Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

**TRABAJO FINAL**

**Curso:**

Complejidad Algorítmica

**Carreras:**

Ingeniería de Software y Ciencias de la Computación

**Profesor:**

REYES SILVA, PATRICIA DANIELA

Sección:

CC41

**Integrantes:**

* Alejandro Olaf López Flores - U201922773
* Sebastian Jesus Ramirez Zapata – U20201C031

Noviembre 2022

**ÍNDICE**

**Índice ………………………………………………………………..………………..…………………. 2**

1. **Descripción del problema …………….……………………………………………… 3**
2. **Descripción y visualización del conjunto de datos (dataset) ………… 4**
3. **Propuesta ………………………………………………………………………………….... 5**
4. **Diseño del Aplicativo …………………………………………………………………… 6**
5. **Conclusiones ….……………………………………………………………………….…... 14**
6. **Bibliografía .…………………………………………………………………………….…... 14**
7. **Descripción del problema**

Nuestro problema planteado hace referencia a un conjunto de datos integrales con respecto a los viajes aéreos tomando en cuenta los vuelos directos o con escala para cierta ruta y también con el precio para llegar de un aeropuerto a otro.

Una persona o familia que desee realizar un viaje de turismo debe de tener en consideración todos los aspectos que esta incluye; tales como el equipaje, el precio por maleta, el precio del o los boletos del avión, gastos de comida y de movilidad, etc. La cual es más importante el precio de los boletos de avión, este puede ser de dos tipos **vuelos directos** (aquel vuelo que te llevará a tu destino sin la necesidad de parar en ningún otro aeropuerto en el trayecto) y **vuelo con escalas** (vuelo que parte desde el aeropuerto más cercano a un punto intermedio o de conexión donde tendrás que descender del primer avión, pasar los filtros necesarios y abordar otro avión para descender a tu destino o a otro aeropuerto(s) antes de hacerlo). Para la toma de esta decisión se tendrán que usar páginas como [www.despegar.com.pe](http://www.despegar.com.pe), [www.expedia.com](http://www.expedia.com), [www.kayak.com.pe](http://www.kayak.com.pe), [www.latamairlines.com](http://www.latamairlines.com) o la sección de vuelos dentro de la página de **Google**, todas estas páginas nos propondrán varios recorridos desde nuestro punto de origen al de destino con la intención de hacer una comparación de recorridos, precios y/o tiempo, lo ideal sería tener un boleto económico que me permita tener un mayor presupuesto para mis demás actividades de turismo. Sin embargo, se puede ver una notable diferencia de precio que existe entre estos vuelos, siendo el **vuelo con escala** de un **menor precio**, también para una mayor experiencia de turismo se puede **alargar** la escala, quedándose en la ciudad de conexión desde algunas horas hasta varios días.

Por ello planteamos la creación de un sistema que nos permita encontrar la ruta aérea con un menor precio para viajar de un aeropuerto a otro, teniendo como parámetros el **código IATA** y el **precio** entre los vuelos. En este informe se realizará la visualización de un grafo que grafica la relación entre los aeropuertos mediante aristas ponderadas que representarán el precio entre cada una de ellas.

Para tener un mayor conocimiento del código IATA se revisó la información de esta fuente: <https://aertecsolutions.com/2018/09/10/codigos-iata-y-oaci-de-los-aerodromos-conceptos-y-diferencias/> .

