Inteligencia Artificial 2020

Informe Actividad Práctica B1: Búsqueda en espacio de estados

Facultad de Ingeniería

Ingeniería en Informática

Alumno: Iván Guerra

Profesor: Juan Francisco Giró

Índice

Introducción

Objetivos

Desarrollar e implementar una aplicación destinada a la solución de problemas en el espacio de estados. Demostrar su buen funcionamiento.

Consignas

La aplicación debe estar orientada a resolver puzles de 9 piezas y opcionalmente puede incluir otros problemas (a criterio del autor).

Requerimientos

- Facilidad para definir el nodo inicial y el nodo objetivo
- Demostrar el buen funcionamiento de la aplicación desarrollada
- Demostrar que el desarrollo es propio.
- Breve informe describiendo el trabajo realizado, los resultados obtenidos y dificultades encontradas.

Descripción del problema

El problema que se busca resolver es un problema de búsqueda en un espacio de estados, donde se tiene un puzzle de 9 piezas (ordenado en forma de grilla de 3x3), que está configurado con un **estado inicial "I"** (ver Figura 1), y se busca a partir de un conjunto finito de operaciones o **movimientos posibles "O"** llegar a un estado final o **estado objetivo (M)**.

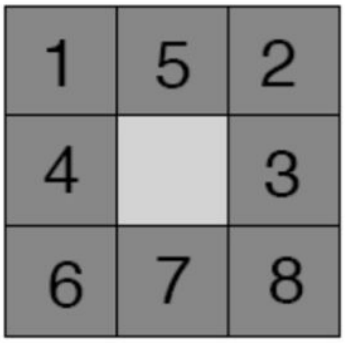


Figura 1. Estado inicial "I"

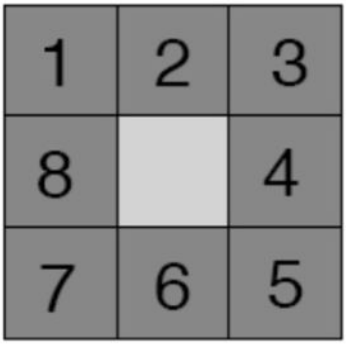


Figura 2. Estado objetivo "O"

Métodos heurísticos de resolución

Estos métodos buscan, a partir del conocimiento sobre el problema que se va a resolver, achicar la búsqueda a solo una porción reducida del espacio de estados posibles, y así poder reducir la complejidad del problema bajando la cantidad de nodos a ser explorados en la búsqueda de la solución o estado objetivo.

Para resolver un problema por algún método heurístico, se necesita expresar el mismo en término de dos parámetros:

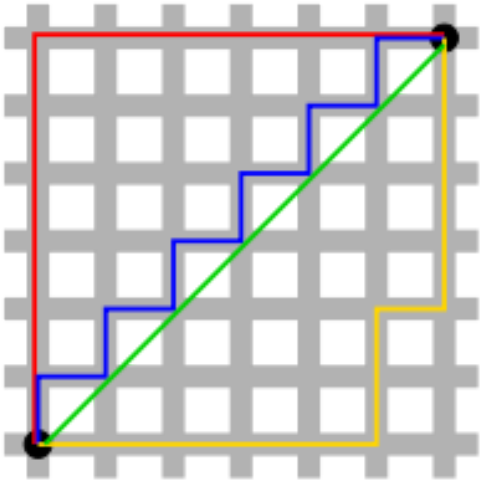
- **g(n):** Costo del camino ya recorrido hasta al nodo "n". En este caso estará representado por el **nivel de profundidad del nodo**

- **h(n):** Distancia del nodo "n" al estado objetivo. En este caso, estará representada por la **distancia Manhattan.**

Distancia manhattan

En esta sección se explicará el concepto de distancia manhattan y su diferencia con la distancia euclídea.

- Si se considera el siguiente diagrama que representa una ciudad como Nueva York, formada en gran medida por una matriz de edificios, en el que hay dos puntos (en negro) cuya distancia queremos calcular:



- Aunque la distancia euclídea sea la correspondiente a la línea verde, los recuadros blancos son edificios, por lo que estamos obligados a bordearlos.

