

## Segundo proyecto

El segundo parcial consiste en implementar el algoritmo A\* de búsqueda informada. El proyecto se puede presentar en C/C++, JAVA, Python trabajando de manera individual o en equipo (de 2 a 4 integrantes). En las siguientes secciones se explican los elementos a implementar:

### Codificación y análisis

Implementar un sistema inteligente capaz de resolver el problema del rompecabezas de 4x4, utilizando las estrategias de búsqueda analizadas en clase. Para ello, tomar en cuenta lo siguiente:

#### Entrada

El tablero de entrada se especifica a partir de un archivo llamado "Datos.txt". El formato del archivo estará dado de la siguiente forma:

- Las 4 primeras líneas con el estado inicial
- Las siguientes 4 líneas con el estado final
- Todos los elementos separados por coma, excepto los últimos que será con un salto de línea (enter)
- El "espacio" será reemplazado por un 0 (cero)

Un ejemplo de archivo de entrada de tableros es el siguiente:

```
1,2,4,7
11,6,12,3
5,10,14,8
9,0,13,15
1,2,3,4
5,6,7,8
9,10,11,12
13,14,15,0
```

#### Salida

La secuencia de movimientos que debe realizar la pieza "espacio en blanco", basados en la siguiente codificación: Up (U), Down (D), Left (L), Right (R). Los caracteres deben de aparecer separados por "," (comas). LA SALIDA DEBERÁ DE MANDARSE A CONSOLA, por ejemplo:

U,L

Si lo consideras conveniente, puedes ir imprimiendo algunos pasos previos, pero asegúrate de al final, agregar el formato correcto

#### Función

Definir dos funciones de evaluación: La realizada en clase y proponer una nueva, con el objetivo de mejorar los resultados. Para la función diseñada, mostrar con un ejemplo como se calcularía un estado actual a partir de una meta previamente definida. Argumentar porqué consideras que la función de evaluación es buena para el proceso de búsqueda.

## Análisis

Presentar un análisis empírico sobre los tiempos de ejecución del algoritmo implementado con ambas funciones, así como un estadístico sobre la relación soluciones encontradas / soluciones no encontradas. (TIP: generar gráficas de tiempo de ejecución, donde se note claramente el comportamiento del algoritmo ante instancias de diferentes tamaños de entrada)

## Contenido del reporte

- Análisis de la función propuesta, con ejemplos de ejecución comparándolos con la función vista en clase
- Análisis empírico de las ejecuciones ambas funciones de evaluación, considerando tiempo, magnitud, soluciones encontradas y no encontradas

## Lista de cotejo

Procedimiento	Puntos
Formato Entradas / Salidas	10
Análisis de la función de evaluación (propuesta)	20
Algoritmo	30
Análisis de tiempos de ejecución	20
Preguntas sobre el problema	20
<b>TOTAL</b>	100