

Sistemas de Inteligencia Artificial

TP1:

Métodos de Búsqueda

Grupo 6:

- + Desiree Melisa Limachi
- + Joseph Rouquette
- + Magdalena Flores Levalle
- + Matias Ezequiel Daneri



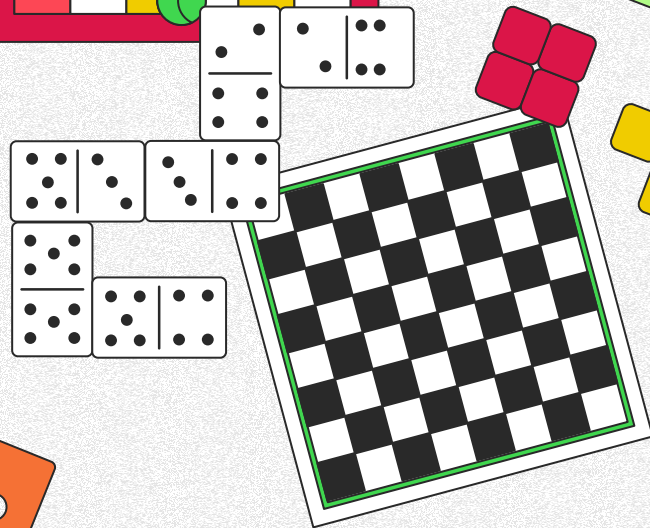
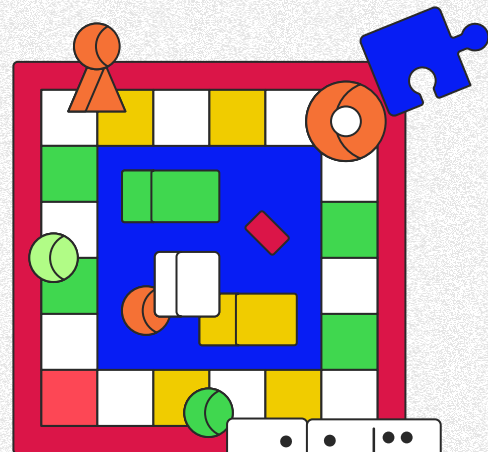
Ejercicio 1

8-puzzle

8		6
5	4	7
2	3	1



	1	2
3	4	5
6	7	8



Estructura de estado



La representación del tablero será una **matriz 3x3**



El valor de cada casillero será un **número entre [0,8]**



Consideramos que todas las **acciones** tienen el **mismo costo: 1**

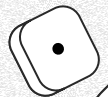
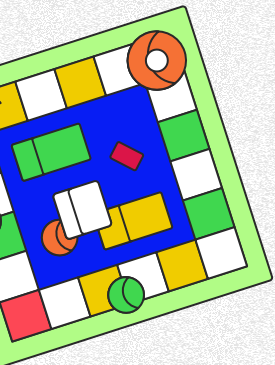


Para mover una pieza, la condición será que una **casilla adyacente** sea la de **valor 0**, es decir, que esté **vacía e intercambiarán lugar**.



La **condición de finalización** se va a dar cuando las casillas se encuentren ordenadas de la siguiente manera:

	1	2
3	4	5
6	7	8

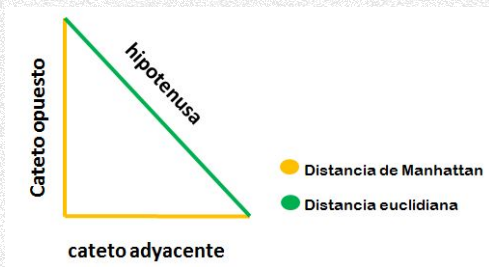




Heurísticas admisibles no triviales

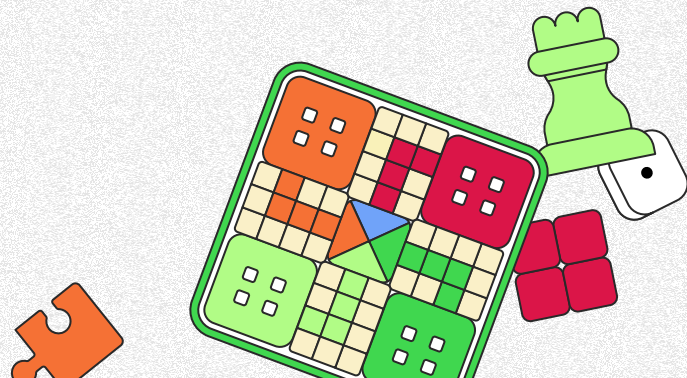
Distancia de Manhattan

- + Calcula la suma de las distancias Manhattan entre la posición actual de cada pieza y su posición objetivo.
- + Es admisible porque $h(n) < \text{costo}$ debido a que en 8-puzzle los movimientos válidos son en horizontal y vertical.



Distancia de Hamming

- + Calcula la cantidad de piezas que no se encuentran en la posición correcta
- + Es admisible porque $h(n) < \text{costo}$ debido a que siempre deberemos hacer al menos $h(n)$ movimientos





Heurísticas admisibles no triviales

Distancia de Manhattan

8		6
5	4	7
2	3	1

1	2	3	4	5	6	7	8
3	4	2	0	2	4	2	4

Manhattan = 21
(3+4+2+2+4+2+4) = 21

	1	2
3	4	5
6	7	8

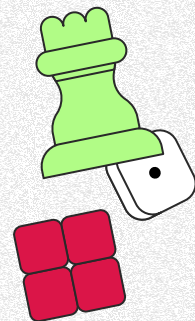
Distancia de Hamming

1	2	3	4	5	6	7	8
x	x	x	✓	x	x	x	x

8		6
5	4	7
2	3	1

Hamming = 7

	1	2
3	4	5
6	7	8



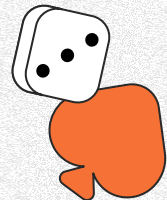
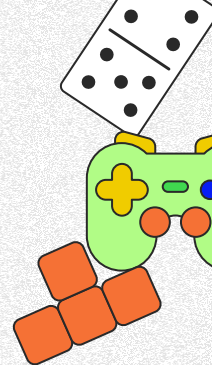
Métodos de búsqueda

Como conocemos el estado objetivo, entonces podemos usar métodos de búsqueda informados y de esta forma minimizar el costo

A* con Distancia de Manhattan

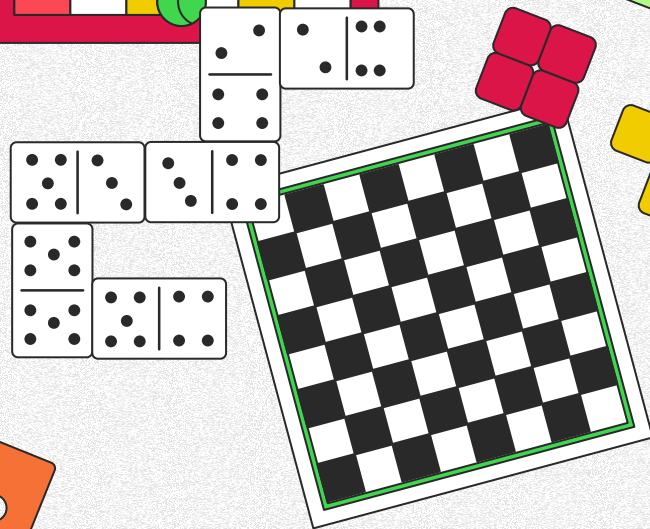
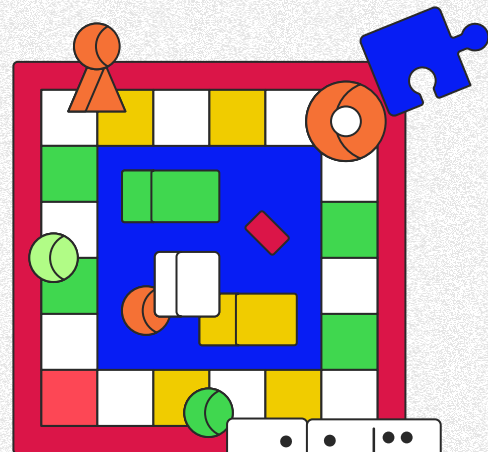
- + Heurística admisible
- + Costo de cada arista > 0
- + Cantidad finita de nodos sucesores

⇒ Solución óptima y completa



Ejercicio 2

Sokoban



Estructura de estado



Se usa la clase “**State**” para almacenar los estados.

