



MACC

Matemáticas Aplicadas y
Ciencias de la Computación

Escala más Óptima.

Teoría de Grafos.

Andrés Felipe Florián Quitián y Sara Gallego Rivera

Noviembre 2019



Contenido

- 1 Introducción.
- 2 Caracterización de la problemática.
- 3 Marco Conceptual.
- 4 Implementación.
- 5 Bibliografía.

Introducción.

En este proyecto se realizó una propuesta para determinar las escalas óptimas a la hora de realizar un vuelo. Se pretende encontrar una ruta alternativa al vuelo directo que tenga la menor duración de tiempo.



Caracterización de la problemática.

El proyecto proporciona rutas alternativas para desplazarse de una ciudad a otra, con escalas de tiempo menores o iguales a siete horas. Este proyecto se desarrolló con el fin de dar una opción de ruta para aviones pequeños que tengan poca capacidad de combustible o personas que por sus condiciones físicas o psicológicas no puedan permanecer gran cantidad de tiempo en un avión.

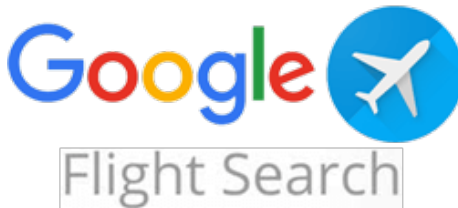


Marco Conceptual.

- **Grafo ponderado:** Un grafo ponderado o grafo con peso es un grafo con etiquetas numéricas en las aristas denominadas pesos en las aristas.
- **Algoritmo de Dijkstra:** Algoritmo que proporciona la distancia mínima entre un par de vértices u, v de un grafo o digrafo ponderado.

Datos.

Los tiempos de vuelo usados en el proyecto fueron tomados de google flights. El cálculo de tiempo está basado en una suposición de que la velocidad media de vuelo del avión es de 500 millas por hora, que son 805 kilómetros por hora. El tiempo de vuelo incluye 30 minutos que es por lo general lo que un avión tarda en despegar y aterrizar.



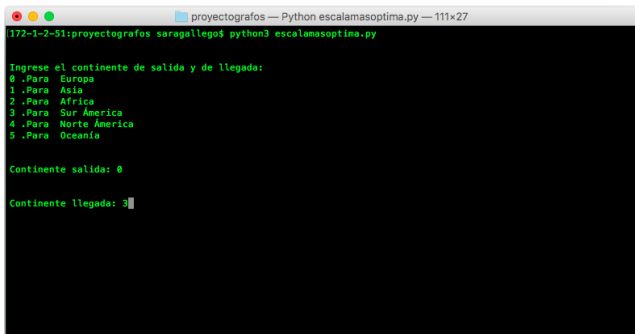
Implementación computacional.

La implementación se llevó a cabo en Python, en esta se usaron librerías como matplotlib y networkx. Para hallar las escalas óptimas de un vuelo se usó el algoritmo de Dijkstra (networkx), para ello se establece una ciudad de salida u y una ciudad de destino v . Al realizar el algoritmo se obtiene un u, v -camino mínimo, este es el que determina cuál es la escala más óptima.



Presentación del modelo y de la solución planteada.

Primero el programa pide el continente de salida y de llegada.



```
projectografos — Python escalamasoptima.py — 111x27
172-1-2-51:projectografos saragallego$ python3 escalamasoptima.py

Ingrese el continente de salida y de llegada:
0 .Para Europa
1 .Para Asia
2 .Para Africa
3 .Para Sur America
4 .Para Norte America
5 .Para Oceanía

Continente salida: 0

Continente llegada: 3
```



Luego pide los países y ciudades de salida.

```
proyectografos — Python escalamasoptima.py — 111x27
Países de salida:
0 .Para Italia
1 .Para Francia
2 .Para España
3 .Para Alemania
4 .Para Polonia
5 .Para Inglaterra
6 .Para Irlanda
7 .Para Portugal
8 .Para Grecia
9 .Para Rumania
10 .Para Austria
11 .Para Finlandia
12 .Para Suecia
13 .Para Países Bajos

Ingrese el país de salida: 0

Ciudades de salida:
0 .Para Roma
1 .Para Milán
2 .Para Venecia

Ingrese la ciudad de salida: 1
```

Posteriormente pide los países y ciudades de destino.

```
proyectografos — Python escalamasoptima.py — 111x27

Ingrese la ciudad de salida: 1

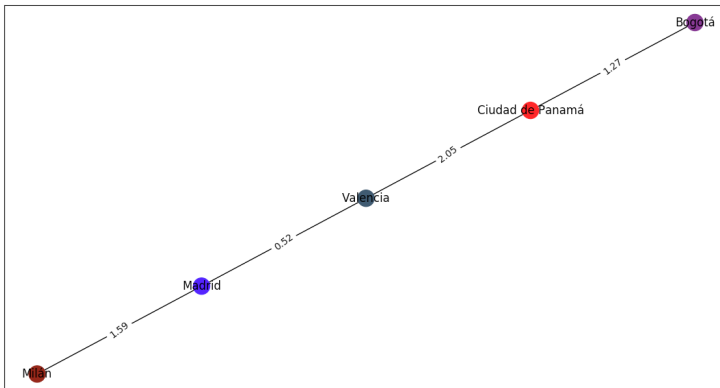
Países de destino:
0 .Para Colombia
1 .Para Ecuador
2 .Para Perú
3 .Para Brasil
4 .Para Chile
5 .Para Argentina
6 .Para Uruguay

Ingrese el pais de destino: 0

Ciudades de destino:
0 .Para Bogotá
1 .Para Medellín
2 .Para Cartagena

Ingrese la ciudad de destino:0
```

Por último, proporciona el camino mínimo entre ambas ciudades.



Bibliografía



Google.

Google flights.



Douglas B. West.

Introduction to graph theory.

Pearson, 2005.



Universidad del
Rosario