```
Algoritmo Dijkstra(nodo_inicial, nodo_final):
    distancias = CrearDiccionarioConValorInfinito()   // Inicializar todas las distancias como infinito
    distancias[nodo_inicial] = 0   // Distancia del nodo inicial a sí mismo es 0
    cola_prioridad = CrearColaDePrioridad()
    cola_prioridad.agregar(nodo_inicial, 0)

    padres = CrearDiccionarioVacio()   // Diccionario para almacenar los padres de los nodos en el camino más corto
    padres[nodo_inicial] = NULO   // Nodo inicial no tiene padre

    WHILE cola_prioridad NO está vacía:
        nodo_actual = cola_prioridad.pop()

        IF nodo_actual == nodo_final:
            SALIR DEL WHILE

        FOR arista EN nodo_actual.adyacentes:
            nodo_destino = arista.destino
            peso_arista = arista.peso
            nueva_distancia = distancias[nodo_actual] + peso_arista
            IF nueva_distancia < distancias[nodo_destino]:
                distancias[nodo_destino] = nueva_distancia
                padres[nodo_destino] = nodo_actual
                cola_prioridad.agregar(nodo_destino, nueva_distancia)

    distancia_total = distancias[nodo_final]
    IMPRIMIR "Distancia más corta desde el nodo", nodo_inicial.id, "al nodo", nodo_final.id, ":", distancia_total

    nodo_actual = nodo_final
    camino = [nodo_actual]
    WHILE nodo_actual != nodo_inicial:
        nodo_actual = padres[nodo_actual]
        agregar nodo_actual al inicio de camino
    IMPRIMIR "Camino más corto:"
    FOR nodo EN camino:
        IMPRIMIR nodo.id, "->"
    FIN PARA
FIN
```