

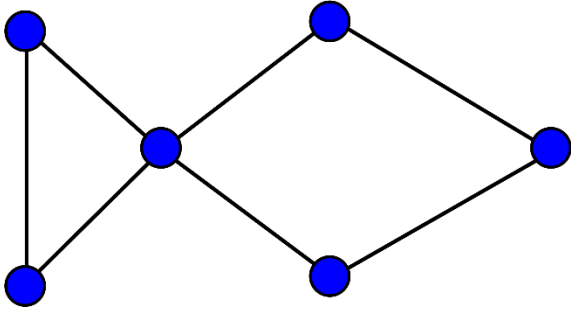
# ALGORITMO DIJKSTRA

INTELIGENCIA ARTIFICIAL 18:00 – 19:00



Flores Soto Gael Eduardo  
N de control: 20170661

# El Algoritmo de Dijkstra: Qué es, Cómo Funciona y Dónde se utiliza



Con él, los navegadores, los robots y los personajes de los videojuegos construyen rutas.

Las aplicaciones de navegación modernas y otros sistemas de apoyo a la toma de decisiones utilizan algoritmos de caminos mínimos para calcular la ruta más eficiente entre un punto de inicio y un destino. Estos algoritmos constituyen una parte esencial de la teoría de grafos y de la optimización en ciencias de la computación.

Dentro de este campo, destacan dos algoritmos ampliamente empleados: **Dijkstra** y **A\***. Ambos permiten encontrar rutas óptimas, pero presentan diferencias significativas en su funcionamiento y en los contextos donde resultan más adecuados.

El algoritmo de **Dijkstra** se basa en la exploración sistemática de nodos a partir de un origen, garantizando siempre la distancia mínima en grafos con pesos no negativos. Por su parte, el algoritmo **A\*** incorpora una función heurística que estima la distancia restante hacia el destino, lo cual le permite enfocar la búsqueda y reducir el espacio explorado, especialmente útil en aplicaciones como videojuegos, robótica o sistemas de navegación en tiempo real.

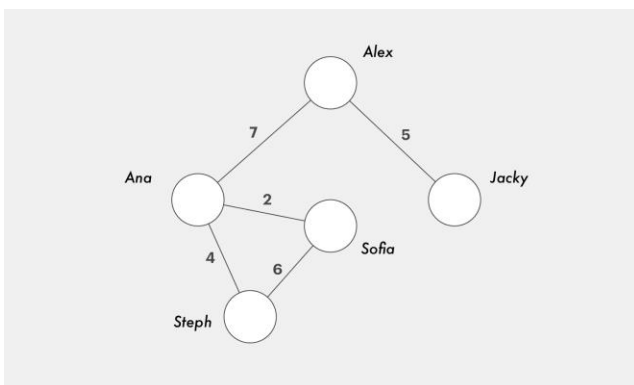
En esta investigación se analizan los fundamentos de ambos algoritmos, sus aplicaciones más relevantes y las diferencias clave que los distinguen.

## El principio: la teoría de los grafos

El algoritmo de Dijkstra es un método para encontrar las rutas más cortas desde un vértice de un grafo hasta todos los demás.

Un grafo es una estructura matemática que consta de vértices (nodos) y aristas (conexiones) entre ellos. Las aristas pueden tener dirección y también pesos: números que indican la fuerza de las conexiones con los vértices.

Los grafos se utilizan para representar objetos, sus interconexiones y las relaciones entre ellos. Por ejemplo, en forma de grafo, se puede representar una red social, donde los vértices son los usuarios, las conexiones de amistad entre ellos son las aristas, y los pesos son, por ejemplo, la frecuencia de intercambio de mensajes.



Pero, por supuesto, el significado práctico de los grafos no se limita a las redes sociales. Por ejemplo, con su ayuda, puedes modelar las conexiones entre:

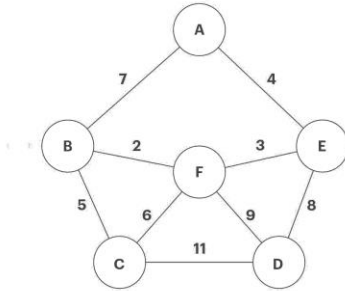
- Páginas web
- Calles y cruces al diseñar una red vial
- Genes, proteínas y moléculas en bioinformática
- Puntos de entrega de mercancías en logística

## Cómo funciona el algoritmo de Dijkstra

La idea del algoritmo de Dijkstra es que podemos encontrar las distancias más pequeñas desde el vértice inicial del grafo hasta todos los demás.

Conociendo estas distancias, se puede construir la ruta más corta entre el punto de inicio y otros puntos.

Supongamos que tenemos varias ciudades conectadas por carreteras. Llamémoslas **A, B, C, D, E, F**. Los números cerca de las aristas son las distancias entre las ciudades en millas.



### Ejemplo de grafo ciudades y carreteras

Al intentar encontrar la ruta más corta, digamos, de A a C. Aunque parece bastante simple: comparamos todos los caminos posibles y elegimos el más corto:  $A \rightarrow B \rightarrow C$ . Una simple enumeración.