- Por lo tanto el camino rojo como el azul y el amarillo son equivalentes y representan la **distancia Manhattan** mínima entre los puntos.

Listas abierta y cerrada

En esta sección, explicaremos el significado de las listas abierta y cerrada en la búsqueda del estado objetivo.

- Inicialmente la **lista abierta** contendrá el estado inicial y la **lista cerrada** estará vacía.

Paso #	Lista abierta	Lista cerrada
1	uuid: d3bfd755-c2d3-4416-a486-179e2d71c546 predec: - f: 5 [2,8,3,1,6,4,7,-1,5]	-

- Sucesivamente, y hasta encontrar el estado objetivo o que se dé una condición de error, las listas se actualizarán de la siguiente manera:

    ➢ En el paso “i”, se retira de la lista abierta el estado **“m”** con el **menor valor** de la función heurística “f” del paso anterior, y se introduce el mismo en la lista cerrada.

Paso #	Lista abierta	Lista cerrada
1	uuid: d3bfd755-c2d3-4416-a486-179e2d71c546 predec: - f: 5 [2,8,3,1,6,4,7,-1,5]	-
2	uuid: aa65528b-9c1e-4ba9-ad79-be9354410967 predec: d3bfd755-c2d3-4416-a486-179e2d71c546 f: 6 [2,8,3,1,6,4,-1,7,5]	uuid: d3bfd755-c2d3-4416-a486-179e2d71c546 predec: - f: 5 [2,8,3,1,6,4,7,-1,5]
	uuid: 7d128833-cbe5-4b41-aff8-1835d20f6b48 predec: d3bfd755-c2d3-4416-a486-179e2d71c546 f: 4 [2,8,3,1,-1,4,7,6,5]	
	uuid: 516e61bb-cc35-41fb-b2a6-7739b6345a90 predec: d3bfd755-c2d3-4416-a486-179e2d71c546 f: 6 [2,8,3,1,6,4,7,5,-1]	

    ➢ Luego, a partir de ese estado “m”, se expande a todos los estados posibles sucesores y se calcula para cada uno su función heurística y se indica su antecesor

Paso #	Lista abierta	Lista cerrada
1	uuid: d3bfd755-c2d3-4416-a486-179e2d71c546 predec: - f: 5 [2,8,3,1,6,4,7,-1,5]	-
2	uuid: aa65528b-9c1e-4ba9-ad79-be9354410967 predec: d3bfd755-c2d3-4416-a486-179e2d71c546 f: 6 [2,8,3,1,6,4,-1,7,5]	uuid: d3bfd755-c2d3-4416-a486-179e2d71c546 predec: - f: 5 [2,8,3,1,6,4,7,-1,5]
	uuid: 7d128833-cbe5-4b41-aff8-1835d20f6b48 predec: d3bfd755-c2d3-4416-a486-179e2d71c546 f: 4 [2,8,3,1,-1,4,7,6,5]	
	uuid: 516e61bb-cc35-41fb-b2a6-7739b6345a90 predec: d3bfd755-c2d3-4416-a486-179e2d71c546 f: 6 [2,8,3,1,6,4,7,5,-1]	

➢ Si algún sucesor de “m” es el estado objetivo, se abandona el proceso iterativo y se devuelve como solución el camino que se obtiene recorriendo los punteros de los antecesores desde el nodo objetivo al inicial y se da por concluido el proceso de búsqueda.