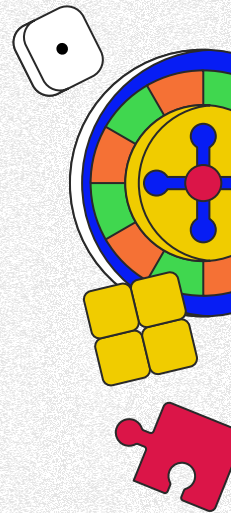
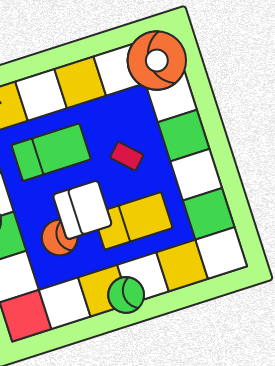


Esta almacena la posiciones del **player**, **boxes** & **goals** según cada uno de los movimientos del juego.



Problema bien definido:

- **Estado inicial**
- **Conjunto de posibles acciones:** $[\uparrow, \downarrow, \leftarrow, \rightarrow]$
- **Transiciones:** Mover elementos
- **Función de costo:** 1
- **Condición de solución:** Cajas sobre los objetivos





Métodos de búsqueda y heurísticas

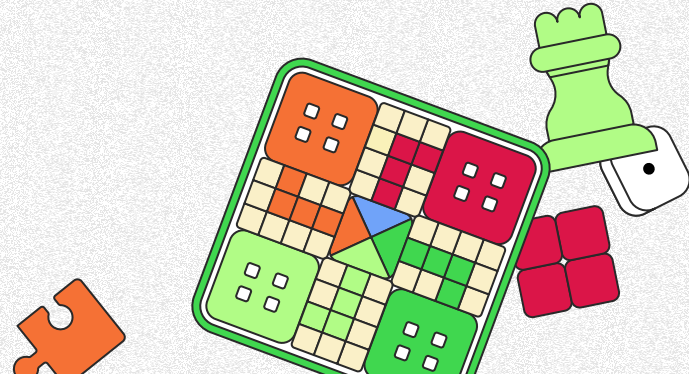
Al usuario se le otorga la posibilidad de elegir entre los siguientes métodos y heurísticas para la resolución al tablero que elija:

Métodos

- + BFS
- + DFS
- + Global Greedy
- + A*

Heurísticas

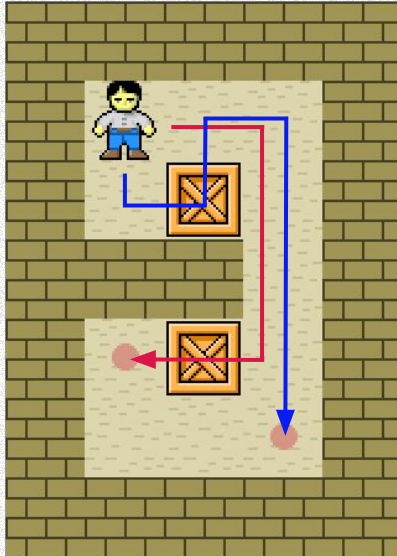
- + Manhattan
- + Combined



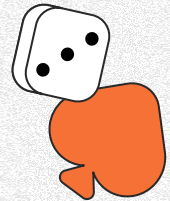
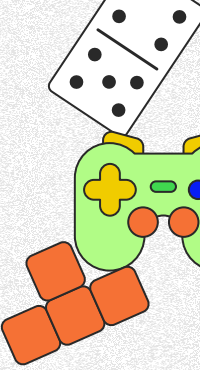
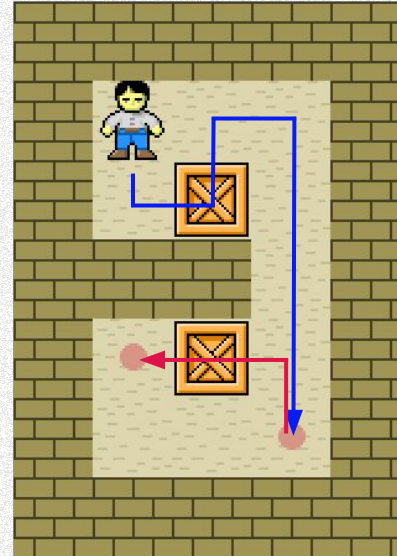
Heurística **NO** admisible

Se considera que una heurística es no admisible si sobreestima el costo real de la solución

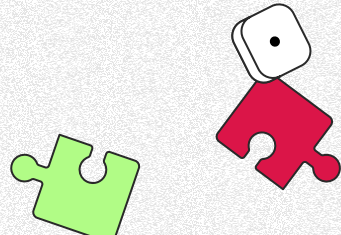
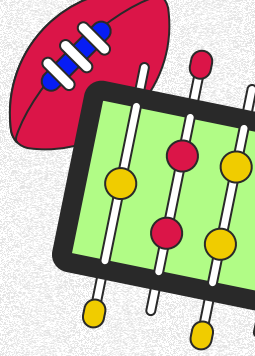
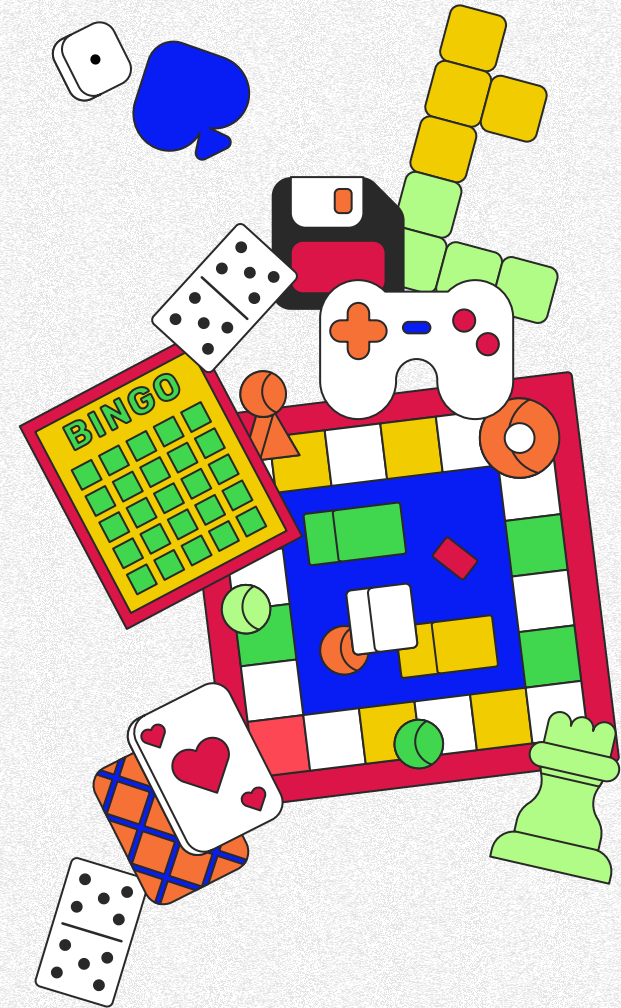
Estimado

