

Trabajo Práctico 1

# Métodos de Busqueda

72.27 – Sistemas de Inteligencia Artificial – 2C 2024

Integrantes del **GRUPO 5**

Kuchukhidze, **Giorgi** – 67262

Madero Torres, Eduardo **Federico** – 59494

Ramos Marca, María **Virginia** – 67200

Plüss, **Ramiro** – 66254

# CONTENIDO



## 8-Puzzle

Ejercicio 1,  
explicación y resolución



Ejercicio 2,  
consideraciones y  
muestras de tableros



## Métodos

Algoritmos de búsqueda y  
heurísticas



## Sokoban

## Resultados y análisis

Comparación de  
algoritmos con variables  
consideradas



## Conclusiones

Hallazgos del análisis  
comparativo

# 8-PUZZLE

Se cuenta con el ejercicio “**8-puzzle**” donde se parte de un **tablero inicial al azar** y, moviendo los números adyacentes al espacio vacío, se busca llegar al **tablero solución**.

- Estructura de estados.
- Dos heurísticas admisibles no triviales.
- Métodos de búsqueda y con qué heurística.

# 8-PUZZLE

- Problema de búsqueda en un espacio de estados.
- Estados ---> Configuración del tablero.
- Acciones posibles ---> Movimientos ortogonales.

$$m_0 = \begin{bmatrix} 5 & 7 & 3 \\ 8 & 2 & 0 \\ 1 & 6 & 4 \end{bmatrix}$$

$$m_{sol} = \begin{bmatrix} 1 & 2 & 3 \\ 8 & 0 & 4 \\ 7 & 6 & 5 \end{bmatrix}$$

# 8-PUZZLE

## Heurísticas admisibles no triviales

### Distancia Manhattan

$$h(n) = |x(n) - x_{\text{objetivo}}| + |y(n) - y_{\text{objetivo}}|$$

*Es la distancia ortogonal de una pieza desde su posición actual hasta su posición objetivo.*

# 8-PUZZLE

## Heurísticas admisibles no triviales

### Distancia Manhattan

Ejemplo: Una ficha está en la posición  $(0, 0)$  y su objetivo es  $(1, 1)$ . Calculemos su distancia Manhattan.

$$d_{\text{Manhattan}} = |1 - 0| + |1 - 0| = 1 + 1 = 2$$

Para mover la ficha desde  $(0, 0)$  hasta  $(1, 1)$ , necesitas 2 movimientos, lo que coincide exactamente con la distancia Manhattan calculada.

# 8-PUZZLE

## Heurísticas admisibles no triviales

### Cantidad de fichas mal colocadas

$$h(n) = \sum_{i=1}^8 \delta(x_i(n), x_i^{\text{objetivo}})$$

Toma en cuenta la suma de todas las fichas mal colocadas ofreciendo una estimación mínima del número de movimientos requeridos.

# 8-PUZZLE

## Heurísticas admisibles no triviales

### Cantidad de fichas mal colocadas

*Ejemplo: Si **3** fichas están mal colocadas provocaría que,*

$$h(n) = 3$$

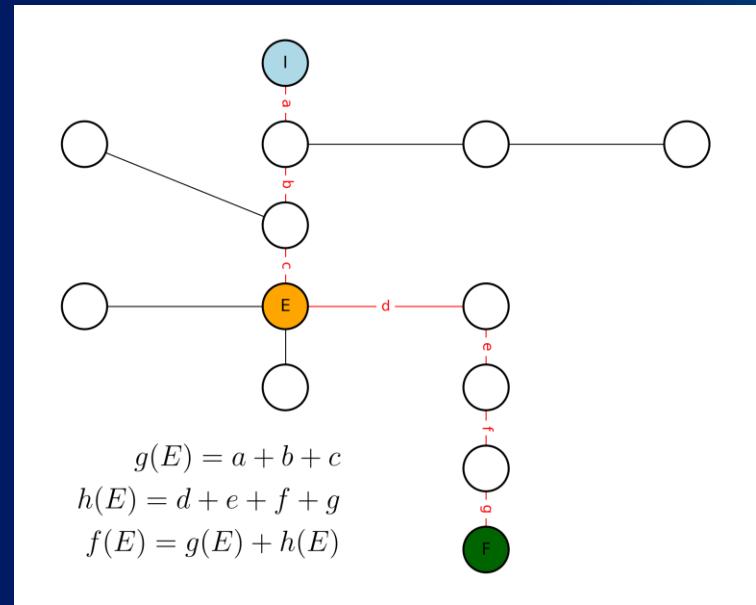
Indicando que se necesitan al menos **3** movimientos para corregir esas fichas. En realidad, podría requerir más movimientos debido a la necesidad de reubicar otras fichas, pero **nunca menos**.

# 8-PUZZLE

## Métodos de Búsqueda

**Método A\***: Este es un algoritmo de búsqueda informada que utiliza una combinación de la función de costo acumulado  $g(n)$  y la heurística  $h(n)$ . Es óptimo y completo cuando se usa con una heurística admisible.

$$f(n) = g(n) + h(n)$$

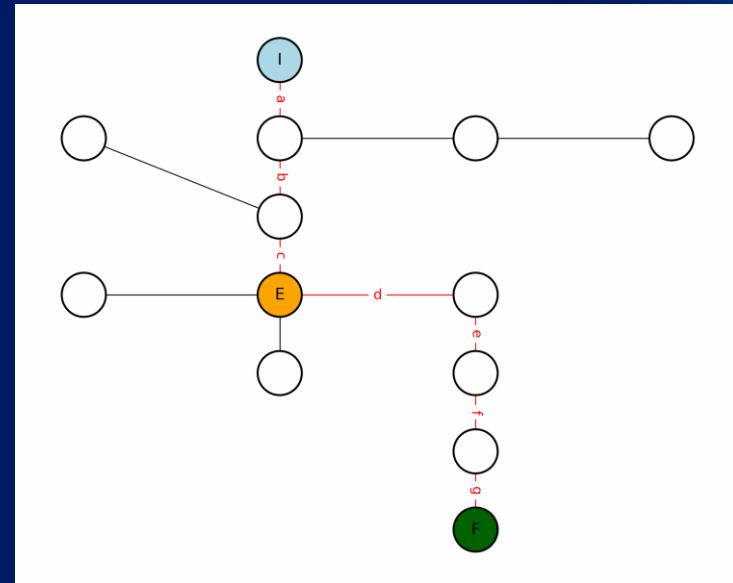


# 8-PUZZLE

## Métodos de Búsqueda

**Método Local Greedy Search:** Este método elige expandir el nodo que parece más cercano al objetivo según la heurística  $h(n)$ . Es más rápido que **A\*** pero no garantiza encontrar la solución óptima.

$$f(n) = h(n)$$



# SOKOBAN

Sokoban es un juego en donde el jugador empuja cajas en un almacén hacia **ubicaciones designadas**.

Las reglas básicas son:

- El jugador solo puede empujar, no tirar.
- Solo se puede empujar una caja a la vez.
- El objetivo es colocar todas las cajas en las **posiciones objetivo**.



# SOKOBAN

Deadlocks: Estados del juego desde los cuales es imposible alcanzar la solución.

Pueden ser:

- Simples: Una caja en una esquina.
- Complejos: Configuraciones de múltiples cajas que bloquean mutuamente el paso.

Espacio de búsqueda enorme: Aproximadamente  $10^{98}$  estados posibles en un tablero típico.



**Nuestro Enfoque: Motor de  
Búsqueda de Soluciones**

# ESTRUCTURA DE ESTADO

- Paradigma Objetos
- Se utiliza la clase "State" para representar el estado del juego.
- Atributos del estado:
  - player
  - boxes
  - targets
  - walls
  - width y height: Dimensiones del tablero de juego
  - parent: Referencia al estado padre

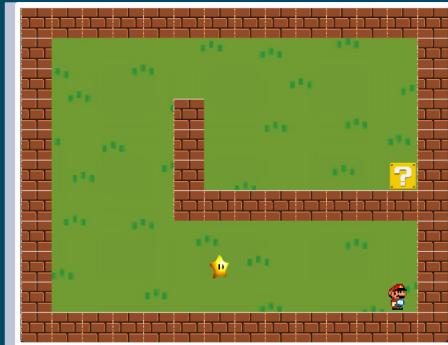
# TABLEROS DEL JUEGO



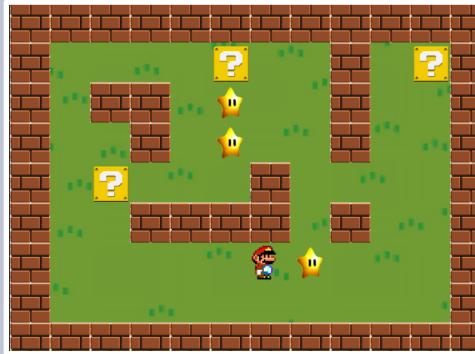
Nivel 1 (Easy)



Nivel 2 (Medium)



Nivel 3 (Difficult)



Jugador



Caja



Goal



Pared



Caja en goal

# ¿CÓMO SE RESUELVE?

Métodos de Búsqueda	
<b>No informados</b>	<b>Informados</b>

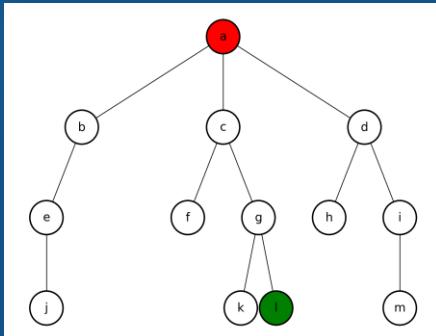
# No informados

- Objetivo
  - Nodo explorado
  - Nodo no explorado
  - Nodo frontera
- r = ramificación  
p = profundidad