Paso #	Lista abierta	Lista cerrada
6	uuid: aa65528b-9c1e-4ba9-ad79-be9354410967 predec: d3bfd755-c2d3-4416-a486-179e2d71c546 f: 6 [2,8,3,1,6,4,-1,7,5]	uuid: d3bfd755-c2d3-4416-a486-179e2d71c546 predec: - f: 5 [2,8,3,1,6,4,7,-1,5]
	uuid: 38617e74-3f2c-4932-9988-a414f41a39c0 predec: 1e6aee4c-e839-4d2c-a4ea-8d1353da783f f: 4 [2,3,-1,1,8,4,7,6,5]	uuid: ab97a483-cdc9-43b4-b955-14d042b065c6 predec: d4efd61f-bc76-4482-bd56-b57d9ec15c8e f: 1 [1,2,3,-1,8,4,7,6,5]
	uuid: 54b26de5-020c-49f2-b2dd-ca9110a9fe84 predec: ab97a483-cdc9-43b4-b955-14d042b065c6 f: 0 [1,2,3,8,-1,4,7,6,5] => OBJETIVO	

Resolución del problema

Métodos heurísticos a utilizar

En la solución del problema se utilizarán dos métodos heurísticos, que se explicaran en forma sintética a continuación:

- Primero el mejor

En este método, el estado seleccionado para continuar es aquél que tiene el menor valor de “h”, siendo “h” el valor de la **distancia Manhattan.**

Es decir que conduce la búsqueda según la proximidad a la meta.

- A*

En este método, el estado seleccionado para continuar es aquél que tiene el menor valor de “g” + “h”, siendo “g” el **nivel de profundidad del nodo** y “h” el valor de la **distancia Manhattan.**

Es decir que considera la proximidad a la meta y también el costo del camino ya recorrido.

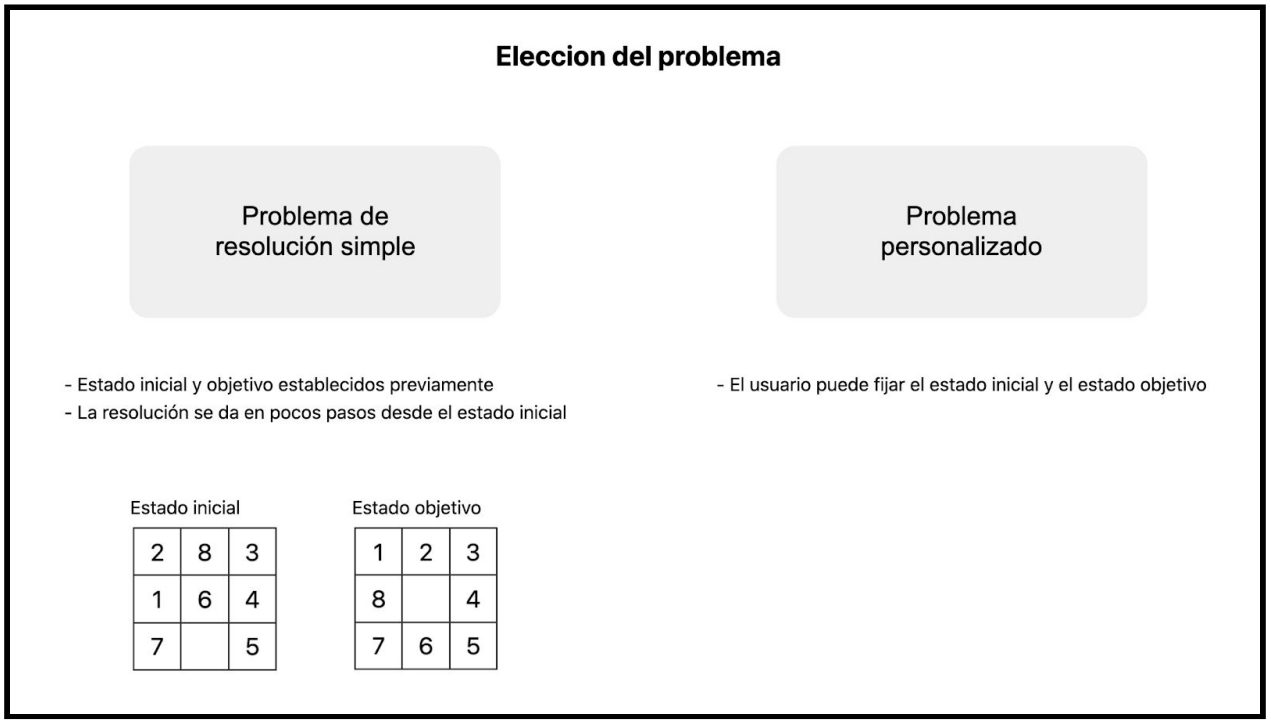
Tecnologías utilizadas

Para la solución del problema, se decidió utilizar la interfaz del navegador para representar el problema en una forma visual

