****Ejercicio de Laboratorio 4. BFS, A\* y SA****

# Ejercicio 1. BFS o A\*

Empleando búsqueda informada (coste uniforme, A\* o una variante) resuelve el problema del 8-puzzle.

## ¿Qué algoritmo utilizamos?

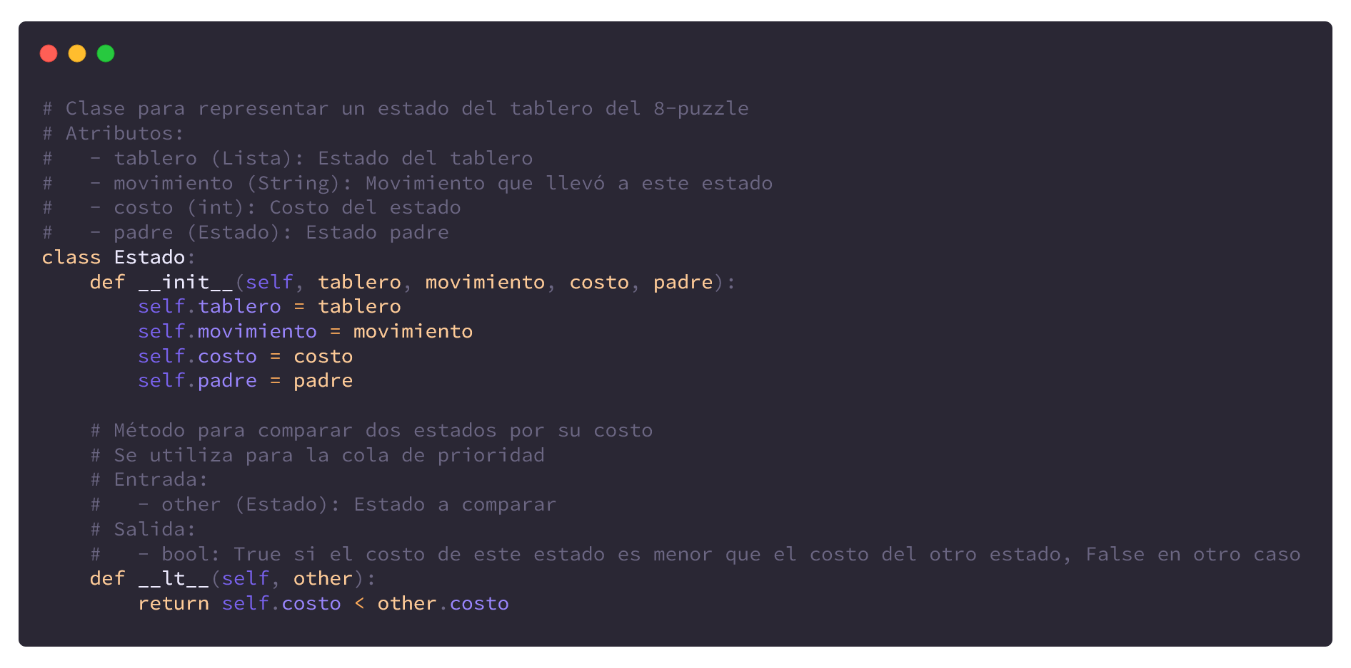
El algoritmo A\* es un algoritmo de búsqueda informada que utiliza una función heurística para encontrar el camino más corto desde un nodo inicial hasta un nodo objetivo. En el caso del problema del 8-puzzle, el algoritmo A\* puede utilizarse para encontrar la secuencia de movimientos necesarios para llegar desde una disposición inicial del tablero hasta una disposición objetivo.