## BFS

Óptimo  
Completo

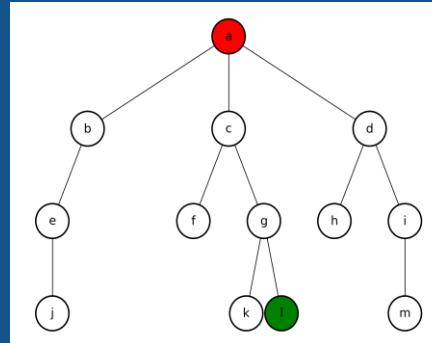
Complejidad t → O( $r^p$ )  
Complejidad s → O( $r^p$ )



## DFS

No óptimo  
Completo

Complejidad t → O( $r^p$ )  
Complejidad s → O( $r^*p$ )



# ¿CÓMO SE RESUELVE?

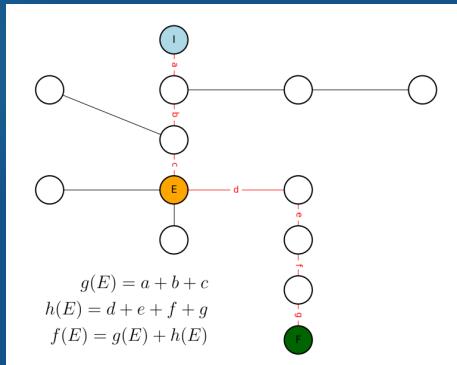
Métodos de Búsqueda	
<b>No informados</b>	<b>Informados</b>

- Objetivo
- Posición de jugador
- Estado inicial
- $r = \text{ramificación}$
- $p = \text{profundidad}$

# Informados

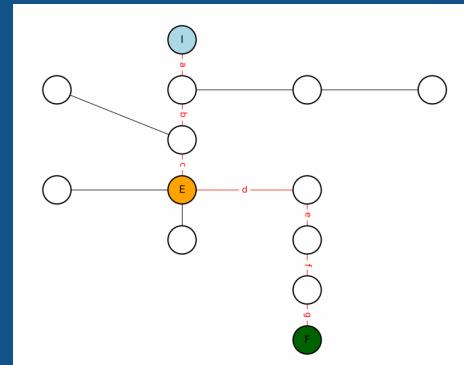
## A\*

Óptimo con  $h(n)$  es admisible  
 Completo con ramificación finita  
 y costo mayor que  $\varepsilon > 0$   
 $O(r^p)$



## GLOBAL GREEDY

No óptimo sin repeticiones de estado  
 Completo dependiendo de  $h(n)$   
 $O(r^p)$

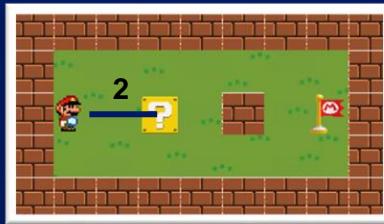


# HEURÍSTICAS

## DISTANCIA MANHATTAN SIN OBSTÁCULOS

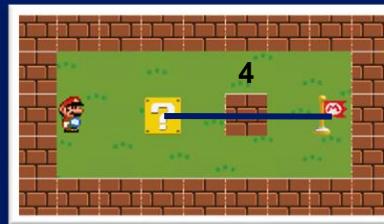
H1

Distancia del jugador a la caja más cercana



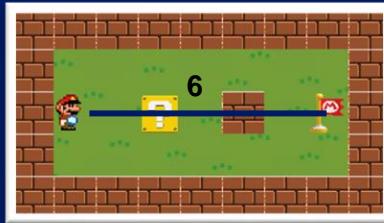
H2

Distancia de la caja a su meta más cercana



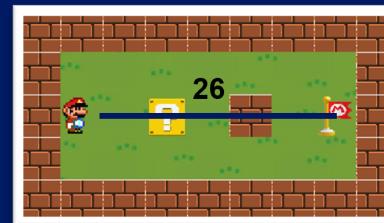
H1 + H2

H3



H3 + penalización de 20 puntos por cercanía con obstáculos

H4



# HEURÍSTICAS

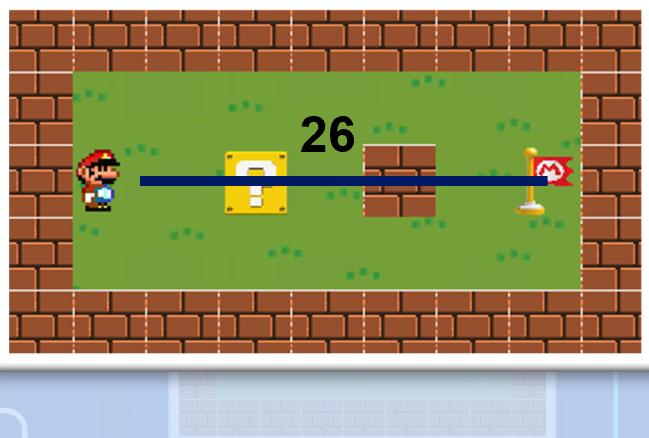
DISTANCIA MANHATTAN SIN OBSTÁCULOS

H1

Distancia del  
jugador a la caja  
más cercana

H2

## NO ADMISIBLE



H1 + H2

H3

H3 + penalización  
de 20 puntos por  
cercanía con  
obstáculos

H4

Distancia de la caja  
a su meta más  
cercana

# MÉTRICAS A ANALIZAR

RESULTADO

NODOS EXPANDIDOS

TIEMPO

01

02

03

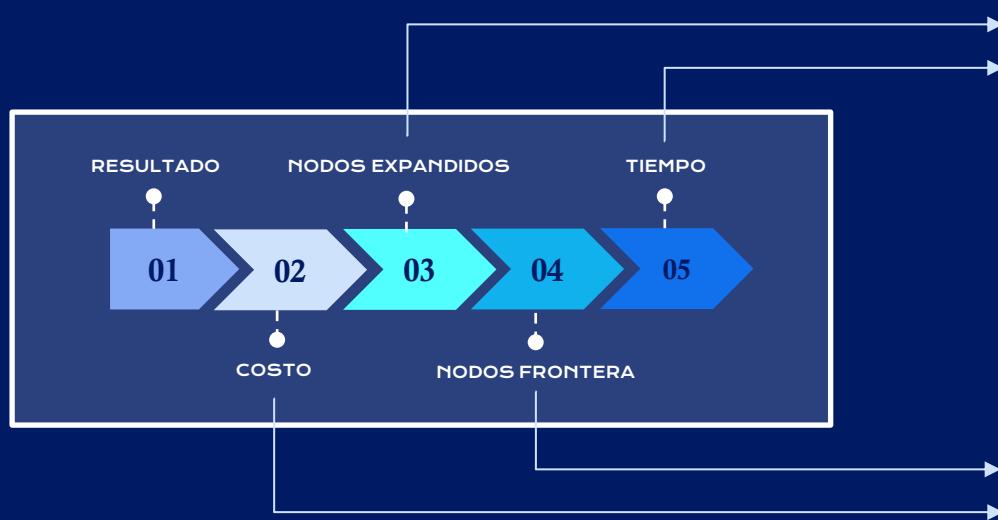
04

05

COSTO

NODOS FRONTERA

# ¿CÓMO SE ANALIZAN?



10 iteraciones con  
cada combinación

Promedio  
y  
variación estándar

# RESULTADOS Y ANÁLISIS

01

02

03

04

05

RESULTADO



Tablero Nivel 1  
(Easy)



EASY

# RESULTADOS Y ANÁLISIS

01

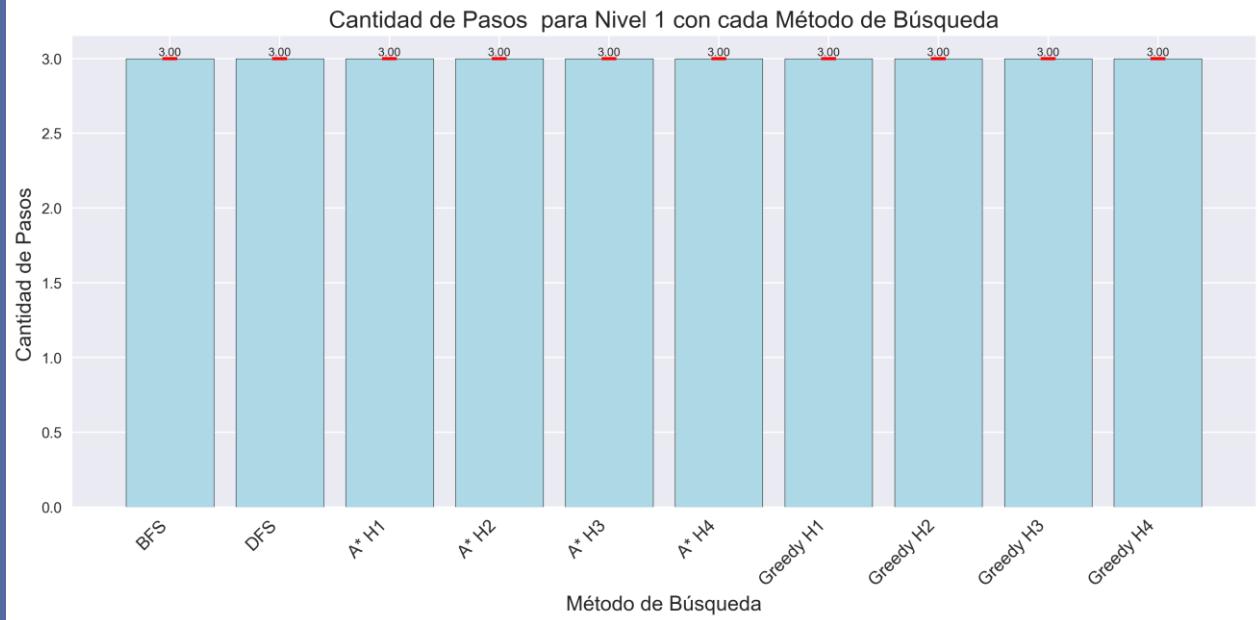
02

03

04

05

COSTO (PASOS)