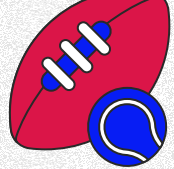


Real

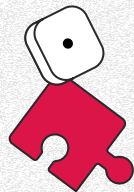
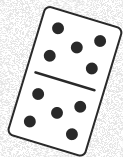
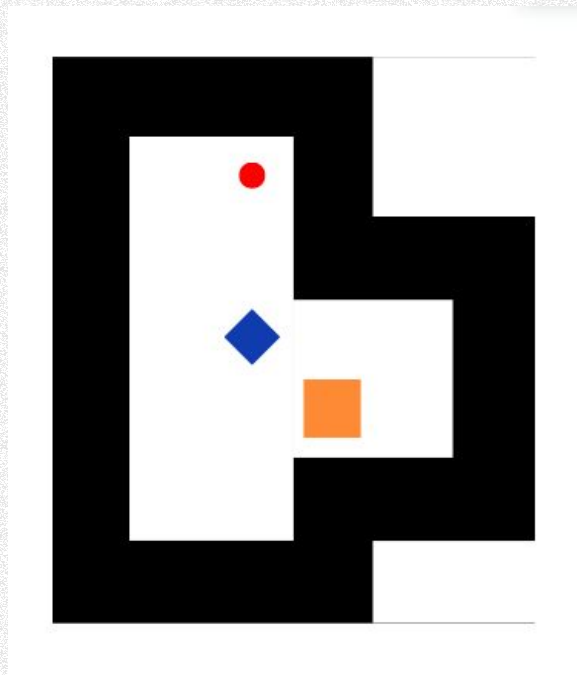


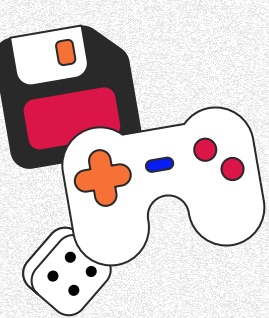
Resultados obtenidos



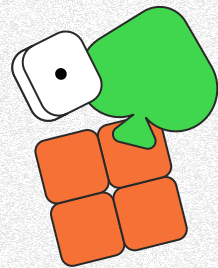
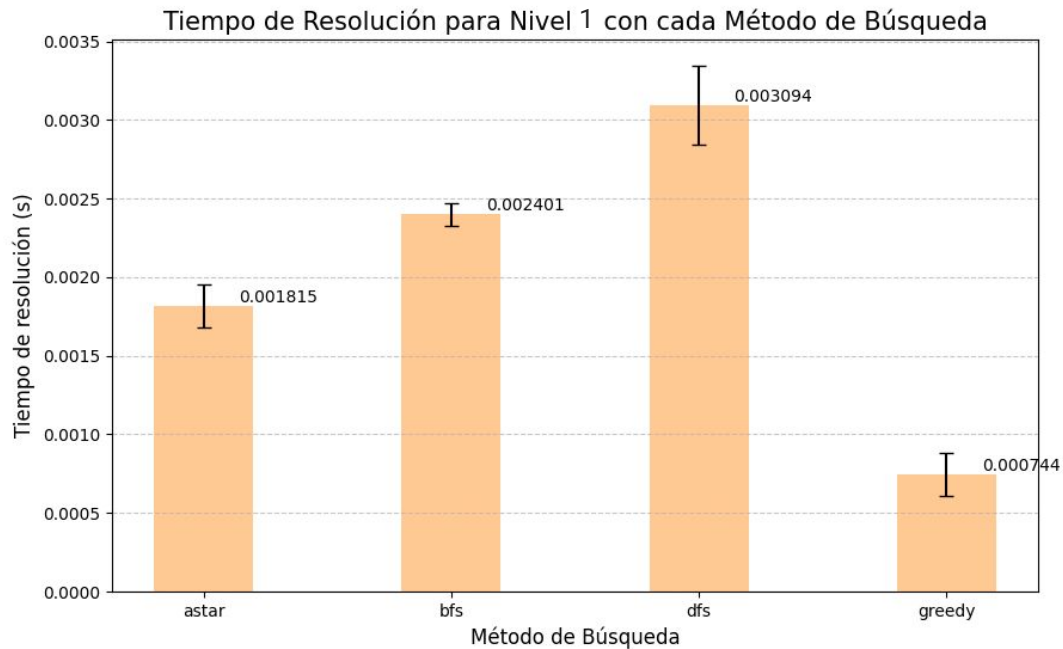


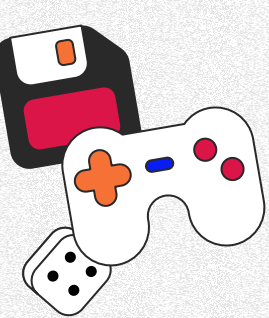
Nivel 1



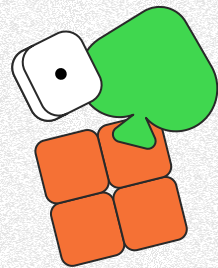
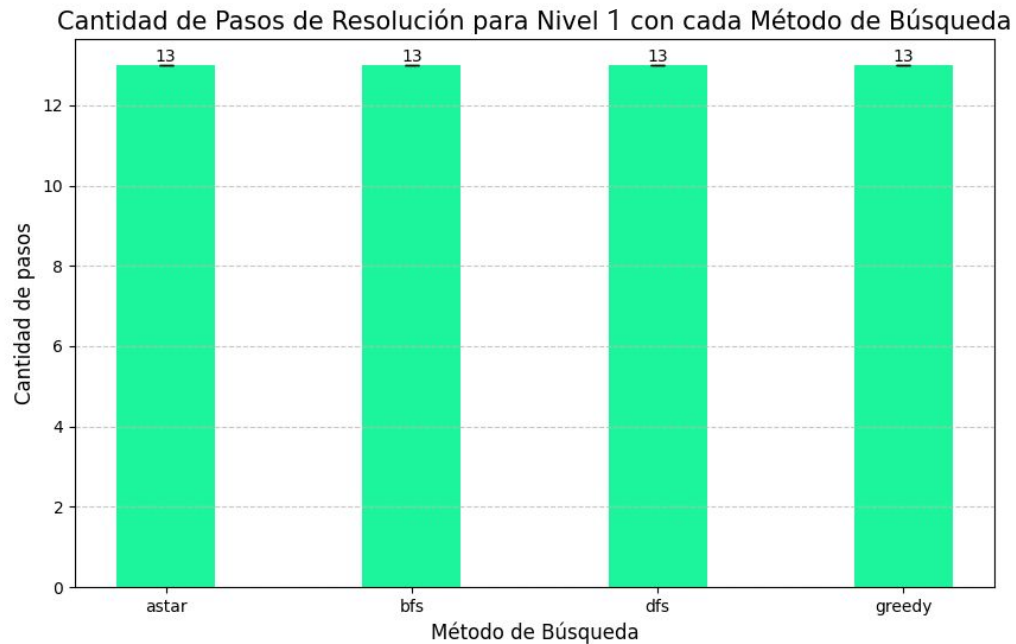