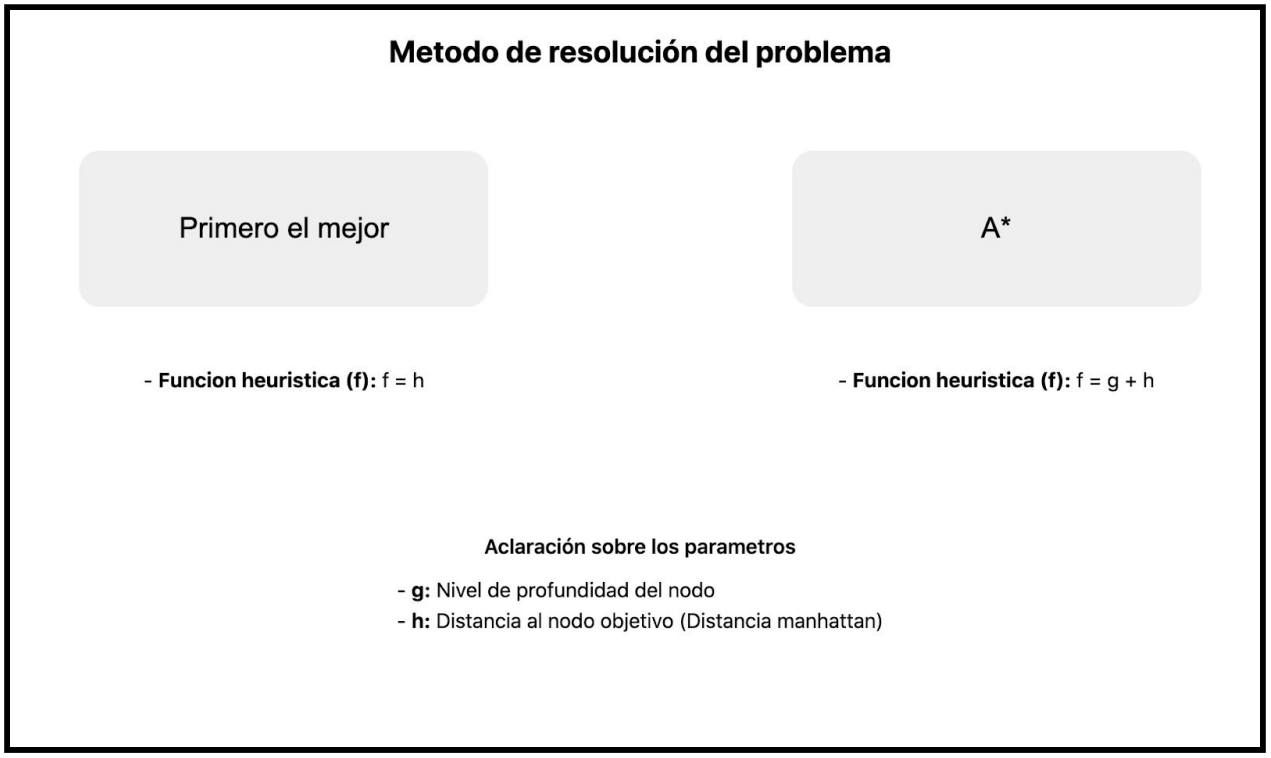
Por lo tanto, se utilizó como lenguaje de programación Javascript y además, para la maquetación de la página el lenguaje HTML y CSS para los estilos.

