**<< Documentacion >>**

Implementacion Dijkstra

Clases y métodos

**Clase Node**

Representa un nodo en el grafo.

**Atributos:**

name: El nombre del nodo (un carácter, como A, B, etc.).

connections: Una lista de conexiones (pares de nodo y peso) hacia otros nodos.

**Constructor:**

**Node(char \_name):** Crea un nodo con un nombre específico.

**Clase Graph**

Maneja la lógica del grafo, como la creación, lectura desde un archivo, y búsqueda de caminos más cortos.

**Atributos:**

nodes: Vector de nodos en el grafo.

num\_nodes: Número de nodos en el grafo.

**Métodos:**

**Graph():** Constructor que inicializa el grafo vacío.

**~Graph():** Destructor que elimina todos los nodos creados dinámicamente.

**readFile(string file\_name):**

Lee una matriz de adyacencia desde un archivo.

Crea los nodos y las conexiones del grafo.

Muestra errores si el archivo no es válido.

**showNodes():**

Muestra en pantalla los nombres de los nodos del grafo.

**askNode():**

Pide al usuario un nodo de destino.

Calcula y muestra el camino más corto desde A al nodo seleccionado.

**selectMinVertex(vector<int>& value, vector<bool>& processed):**

Encuentra el nodo no procesado con la menor distancia calculada.

**dijkstra(int start):**

Aplica el algoritmo de Dijkstra desde un nodo inicial (A).

Devuelve un vector que indica el nodo anterior en el camino más corto hacia cada nodo.

**printPath(vector<int>& prev, int node):**

Muestra el camino más corto desde el nodo inicial al nodo de destino.

**printDistance(vector<int>& prev, int node):**

Calcula y muestra la distancia total del camino más corto al nodo de destino.