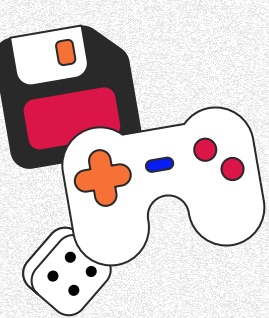
Tiempo de resolución



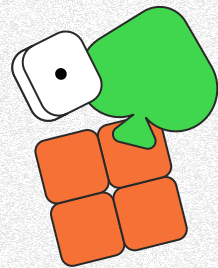
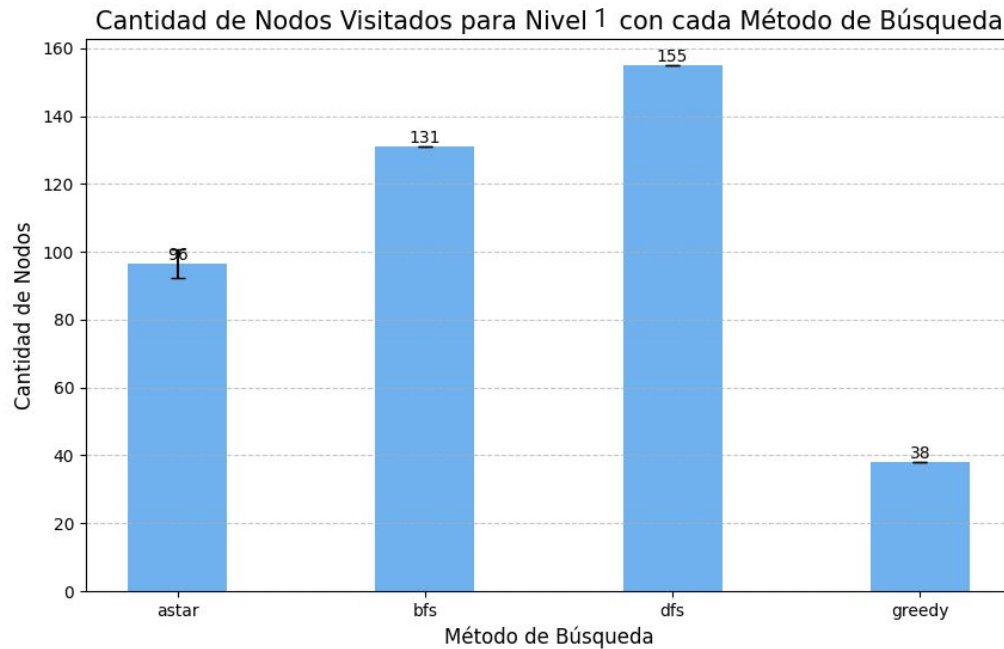


Cantidad de pasos

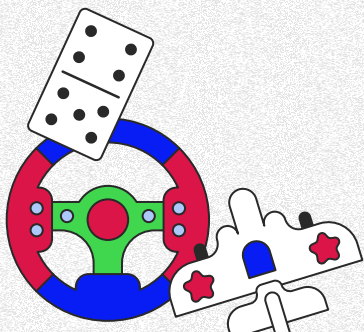
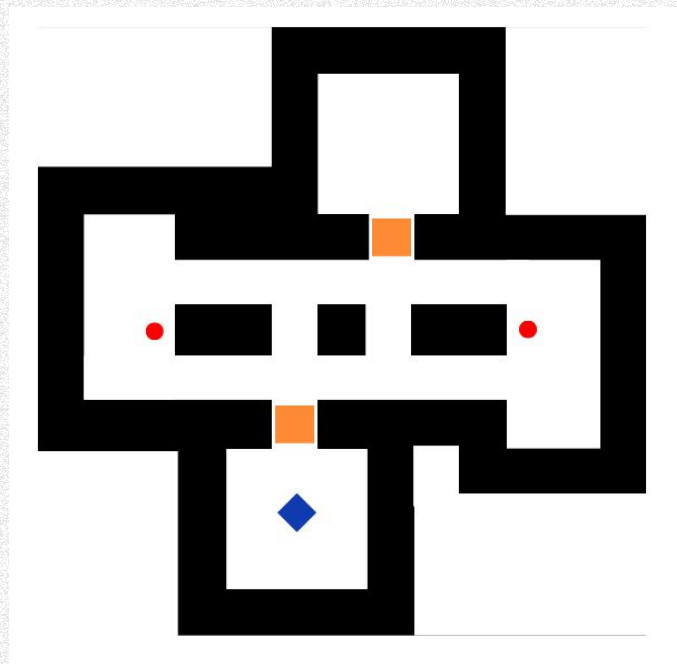
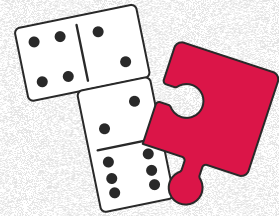




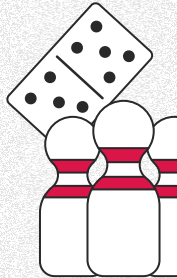
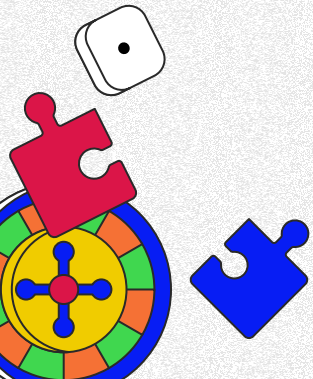
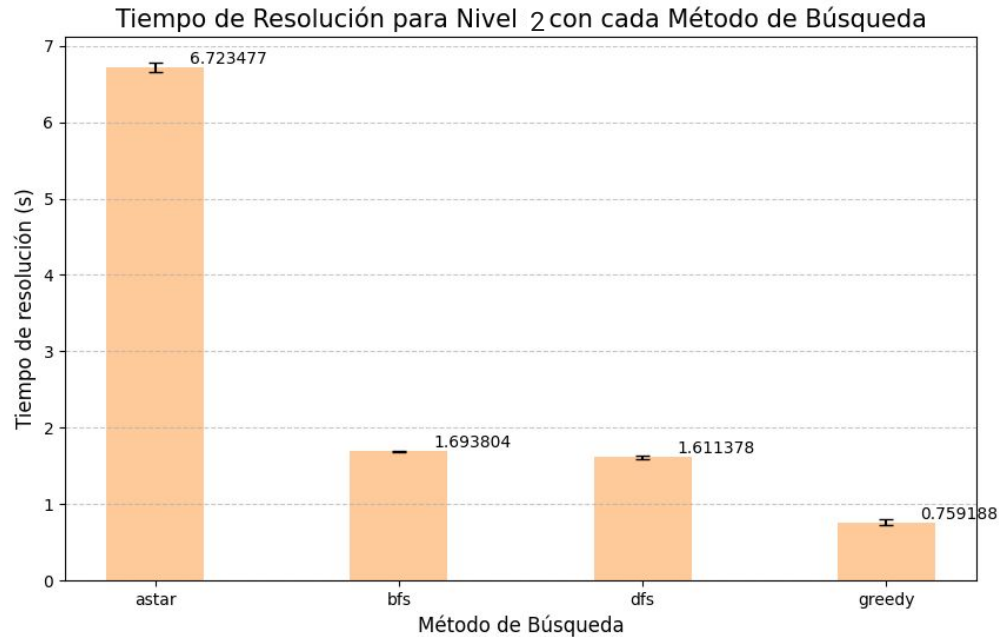
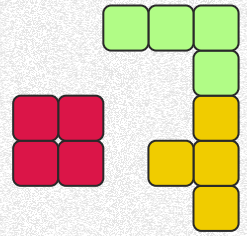
Cantidad de nodos visitados