## ¿Qué heurística utilizamos?

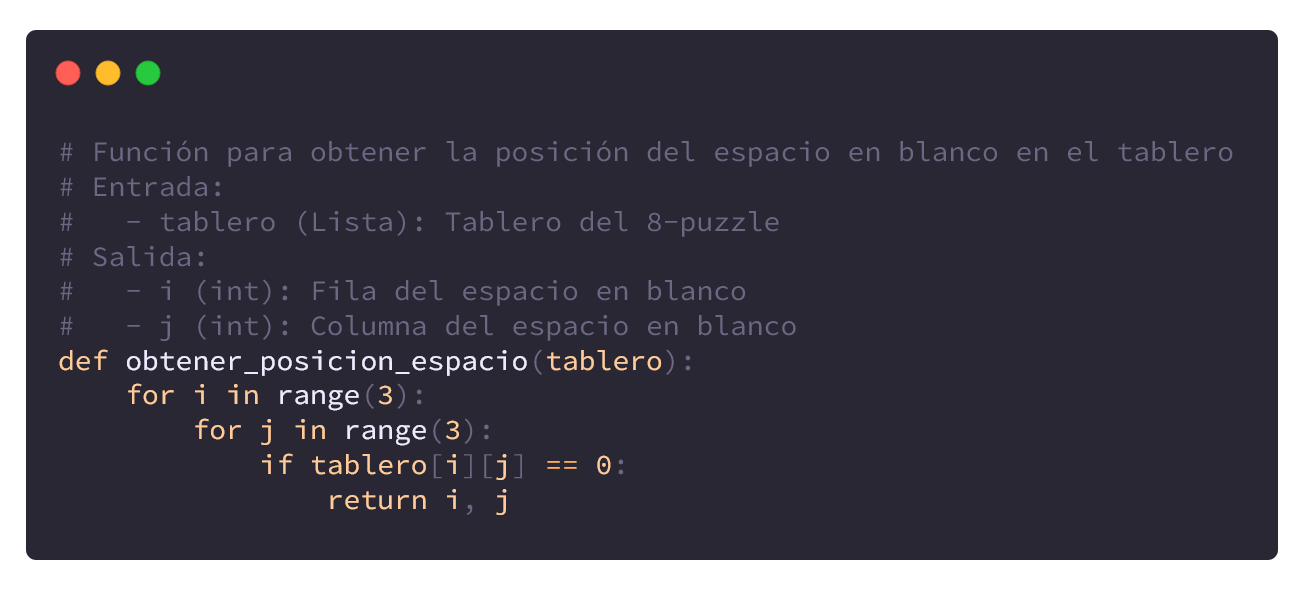
Se utiliza la heurística de la distancia Manhattan para calcular el costo estimado desde un estado del tablero hasta el estado objetivo. El algoritmo A\* combina este costo estimado con el costo acumulado de los movimientos realizados para tomar decisiones informadas sobre qué nodos explorar primero en la búsqueda.

