



Inteligencia Artificial

Unidad 1: Representación Avanzada del Conocimiento

TEMA 2: Algoritmos de la IA Clásica

Módulo 2: Algoritmo de Búsqueda Heurística A-Star (A*)

Unidad 1

Representación Avanzada del Conocimiento

TEMA 2: Algoritmos de la IA Clásica

Sesión 5

MÓDULO 2: Algoritmo de Búsqueda Heurística A-Star (A^*)



Contenido

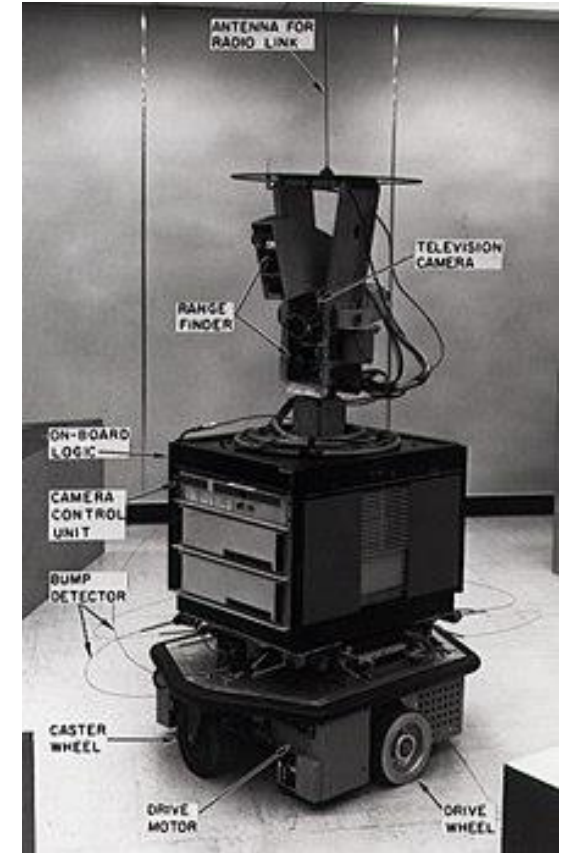
1. Breve historia del Algoritmo A-star
2. Objetivos del Algoritmo A^*
3. ¿Cómo funciona el Algoritmo A^* ?
4. Terminología del Algoritmo A^*
5. Ejemplos del Algoritmo A^*
6. Aplicaciones del Algoritmo A^*



Preguntas

1. Breve historia del Algoritmo de búsqueda A*

- Fue publicado por primera vez en 1968 por Peter Hart, **Nils Nilsson** y **Bertram Raphael**.
- **A *** fue inventado por investigadores que trabajaban en la planificación del camino de Shakey the Robot.
- Inicialmente fue considerado una extensión del algoritmo de Dijkstra, pero resulta ser más rápido que este debido a que utiliza la función heurística.



2. Objetivos del Algoritmo de búsqueda A*

UTILIZADO PARA LA SOLUCION DE PROBLEMAS

Objetivo: lograr la **optimización** y la **completitud**, dos propiedades valiosas de los algoritmos de búsqueda.

Optimización: significa la garantía de encontrar la mejor solución posible.

Completitud: significa que si existe una solución a un problema dado, el algoritmo garantiza encontrarla.

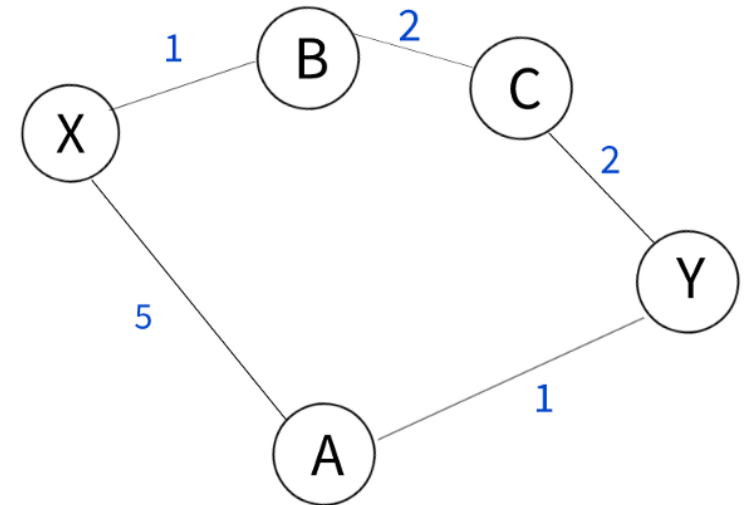
3. ¿Cómo funciona el Algoritmo A-star?