EASY

# RESULTADOS Y ANÁLISIS

01

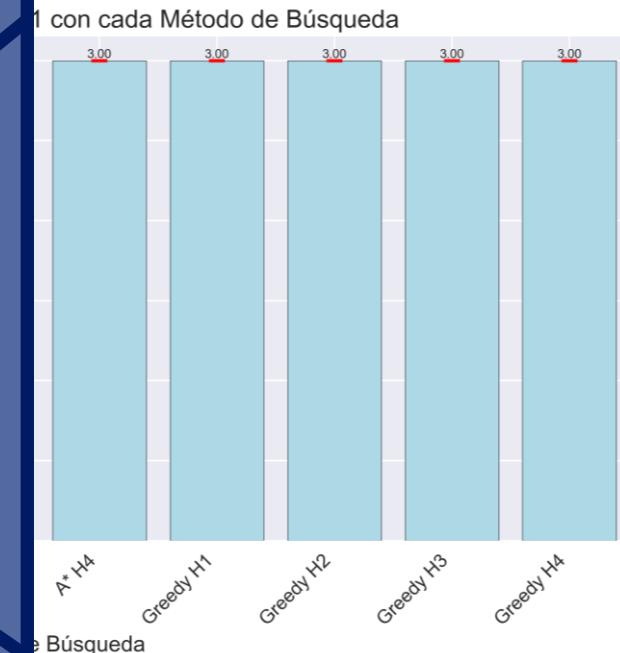
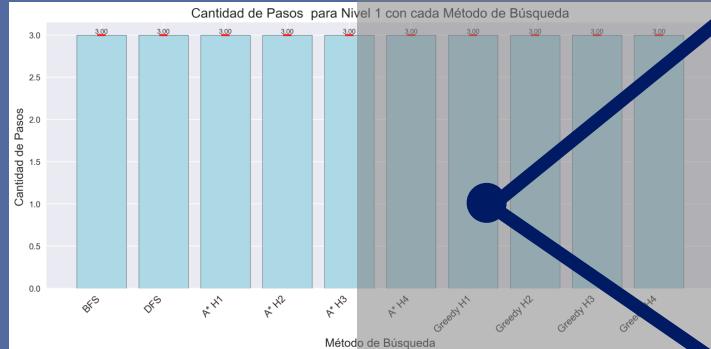
02

03

04

05

COSTO (PASOS)