## Código

### Clase de estado del tablero



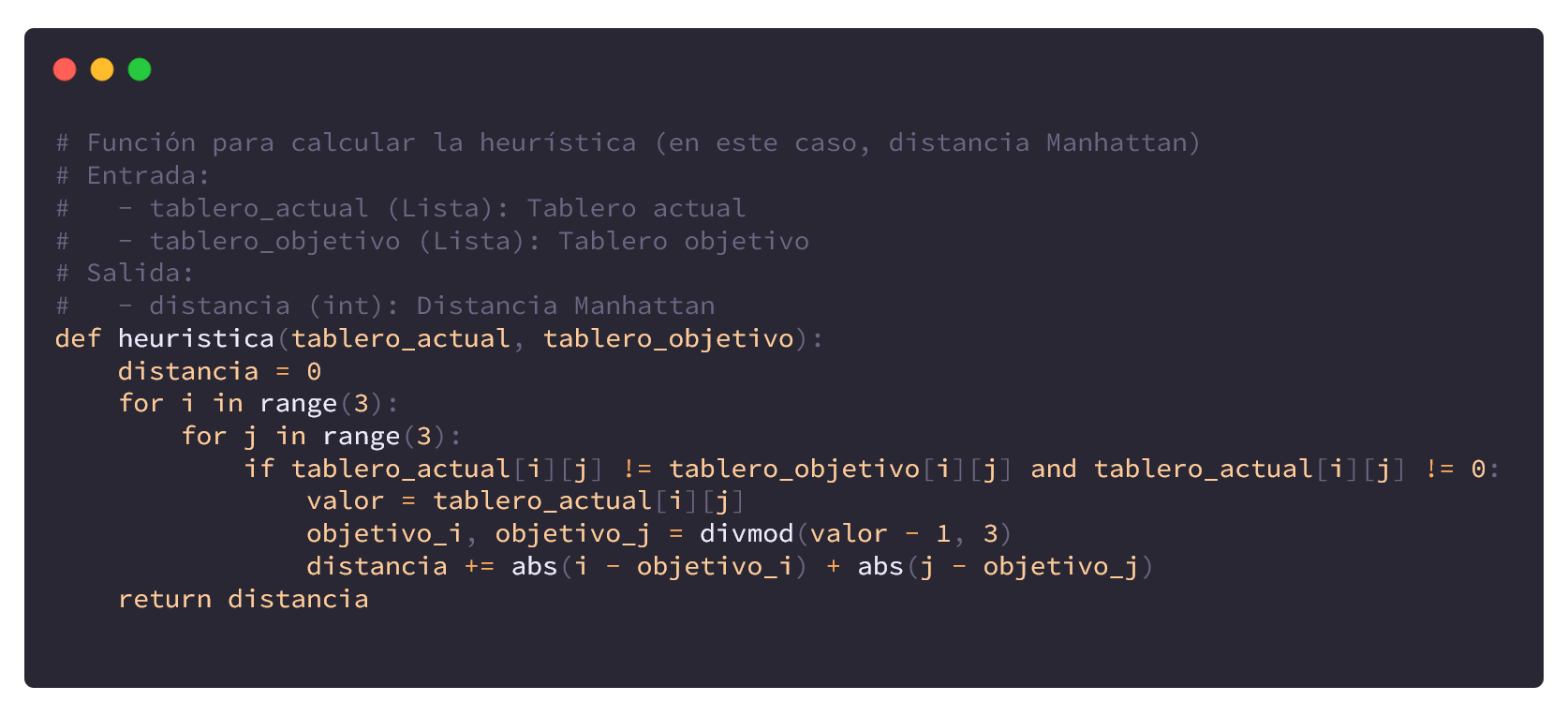
### Función obtener\_posicion\_espacio(tablero)

****

### Función generar\_movimientos(estado)

****

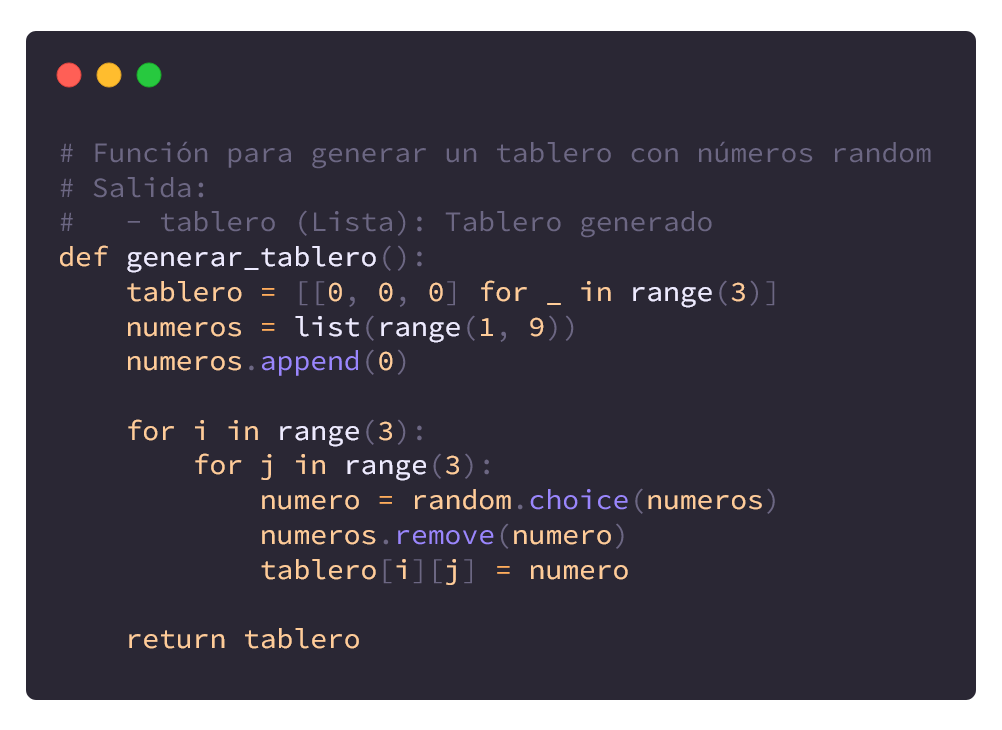
### Función heuristica(tablero\_actual, tablero\_objetivo)