Pasos para realizar una búsqueda

1) Al iniciar el programa, se podrá elegir entre resolver un **problema simple**, donde el estado inicial y objetivo están establecidos previamente y un **problema personalizado**, donde el usuario puede fijar el estado inicial y el estado objetivo.

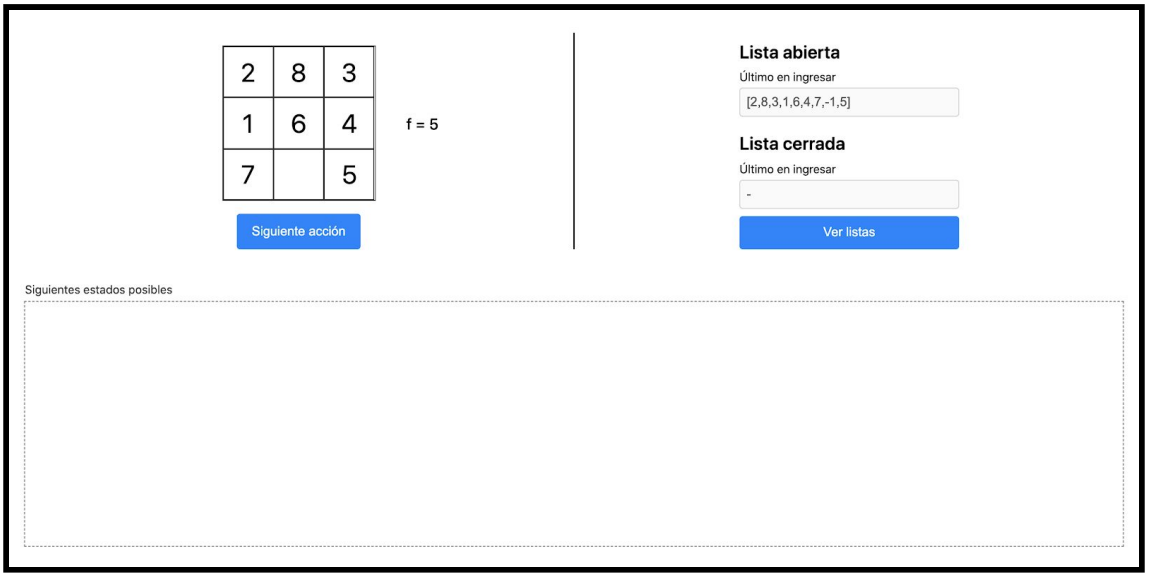


2) Luego, se debe elegir el metodo de resolucion del problema: **Primero el mejor** o **A***

