



## **Laboratorio Dijkstra**

### **Participantes:**

Daniel Gonzalez

Nicolas Algarra

### **Profesor:**

John Corredor

### **Creado**

30/10/2024

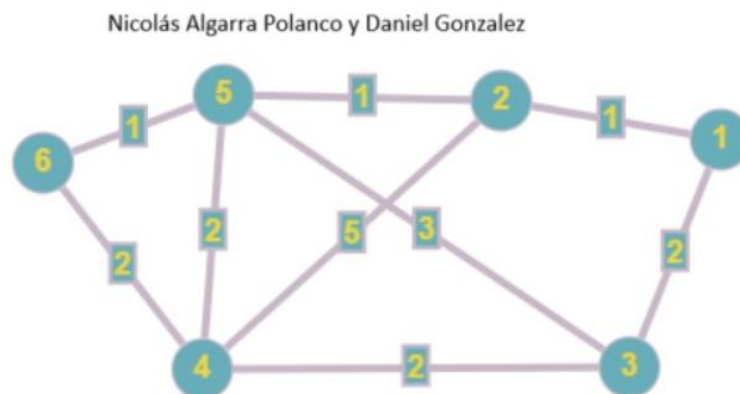
## 1. Imagen del grafo

El primer punto fue graficar el siguiente grafo dada una matriz de adyacencia el cual observamos 6 vértices.

**MATRIZ:**

```
int graph[6][6]={
    {0, 1, 2, 0, 0, 0},
    {1, 0, 0, 5, 1, 0},
    {2, 0, 0, 2, 3, 0},
    {0, 5, 2, 0, 2, 2},
    {0, 1, 3, 2, 0, 1},
    {0, 0, 0, 2, 1, 0}};
```

**GRAFO:**



## 2. Corrección de código dado por el profesor.

- Analizando el código dado notamos que se encontraba un tipo de dato que era inconsistente con las librerías dadas, por lo tanto, agregamos la librería correspondiente a este dato que es <limits>, esto para hacer referencia a un tamaño infinito en el peso de las aristas en la primera iteración
- Además, se creó una constante para poder realizar un cambio de manera fácil para el código según el grafo.

## 3. Agregamos un nuevo grafo en el código.

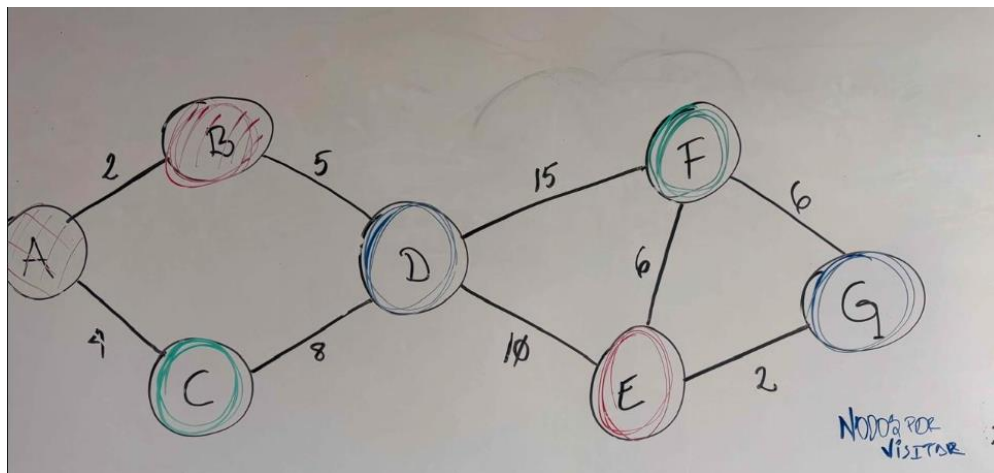
Para este nuevo grafo nos basamos con esta matriz de adyacencia con el fin de poder graficarlo visualmente.

**MATRIX:**

	A	B	C	D	E	F	G
A	0	2	4	0	0	0	0
B	2	0	0	5	0	0	0
C	4	0	0	8	0	0	0
D	0	5	8	0	10	15	0
E	0	0	0	10	0	6	2
F	0	0	0	15	6	0	6
G	0	0	0	0	2	6	0

Posteriormente procedemos a realizar visualmente el grafo por lo cual quedaria de la siguiente manera.

**GRAFO:**



#### 4. Compilación de código.

El siguiente comando es para compilar

g++ \*.cpp -o ejecutable

Este comando es para ejecutar

./ejecutable