### Función resolver\_8\_puzzle(inicial, objetivo)

****

### Función generar\_tablero()



### Programa Principal



## Salida

### Si no hay solución

En el problema del 8-puzzle, específicamente en el contexto de la búsqueda de soluciones utilizando algoritmos como A\* o BFS, puede haber situaciones en las que no exista una solución válida. Estas situaciones se deben a las restricciones del problema y la configuración inicial del tablero. Algunas son inversión de paridad, bloqueo de espacio, configuración no conectada al tablero objetivo o estado inalcanzable.

Texto

Descripción generada automáticamente

### Si existe solución

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente

El algoritmo comienza explorando el estado inicial y expandiendo los movimientos posibles desde ese estado. Luego, evalúa la heurística para cada uno de estos estados expandidos y los agrega a una cola de prioridad. En cada iteración, el algoritmo selecciona el estado con el menor costo total estimado (suma del costo acumulado y la heurística) para explorar a continuación.

Así, el algoritmo A\* continúa explorando y expandiendo estados hasta encontrar el estado objetivo, garantizando que el camino encontrado sea óptimo en términos de la cantidad de movimientos necesarios para llegar al estado objetivo desde el estado inicial.

# Ejercicio 2. Simulated Annealing

Empleando templado simulado (simulated annealing) encuentra el valor mínimo de las siguientes funciones:

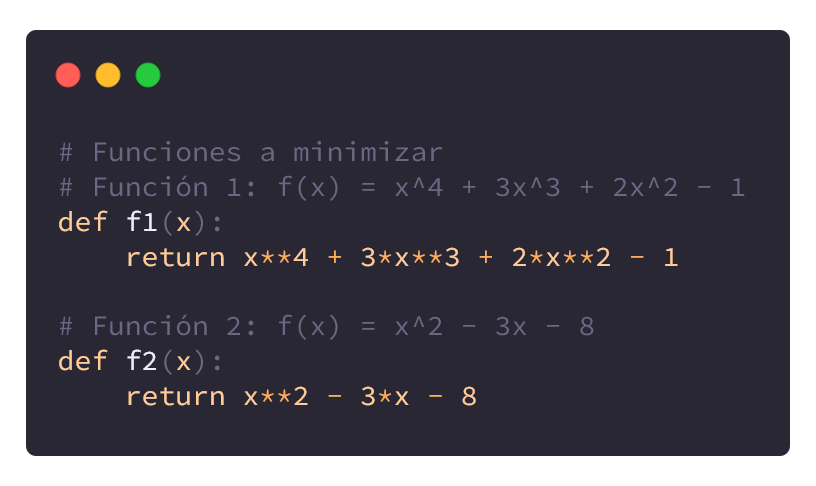
## Solución

Para encontrar el valor mínimo de las funciones dadas utilizando el método de templado simulado (simulated annealing), necesitamos seguir estos pasos:

1. Definir la función objetivo .
2. Inicializar un punto aleatorio como punto de partida.
3. Definir el rango de valores de en el que se buscara el mínimo.
4. Definir la función de probabilidad para aceptar un nuevo punto basado en la diferencia de valores de la función objetivo.
5. Implementar el algoritmo de templado simulado con iteraciones y un esquema para reducir la “temperatura”.

## Código

### Funciones para minimizar



### Función simulated\_annealing(función, temperatura\_inicial, temperatura\_final, alpha, iteraciones\_por\_temp)

****

### Programa Principal

## Salida

Texto

Descripción generada automáticamente