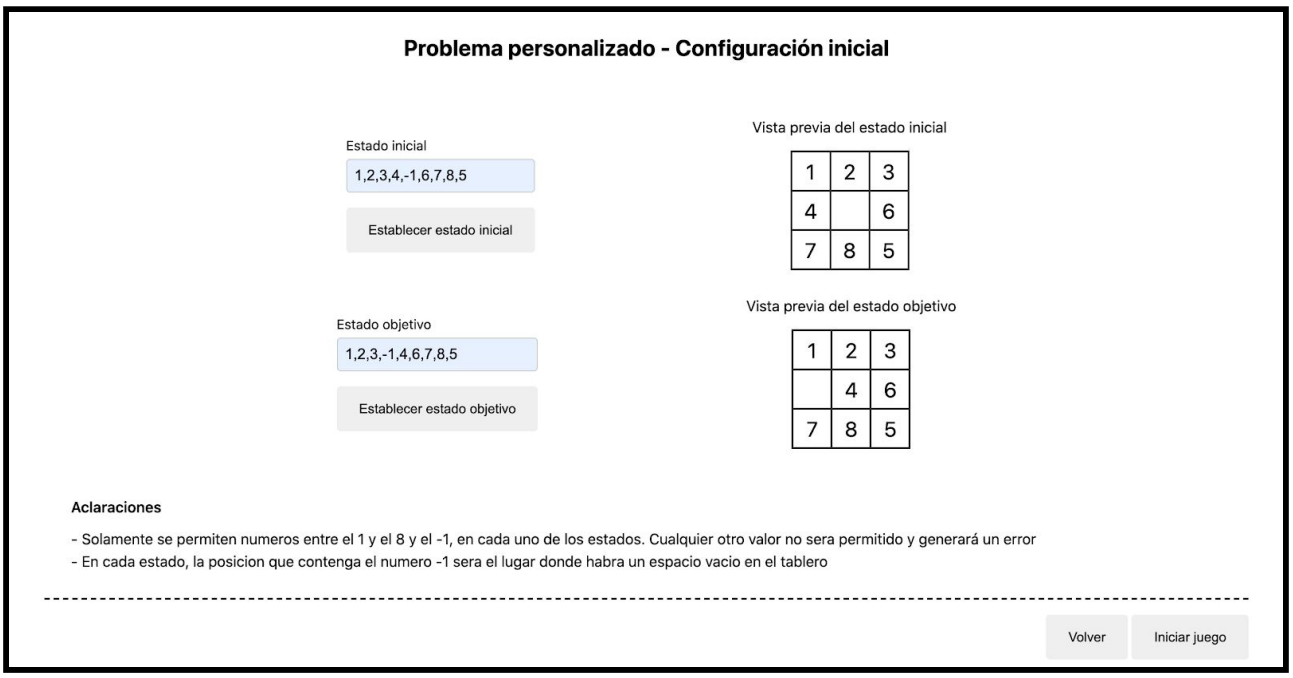


3) Paso siguiente, si se eligio la resolución del problema simple, se mostrará la interfaz de inicio del juego **(a)** ó, si se eligio la resolución del problema personalizado se podrá indicar el estado de inicio y estado objetivo y luego iniciar el juego **(b)**:

   a)