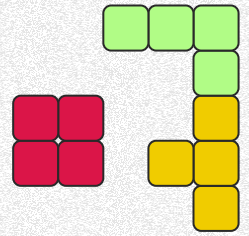
Nivel 2



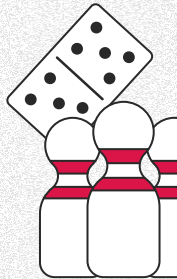
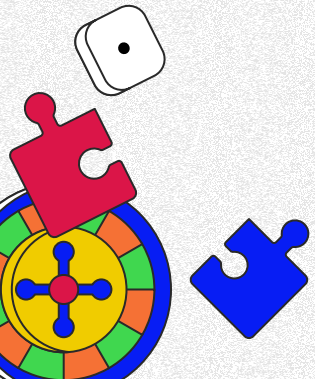
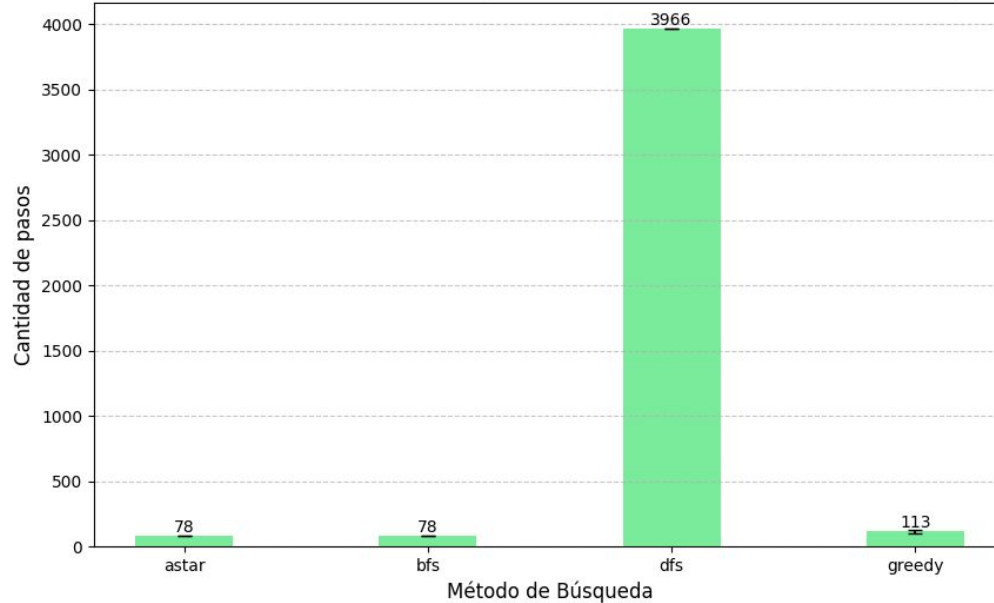
Tiempo de resolución



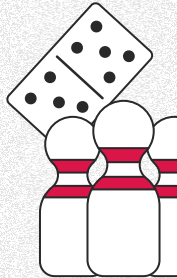
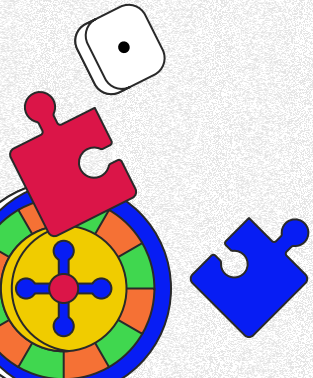
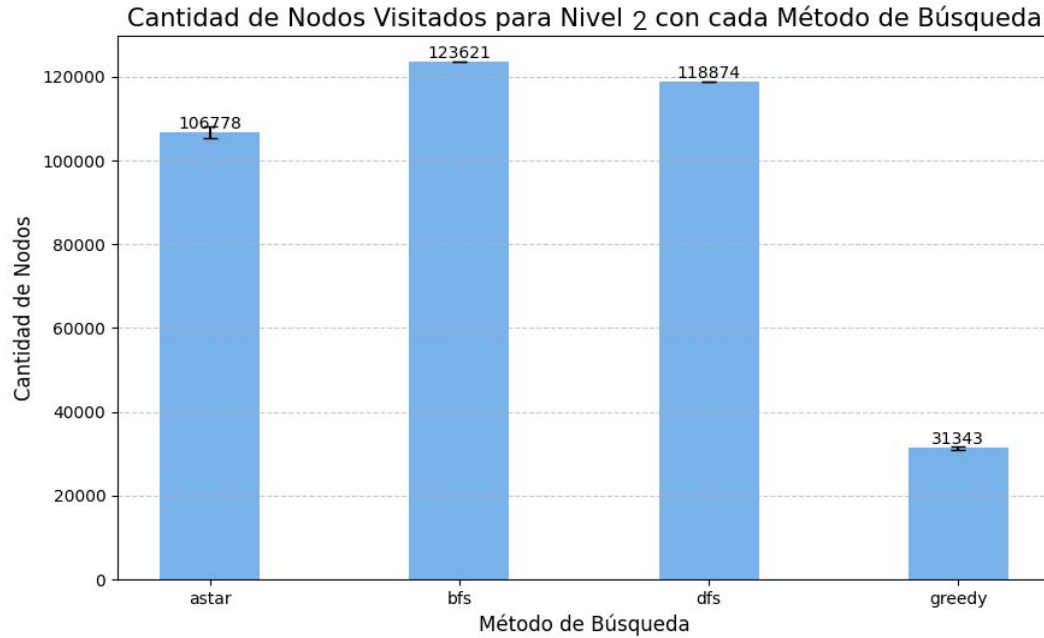
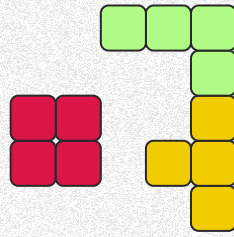
Cantidad de pasos



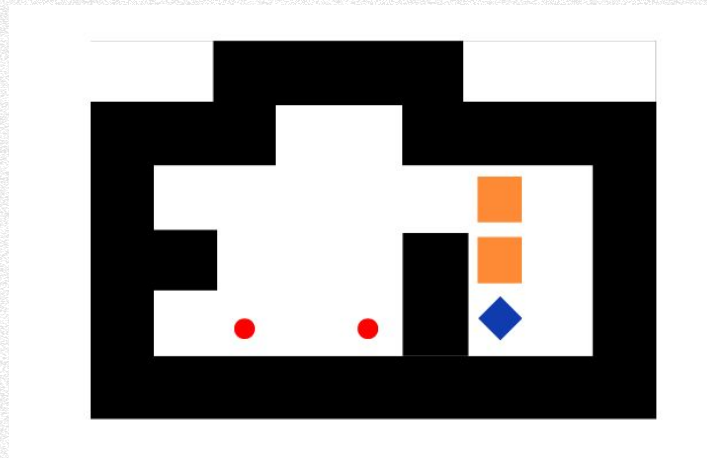
Cantidad de Pasos de Resolución para Nivel 2 con cada Método de Búsqueda



