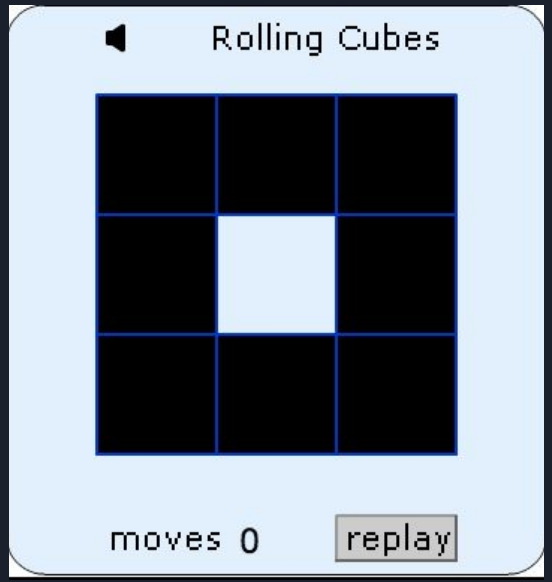
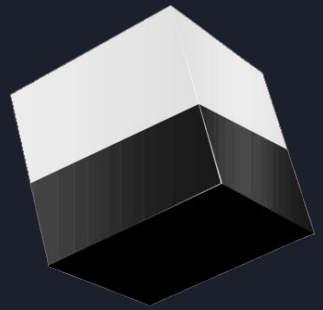
Métodos de búsqueda

Grupo 12 - Sistemas de Inteligencia Artificial
Romain Thibaut Marie Latron
Kevin Chidiac
Ramiro Hernán Olivera Fedi
Alan Arturo Hernández Gutiérrez

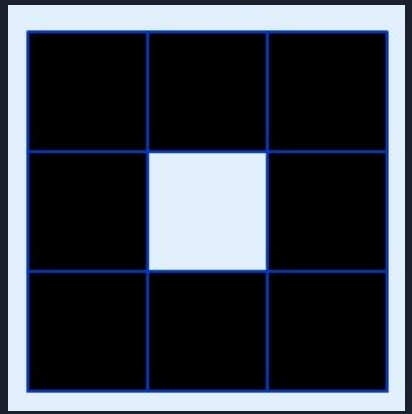
El juego



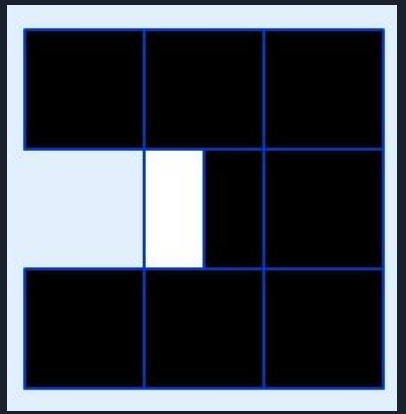
El juego ***Rolling Cubes*** consiste de un tablero cuadrado de ***9*** casilleros y ***8*** piezas. El objetivo es rotar todas las piezas negras haciéndolas girar sobre el espacio vacío, hasta que todos muestren la cara blanca.



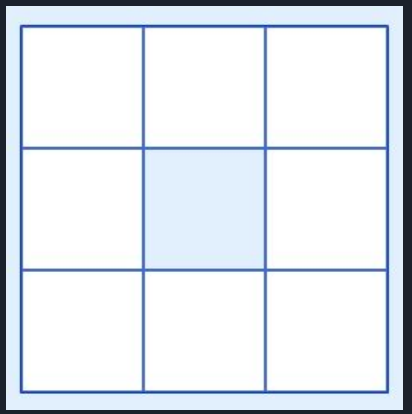
El juego



Estado inicial



Estado intermedio



Estado objetivo

Definición del problema

01 **Estado inicial.** Tablero con ***8*** piezas mostrando su cara negra, sin pieza en el casillero del medio.

02 **Conjunto de posibles acciones.** Dado un estado se puede rotar cualquier pieza que cumpla con la condición de que comparta uno de sus lados con el casillero vacío.