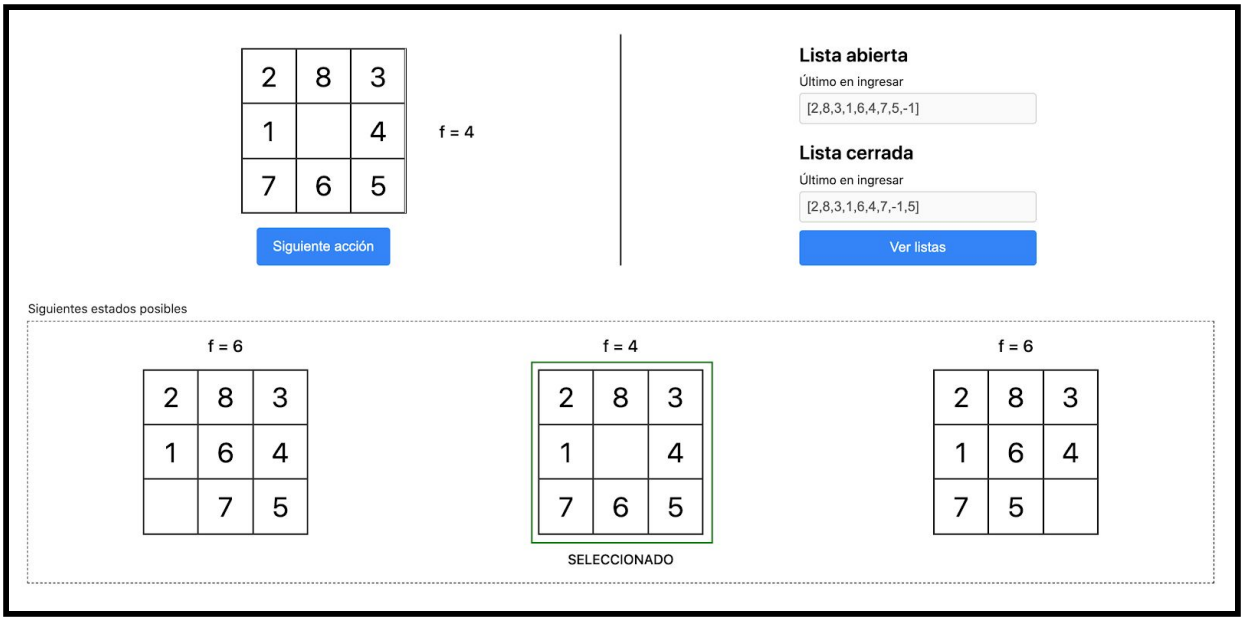


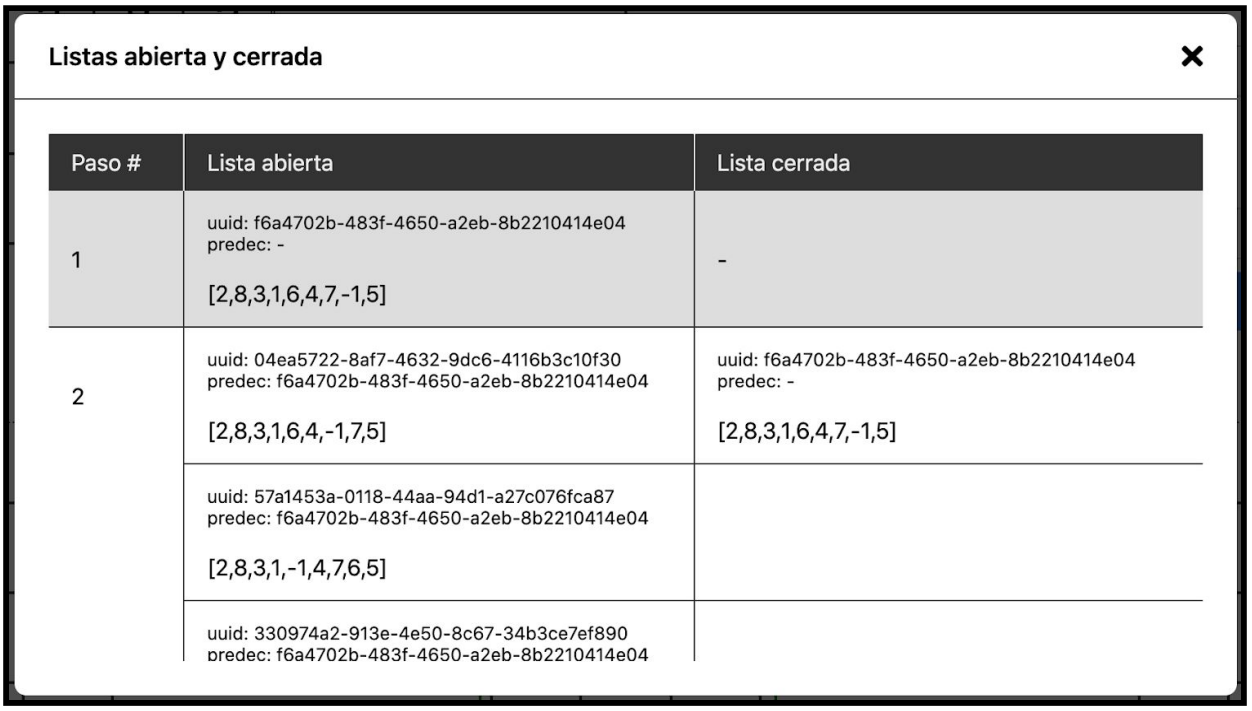
b)



4) Una vez comenzado el juego, se puede ver:

   a) En la parte **superior izquierda,** el tablero con el estado actual y al lado el valor de la función heurística para ese estado.
   b) En la parte **superior derecha**, se pueden ver dos inputs de tipo texto con los últimos elementos en ingresar a las listas abiertas y cerradas.
   Si se hace click en el botón “Ver listas”, se puede ver el contenido de cada lista.
   c) En la parte **inferior**, se puede ver los siguientes estados posibles donde se puede avanzar, donde cada uno tiene indicado su valor de función heurística.