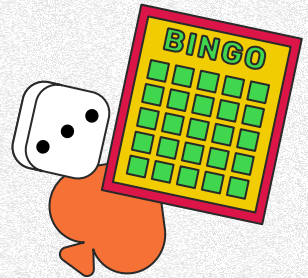
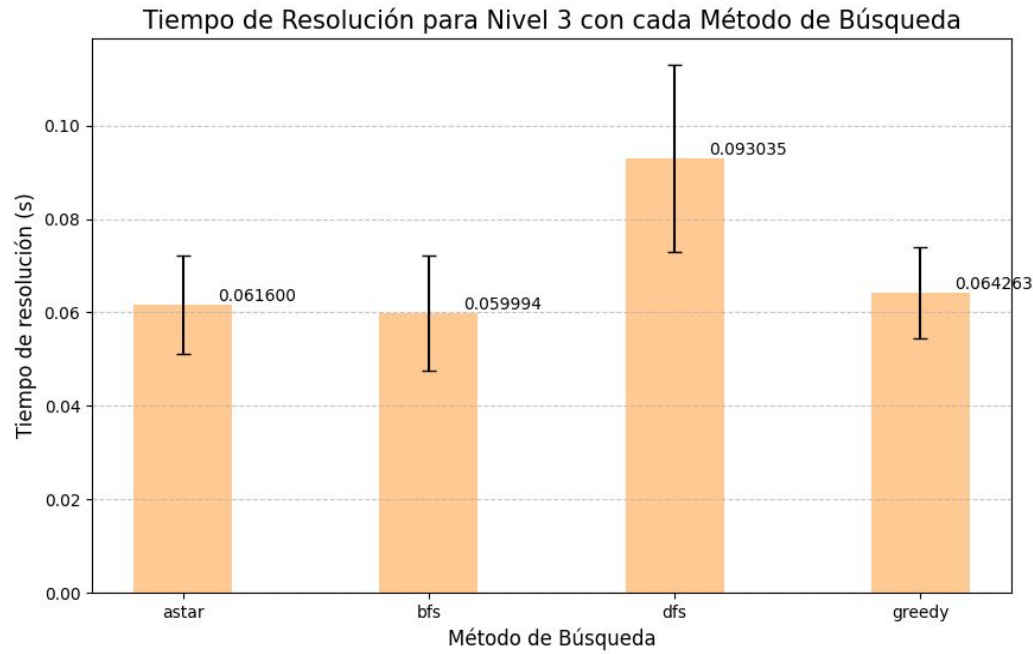
Cantidad de nodos visitados



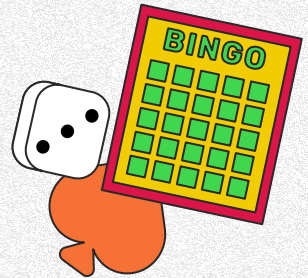
Nivel 3



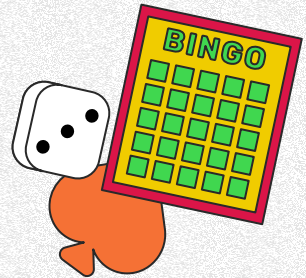
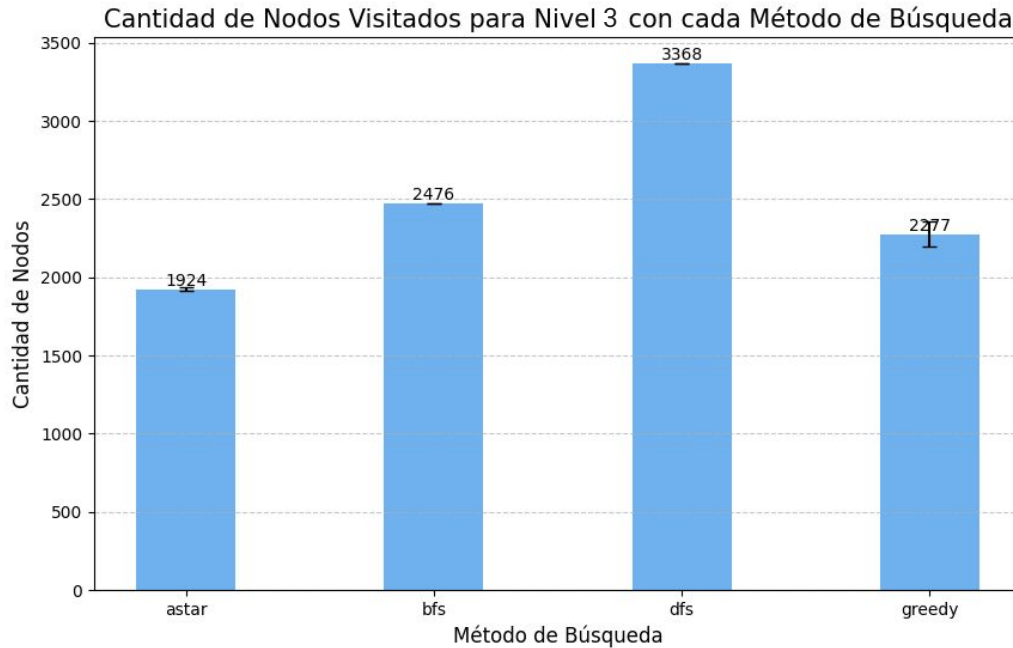
Tiempo de resolución