EASY

# RESULTADOS Y ANÁLISIS

01

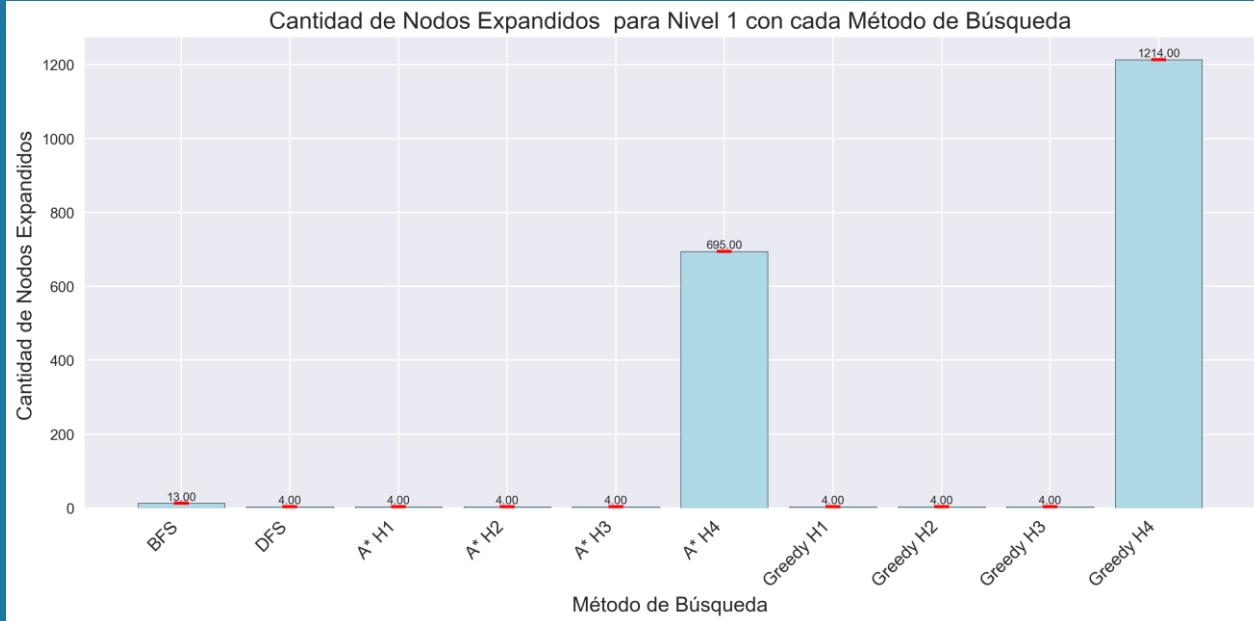
02

03

04

05

## NODOS EXP





EASY

# RESULTADOS Y ANÁLISIS

01

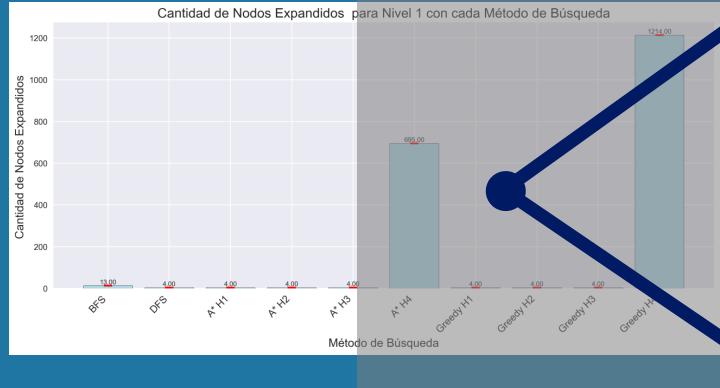
02

03

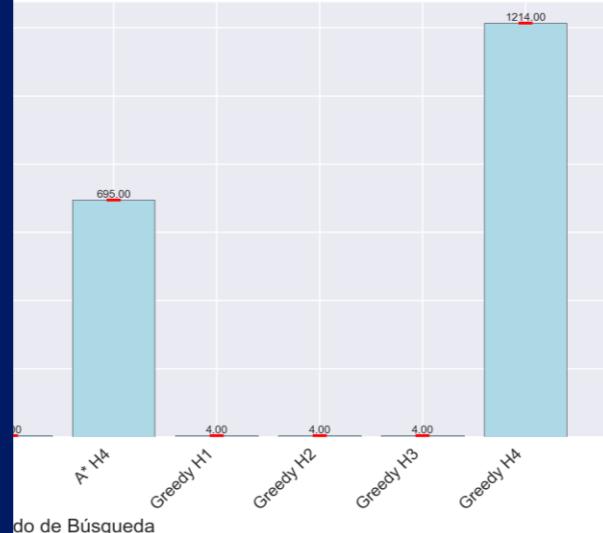
04

05

## NODOS EXP



para Nivel 1 con cada Método de Búsqueda





EASY

# RESULTADOS Y ANÁLISIS

01

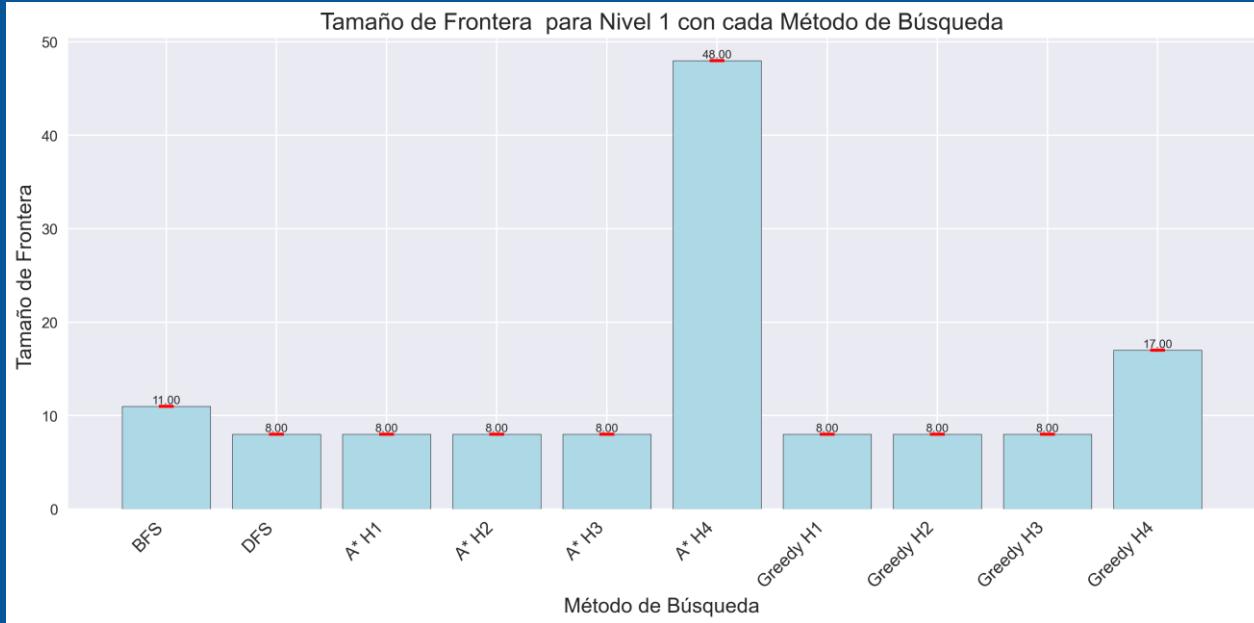
02

03

04

05

NODOS FR





EASY

# RESULTADOS Y ANÁLISIS

01

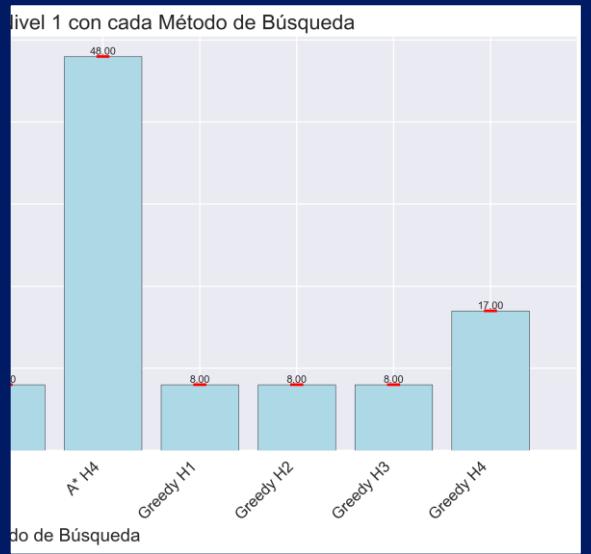
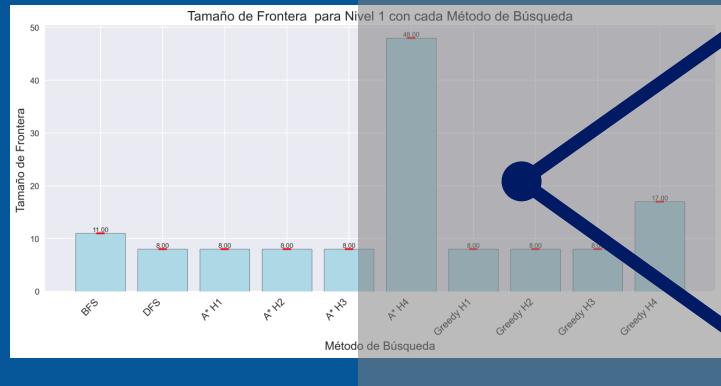
02

03

04

05

NODOS FR





EASY

# RESULTADOS Y ANÁLISIS

01

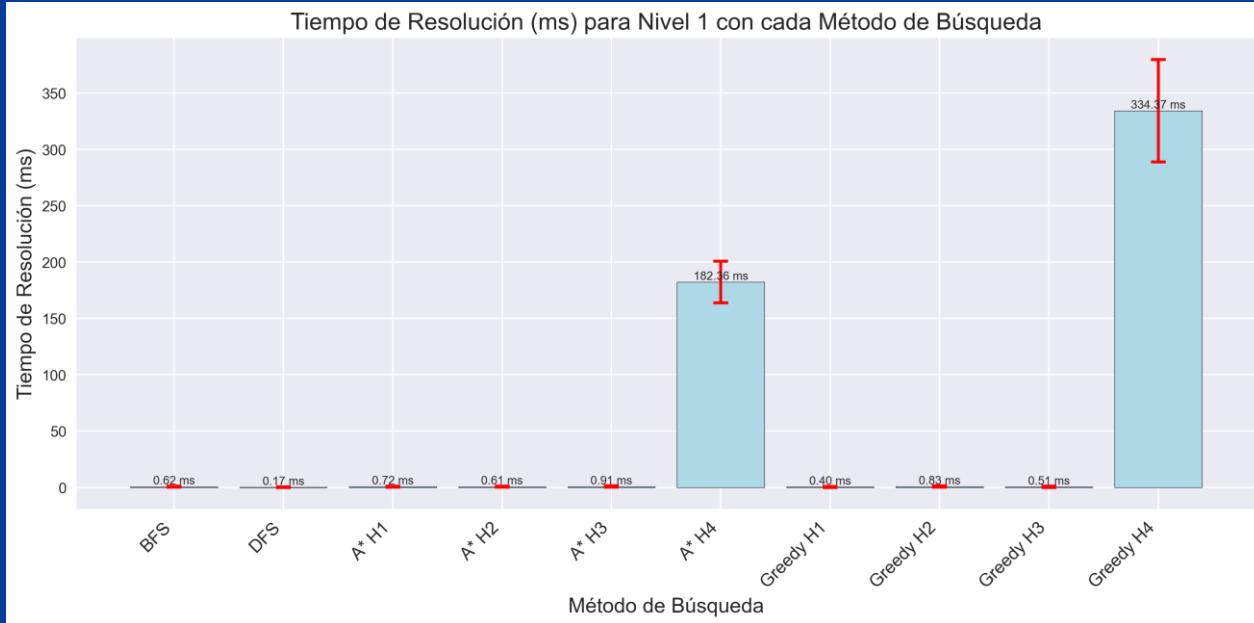
02

03

04

05

TIEMPO





EASY

# RESULTADOS Y ANÁLISIS

01

02

03

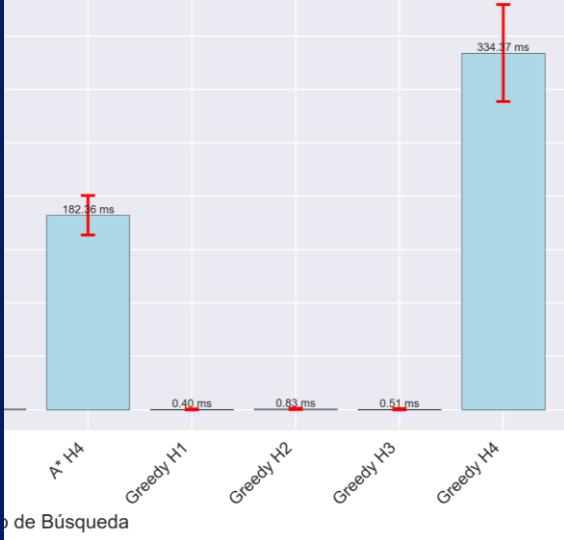
04

05

TIEMPO



Nivel 1 con cada Método de Búsqueda



# RESULTADOS Y ANÁLISIS

01

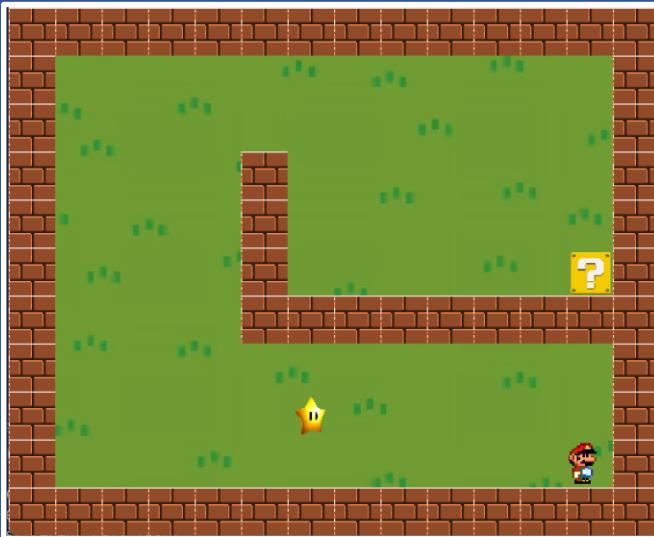
02

03

04

05

RESULTADO



Tablero Nivel 2  
(Medium)



