



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



# **Tecnológico Nacional de México Campus Culiacán**

## **Ingeniería en Sistemas Computacionales**

**Materia:**

**Inteligencia artificial**

**Algoritmo de Dijkstra**

**Profesor:**

**JOSE MARIO RIOS FELIX**

**Alumno:**

**CAMPOS LOPEZ LUIS FERNANDO**

El algoritmo de Dijkstra, propuesto por el científico holandés Edsger W. Dijkstra en 1956, es un método que permite encontrar la ruta más corta desde un nodo origen hasta todos los demás nodos en un grafo ponderado con pesos no negativos. Se considera uno de los algoritmos más importantes en la teoría de grafos y es ampliamente utilizado en aplicaciones prácticas.

### **Características principales**

- Trabaja sobre un grafo dirigido o no dirigido.
- Los pesos de las aristas deben ser no negativos.
- Encuentra el camino más corto desde el nodo de inicio a todos los demás nodos.
- Complejidad:
  - Usando matriz de adyacencia:  $O(n^2)$ .
  - Usando cola de prioridad (Heap):  $O((V+E) \log V)$ , donde  $V$  son vértices y  $E$  son aristas.

### **Funcionamiento paso a paso**

1. Inicializar
  - La distancia del nodo origen es 0.
  - La distancia del resto de nodos es infinita.
2. Marcar el nodo inicial como visitado.
3. Para cada nodo vecino, actualizar su distancia si se encuentra un camino más corto.
4. Escoger el nodo no visitado con menor distancia.
5. Repetir el proceso hasta haber visitado todos los nodos.

### **Aplicaciones del algoritmo de Dijkstra**

1. GPS y navegación: encontrar la ruta más corta entre ciudades o calles.
2. Redes de computadoras: protocolos de enrutamiento como OSPF (Open Shortest Path First).
3. Inteligencia artificial y videojuegos: personajes que buscan el camino más corto en mapas.
4. Logística y transporte: optimización de rutas de distribución.
5. Planificación de proyectos: encontrar caminos más cortos en grafos de tareas.

## **Ventajas**

- Encuentra siempre la solución óptima en grafos con pesos no negativos.
- Relativamente sencillo de implementar.
- Versátil para diferentes áreas.

## **Desventajas**

- No funciona correctamente si el grafo tiene aristas con peso negativo.
- Puede ser más lento en grafos muy grandes comparado con algoritmos como A\*.
- Requiere estructuras adicionales (colas de prioridad) para ser eficiente en grandes grafos.