03 **Modelo de transición.** Mover una pieza sobre el casillero vacío genera un estado en el que la posición en la que se encontraba ahora está vacía, y la posición que estaba vacía ahora contiene una rotación de la pieza movida.

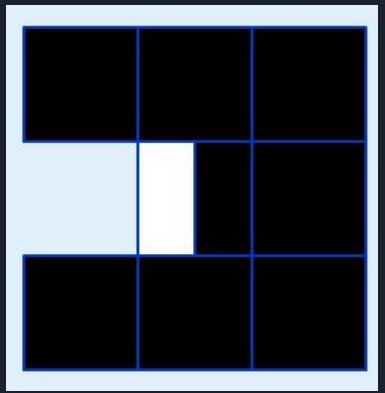
04 **Condición de solución.** Un estado es solución si todas las piezas del tablero son blancas.

Representación del problema: *Estado*

Cubo. Modelo representativo de cada pieza, compuesto únicamente por el color de su cara visible.

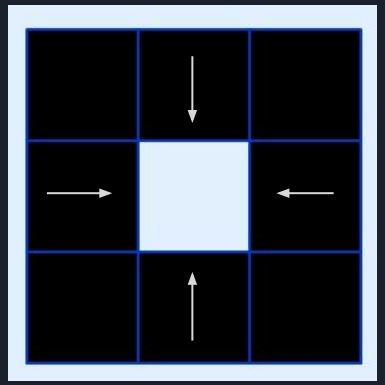
Tablero. Lista de ***Cubos***, representando la posición de cada cubo en el tablero.

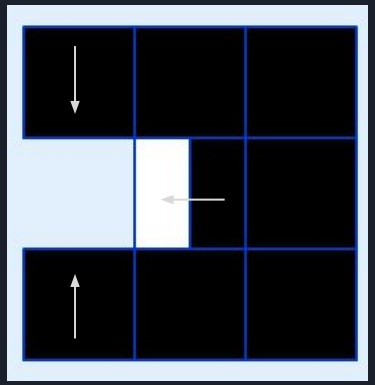
Casillero vacío. Índice del casillero vacío en el tablero. El casillero vacío se modeló como una pieza más, cuyo color es ***EMPTY.***

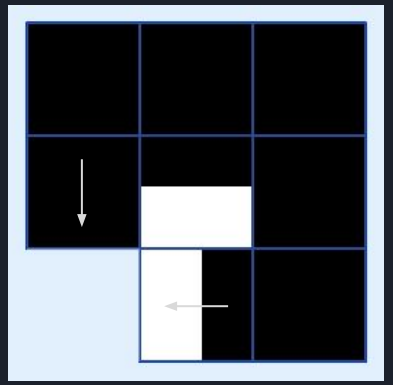


Reglas. Lista de reglas permitidas para el tablero de este estado.

Representación del problema: *Reglas*

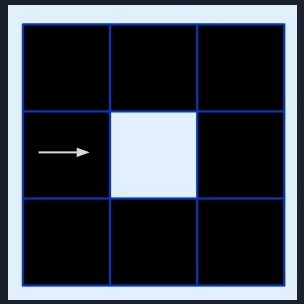


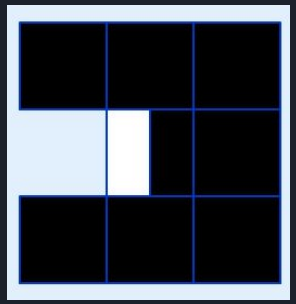




Reglas: Se representan todos los movimientos posibles para las piezas adyacentes al casillero vacío (4 movimientos). Pero las rotaciones permitidas se restringen de acuerdo a la posición del mismo. Hay entonces ***4 reglas por estado***.

