***Tarea Integradora3***

Juan Manuel Velosa Valencia

A00376621

Sebastián Zapata Valencia

A00371770

Gabriel Kremer

A00371578

Santiago de Cali, Colombia

**Etapa 1. Contexto Problemático.**

Los estudiantes de ICESI han notado que últimamente el sistema de rutas aéreas que están tomando las diferentes aerolíneas no son las más eficientes, tienden a demorarse más en llegar a su destino o más de lo habitual. Por ende, deciden desarrollar un software que permita al piloto ingresar un punto de partida para después ingresar su destino, y que como resultado el sistema calcule la ruta más eficiente de llegar al destino deseado.

*Definición del Problema:*

Las aerolíneas requieren de un software que les permita identificar la ruta más eficiente con respecto a la distancia de un vuelo en un viaje.

**Desarrollo de la solución**

Para sobrellevar el problema anteriormente mencionado se seleccionó el Método de la Ingeniería para desarrollar una solución, mediante un enfoque sistemático y acorde a la situación problemática planteada.

Para esto, se implementó la descripción del Método de la Ingeniería del libro “Introduction to Engineering” de Paul Wright, se define el siguiente diagrama de flujo, para seguir los pasos en el desarrollo de la solución.

Diagrama

Descripción generada automáticamente

**Etapa 2. Marco teórico.**

Con el objetivo de afrontar el problema con la mayor claridad posible, se buscarán las definiciones relacionadas con el problema previamente definido.

*Definiciones:*

*Aerolínea:* es una empresa que se dedica al transporte de pasajeros o carga.

*Vuelo directo:* un vuelo directo es aquel que te llevará desde el aeropuerto más cercano hasta el aeropuerto de tu destino final, sin parar en ningún otro aeropuerto en el trayecto.

*Vuelo a escala:* un vuelo con escalas te llevará desde el aeropuerto más cercano a un punto intermedio o de conexión donde tendrás que descender del primer avión, pasar por los filtros de seguridad o migración necesarios y abordar otro avión que te llevará ya sea a tu destino final o a otro(s) aeropuerto(s) antes de hacerlo.

*Grafo:* es un tipo abstracto de datos (TAD), que consiste en un conjunto de nodos (también llamados vértices) y un conjunto de arcos (aristas) que establecen relaciones entre los nodos. El concepto de grafo TAD desciende directamente del concepto matemático de grafo.

*Recorrido:* una aeronave no vuela en línea recta; se mueve de un lugar a otro. En distancias más grandes, dicha ruta en forma poligonal casi se ajusta a la línea directa. La razón es simple y lógica: cuanto más corta es la distancia, menor es la cantidad de combustible consumido.

Fuentes:

[*https://es.wikipedia.org/wiki/Wikipedia*](https://es.wikipedia.org/wiki/Wikipedia)

[*https://latam.kaspersky.com/blog/*](https://latam.kaspersky.com/blog/%20)

[*https://concepto.de/*](https://concepto.de/%20) [*https://techlib.net/index.html*](https://techlib.net/index.html)

**Etapa 3. Búsqueda de soluciones creativas.**

Se realiza la búsqueda de diferentes soluciones que puedan resolver el problema de la ruta aérea ms eficiente, el cual es identificar el menor recorrido posible en un viaje de un punto de partida a otro, teniendo en cuenta que el viaje debe tener un punto de partida como tal. Para plantear las ideas que puedan solucionar el problema anteriormente mencionado, se optó por realizar una **lluvia de ideas**, para así poder generar una cantidad considerable de opciones, y en donde se plantearon las siguientes:

***Alternativa 1.*** *Búsqueda del recorrido mínimo de un determinado viaje por medio del algoritmo Dijkstra usando grafos.*

Este algoritmo es parecido al de Floyd Warshall, ya que también busca la menor distancia entre dos vértices, con la diferencia de que el algoritmo Dijsktra no usa matrices para encontrar el camino mínimo. Este algoritmo lo hace por medio de colas ordenadas, de menor a mayor costo de aristas y un arreglo de distancias, donde cada vértice tiene su propio valor de distancia mínima. Este algoritmo tiene una complejidad temporal de O(𝑉2) o O(E log V), donde E es el número de aristas y V el número de vértices. Además, por medio de este método se usan grafos dirigidos, en donde cada relación representa el país de origen y el país de destino de cada viaje.

***Alternativa 2.*** *Búsqueda del recorrido mínimo de un determinado viaje por medio del algoritmo Floyd Warshall usando grafos.*

Este algoritmo es uno de los mejores para realizar la búsqueda de valores mínimos, ya que para este se implementa una matriz en donde la intersección de cada uno de los vértices contiene el camino más eficiente. Este algoritmo tiene una complejidad temporal de Θ(𝑉3) donde V es el número de vértices que contiene el grafo. Además, al utilizar grafos, se pueden representar los viajes posibles entre cada punto de partido y destino con la implementación de grafos dirigidos, donde se identifica el punto de partida y de llegada del viaje.

