En el siguiente escrito acerca del programa de grafos analizaremos lo siguiente:

Clase main

Al principio tenemos nuestra clase principal llamada maiin método main para ejecutar el programa, tenemos una variable que se llama initial tipo string que va a guardar el puntu inicial para arrancar la ruta. Tenemos un objeto de la clase graph que va a guardar el grafo en si. Después tenemos otra variable string current que es el punto final de la ruta.

Un objeto scaaner para poder acerca la lectura de los datos.

Tenemos un arreglo de cadenas de texto que va a guardar el nombre de todas las ciudades.

Dentro de los métodos que tengamos para el funcionamiento tenemos el constructor que se encargara de llenar el grafo con las ciudades y sus conexiones entre si. Luego tenemos otro método que se llama imprimir contenido que imprime las conexiones entre ciudad y ciudad mas la distancia entre cada una de ellas.

Después tendremos el método Dijkstra que se encargara de hacer la búsqueda dentro del grafo para obtener la ruta mas optima.

El main que incluimos el cual se encarga de acernos el menú principal para dar inicio a lo que requiera el usuario.

La clase edge te va a unir un vértice con otro, por ejemplo hara la ciudad que la línea es el vértice. Referencia hacia donde llega. Guardara el destino y el peso que significa el kilometraje. Tiene un método que te devulve el destino y los kilometrajes o peso. Como conecta tu tienes destino y peso.

La clase vertex te va a guarda la información de los nodos como nombre de ciudad y te dice el nodo previo con el siguiente adyacente para despues poder hacer la suma.

La clase Grafo guarda los vértices conectados con otros vértice, en si es la que crea el grafo. Tenemos un método addvertex para agregar vértices (nodos) y tenemos otro que se llama getvertex que nos devuleve en base a una cadena de texto un nodo. Metodo Djstrak el cual se encargara de hacer la búsqueda en nuestro grafo. Metodo compact edge que se va a encargar de elegir cual de las uniones es mas liviana en este caso corta. Metodo printpad imprimir el vértice siguiente del actual y esto nos va a servir para cuando queramos imprimir la ruta optima.