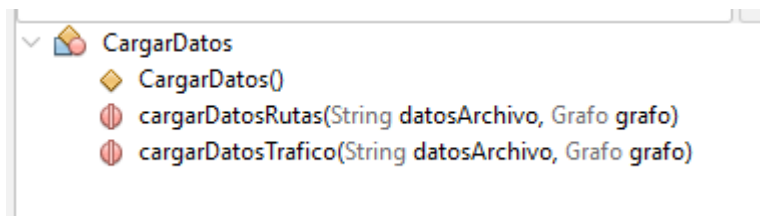
Esta clase nos permite manejar el árbol b del sistema, contiene todos los metodos necesarios, para la insercion y busqueda de datos, ademas del balanceo automatico para que siempre se mantenga ordenado.

Controlador:



Clases que nos permiten estructurar los datos que el usuario nos manda desde la vista y agregarlos a los grafos.

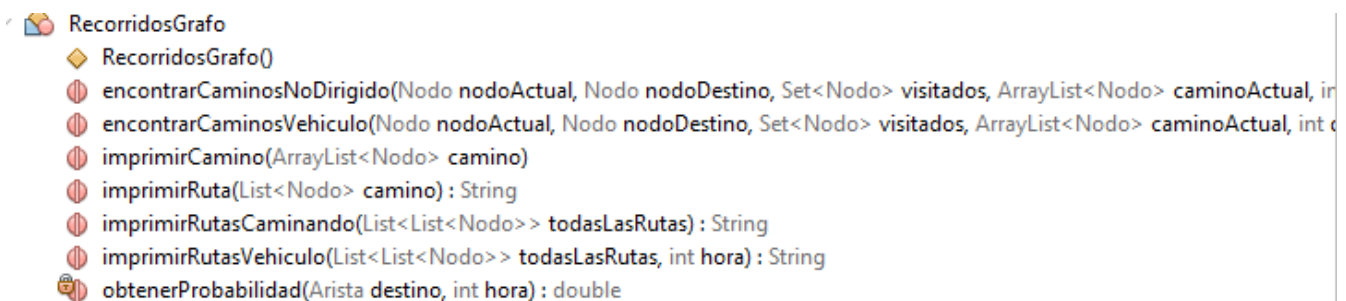
Clase CargarDatos:



Esta clase contiene solo dos metodos, pero son los mas importantes del sistema, ya que permite obtener las rutas que el usuario ingrese, e

ingresarlas al grafo, prácticamente divide los datos, obtiene los Nodos Origen y Destino, y los ingresa al grafo con sus respectivas aristas.

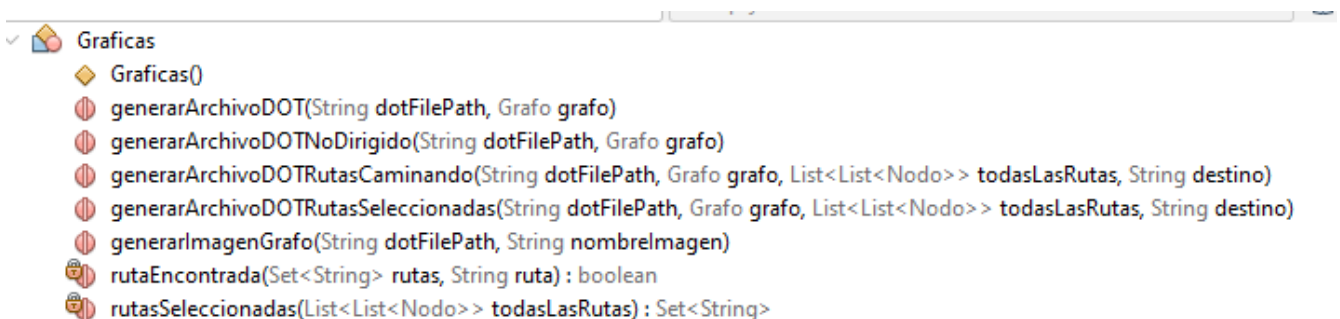
Clase RecorridosGrafo:



Esta sin duda es la segunda clase mas importante del software ya que nos permite obtener los recorridos de un NodoInicio y un NodoFinal, los dos metodos importantes son:

encontrarCaminosVehiculo: Nos devuelve un listado de rutas posibles si el usuario va en vehiculo.

Clase Graficas:



Esta clase nos genera las imagenes de los grafos, nos puede generar 4 tipos de imagenes:






generarArchivoDot: Nos genera el Dot del grafo de todo el sistema con direcciones.

generarArchivoDotNoDirigido: Nos genera el dot del grafo completo no dirigido.

generarArchivoDotRutasCaminando: Nos genera el grafo con la diferencia de que pueden apreciar todas las rutas de un punto inicial a otro, este dot generado es un grafo no dirigido.
















generarArchivoDotRutasSeleccionadas: Este es igual al anterior solo que es un grafo dirigido.

Clase GraficasArbolB:

 GraficasArbolB	Clase que contiene los métodos necesarios para generar la imagen del árbol
 GraficasArbolB()	
 buscarPaginas(NodoArbolB n, FileWriter writer, List<String> rutas)	
 generarDOT(String archivoDot, NodoArbolB root, List<String> ruta)	
 generarImagenGrafo(String dotFilePath, String nombrelimagen)	

Vista:

Clase VentanaPrincipal:

 VentanaPrincipal :: JFrame	Esta clase contiene los métodos que nos permitirán obtener las acciones del usuario desde la interfaz gráfica.
 VentanaPrincipal()	
 activarScroll()	
 botonAvanzarActionPerformed(ActionEvent evt)	
 botonEmpezarViajeActionPerformed(ActionEvent evt)	
 botonPararHoraActionPerformed(ActionEvent evt)	
 cargarRutasActionPerformed(ActionEvent evt)	
 cargarTraficoActionPerformed(ActionEvent evt)	
 generarArbolB()	
 generarGrafoPrincipal()	
 generarGrafoRutas()	
 initComponents()	
 main(String[] args)	
 obtenerHora() : int	
 recorrerViaje(String origen, String destino)	