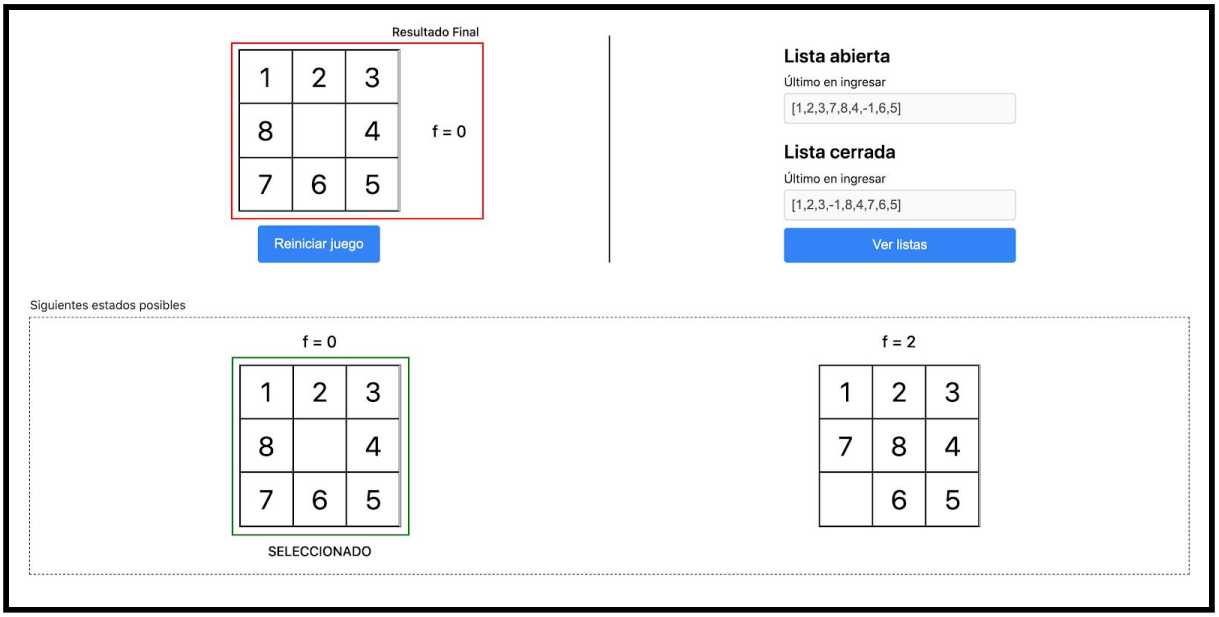


Listas abierta y cerrada

Paso #	Lista abierta	Lista cerrada
1	uuid: f6a4702b-483f-4650-a2eb-8b2210414e04 predec: - [2,8,3,1,6,4,7,-1,5]	-
2	uuid: 04ea5722-8af7-4632-9dc6-4116b3c10f30 predec: f6a4702b-483f-4650-a2eb-8b2210414e04 [2,8,3,1,6,4,-1,7,5]	uuid: f6a4702b-483f-4650-a2eb-8b2210414e04 predec: - [2,8,3,1,6,4,7,-1,5]
	uuid: 57a1453a-0118-44aa-94d1-a27c076fca87 predec: f6a4702b-483f-4650-a2eb-8b2210414e04 [2,8,3,1,-1,4,7,6,5]	
	uuid: 330974a2-913e-4e50-8c67-34b3ce7ef890 predec: f6a4702b-483f-4650-a2eb-8b2210414e04	

5) En este momento, ya se puede comenzar a jugar, presionando el botón **"Siguiente acción"** y observando el movimiento de las piezas del tablero hasta llegar al estado objetivo.

   También, se irán actualizando las listas "abierta" y "cerrada" con los diferentes movimientos.



Ejecución del proyecto terminado

Para la ejecución del proyecto, dirigirse al repositorio del proyecto indicado al final de este informe en este informe y seguir las instrucciones detalladas en el archivo README.md.

Repositorio del proyecto terminado

- **Enlace al repositorio del proyecto terminado:**

  https://github.com/iguerra94/NinePiecesPuzzle