Icono

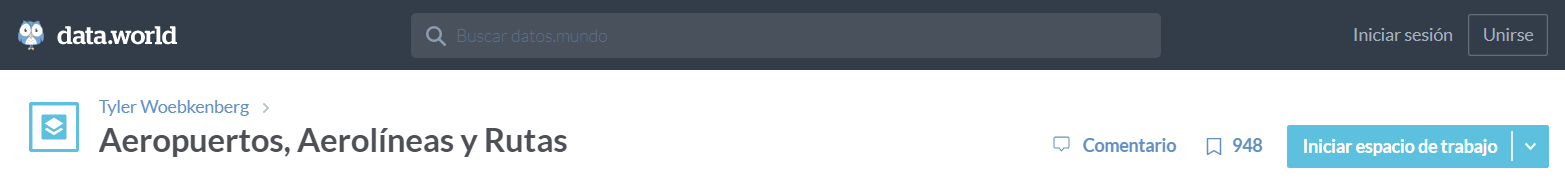
Descripción generada automáticamente

El código de aeropuertos de IATA Precios de los vuelos

está formado por grupos de tres letras

1. **Descripción y visualización del conjunto de datos (dataset):**

Para buscar los datos se usó la página web de “data.world” en el cual el usuario llamado Tyler Woebkenberg publicó dataset’s de aerolíneas, aviones y lo que nos interesó para presentarlo en este trabajo, dataset de aeropuertos y rutas. El usuario comenta que recopiló esta información de la página <https://openflights.org/data.html#route>.



La cantidad de datos excedían a los que necesitábamos, por ello realizamos un recorte en el dataset de aeropuertos y para que esté en sincronización con el otro dataset, se filtró cada ruta para que no presente conexiones inexistentes.

Presentamos 1000 aeropuertos identificados con su respectivo código IATA y nombre, además de ello tenemos 469 rutas que interconectan entre sí a los nodos (Aeropuertos).

**Descripción de Datos:**

Cada dato de nuestros datasets es utilizado para la resolución del programa representando las diferentes partes del grafo:

* **Nodos**:

Los nodos estarán representados por el código IATA de los aeropuertos obtenidas del primer dataset.

* **Aristas**:

Las aristas serán aristas ponderadas y obtendrán el valor del precio de los diferentes vuelos obtenidos del segundo dataset.

* **Unir Nodos y Aristas**:

Los uniremos en nuestro código a través de una lista de adyacencia, con lo cual se dibujará el gráfico que representa todas las conexiones de los vuelos del dataset, después se utilizará el algoritmo Dijkstra para poder hallar la ruta más barata y con un menor recorrido de un aeropuerto a otro.

1. **Propuesta**

Nuestro sistema buscará diferentes rutas por las cuales se pueda viajar ya sea en vuelo de escala o directo teniendo como diferenciador el precio que existe entre el viaje de un aeropuerto a otro. Para ello el trabajo se dividió en hitos:

Para el primer hito se realizó la representación de nuestros dataset en un grafo, para ello se optó por utilizar una lista de adyacencia debido a su mayor facilidad para manipular nuestros datos.

Para crear nuestra lista de adyacencia, primero se crearon dos listas, una de adyacencia y otra lista simple para los nodos (Aeropuertos) luego de ello se crean funciones como la de añadir, verificar redundancia e imprimir. Después de ello se realiza la lectura de nuestros .csv, y añadimos a la lista simple los aeropuertos(nodos). Para las rutas se utiliza la función ad\_arista que añade directamente las conexiones entre los nodos dentro de nuestra de adyacencia. Para imprimir la lista se usa una la función imprimirlistadeadyacencia.

Para graficar el grafo se utiliza nuestra clase Graph, ya visto anteriormente en clase. Sin embargo, nos tocaba implementarlo para que vaya a la par de nuestra de lista adyacencia, separamos en keys y values nuestra lista de adyacencia y lo imprimimos usando la función Dot de la clase Graph.

Además, el programa contara con la variación global de los precios de los pasajes aleatoriamente cada vez que busques una ruta.

**Solución en Collab:**

<https://colab.research.google.com/drive/10S02d0wlDURxSpiY5dO5zPgqHWtXKSVS?usp=sharing>.

**Dataset’s necesarios para corroborar la solución en Collab:**

<https://drive.google.com/drive/folders/1ljgFFquLfbkt5v24MQAlypnzoXpTcMUi?usp=sharing>.

**Dibujo del Grafo:**

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

Para una mejor resolución:

<https://drive.google.com/file/d/1DrH6Rh9jaWpZqgY-4cMf--Qzps36plUN/view?usp=sharing>.

