

MÉTODO DE LA INGENIERÍA

Fase 1. Identificación del problema.

Identificación de necesidades y síntomas:

- Identificar el camino adecuado que une a un par de vértices que requiera atravesar a la arista especificada.
- Definir el grafo como no dirigido.
- Encontrar el número de pares adecuados que unen a los vértices con la ruta única que existe para ello.

Definición del problema:

- La solución debe encontrar la cuenta total de pares de vértices cuyo camino que los une, debe involucrar a una arista específica.

Fase 2. Recopilación de la información necesaria.

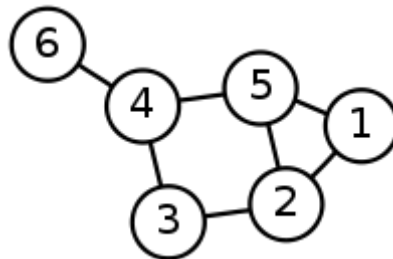
Fuentes:

<https://matematicasies.com/Grafos-Definicion>

<https://xcodigoinformatico.blogspot.com/>

<https://www.ecured.cu/>

Un grafo es una pareja de conjuntos $G = (V, A)$, donde V es el conjunto de vértices, y A es el conjunto de aristas conjunto de objetos llamados nodos o vértices.



En la imagen anterior vemos un grafo con 6 vértices (numerados desde el 1 hasta el 6).

También vemos algunas aristas que unen (o conectan) parejas de vértices.

Por ejemplo: el vértice 1 está conectado con los vértices 2 y 5, pero no está conectado con el resto de vértices.

En los grafos de tipo «simple», no puede haber más de una arista que una dos vértices cualesquiera.

Por otro lado, es posible encontrar otro tipo de grafos que pueden ser:

- Grafo dirigido: se refiere a los grafos que presentan sus grafos orientados por flechas.
- Grafo no dirigido: dicese del tipo de grafo que no tienen flecha, o sea no están orientados.

- **Multigrafo:** refiere al tipo de grafo que puede aceptar entre sus dos vértices a más de una arista, las cuales se nombran como lazos o aristas múltiples. De estos tipos de grafos se destacan los grafos simples, los cuales se les suele llamar como grafo general.
- **Grafo bipartido:** se refiere al tipo de grafo que con sus vértices se pueden crear dos conjuntos disjuntos, donde no habrá adyacencias entre los vértices que conforman el mismo conjunto.

Pero, para solución de este problema en particular, es importante tener en cuenta que se trabajará con un grafo no dirigido.

Del mismo modo, es posible que se pueda interactuar con los vértices y aristas del grafo y con los caminos que une a cada vértice con distintos algoritmos, cada uno con un enfoque distinto, entre los cuales se puede encontrar:

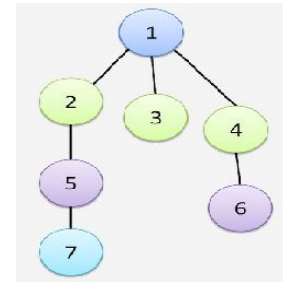
- **Algoritmo de Kruskal:** el algoritmo de kruskal es un algoritmo de recubrimiento mínimo conexo ponderado, o sea que va unir todos los nodos formando un árbol, tomando la aristas que tengan un peso que siempre sea menor.
- **Algoritmo de Dijkstra:** el algoritmo de dijkstra determina la ruta más corta desde un nodo origen hacia los demás nodos para ello es requerido como entrada un grafo cuyas aristas posean pesos.
- **Algoritmo de Floyd-Warshall:** es un algoritmo de análisis sobre grafos que permite encontrar el camino mínimo en grafos dirigidos ponderados. El algoritmo encuentra el camino entre todos los pares de vértices en una única ejecución, constituyendo un ejemplo de programación dinámica.
- **Algoritmo de Prim:** el algoritmo encuentra un subconjunto de aristas que forman un árbol con todos los vértices, donde el peso total de todas las aristas en el árbol es el mínimo posible. Si el grafo no es conexo, entonces el algoritmo encontrará el árbol recubridor mínimo para uno de los componentes conexos que forman dicho grafo no conexo.

Fase 3. Búsqueda de soluciones creativas.

- Utilizar los recorrido por anchura y profundidad para identificar qué caminos compone cada arista
- Utilizar el algoritmo dijkstra para hallar el camino entre cada vértice y saber en cuantos la arista hace parte del camino
- usar el algoritmo prim para sacar el árbol que una todos los vértices y ver la arista como forma camino
- Usar el algoritmo Kruskal para hallar el bosque del grafo y ver cada arista cuántos caminos conforma entre vértices
- Con el algoritmo floyd-warshall conseguir la matriz de adyacencia y ver la arista en cuántos caminos hace parte.

Fase 4. Transición de las Ideas a los Diseños Preliminares.

Son algoritmos que nos permiten recorrer todos los vértices del grafo exactamente una vez y en un orden específico dando como resultado un árbol cuya raíz es el nodo inicial y sus ramas son todos los caminos que este tiene hacia los otros nodos.



El algoritmo Kruskal nos es muy útil debido a que encuentra el árbol recubridor mínimo en un grafo conexo y ponderado, buscando el conjunto aristas que, siendo del menor peso posible, conectan todos los vértices del grafo. Si el grafo no es conexo crea el bosque mínimo.

Criterios. A partir de las alternativas planteadas y la definición del problema, se plantean los siguientes criterios para escoger la solución más apropiada:

- **Criterio 1:** Facilidad de implementación. Se refiere a la posibilidad que hay de terminar la implementación a un plazo corto de tiempo. Esto puede ser:
 - [3] Muy probable.
 - [2] Probable

- [1] Improbable.
- **Criterio 2:** Consumo de memoria. Se refiere a la cantidad de almacenamiento que implementa el algoritmo para su debido funcionamiento. Puede ser:
 - [3] Baja.
 - [2] Normal.
 - [1] Excesiva.
- **Criterio 3:** Accesibilidad. Se refiere a la dificultad de acceder a algún dato de la estructura
 - [3] Fácil.
 - [2] Media.
 - [1] Difícil.
- **Criterio 4:** Efectividad. Se refiere a la eficacia con la que los métodos resuelven el problema
 - [3] Alta
 - [2] Media.
 - [1] Baja

Evaluación:

	Criterio 1	Criterio 2	Criterio 3	Criterio 4	Total
<i>Alternativa 1.</i>	3	2	3	3	11
<i>Alternativa 2.</i>	2	2	2	3	9

Selección:

De acuerdo a los criterios planteados y la obtención de los resultados finales, se considera como mejor opción de diseño de solución la Alternativa 1. Teniendo en cuenta que es importante fortalecer los criterios en los que se presente mayor dificultad y, además, implementar la solución con las modificaciones que sean necesarias para lograr el objetivo final, manteniendo la esencia de esa Alternativa.