Medium

# RESULTADOS Y ANÁLISIS

01

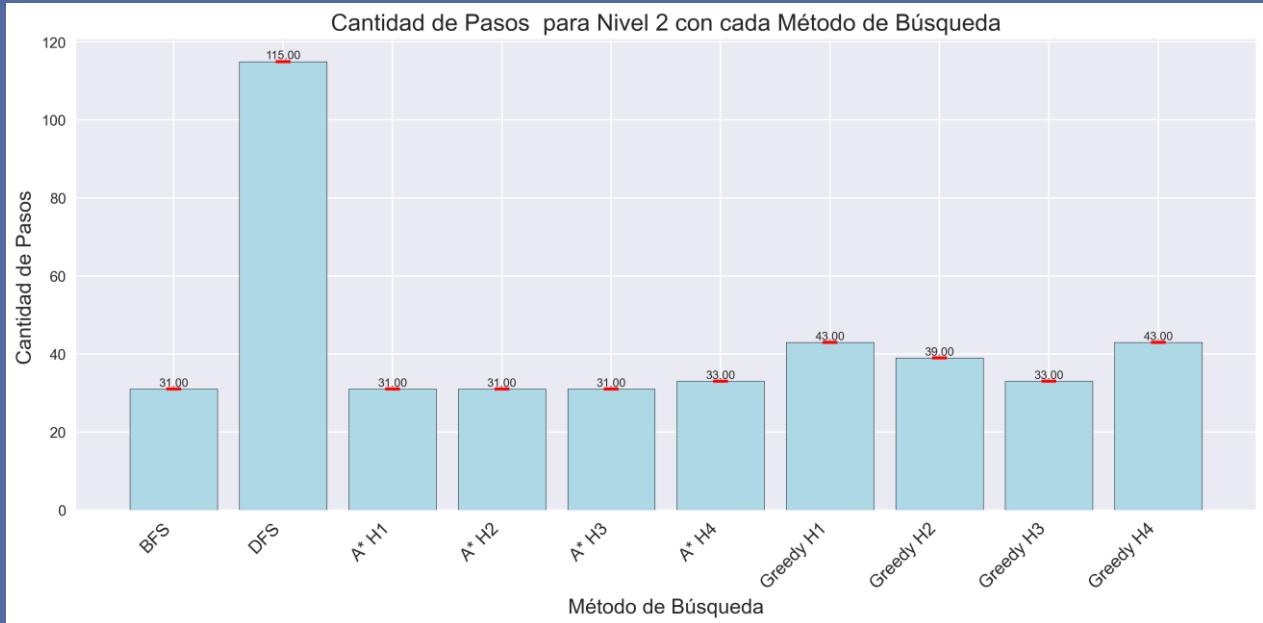
02

03

04

05

COSTO (PASOS)





# RESULTADOS Y ANÁLISIS

01

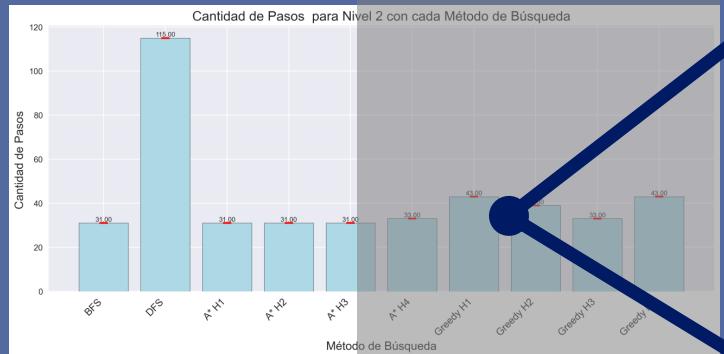
02

03

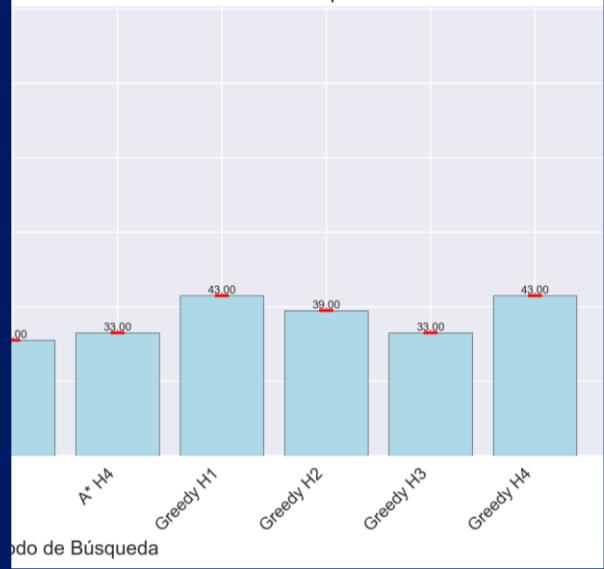
04

05

COSTO (PASOS)