Cantidad de pasos

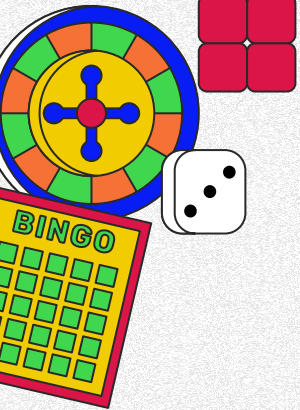


Cantidad de nodos visitados

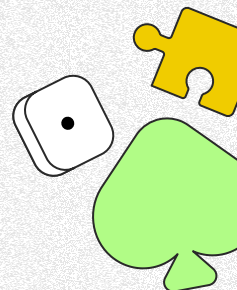
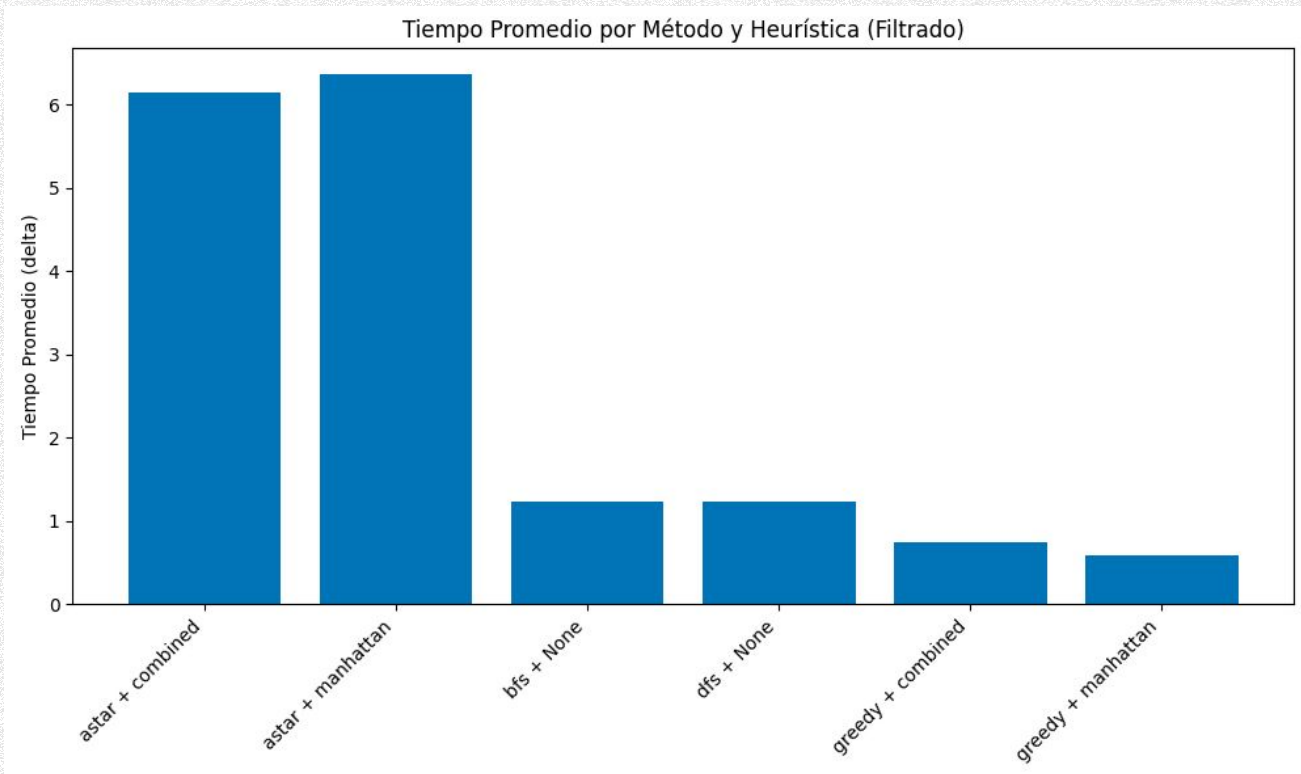


Promedio de los niveles

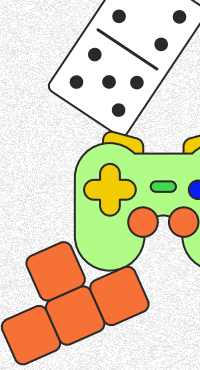
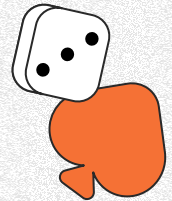
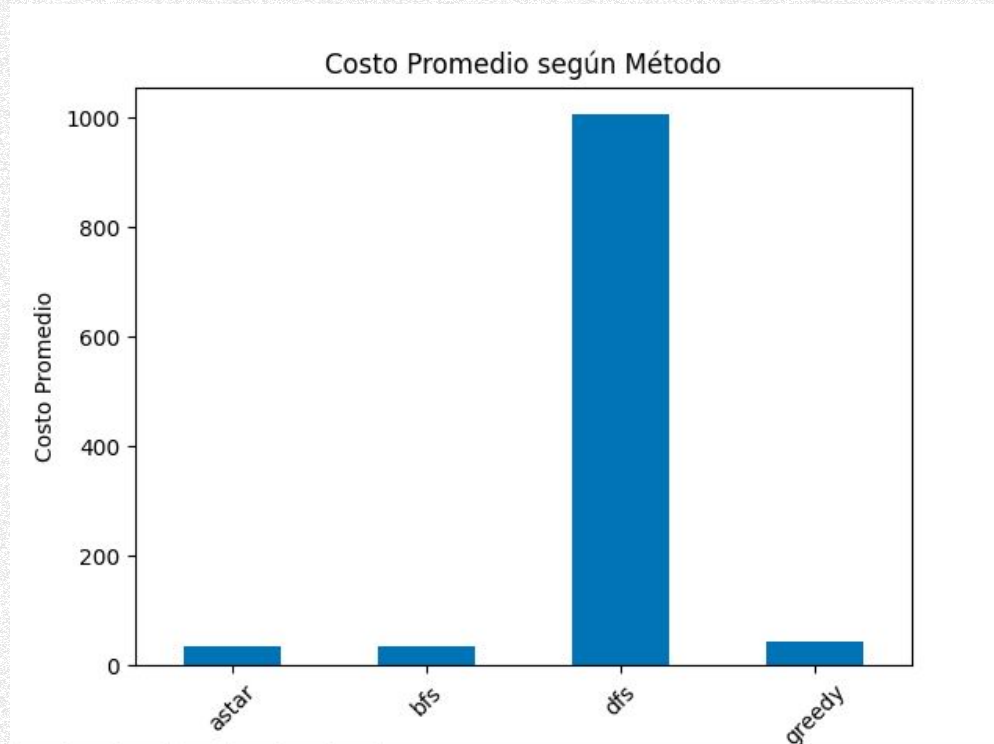




