Proyecto

* 1. Especificación

-Entrada : Ciudad a donde se dirige el usuario y su ciudad de origen

- Salida: Camino más eficiente desde su ciudad de origen hasta la ciudad de destino

2. Diseño

Se crean 3 archivos, los cuales contienen la información de aeropuertos y ciudades. Luego otro con las funciones utilizadas en la búsqueda por anchura y por último un archivo de interfaz de usuario.

2.1 Estrategia

En el primer archivo de datos se generan dos diccionarios los cuales tendrán toda la información necesaria, así como las conexiones entre nodos. Con esto en cuenta, a partir de la función de búsqueda por anchura, se deja como nodo raíz la ciudad de origen. Se crea una función que lo que hace es dejarme en una lista el camino más eficiente entre la ciudad de origen y la ciudad de destino.

2.2 Modulos y subprogramas

Class GraphL (Clase encontrada en Taller\_grafos.py)

|  |  |
| --- | --- |
| BFS | A partir de un nodo raíz, se exploran todos los posibles caminos a los nodos del grafo |
| printBFS | Me retorna un arreglo que contiene el camino mas eficiente desde el nodo raíz hasta el nodo requerido |
| Ret\_values | Me retorna el valor de la llave ‘phi’ de un diccionario, el cual es el antecesor de la llave |

Datos.py

|  |  |
| --- | --- |
| Take\_position | Crea una lista con las llaves del diccionario y retorna a partir de un numero recibido, el valor en la lista creada |
| Connect | Función principal del proyecto. Genera las conexiones en el grafo y retorna, en conjunto con el archivo de BFS, la ruta mas eficiente entre la ciudad de origen y la ciudad de destino |
| answer | Encuentra el aeropuerto correspondiente a la ciudad y lo imprime en el orden de la ruta |