A-Star (también conocida como **A ***) es uno de los **algoritmos de búsqueda** más exitosos utilizado para encontrar la ruta más corta entre **nodos** o **gráficos ponderados**.

Es un algoritmo de **búsqueda informada**, ya que usa:

- Información sobre el costo de la ruta
- **Heurísticas** (funciones o reglas) para encontrar la solución.

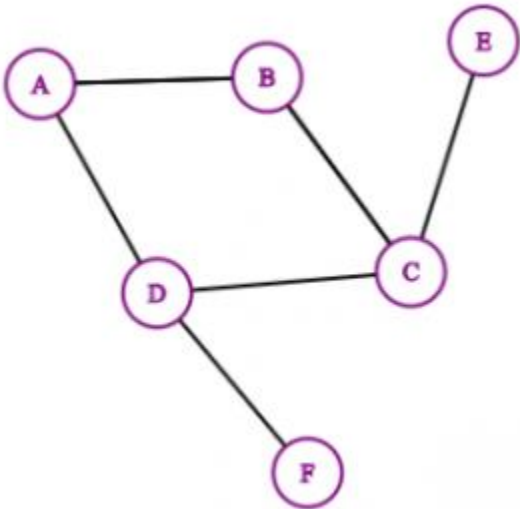
El **algoritmo A *** básicamente alcanza el resultado óptimo calculando las posiciones de todos los demás nodos entre el nodo inicial y el nodo final.



4. Terminología del Algoritmo A*

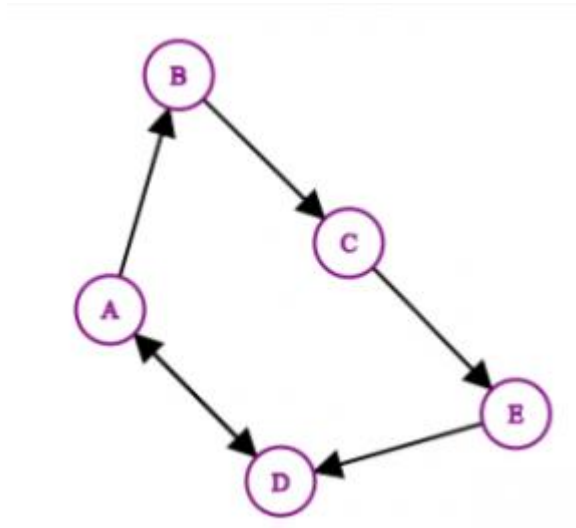
ENTENDAMOS COMO ES LA ESTRUCTURA DE UN GRAFO

Grafo no dirigido



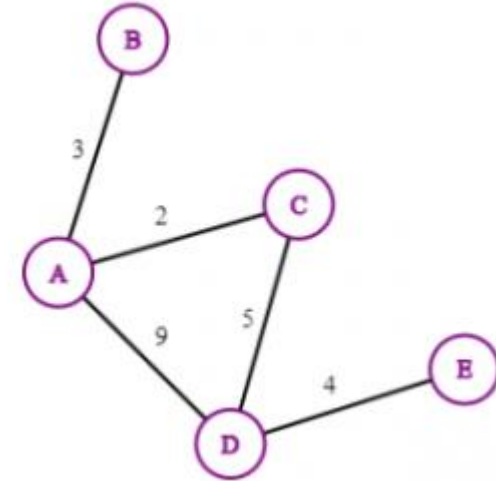
Cuando los bordes (también llamados aristas o ramas) no tienen una dirección.

Grafo dirigido



Cuando los bordes tienen una dirección.