Representación del problema: *Transición*





Al aplicar una regla a un ***nodo*** (estado), se obtiene un nuevo nodo con una copia del tablero con la regla aplicada. De esta manera, el ***modelo de transición,*** aunque dependiente de la regla, puede describirse como:

1. Ubicar en la posición de la pieza afectada por la regla una pieza vacía

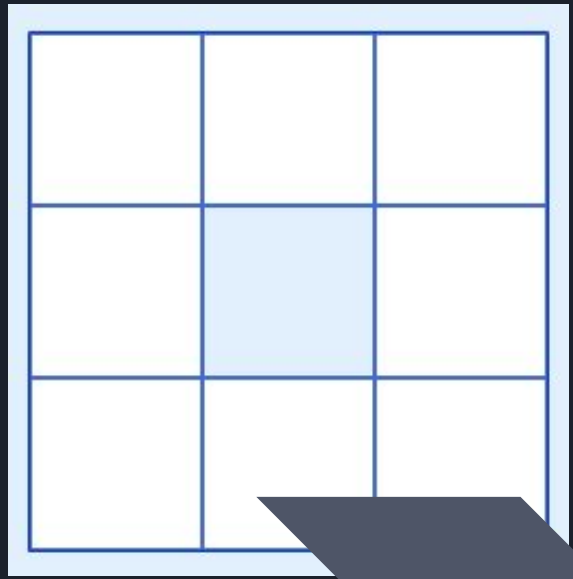
2. Ubicar en la posición de la pieza vacía una rotación de la pieza afectada con la regla, de acuerdo a una tabla de rotaciones

Representación del problema: *Terminación*

Se define como la ***condición de terminación*** a la siguiente premisa:

Siendo el estado uno válido, todos los cubos son de color blanco, salvo uno que es vacío.

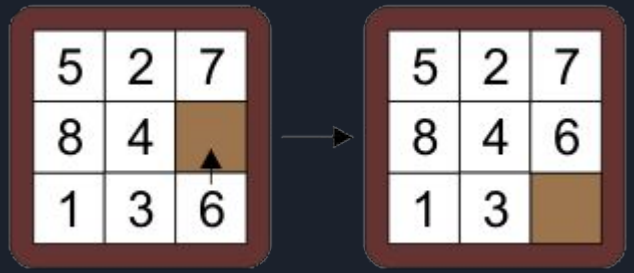
Si la condición se cumple, hemos encontrado una solución al problema planteado.



Reducción a N-Puzzle

Para comenzar a abordar el problema de la definición de funciones de costos y de heurísticas, buscamos soluciones a problemas similares.

El ***N-Puzzle*** era un buen candidato para la reducción del problema de ***Rolling Cubes.*** Si bien no se logró la reducción, las heurísticas utilizadas para este juego sirvieron de inspiración para el nuestro.



N-Puzzle

Algoritmos de búsqueda

Algoritmos no informados

- BFS
- DFS
- Iterative Deepening

Algoritmos informados

- A*
- Greedy search

Función de costos

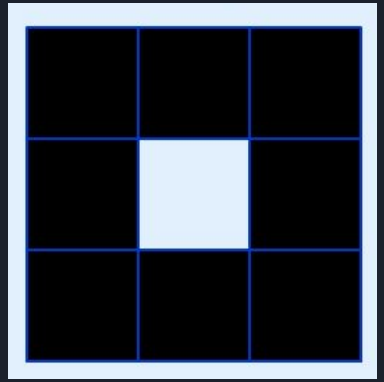
Se utilizará como ***función de costos*** una función que incrementará en una unidad con cada regla aplicada. Es decir, utilizando los conceptos del juego ***Rolling Cubes***, la cantidad de movimientos.

Heurísticas: *Cubos negros*

Surge a través de la idea de representar al objetivo del juego como la minimización de la cantidad de piezas con cara negra en el tablero.