Nivel 2 con cada Método de Búsqueda





Medium

# RESULTADOS Y ANÁLISIS

01

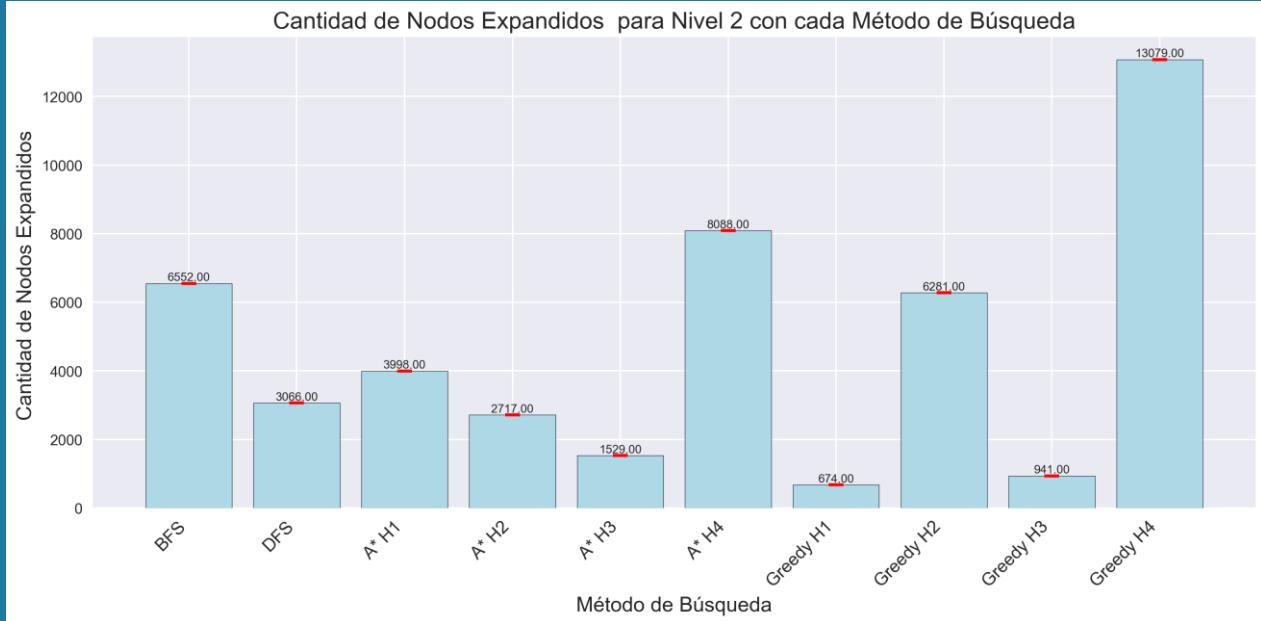
02

03

04

05

NODOS EXP





Medium

# RESULTADOS Y ANÁLISIS

01

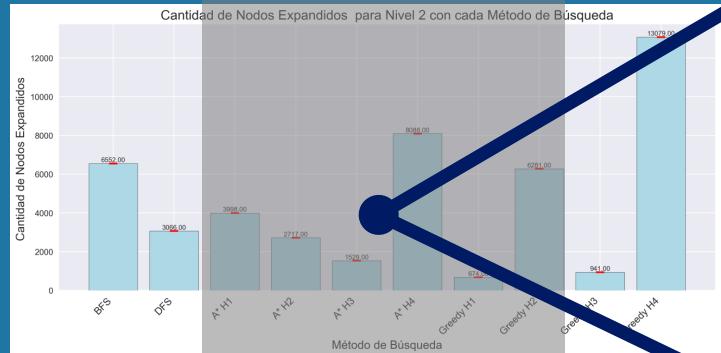
02

03

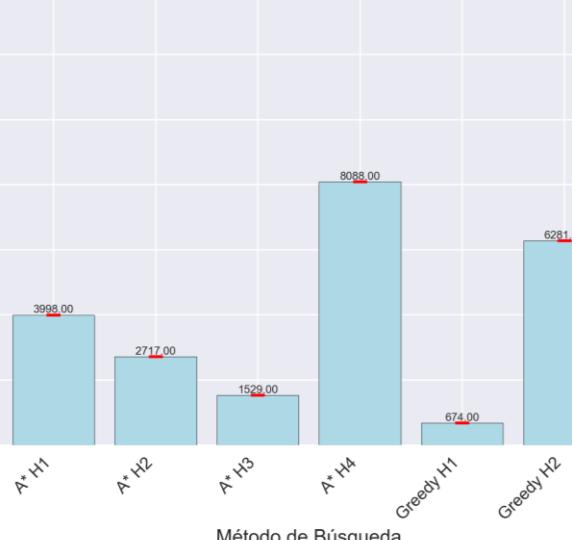
04

05

## NODOS EXP



Cantidad de Nodos Expandidos para Nivel 2 con cada Método de Búsqueda





Medium

# RESULTADOS Y ANÁLISIS

01

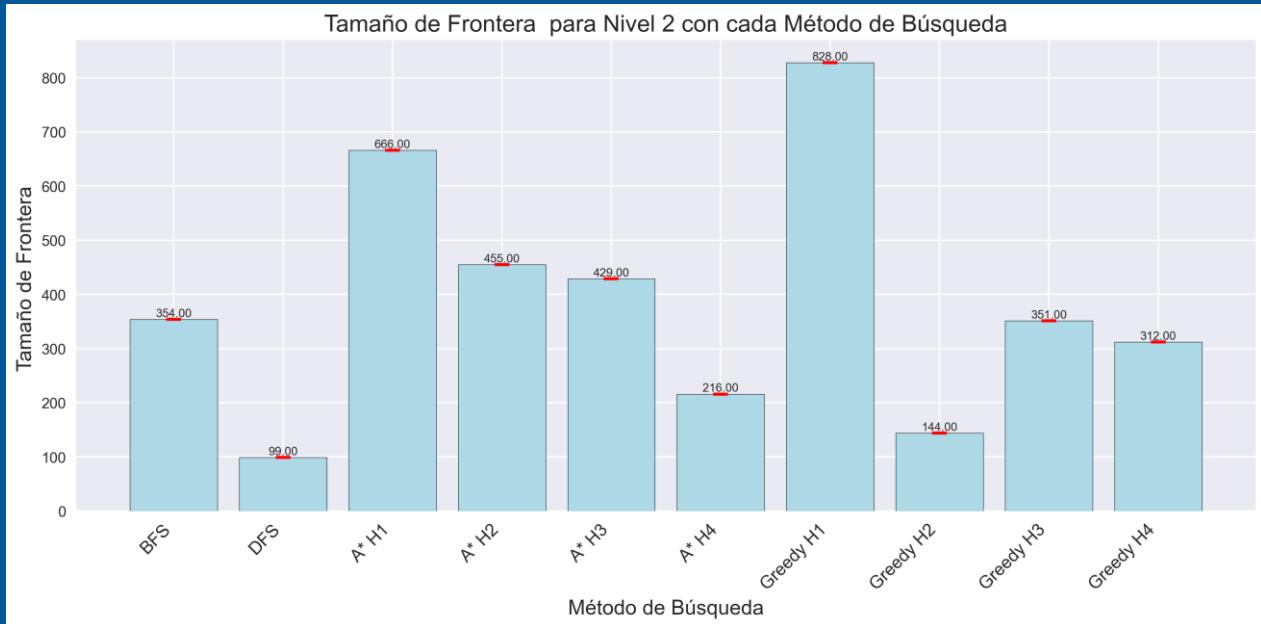
02

03

04

05

NODOS FR





Medium

# RESULTADOS Y ANÁLISIS

01

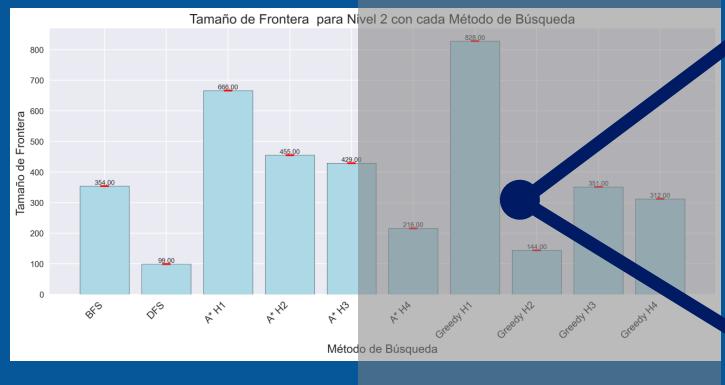
02

03

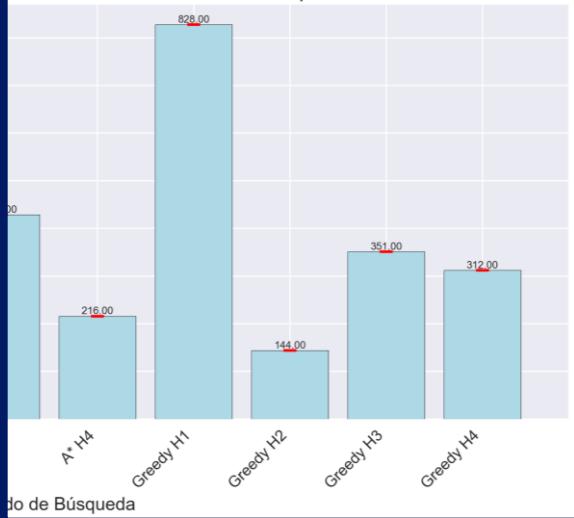
04

05

NODOS FR



Nivel 2 con cada Método de Búsqueda





Medium

# RESULTADOS Y ANÁLISIS

01

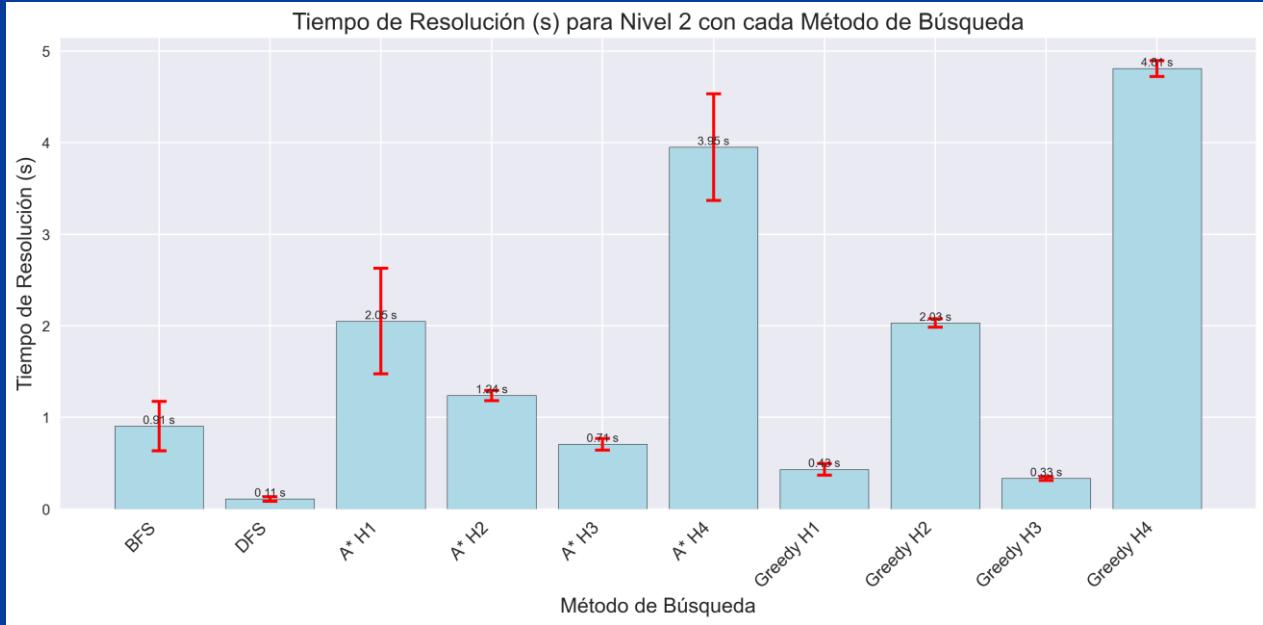
02

03

04

05

TIEMPO





Medium

# RESULTADOS Y ANÁLISIS

01

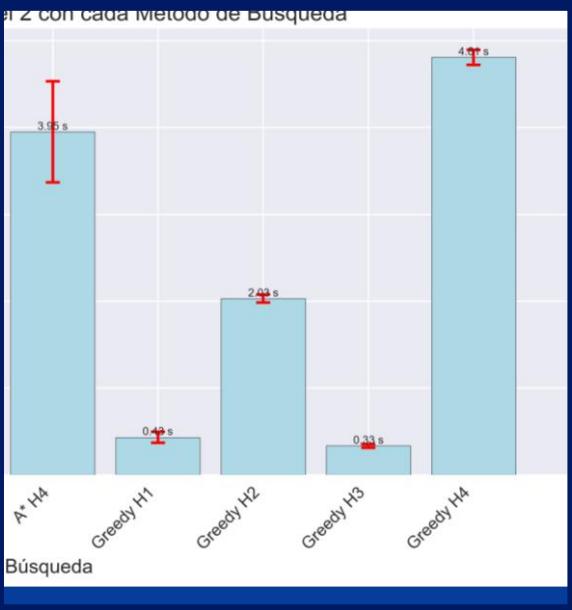
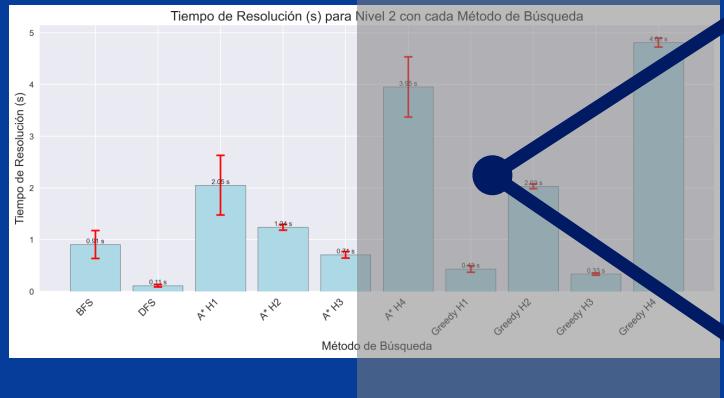
02

03

04

05

TIEMPO



# RESULTADOS Y ANÁLISIS

01

02

03

04

05

RESULTADO



Tablero Nivel 3  
(Difficult)