Grafo ponderado

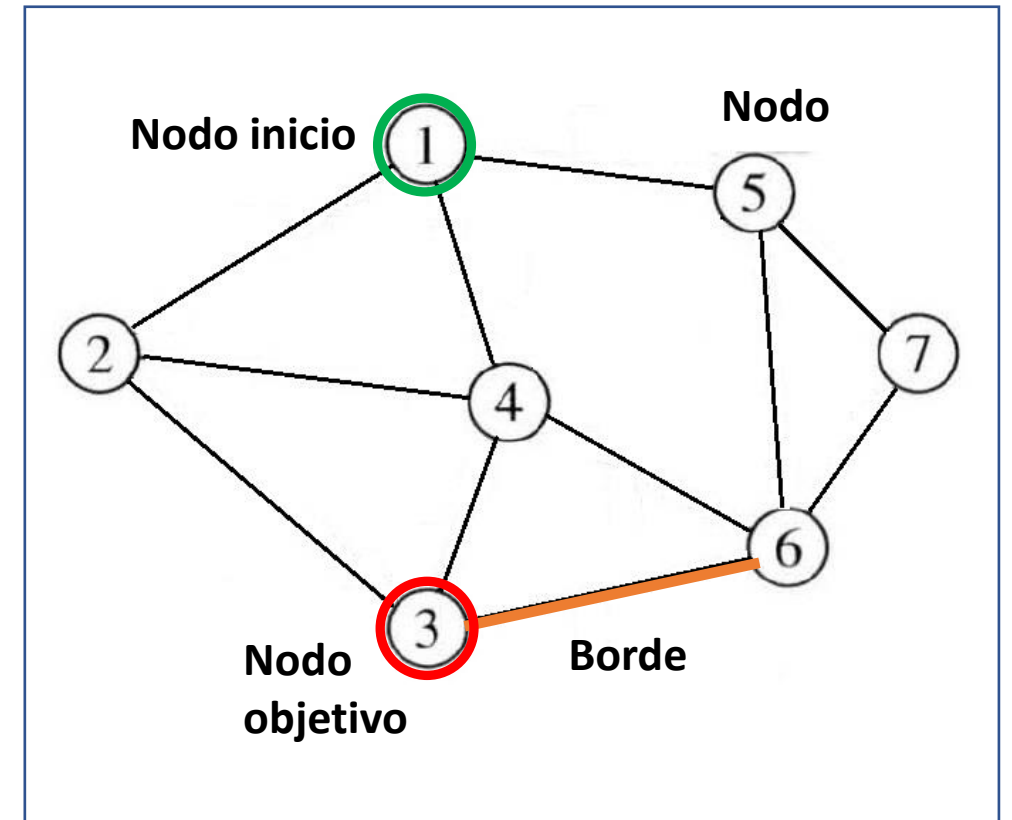


Cuando los bordes tienen un valor numérico.

4. Terminología del Algoritmo A*

ENTENDAMOS COMO ES LA ESTRUCTURA DE UN GRAFO

Nodo : (Estado o Vértice)	Todas las posibles posiciones o paradas con una identificación única.
Borde : (Arista o rama)	Cada conexión entre dos nodos.
Transición :	El acto de moverse entre nodos (o vértices)
Nodo inicio:	Dónde empezar a buscar (nodo inicial).
Nodo objetivo:	El objetivo para detener la búsqueda (nodo final).
Espacio de búsqueda:	Una colección de nodos, como todas las posiciones del tablero de un juego de mesa