Se define la ***heurística*** como:

h(n) *= 2 × ∑ cubos con cara negra + ∑ cubos con media cara negra*



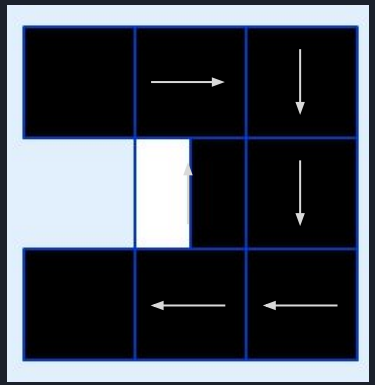
La ***heurística*** es bastante básica y no brinda mucha información sobre el juego, por lo que no es muy efectiva para los métodos de búsqueda informados.

Esta heurística ***es admisible***

Heurísticas: *Mejorada*

Si bien la última heurística es buena, demora demasiado tiempo y es por lo tanto muy ineficiente por ser demasiado aproximativa.

Para este heurística se tuvo en cuenta un conjunto de casos especiales. Por ejemplo, si tenemos un cubo en un borde que no puede ser rotado a blanco en un solo paso, notamos que la cantidad mínima de movimientos era de 12 aplicaciones de reglas, y desplazandolo 4 veces. Repitiendo este mecanismo para cada cubo, y teniendo en cuenta el color y la posición de cada cubo, tomamos el valor máximo como aproximación.



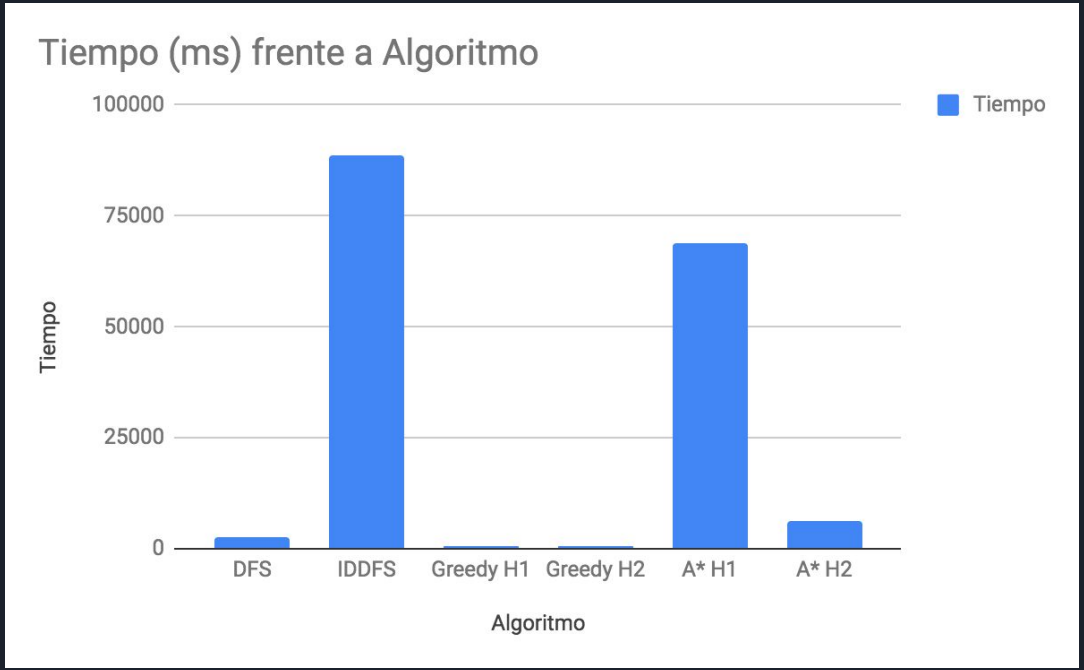
La ***heurística*** representa mucho mejor el estado del juego, por lo que se esperan buenos resultados. Esta heurística ***es admisible***.

