

## **MÉTODO DE INGENIERÍA**

### **FASE 1: IDENTIFICACIÓN DEL PROBLEMA**

- Planificación y gestión ineficientes de rutas: La situación actual carece de un sistema efectivo para planificar y gestionar las rutas de transporte a nivel nacional. Esto conlleva dificultades para optimizar las rutas, calcular los tiempos de viaje y tomar decisiones informadas para el transporte nacional.
- Falta de un modelo de datos adecuado: Existe la necesidad de diseñar y desarrollar un modelo de datos apropiado que pueda representar las ciudades. Es posible que el sistema actual no tenga un enfoque estructurado para almacenar y organizar los datos, lo cual puede dificultar el acceso y recuperación eficiente de la información.
- Ausencia de algoritmos de optimización de rutas: resulta difícil encontrar las rutas más cortas entre ciudades específicas y determinar las conexiones más eficientes entre todas las ciudades. Esto conduce a una planificación poco óptima y posibles ineficiencias en el transporte nacional.

### **FASE 2: RECOPIACIÓN DE LA INFORMACIÓN NECESARIA**

- Estructuras de datos: son formas de organizar y almacenar datos en una computadora de manera eficiente para que puedan ser manipulados y recuperados fácilmente. Algunas de las estructuras de datos que se pueden utilizar en este problema son arreglos, listas enlazadas, colas, pilas, árboles o grafos.
- Grafos: estructura de datos compuesta por un conjunto de nodos (vértices) que están conectados entre sí mediante enlaces (aristas). Los grafos se utilizan para representar relaciones y conexiones entre diferentes elementos, donde cada nodo puede estar conectado a uno o varios nodos mediante aristas. Los grafos son ampliamente utilizados en diversas áreas, como redes informáticas, algoritmos de búsqueda y optimización, análisis de redes sociales y representación de sistemas complejos.
- Árboles: estructura de datos jerárquica en la que los elementos, llamados nodos, están organizados de forma que exista un nodo principal, llamado raíz, y los demás nodos se distribuyen en niveles inferiores de manera ordenada. Cada nodo puede tener cero o más nodos hijos, que se ramifican a partir de él. Los árboles se utilizan ampliamente en informática para representar estructuras de datos como árboles de búsqueda, estructuras de directorios en sistemas de archivos y representaciones de jerarquías en algoritmos y bases de datos.

- BFS (Breadth-First Search) es un algoritmo de búsqueda en grafos que recorre o explora un grafo de manera nivel por nivel, comenzando desde un nodo inicial y visitando todos sus vecinos antes de pasar a los nodos vecinos del siguiente nivel. En otras palabras, el BFS explora primero todos los nodos directamente conectados al nodo inicial, luego los nodos que están a un nivel de distancia, y así sucesivamente. Es utilizado para encontrar la ruta más corta entre dos nodos en un grafo no ponderado y también es útil para la exploración exhaustiva de un grafo.
- DFS (Depth-First Search) es un algoritmo de búsqueda en grafos que explora o recorre un grafo siguiendo una ruta específica, profundizando tanto como sea posible antes de retroceder. Comienza en un nodo inicial y visita uno de sus vecinos no visitados. A continuación, se adentra en ese vecino y continúa explorando hasta que ya no haya nodos no visitados en esa rama. Luego retrocede al nodo anterior y continúa explorando otras ramas no visitadas. El DFS se utiliza para realizar búsquedas exhaustivas en grafos y se puede implementar tanto de manera recursiva como mediante el uso de una pila (stack).
- Dijkstra es un algoritmo utilizado para encontrar el camino más corto entre un nodo de origen y todos los demás nodos en un grafo ponderado. Este algoritmo asigna a cada nodo un valor provisional de distancia desde el nodo de origen y luego actualiza iterativamente estos valores a medida que encuentra rutas más cortas. El algoritmo Dijkstra utiliza una estrategia de exploración por niveles y selecciona en cada iteración el nodo con la distancia provisional más pequeña como siguiente nodo a evaluar. A medida que avanza, va calculando y actualizando las distancias mínimas hasta alcanzar todos los nodos alcanzables desde el nodo de origen. Al finalizar, se obtiene la distancia más corta desde el nodo de origen a todos los demás nodos del grafo.
- El algoritmo de Floyd-Warshall es un algoritmo utilizado para encontrar los caminos más cortos entre todos los pares de nodos en un grafo ponderado, ya sea dirigido o no dirigido. A diferencia de otros algoritmos que encuentran el camino más corto entre un par de nodos específicos, el algoritmo de Floyd-Warshall encuentra la distancia más corta entre todos los pares de nodos posibles en el grafo. Utiliza una matriz para almacenar y actualizar las

distancias a medida que se tira sobre los nodos del grafo. Al finalizar, la matriz contiene las distancias más cortas entre todos los pares de nodos.