Difficult

# RESULTADOS Y ANÁLISIS

01

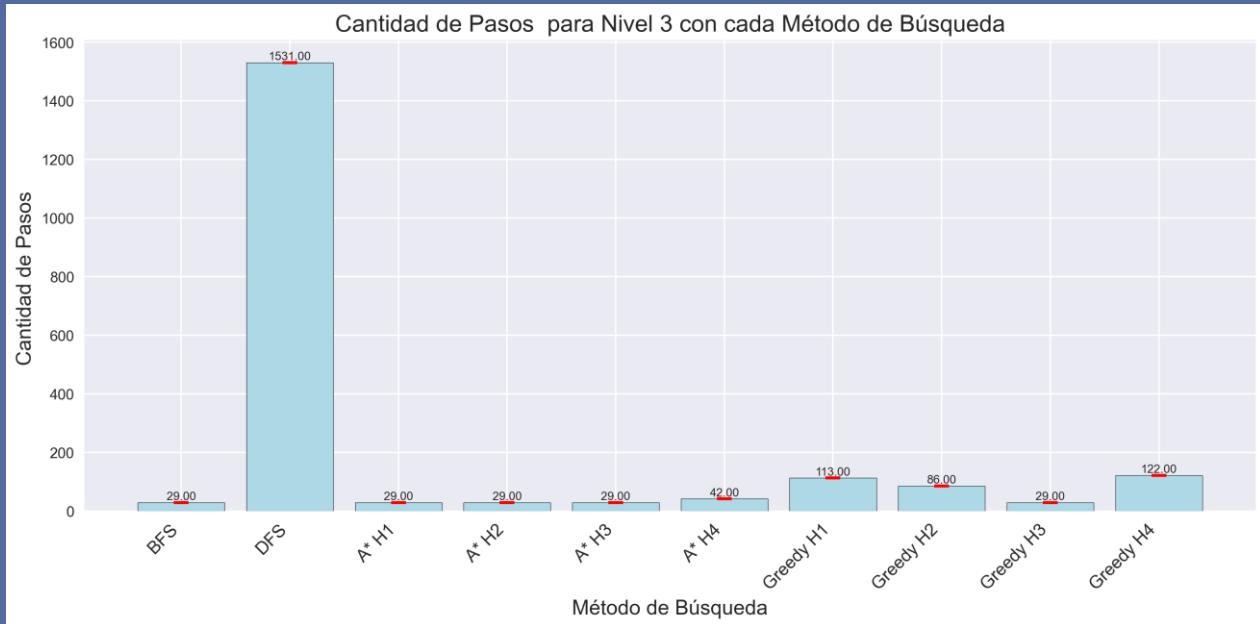
02

03

04

05

COSTO (PASOS)