1. **Diseño del Aplicativo**

* **Análisis de los requisitos:**
* Mapeo total de la mayoría de los aeropuertos existentes en el mundo mediante un grafo.
* Se realice la búsqueda de la mejor ruta teniendo como factor decisivo la ruta con menor precio.
* Poder escoger el aeropuerto de origen y además del aeropuerto de destino.
* Tener una interfaz intuitiva y fácil de manipular.
* **Diseño del sistema:**
  + **Back-End:** Para la creación de nuestro grafo se usarán una lista de adyacencia y una lista simple, donde esta se podrá graficar mediante el conjunto de herramientas **graphviz.** Para el cálculo de la ruta se usará el **algoritmo de Dijkstra** aplicado en nuestra lista de adyacencia.
  + **Front-End:** La visualización de nuestro sistema se realizará mediante el uso de la librería tkinter, esta librería nos permitirá realizar la interfaz gráfica que satisfaga nuestro proyecto.
* **Implementación:**

El desarrollo del código se realizó dentro del editor de código de Google Collab en la cual se utilizaron librerías como graphviz y numpy.

* **Solución en Collab:**

<https://colab.research.google.com/drive/10S02d0wlDURxSpiY5dO5zPgqHWtXKSVS?usp=sharing>.

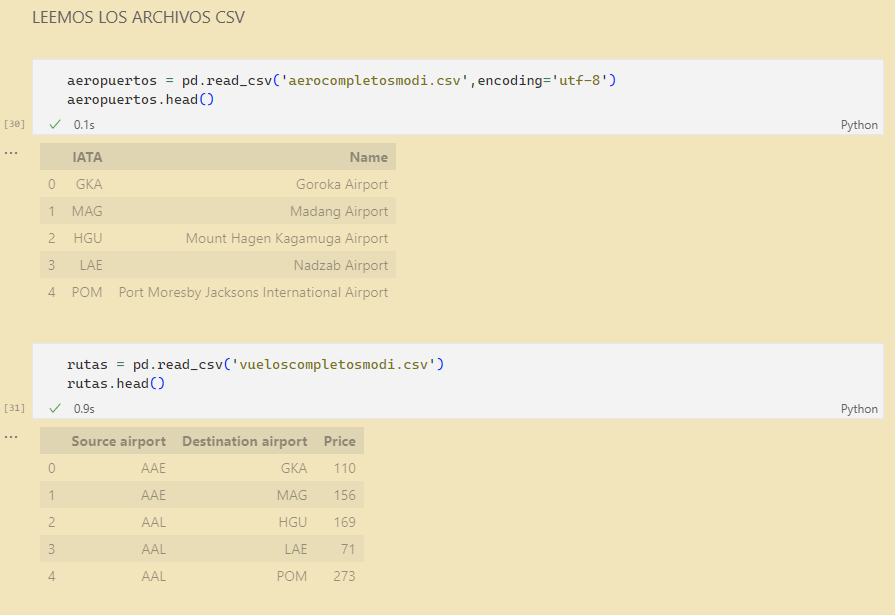
* **Repositorio de GITHUB:**

<https://github.com/olafLopezF/Trabajo-de-complejidad-algoritmica>

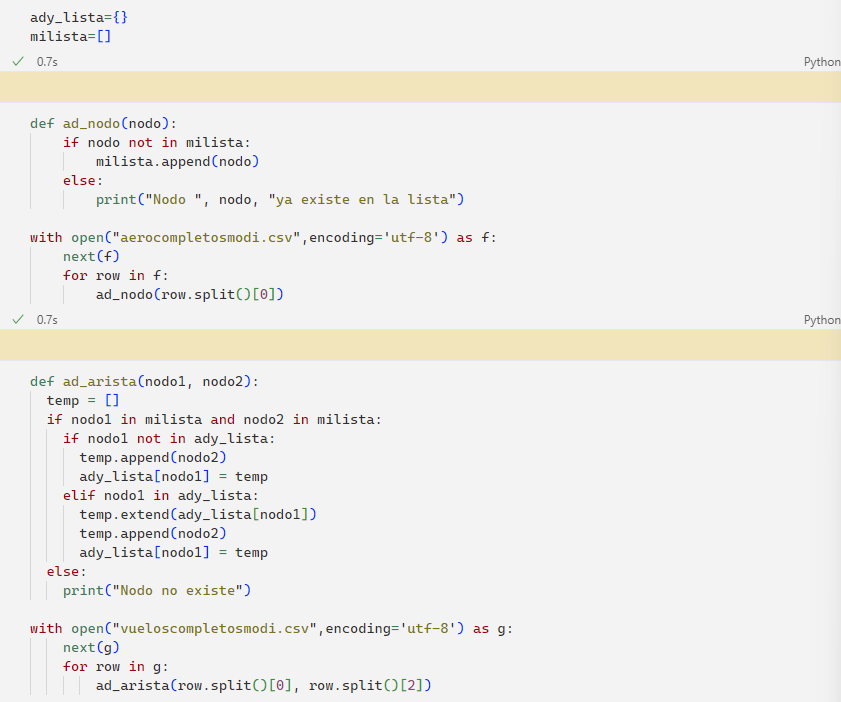
* **Verificación:**

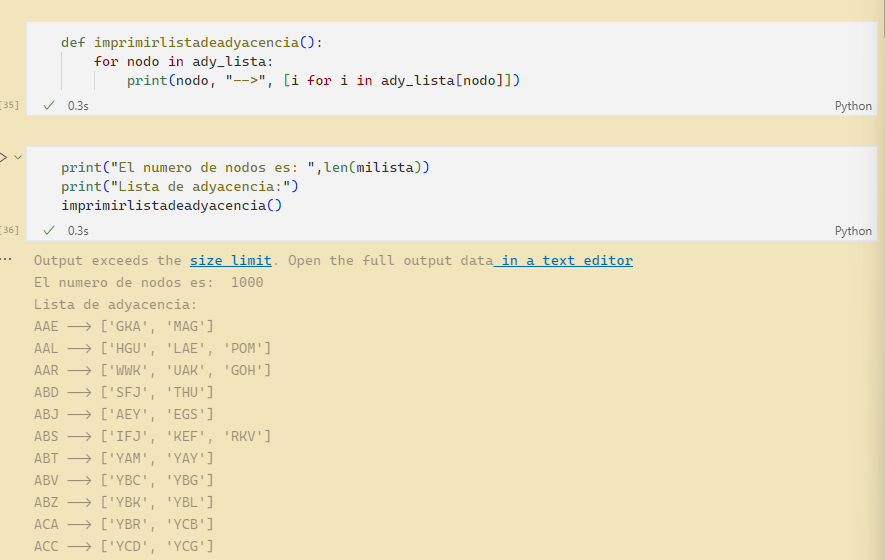
Capturas del código del programa:

**Lectura de archivos .csv**



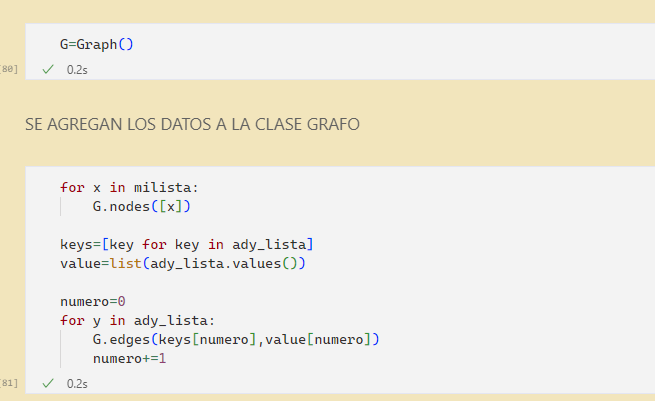
**Creación e impresión de la lista de adyacencia con la información de los archivos .csv**