4. Terminología del Algoritmo A*

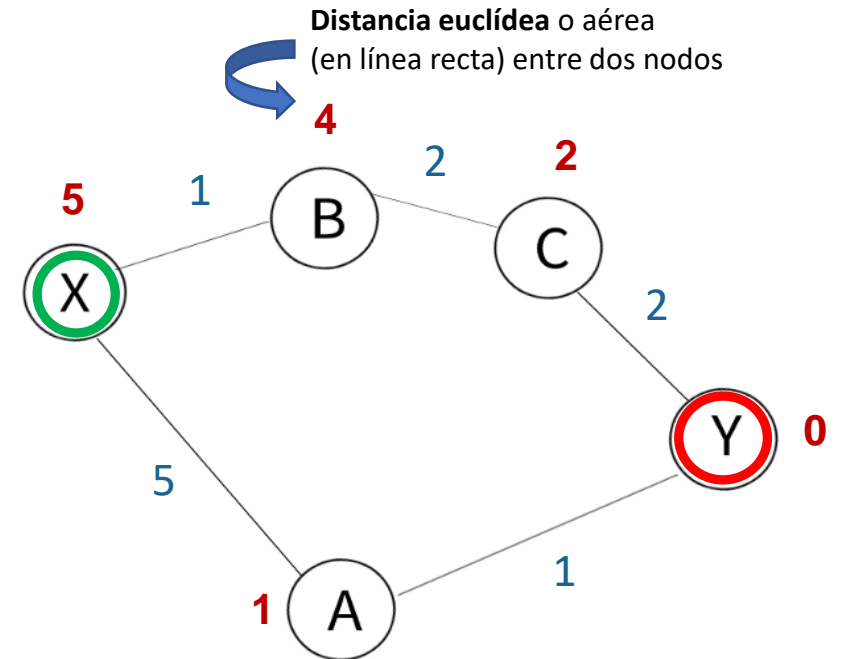
ENTENDAMOS COMO ES LA ESTRUCTURA DE UN GRAFO PONDERADO

Costo: valor numérico (por ejemplo, distancia, tiempo o gasto financiero) para la ruta de un nodo a otro nodo.

$g(n)$: el costo de la ruta entre el primer nodo y el nodo actual (dato del problema)

$h(n)$: función heurística (costo estimado heurístico desde el nodo n hasta el nodo objetivo)

$f(n)$: Costo total calculado de la ruta (elegiremos el costo menor)



4. Terminología del Algoritmo A*

Costo total calculado de la ruta

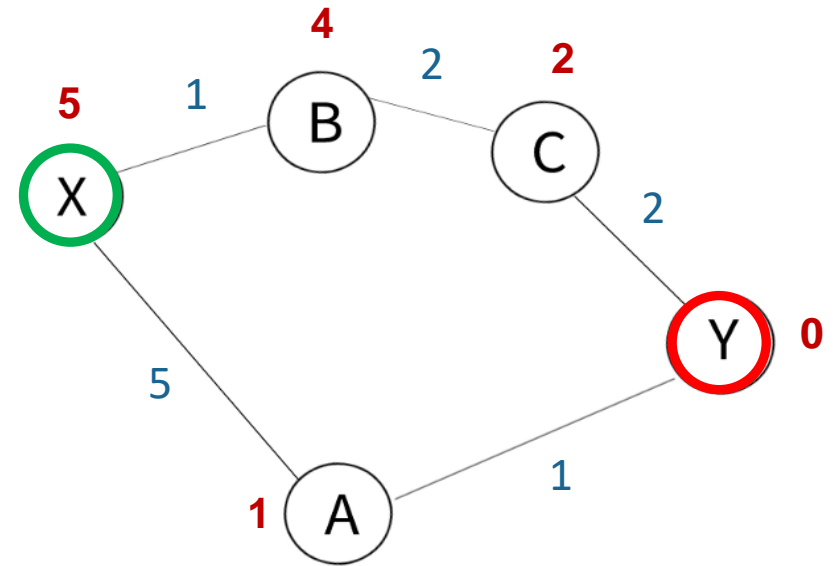
$$f(n) = g(n) + h(n) = \text{Costo total calculado de la ruta}$$

Donde:

g(n): el costo de la ruta entre el primer nodo y el nodo actual
(dato del problema)

h(n): función heurística (costo estimado heurístico desde el
nodo n hasta el nodo objetivo)

Pregunta: Usando la **función f(n)** para calcular el costo de la ruta
¿Cuál sería la ruta más corta para ir de X-Y?



5. Ejemplos del Algoritmo A*

EJEMPLO #1: Utilizando la distancia euclidiana.

Pregunta: Usando la **función f(n)** para calcular el costo de la ruta
¿Cuál sería la ruta más corta para ir de **X**=>**Y**?