- El algoritmo de Prim es utilizado para encontrar el árbol de expansión mínima en un grafo ponderado no dirigido. Este algoritmo comienza con un nodo inicial y de forma iterativa agrega aristas que conectan el árbol existente con un nuevo nodo, siempre seleccionando la arista de menor peso posible. De esta manera, el algoritmo construye un árbol de expansión que abarca todos los nodos del grafo y tiene el peso total mínimo. El algoritmo de Prim garantiza que el árbol de expansión mínimo sea encontrado, lo que significa que contiene todas las aristas necesarias para conectar todos los nodos y su suma de pesos es la mínima posible.
- El algoritmo de Kruskal es utilizado para encontrar el árbol de expansión mínima en un grafo ponderado no dirigido. Este algoritmo selecciona las aristas del grafo en orden ascendente según su peso y las agrega al árbol de expansión mínimo, siempre y cuando no formen un ciclo. De esta manera, se construye un árbol que conecta todos los nodos del grafo con el menor peso total posible. El algoritmo de Kruskal es eficiente y garantiza la obtención del árbol de expansión mínimo.

#### **Referencias:**

[https://www.utm.mx/~mgarcia/ED4\(Grafos\).pdf](https://www.utm.mx/~mgarcia/ED4(Grafos).pdf)  
[https://www.unipamplona.edu.co/unipamplona/portallG/home\\_23/recursos/geral/11072012/grafos3.pdf](https://www.unipamplona.edu.co/unipamplona/portallG/home_23/recursos/geral/11072012/grafos3.pdf)  
[https://www6.uniovi.es/usr/cesar/Uned/EDA/Apuntes/TAD\\_apUM\\_07.pdf](https://www6.uniovi.es/usr/cesar/Uned/EDA/Apuntes/TAD_apUM_07.pdf)

### **FASE 3: BÚSQUEDA DE SOLUCIONES CREATIVAS**

Implementar estructuras de grafos, y, para abordar el problema de planificación de rutas, se podrían utilizar algoritmos como Dijkstra y Kruskal:

**Algoritmo de Dijkstra:** Este algoritmo se puede implementar para encontrar las rutas más cortas entre dos ciudades específicas en el grafo ponderado. Proporcionará información sobre las mejores rutas para el transporte nacional, considerando la distancia o el tiempo de viaje como criterios de optimización.

Utilizar el algoritmo de Floyd-Warshall para encontrar las distancias más cortas entre todas las parejas de ciudades en el grafo ponderado. Esto nos ayudará a obtener la información completa sobre las distancias y tiempos de viaje entre todas las ciudades del país incluidas en el programa.

**Algoritmo de Kruskal:** Este algoritmo se puede utilizar para encontrar el árbol de

recubrimiento mínimo que conecte todas las ciudades de manera eficiente. Ayudará a identificar las conexiones más importantes entre las ciudades y brindará información valiosa sobre las rutas óptimas para el transporte nacional.

Usar el algoritmo de búsqueda en profundidad (DFS) con el fin de encontrar rutas que permitan visitar la mayor cantidad de ciudades, sin importar la longitud total del recorrido. Esta estrategia podría ser de gran utilidad para aquellos viajeros que buscan explorar diferentes lugares y descubrir nuevas ciudades en su camino. El objetivo de utilizar el algoritmo DFS es maximizar la experiencia del usuario durante su recorrido, permitiéndole visitar la mayor cantidad de destinos posibles. Con esta técnica de búsqueda, se espera encontrar soluciones eficientes y creativas para el sistema de planificación y gestión de rutas de transporte a nivel nacional.

Estos algoritmos, junto con las estructuras de datos adecuadas, permitirán mejorar la planificación y gestión de las rutas de transporte a nivel nacional, proporcionando información precisa sobre las mejores rutas y conexiones entre las ciudades.

#### **FASE 4: TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS PRELIMINARES, EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN**

1. Utilizar el algoritmo de Dijkstra para encontrar las rutas más cortas entre dos ciudades específicas. Esto permitirá calcular los tiempos de viaje y determinar las conexiones más eficientes entre las ciudades seleccionadas.
2. Aplicar el algoritmo de Kruskal para encontrar un árbol de recubrimiento mínimo que conecte todas las ciudades de manera eficiente. Esto brindará información sobre las mejores rutas para el transporte nacional, garantizando una cobertura óptima con un peso total mínimo.
3. Utilizar el algoritmo de Floyd-Warshall para encontrar las distancias más cortas entre todas las parejas de ciudades en el grafo ponderado. Esto nos ayudará a obtener la información completa sobre las distancias y tiempos de viaje entre todas las ciudades del país incluidas en el programa.