Resultados

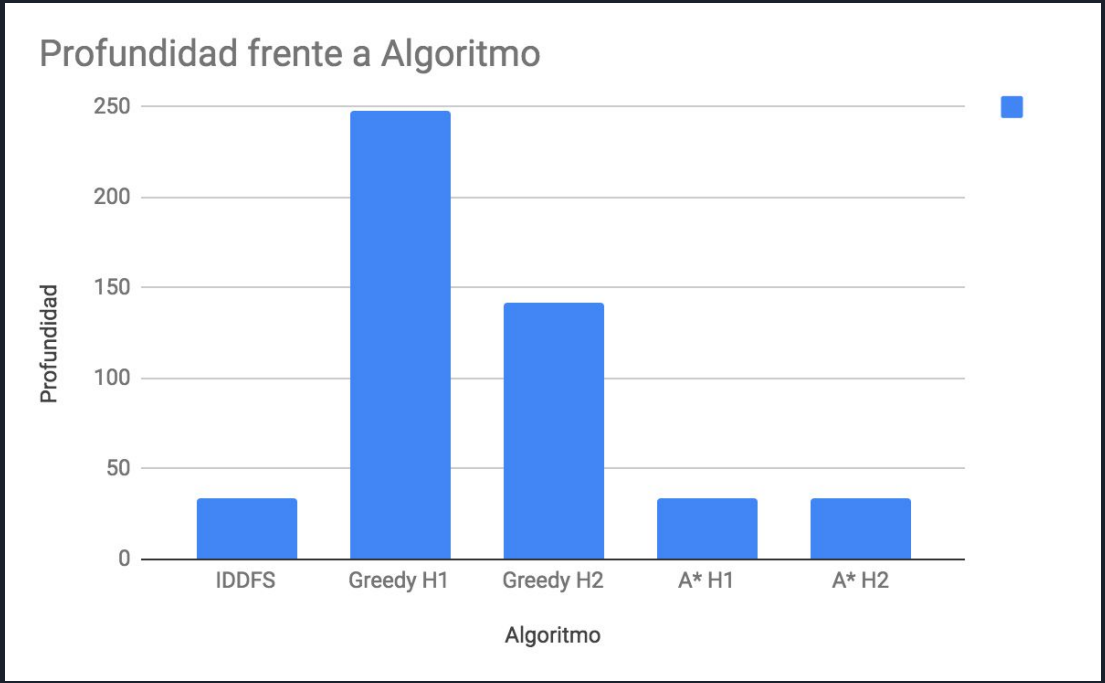
La siguiente tabla resume los resultados obtenidos de la ejecución de los diferentes métodos de búsqueda con las diferentes heurísticas.

Algoritmo	Profundidad	Estados generados	Nodos frontera	Nodos expandidos	Tiempo
DFS	74789	135837	60317	75520	2649 ms
BFS	31	2225094	358289	1866805	1 hs
IDDFS	34	1537503	19	1537485	88784 ms
Greedy H1	111261	192233	77772	114461	2801 ms
Greedy H2	14630	24492	10395	14097	340 ms
A* H1	34	787094	250168	536926	64690 ms
A* H2	34	137919	49628	88291	6388 ms

Resultados


Tiempo (ms) frente a Algoritmo
Tiempo
100000
75000
50000
25000
0
Tiempo
DFS
IDDFS
Greedy H1
Greedy H2
A* H1
A* H2
Algoritmo

Resultados


Profundidad frente a Algoritmo
250
200
150
100
50
0
Profundidad
IDDFS
Greedy H1
Greedy H2
A* H1
A* H2
Algoritmo

Conclusiones

01 **La representación del problema** es muy importante para el ahorro de memoria. La explosión combinatoria de la cantidad de estados en el ***game tree*** fuerzan a la optimización de la representación con respecto al espacio. Pudimos haber utilizado una representación con “Versionado de cambios”.

02 **Las diferentes heurísticas** brindan mayor o menor información sobre el problema. Es muy relevante encontrar buenas heurísticas para poder alcanzar una buena solución.

03 **Los algoritmos de búsqueda no informados** no dan buenos resultados ya que exploran una gran cantidad de nodos. Los **informados** son mucho más eficientes a este respecto. A* tarda mucho más ya que busca el camino óptimo.

04 **El juego *Rolling Cubes*** presenta dificultades a la hora de generar buenas heurísticas.

¡Gracias!