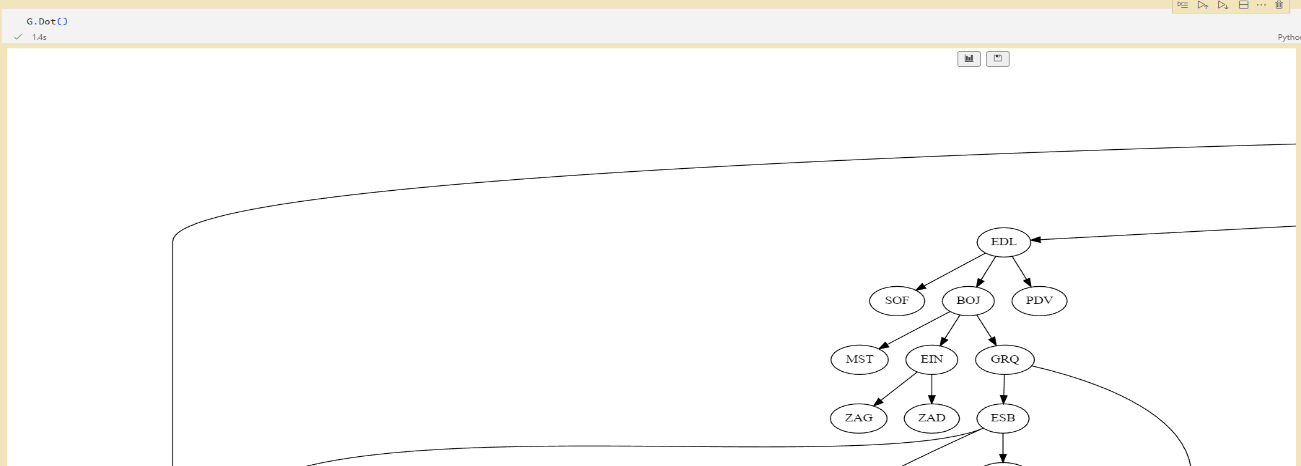


**Clase grafo y agregación de datos al grafo:**

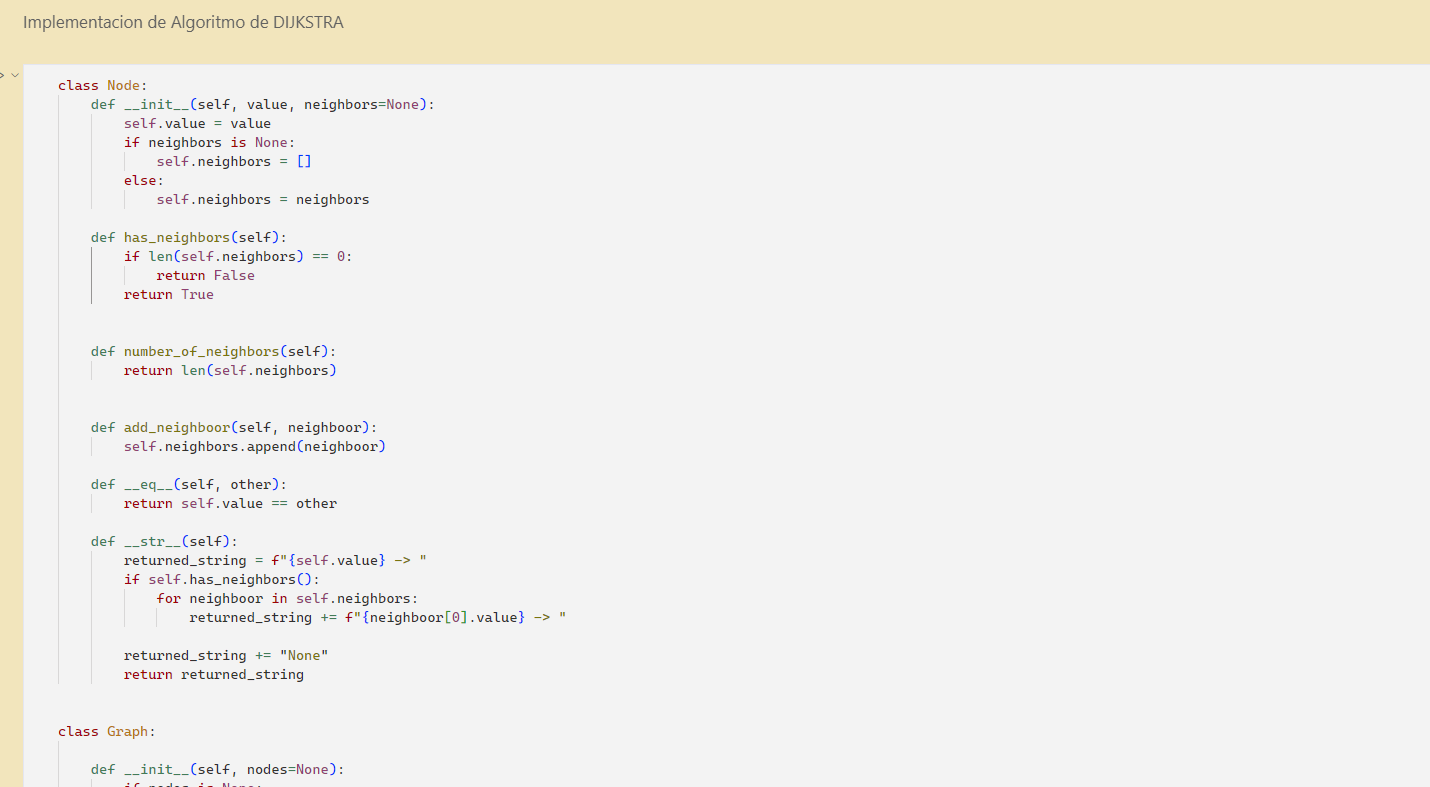




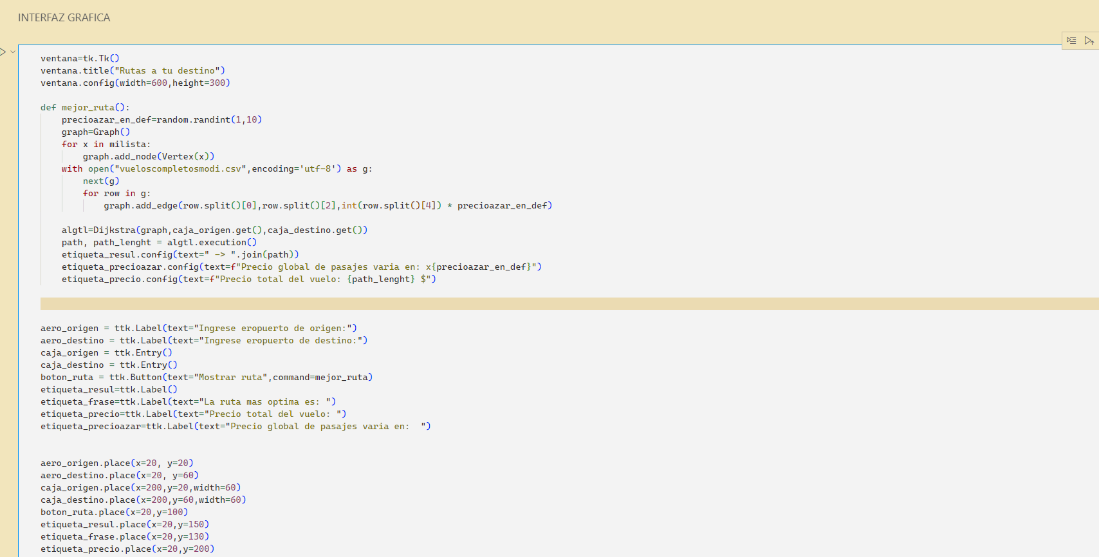
**Dibujo del grafo con los datos ingresados:**



**Implementación del algoritmo de