Ruta 1: XAY

$$X - A \Rightarrow g(A) + h(A) = 5 + 1 = 6$$

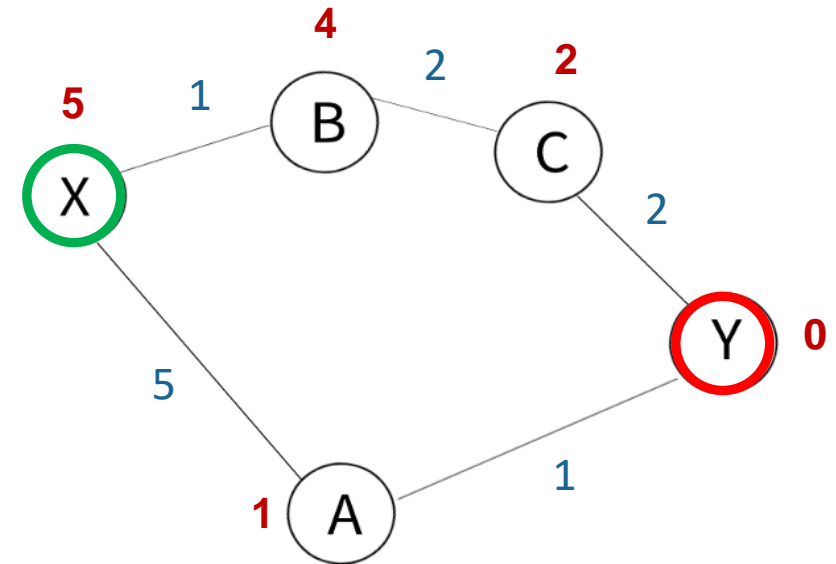
$$A - Y \Rightarrow g(Y) + h(Y) = 6 + 0 = 6$$

Ruta 2: XBCY

$$X - B \Rightarrow g(B) + h(B) = 1 + 4 = 5$$

$$B - C \Rightarrow g(C) + h(C) = 3 + 2 = 5$$

$$C - Y \Rightarrow g(Y) + h(Y) = 5 + 0 = 5$$



Solución: la ruta más corta es la ruta 2 = **XBCY**. El costo de esta vía es de 5 unidades, mientras que el costo de la ruta alternativa XAY es de 6 unidades.

g: recordar que se calcula el costo desde el nodo inicial (X)

h: recordar que se calcula el costo desde el nodo actual (n)

5. Ejemplos del Algoritmo A*

Pregunta:

Usando la **función** $f(n) = g(n) + h(n)$ para calcular el costo de la ruta
¿Cuál sería la ruta más corta para ir de **A**=>**J**?

Solución:

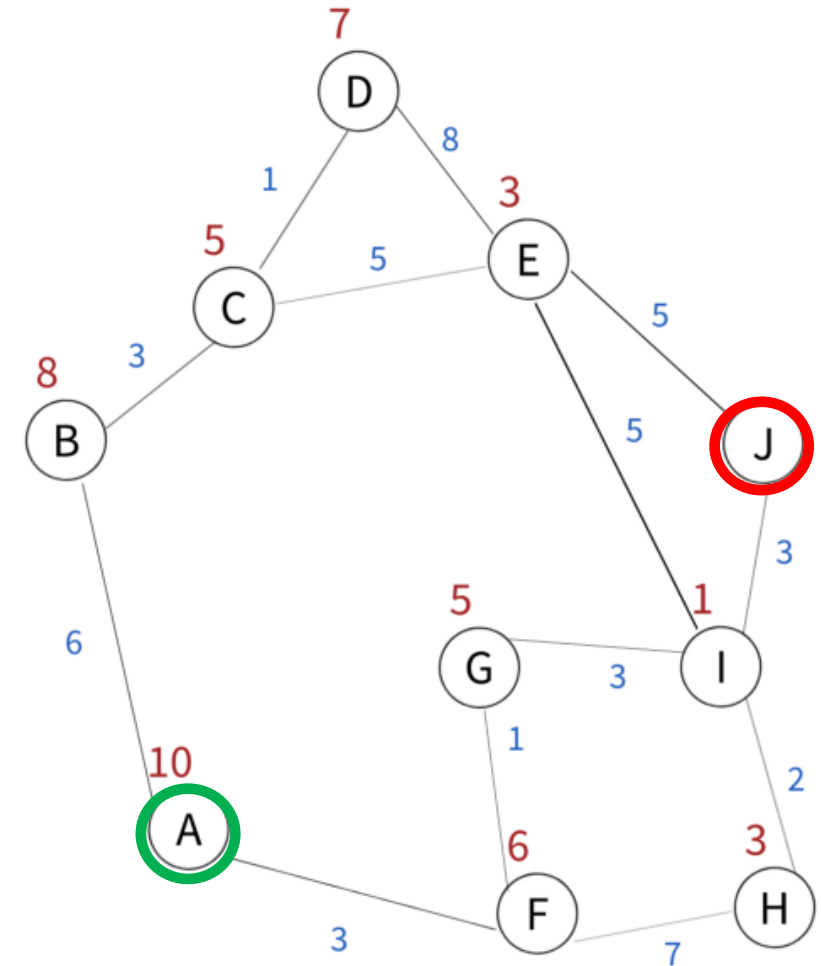
- Nos dan el costo $g(n)$ de cada borde y el $h(n)$ para cada nodo
- Desde el nodo A hay 2 puntos (B y F) para llegar a J.
- Calculamos los costos generales para elegir el de menor costo:

$$A - B \Rightarrow f(B) = g(B) + h(B) = 6 + 8 = 14$$

$$A - F \Rightarrow f(F) = g(F) + h(F) = 3 + 6 = 9$$



A* continúa desde aquí por ser el de menor costo.



5. Ejemplos del Algoritmo A*

Continuación:

- Desde el **nodo F** hay 2 puntos (G y H) para llegar a J.

$$F - G \Rightarrow f(G) = g(G) + h(G) = 4 + 5 = 9$$



A* continúa desde aquí por ser el de menor costo.

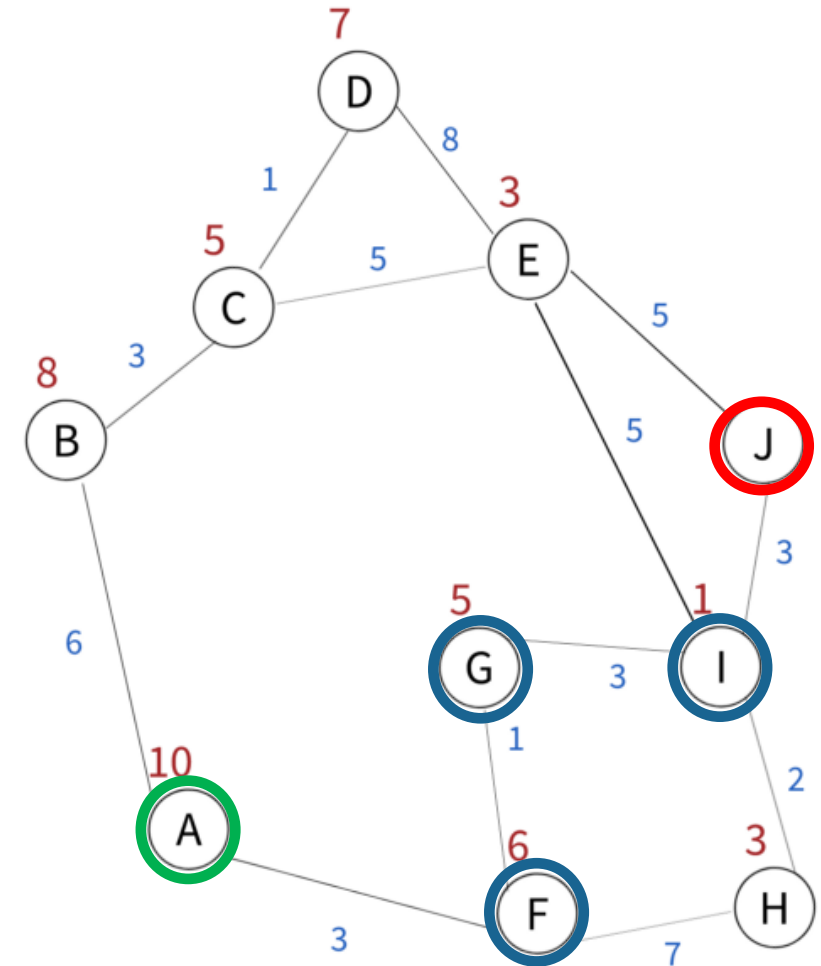
$$F - H \Rightarrow f(H) = g(H) + h(H) = 10 + 3 = 13$$

- Faltaría seguir de G – I para llegar a J

$$G - I \Rightarrow f(I) = g(I) + h(I) = 7 + 1 = 8$$

$$I - J \Rightarrow f(J) = g(J) + h(J) = 10 + 0 = 10$$

- Dado que todos los valores obtenidos después de ir al nodo F resultaron menores que $f(B) = 14$, no regresamos al nodo B.
- Bastaría que $f(G)$ o $f(I)$ sean mayores a $f(B) = 14$ para que el algoritmo se interrumpa. En este caso, según el algoritmo A*, el proceso se interrumpe aquí y la ruta continúa con el nodo B. Aquí, tan pronto como $f(C) > f(I)$.