***Alternativa 3.*** *Búsqueda del recorrido mínimo de un determinado viaje por medio de una Stack de rutas .*

En esta solución se propone el uso de la estructura de datos llamada Stack. Cada ruta tendrá una stack por cada viaje en donde se almacenarán todas estas. Al momento de comparar los las rutas, se hará el uso de una variable global que tendrá tramo más corto del viaje, pero que al iniciar, tomará el primer valor proveniente de la stack. Al obtener el primer valor, la stack empezará a sacar cada uno de sus elementos para que se compare con la variable global y se van a asignar a esta solo si su distancia es menor. Al finalizar la búsqueda de la mínima distancia del viaje.

***Alternativa 4.*** *Búsqueda del recorrido mínimo de un determinado viaje por medio del algoritmo BFS*

Es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo. Su funcionamiento consiste en ir expandiendo cada uno de los nodos que va localizando, de forma recurrente (desde el nodo padre hacia el nodo hijo). Cuando ya no quedan más nodos que visitar en dicho camino, regresa al nodo predecesor, de modo que repite el mismo proceso con cada uno de los vecinos del nodo. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

**Etapa 4. Transición de las Ideas a los Diseños Preliminares.**

La alternativa 2 y 3 son descartadas, porque aunque encuentren una solución al problema, estas cuentan con una complejidad temporal y espacial muy elevada, ya que necesitan de procesos poco eficientes, y además, para poder resolver con la problemática necesitan de un mayor espacio dentro de la memoria, lo que las hace nada eficientes, es por todo esto que quedan eliminadas dentro de las opciones viables.

Luego de descartar las opciones no factibles para desarrollar el problema, se realizará una amplia revisión de las alternativas restantes, para así definir una de ellas como la solución pertinente al problema

***Alternativa 1.*** *Búsqueda del recorrido mínimo de un determinado viaje por medio del algoritmo Dijkstra usando grafos.*

* Este algoritmo puede tomar una complejidad temporal de O(𝑉2) o O(E log V). La primera complejidad temporal se genera con una implementación normal del algoritmo. Con la segunda complejidad temporal, la cual es más eficiente, se requiere la implementación de un binary heap, lo que aumenta el nivel de complejidad de la implementación del algoritmo.
* Su implementación requiere el uso de colas y arreglos con un tamaño V, lo cual aumenta la complejidad espacial del sistema.
* Esta solución informa el recorrido a seguir para el viaje con un costo mínimo.

***Alternativa 4.*** *Búsqueda del recorrido mínimo de un determinado viaje por medio del algoritmo BFS*

* En este algoritmo se propone recorrer los valores por niveles introducidos logrando así verificarlos y recorrer los valores de cada nodo y sus respectivos valores adyacentes.
* Cuando ya no quedan más nodos que visitar en dicho camino, regresa al nodo predecesor, de modo que repite el mismo proceso con cada uno de los vecinos del nodo
* Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

**Etapa 5. Evaluación y Selección de la Mejor Solución.**

*Criterios:*

Se van a definir criterios que permitirán evaluar con facilidad las alternativas propuestas para el desarrollo del problema previamente definido. Esto ayudará a definir las características que tienen prioridad al momento de desarrollar una solución para el problema. Cada característica será evaluada por medio de un valor, en donde el mayor valor significa que es muy importante para la solución y el último significa que es el que menos se espera para el desarrollo de la solución.

***Criterio A. Eficiencia.*** Se busca una solución que tenga una complejidad temporal reducida. La complejidad temporal puede ser: - [5] Constante.

* [4] Logarítmica.
* [3] Líneal
* [2] Cuadrática
* [1] Cúbica

***Criterio B. Claridad.*** Se busca que la solución sea lo más intuitiva y clara posible para el usuario. La solución puede ser:

* [3] Intuitiva y clara. - [2] Poco clara - [1] Ambigua.

***Criterio C. Optimización.*** Se busca que la solución sea lo más óptima posible, ocupando el menor espacio posible en memoria. La solución puede ser:

* [3] Óptima, ocupa el menor espacio posible en la memoria.
* [2] Regular, ocupa un espacio prudente en memoria, sin ser óptimo pero tampoco pésimo.
* [1] Pésima, ocupa mucho espacio en la memoria.

*Evaluación:*

Al evaluar las alternativas sobre los tres criterios previamente definidos, se obtuvo la siguiente tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Criterio A | Criterio B | Criterio C | Total |
| Alternativa 1 | 3 | 3 | 2 | 8 |
| Alternativa 2 | 3 | 3 | 3 | 9 |

*Selección:*

A partir de los resultados obtenidos en la evaluación, podemos concluir que la alternativa 1 (Búsqueda del costo mínimo de un determinado viaje por medio del algoritmo Dijkstra usando grafos) es la mejor solución, ya que obtuvo el mayor puntaje cuando fue evaluado con los criterios previamente definidos.