DIJKSTRA:**



**Implementación de interfaz gráfica y pedido de datos al usuario:**

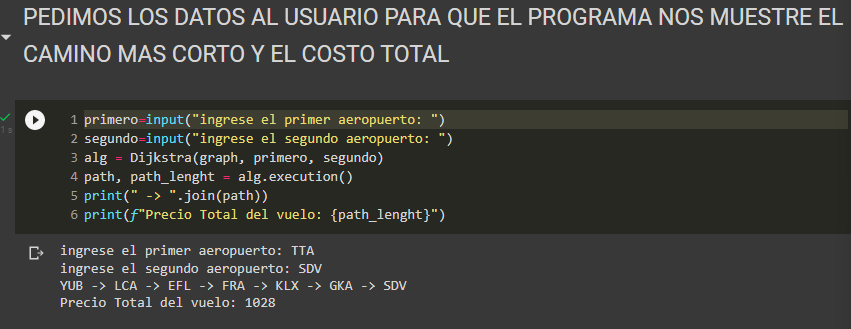


* **Mantenimiento o puesta en producción:**

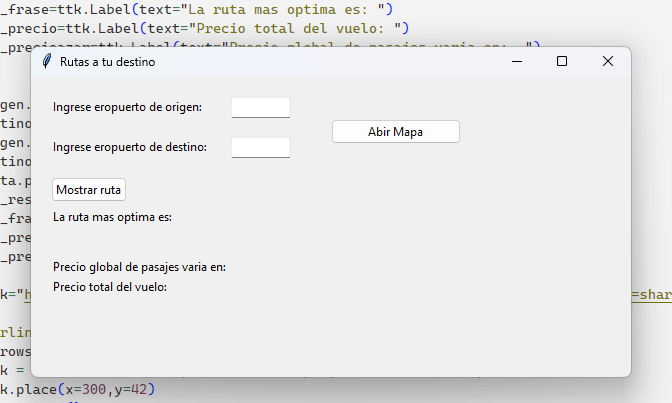
**Ejecución del código para la creación del grafo:**