Difficulty

# RESULTADOS Y ANÁLISIS

01

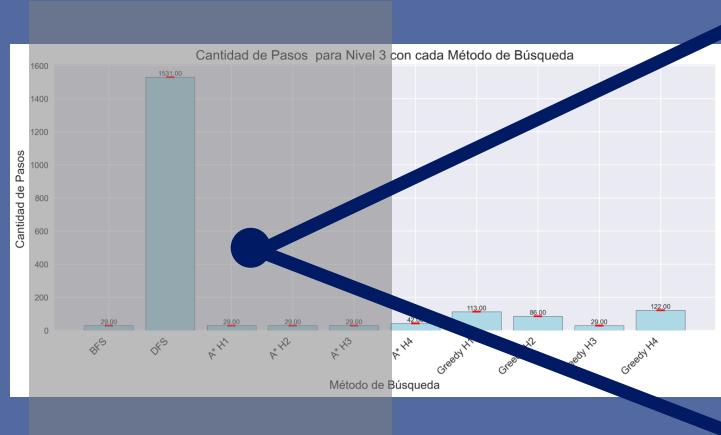
02

03

04

05

## COSTO (PASOS)





Difficult

# RESULTADOS Y ANÁLISIS

01

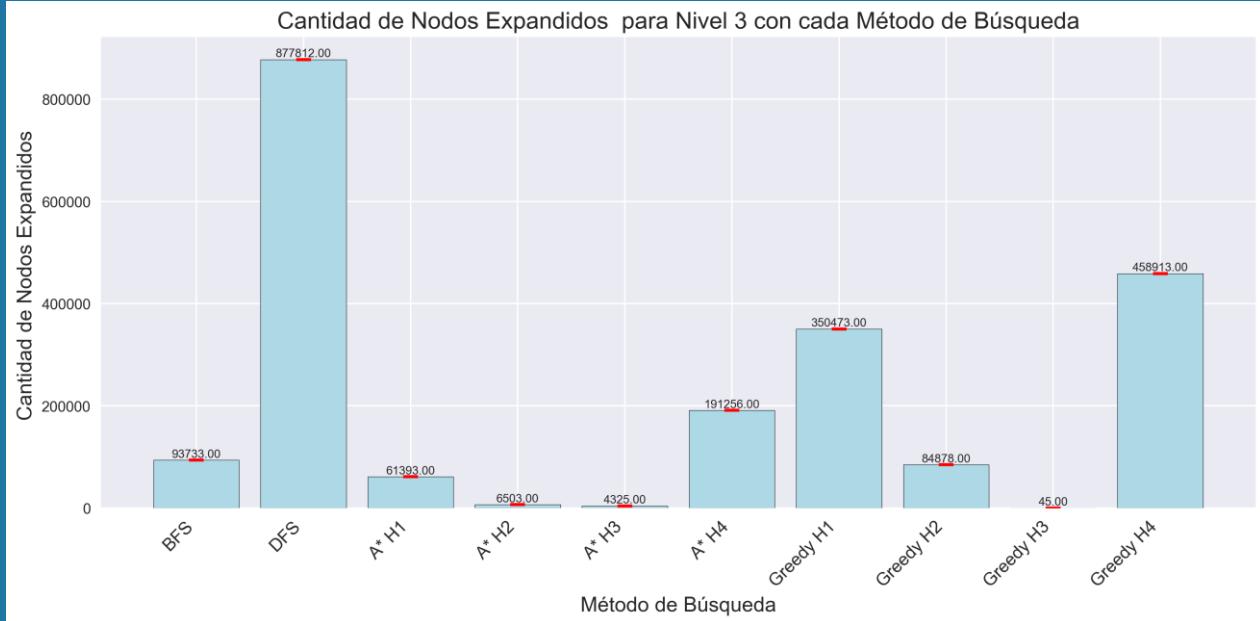
02

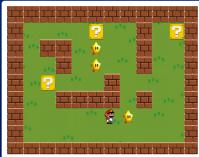
03

04

05

NODOS EXP





Difficult

# RESULTADOS Y ANÁLISIS

01

02

03

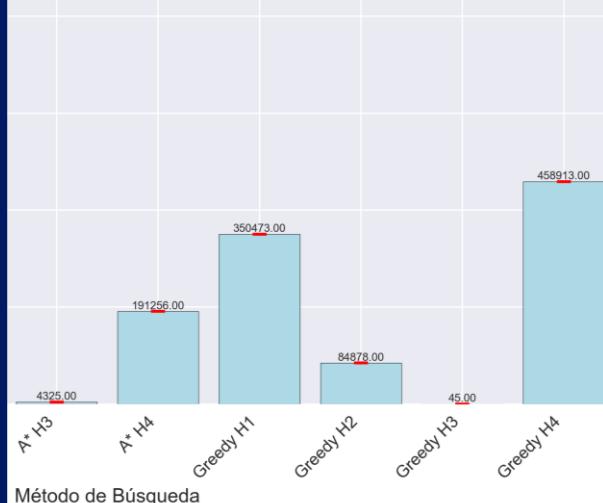
04

05

## NODOS EXP



Nodos Expandidos para Nivel 3 con cada Método de Búsqueda





Difficult

# RESULTADOS Y ANÁLISIS

01

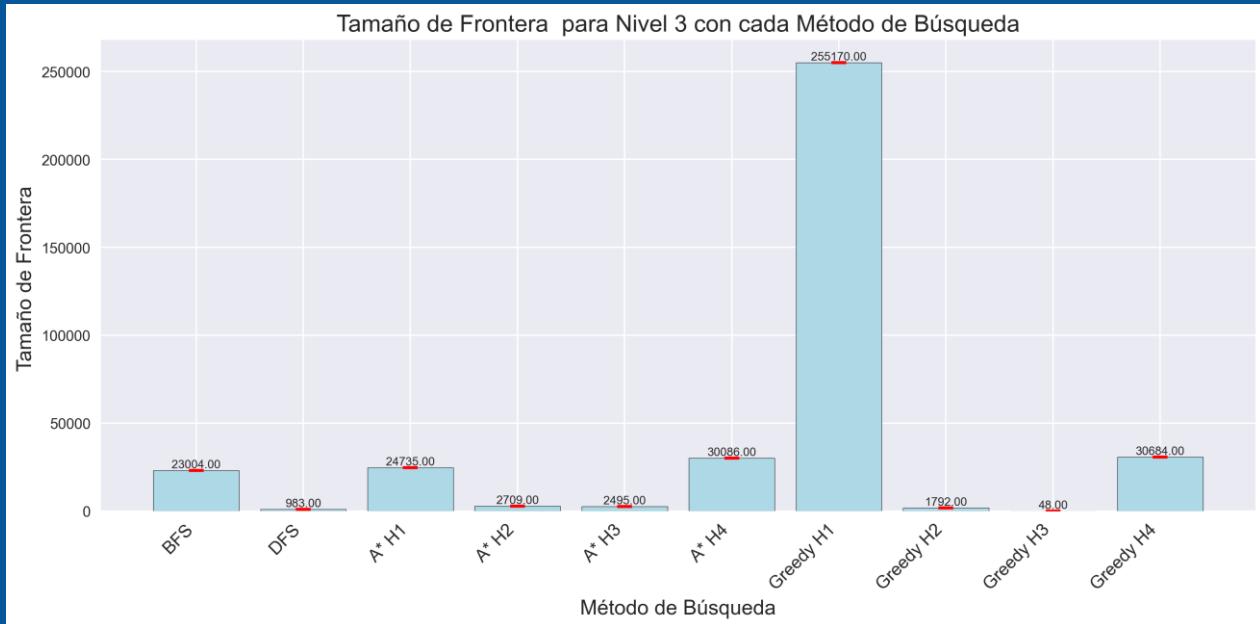
02

03

04

05

NODOS FR





Difficult

# RESULTADOS Y ANÁLISIS

01

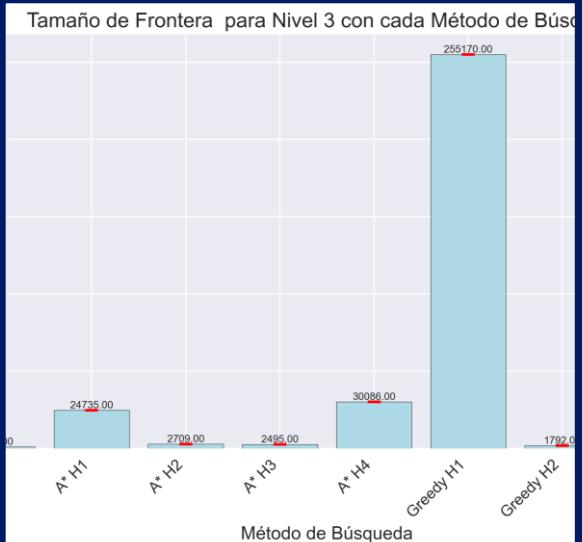
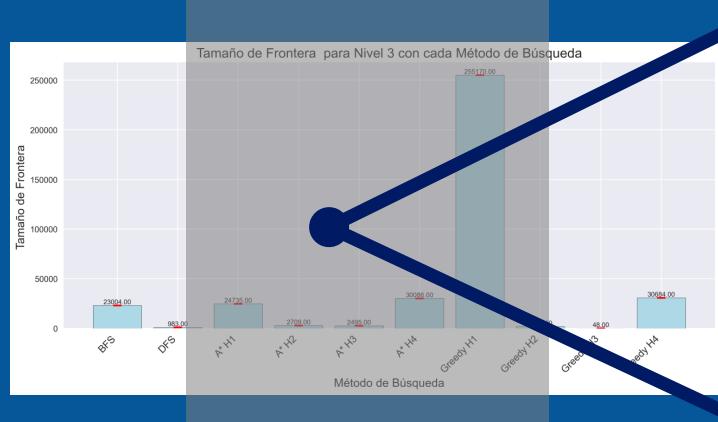
02

03

04

05

NODOS FR





Difficulty

# RESULTADOS Y ANÁLISIS

01

02

03

04

05

TIEMPO





Difficulty

# RESULTADOS Y ANÁLISIS

01

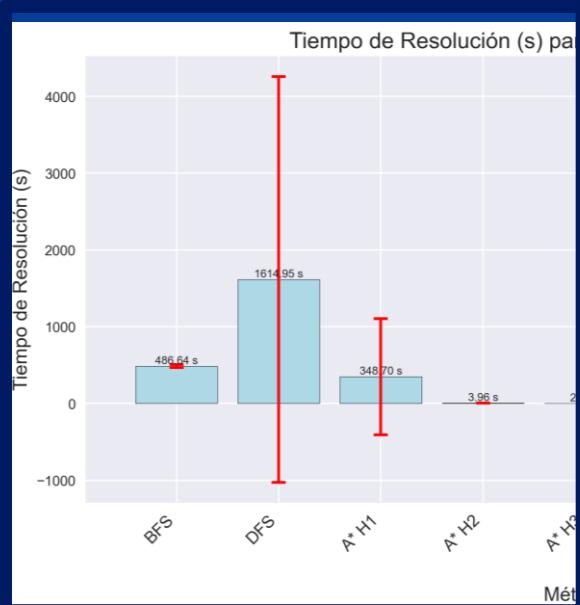
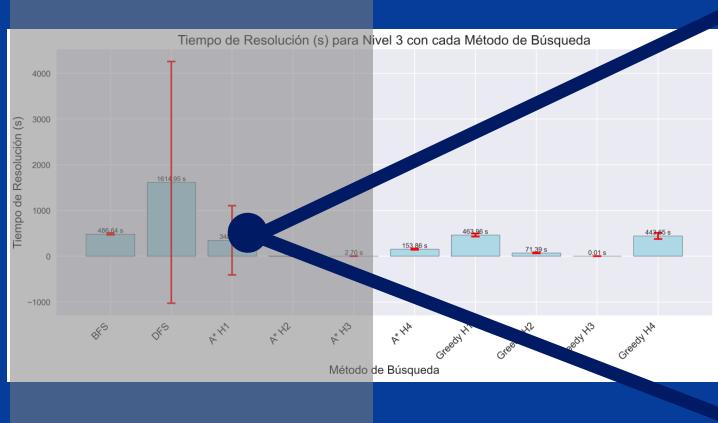
02

03

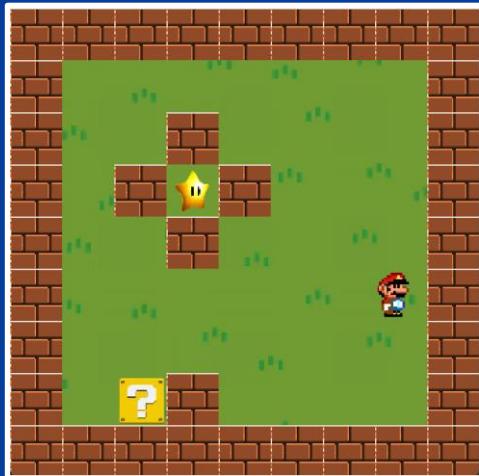
04

05

TIEMPO



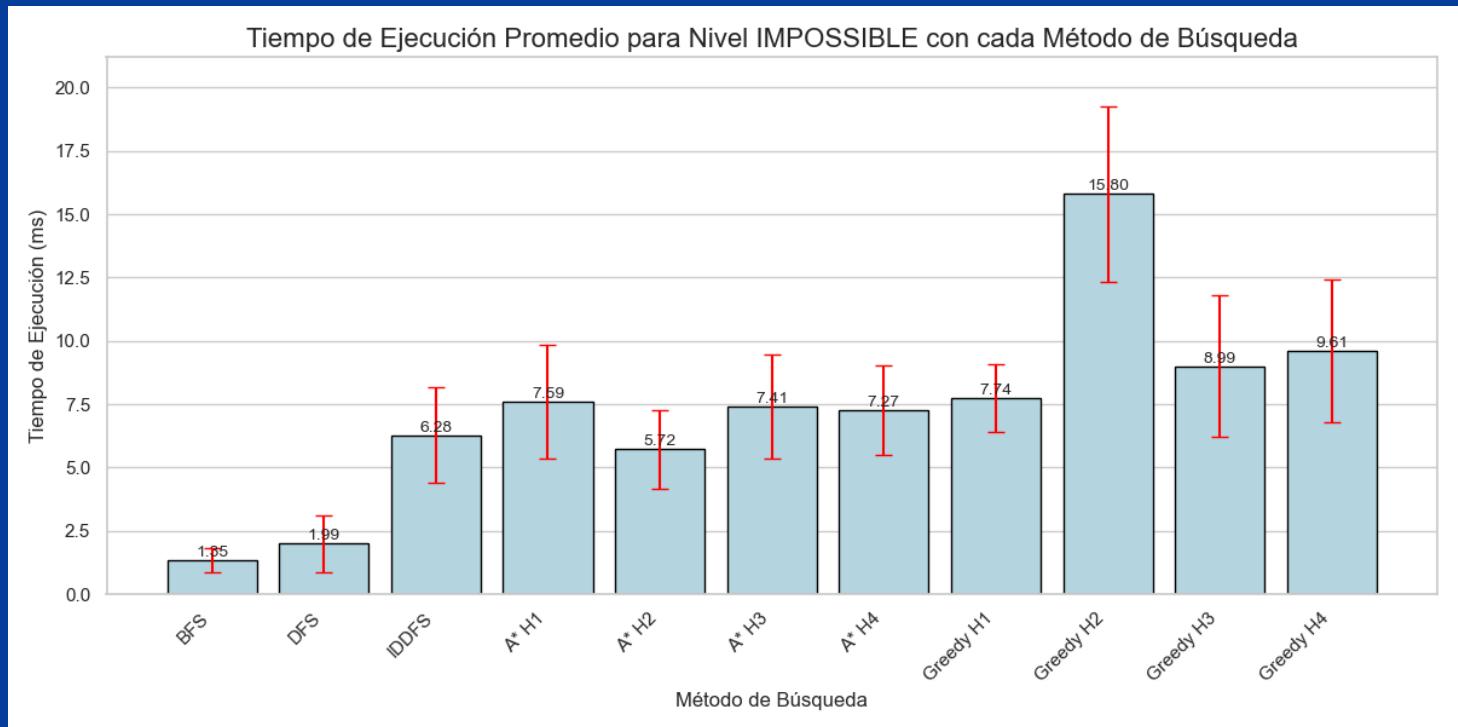
# Resultados Nivel Impossible



¿Qué algoritmo "tardó" menos tiempo en darse cuenta que NO había solución?



# Resultados Nivel Impossible



# CONCLUSIONES

## COSTO

Complejidad  $\neq$  n° pasos  
Sin aumento lineal

A\* con H3 y BFS

DFS

## NODOS EXP

Aumenta con el nivel

H3

H4 y DFS (difficult)

## NODOS FR

DFS  
(explora un camino)

GREEDY H1

## TIEMPO

GREEDY H3  
(no óptimo)

H4 Y DFS (difficult)

## HEURÍSTICAS

Efectividad dependiendo  
del nivel de tablero

H3

H4 es no admisible

## MÉTODOS

Informados con H  
admisibles para niveles  
complejos  
A\* H3

No informados para simples  
BFS

# ¡GRACIAS!

¿Preguntas?

