

## Goku Smart IA

Carlos Alberto Henao - 2110339  
Oscar Trujillo - 2110318  
Sebastián Tutistar Valencia - 2110309

### - Avara & A\*

La heurística utilizada en este algoritmo se llama "heurística de distancia de Manhattan".

La heurística de distancia de Manhattan se calcula encontrando la distancia entre el nodo actual y el objetivo (que puede ser uno o varios, como en este caso) sumando las diferencias absolutas de sus coordenadas x e y, tal como se muestra en:

```
// Función heurística para estimar el costo restante desde un nodo hasta el objetivo
function heuristic(node, goals) {
  let lowerEstimatedCost = Infinity;
  for (let i = 0; i < goals.length; i++) {
    const goal = goals[i];
    const dx = Math.abs(node[0] - goal[0]);
    const dy = Math.abs(node[1] - goal[1]);
    const estimatedCost = dx + dy;
    if (estimatedCost < lowerEstimatedCost) {
      lowerEstimatedCost = estimatedCost;
    }
  }
  return lowerEstimatedCost;
}
```

El algoritmo utiliza una cola de prioridad para ordenar los nodos en función de la suma de la distancia de Manhattan desde el nodo actual hasta los objetivos y la distancia recorrida hasta el nodo actual. Esto permite al algoritmo expandir primero los nodos que tienen una distancia estimada menor a los objetivos y una distancia menor recorrida. El algoritmo también utiliza una lista de nodos visitados para evitar expandir los mismos nodos dos veces.

En esencia se usa una cola de prioridad para encontrar la mejor solución en la matriz que contiene 2 nodos meta, en este caso, las 2 esferas del dragón.

### **Justificación de la admisibilidad de la heurística planteada**

Esta heurística es admisible, lo que significa que nunca sobreestima el costo real del camino para alcanzar la meta.

Como cada movimiento en una cuadrícula sólo puede ser horizontal o vertical, pero no diagonal, la distancia Manhattan siempre es menor o igual que la distancia real entre dos nodos. Por lo tanto, esta heurística nunca sobreestima la distancia restante para alcanzar el objetivo.

En este caso, la función de distancia Manhattan se utiliza para calcular la distancia entre un nodo y todos los objetivos restantes, y se selecciona el objetivo más cercano para expandir. Debido a que la heurística nunca sobreestima la distancia real a la meta, el algoritmo A\* puede encontrar la solución óptima en el menor número posible de pasos.

Además, en el algoritmo Avara (Greedy) adaptado para búsqueda informada, se utiliza la heurística para elegir el siguiente nodo que se expandirá, y se elige el nodo con la menor estimación de costo, por lo que una heurística admisible garantiza que el algoritmo encontrará una solución óptima si esta existe.

Link del Repositorio: <https://github.com/klark003/Proyecto-1-IA>