Diagrama

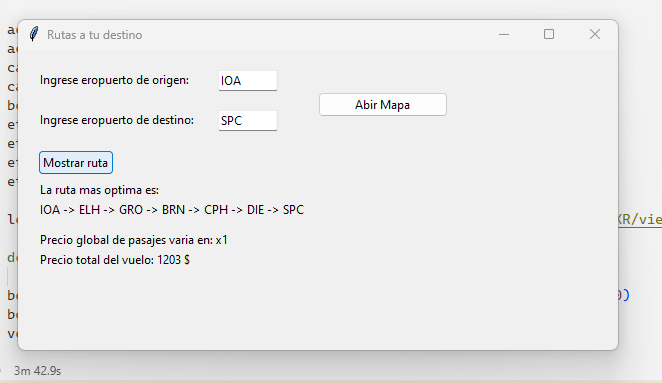
Descripción generada automáticamente  
**Ejecución del código DIJKSTRA pidiendo los datos al usuario:**



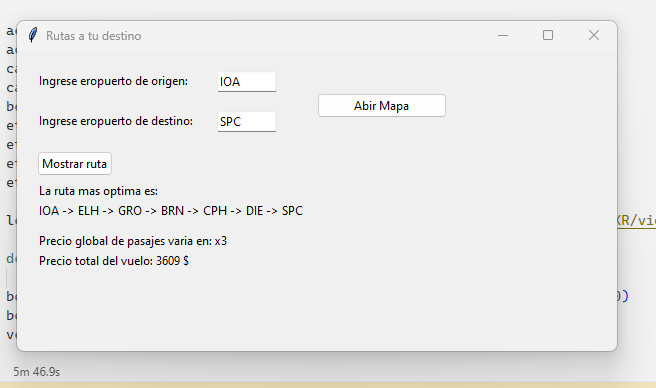
**Ejecución del código completo para la visualización de la interfaz gráfica:**



El usuario ingresa datos y le muestra la ruta con el mejor precio:



El precio global de los pasajes cambia cada vez que buscas una ruta:



1. **Conclusiones:**

Como resultado de lo elaborado, se reconoce de gran importancia de un algoritmo que nos permita encontrar rutas eficaces para una cantidad de datos muy grande. En nuestro caso se usó el algoritmo Dijkstra para el grafo ponderado donde se usó como nodo los códigos IATA de aeropuertos alrededor del mundo y las aristas ponderadas como el precio de un nodo a otro.

En cuanto al desarrollo del algoritmo, se logró desarrollar con una menor dificultad debido a lo aprendido en clase y más aún con la gráfica del grafo ponderado que nos permitía tener un mapeo con nodos dirigidos.

En conclusión, gracias a lo mencionado anteriormente se logró tener un sistema que satisface los requerimientos del cliente y funciona correctamente, siendo capaces además de ser adaptable a cualquier entorno visual y escalable.

1. **Bibliografía:**

Cabrera, R. (2021, 24 septiembre). *¿Qué es mejor un vuelo directo o con escalas? ✈️*. Skyscanner Español. <https://www.espanol.skyscanner.com/noticias/que-es-mejor-un-vuelo-directo-o-con-escalas>

Lovro (s. f.). *Binary Search*. Recuperado 7 de noviembre de 2022, de <https://www.topcoder.com/thrive/articles/Binary+Search>

Sánchez, M. (2019, 22 octubre). *Códigos IATA y OACI de los aeródromos, conceptos y diferencias*. AERTEC. <https://aertecsolutions.com/2018/09/10/codigos-iata-y-oaci-de-los-aerodromos-conceptos-y-diferencias/>