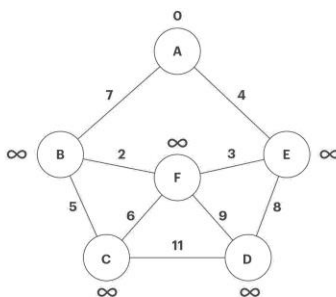
**Las dificultades comienzan cuando hay demasiados datos.** Si hubiera más de 26 ciudades en nuestro mapa, incluso la computadora más potente necesitaría miles de millones de años para calcular todas las opciones.

El algoritmo de Dijkstra funciona de manera diferente: **no enumera “mentalmente” todas las opciones posibles, sino que construye la ruta paso a paso.** En cada paso, el algoritmo elige el vértice menos distante y se mueve hacia él, luego hacia el siguiente, y así sucesivamente hasta llegar al destino.

El significado clave es este: si en cada etapa se toman decisiones óptimas, la decisión final, lo más probable, también será óptima. Los algoritmos que funcionan de acuerdo con este principio se llaman [algoritmos voraces](#).

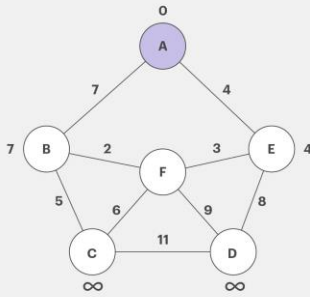
Volviendo al grafo con ciudades. Encontremos con la ayuda del algoritmo de Dijkstra las distancias más cortas de A a B, C, D, E y F.

**Paso 1.** Establezcamos una estimación inicial del camino hasta A para cada vértice. Para la propia A, la estimación es 0, al resto se les asigna infinito, ya que aún no conocemos sus valores.



### Paso 1 ruta más corta en ejemplo de ciudades

Consideremos los vértices adyacentes a A, es decir, aquellos que están conectados a A mediante aristas directamente. Estos son **B** y **E**: sus distancias a A son **7** y **4** respectivamente. Dado que estos valores, obviamente, son menores que el infinito, los actualizaremos en el diagrama. Consideraremos el vértice A como visitado, lo coloreamos y ya no lo consideramos.



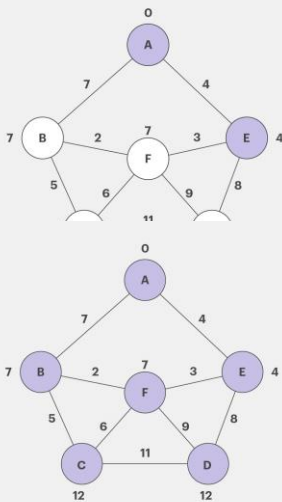
### Paso 2 ruta más corta en ejemplo de ciudades

Ahora pasemos al vértice no visitado con la distancia más pequeña a A. Este es el vértice **E**. Los vértices adyacentes a él no visitados son **F** y **D**. Sus distancias a A serán iguales a la estimación de E (es decir, la distancia de E a A) más los pesos de las aristas de E a estos vértices.

Resulta así:

- Para F:  $4 + 3 = 7$
- Para D:  $4 + 8 = 12$

Las distancias obtenidas son menores que las estimaciones anteriores, por lo que las registraremos cerca de los vértices F y D. Consideraremos el vértice E como visitado. Lo coloreamos.



### Paso 3 ruta más corta en ejemplo de ciudades

Lo siguiente, por analogía: el algoritmo elige los vértices no visitados con la estimación más pequeña y calcula las distancias desde los vértices adyacentes a él hasta A. Y esto continúa hasta que el algoritmo calcula las distancias más cortas hasta A para todos los vértices.

### Paso 4 ruta más corta en ejemplo de ciudades

¡Listo! Ahora podemos construir la ruta más corta de A a cualquier otro vértice. Por ejemplo:

- De A a F: A — E — F
- De A a D: A — E — D
- De A a C: A — B — C

✗

Pero el algoritmo de Dijkstra también tiene limitaciones: en primer lugar, solo funciona con grafos ponderados, es decir, aquellos donde los pesos entre las aristas se conocen de antemano. Y en segundo lugar, estas distancias deben ser no negativas.

## Cómo surgió el algoritmo de Dijkstra

El algoritmo de búsqueda de la ruta más corta fue desarrollado por el científico holandés **Edsger Dijkstra** en 1956. En ese momento, estaba buscando una forma de demostrar las capacidades de la nueva computadora ARMAC y estaba buscando un problema que ARMAC pudiera resolver y que fuera comprensible para las personas que no estaban familiarizadas con las computadoras.

Dijkstra tomó el problema de encontrar la ruta más corta y desarrolló un algoritmo para resolverlo. Basándose en el algoritmo, desarrolló un programa para construir rutas entre ciudades en un mapa de transporte de los Países Bajos.

Más tarde, Dijkstra contó:

*“Una vez, mi novia y yo bebíamos café en la terraza de un café en Ámsterdam. Allí, en unos veinte minutos, desarrollé el algoritmo para encontrar la ruta más corta. Fue publicado tres años después, en 1959.*

*Una de las razones por las que el algoritmo resultó tan hermoso fue que lo creé en mi mente, sin lápiz ni papel. Luego me enteré de que una de las ventajas de este tipo de diseño es que, al hacerlo, intentas evitar al máximo las dificultades.*

*Este algoritmo se convirtió, para mi sorpresa, en una de las piedras angulares de mi fama”.*

**Edsger Dijkstra**

**en una entrevista con Philip L. Fran**