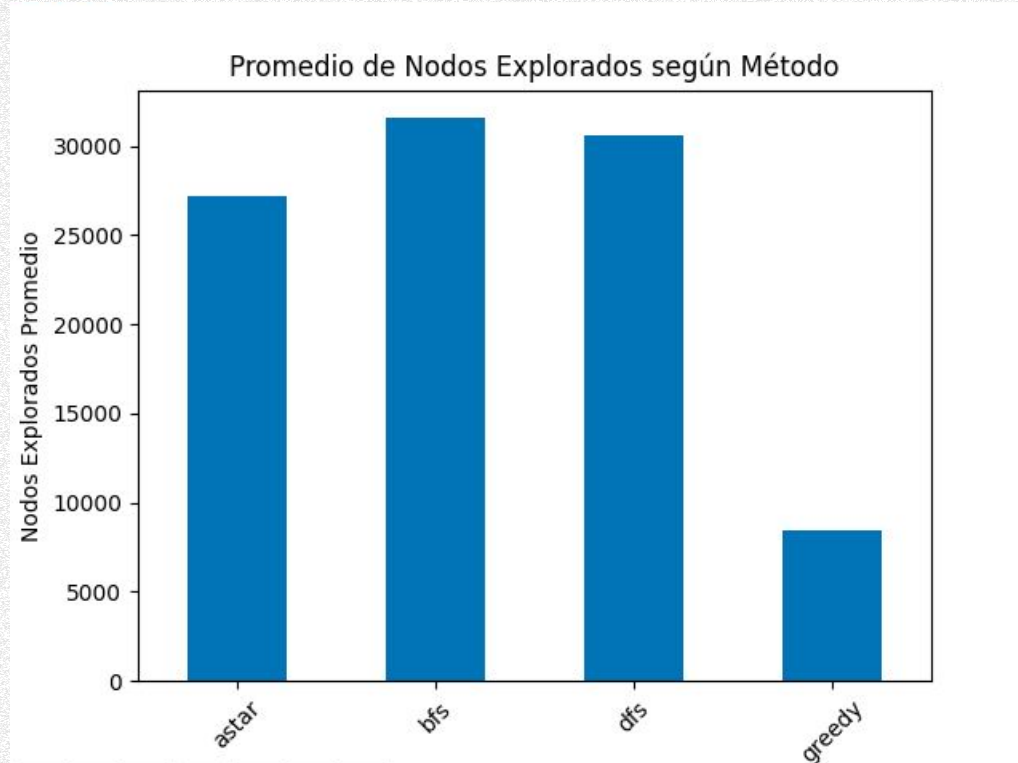
Tiempo



Costo

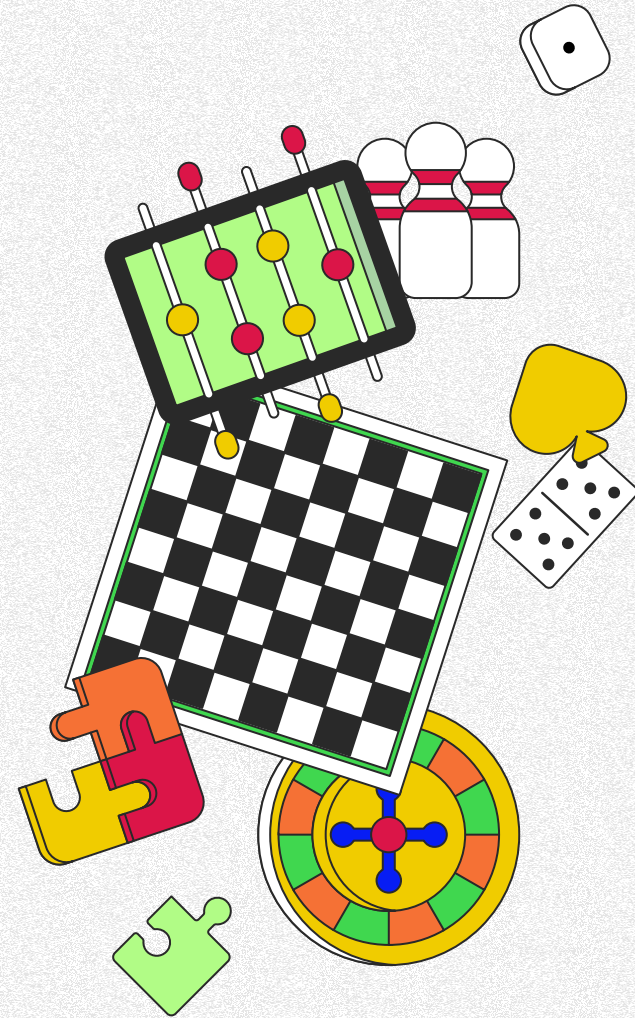


Nodos explorados



Conclusiones

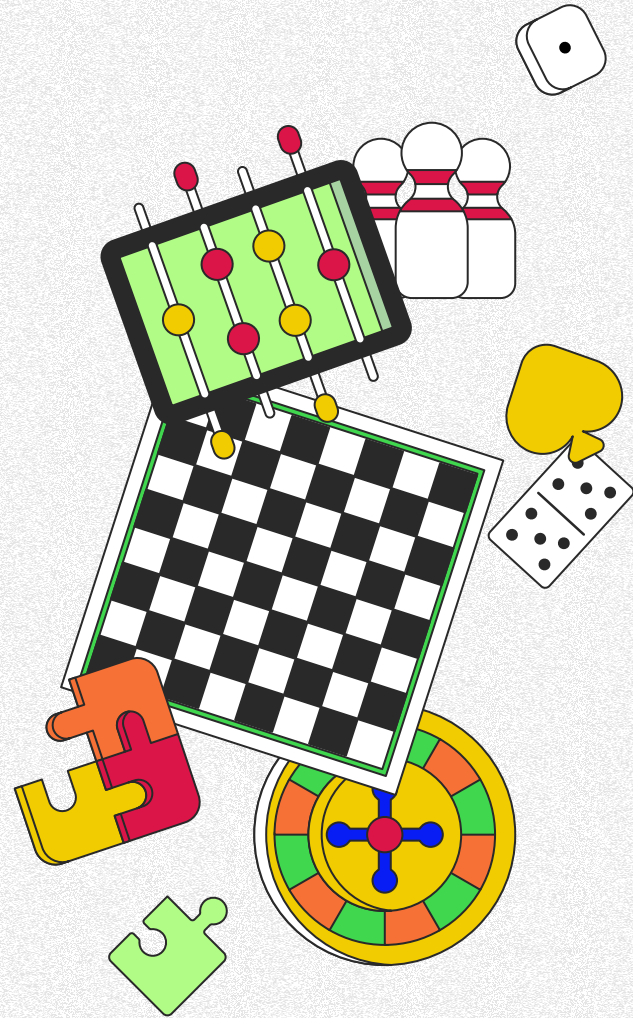
- + A* con heurísticas admisibles logra soluciones óptimas y completas, al igual que BFS
- + Sin embargo, el uso de la heurística combinada suele resultar en un tiempo de resolución más largo que el de la heurística de Manhattan.
- + Por otro lado, Greedy ofrece un tiempo de resolución considerablemente más corto que otras alternativas, aunque no garantiza siempre una solución óptima.

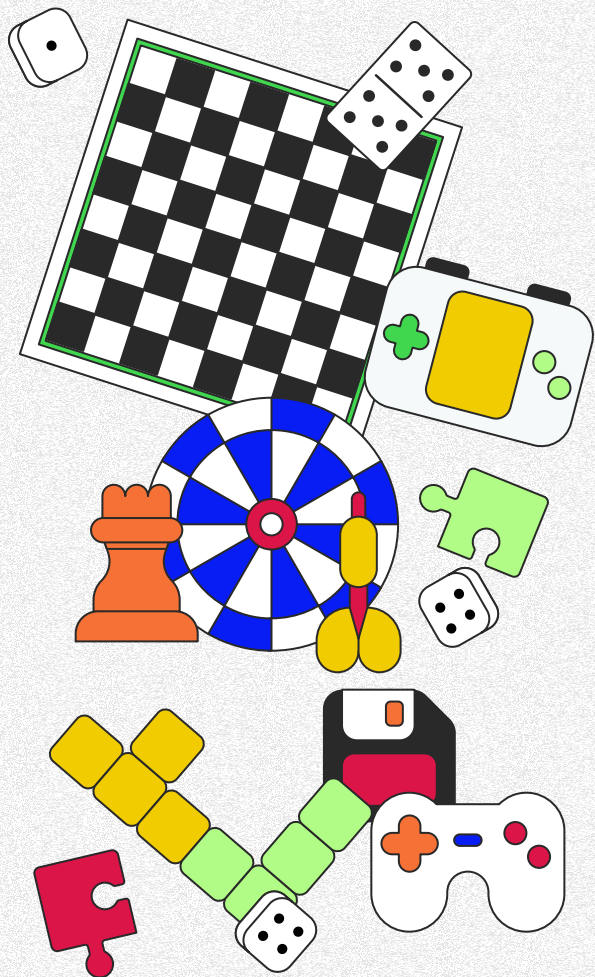


Conclusiones (focus DFS/BFS)

Se observa que DFS es un Algo que no solo no ofrece una solución óptima a diferencia de BFS sino que también es más largo y recorrido más nudos. En cambio, en algunos casos (nivel 3), es más rápido que BFS

Se puede concluir que el Algo perfecto depende a menudo del nivel





FIN