

## Pseudocódigo Dijkstra

nombre del  
nodo inicial

Lista de adyacencia

```
void dijkstra ( string inicio, map<string, vector<pair<int, string>>> lista ) {  
    map<string, int> distancia // distancia para llegar a cada  
    // nodo desde el inicial  
    // Inicializar todas las distancias en infinito  
    for (int i=0; i<distancia.size(); i++) {  
        distancia[i][1] = 0  
    }  
  
    // Inicializar distancia del nodo inicial en 0  
    distancia[inicio] = 0  
  
    // Crear cola para ir guardando los nodos y poder obtener el menor  
    Vector<pair<int, string>> cola // costo, nodo  
    cola.push_back (0, inicio)  
  
    // Mientras la cola no este vacía, procesar  
    while (!cola.empty()) {  
  
        // Obtener nodo menor  
        int distActual = cola.pop(menor).first  
        string nodoActual = cola.pop(menor).second  
        cola.delete(menor)  
  
        // Explorar vecinos  
        for (int i=0; i<lista[nodoActual].size(); i++) {  
  
            // Obtener vecino y costo para llegar a él  
            string vecino = lista[nodoActual][i].nodo2  
            int distNueva = distActual + lista[nodoActual][i].costo  
  
            // Si la distancia nueva es menor, actualizar  
            if (distNueva < distancia[vecino]) {  
                distancia[vecino] = distNueva  
                cola.push_back (distNueva, vecino)  
            }  
        }  
    }  
    print (distancia)
```

Nota: Realizamos este pseudocódigo / código con el objetivo de poder pasar los nodos por nombre y no por índice, para que sea más flexible