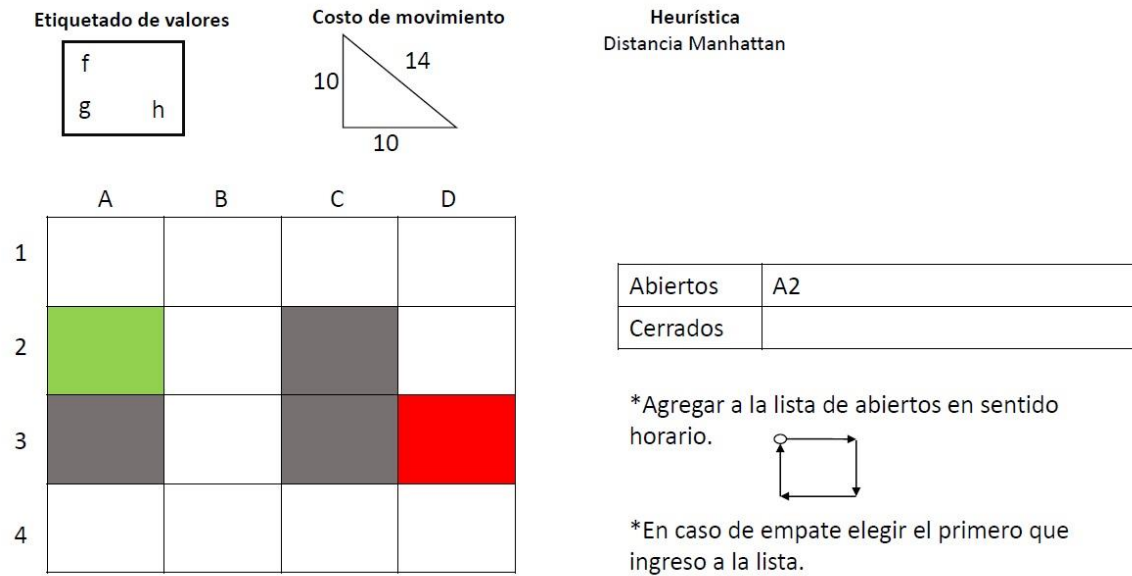


5. Ejemplos del Algoritmo A*

EJEMPLO #2: Utilizando la distancia Manhattan (Taxicab o City Block)

La distancia Manhattan, es la distancia que recorrería un automóvil en una ciudad (por ejemplo, Manhattan) donde los edificios están dispuestos en bloques cuadrados y las calles rectas se cruzan en ángulos rectos.

Planificación de ruta con A*



Donde el Costo

Movimiento horizontal: 10

Movimiento vertical: 14

Función de Coste

$$f(n) = g(n) + h(n)$$

Objetivo:

Llegar al objetivo final por el camino menos costoso.

5. Ejemplos del Algoritmo A*

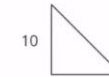
Heurística A*: Atajos para la solución de problemas

(Obtener la mejor ruta para ir de K a P – Heurística: Distancia Manhattan)



A	B	C	D	E
F	G	H	I	J
K	L	M	O	P
Q	R	R	S	T
U	V	W	X	Y

ID	F
G	H



$$G = \sqrt{a^2 + b^2}$$

ListaAbierta
Lista cerrada

Video: <https://youtu.be/JvgKPtGKmao> (30 min)

6. Aplicaciones del Algoritmo A*

A-star (A *) es un poderoso algoritmo en Inteligencia Artificial con una amplia gama de usos.

- **A *** es la opción más popular para **la búsqueda de rutas (mapas)**.

En el planeamiento de rutas, estimar el costo del camino más barato puede ser la distancia en línea recta entre dos ciudades.

Aplicado a problemas de caminos/trayectorias se utiliza la heurística:

- ✓ **Distancia Euclídea (o aérea):** distancia en línea recta entre dos nodos
- ✓ **Distancia Manhattan:** distancia en línea recta entre dos nodos
- En Machine Learning y optimización de búsqueda.
- En el desarrollo de juegos donde los personajes navegan a través de terrenos complejos y obstáculos para llegar al jugador.

PREGUNTAS

Dudas y opiniones