



Sistemas de inteligencia artificial

TP1: Métodos de búsqueda

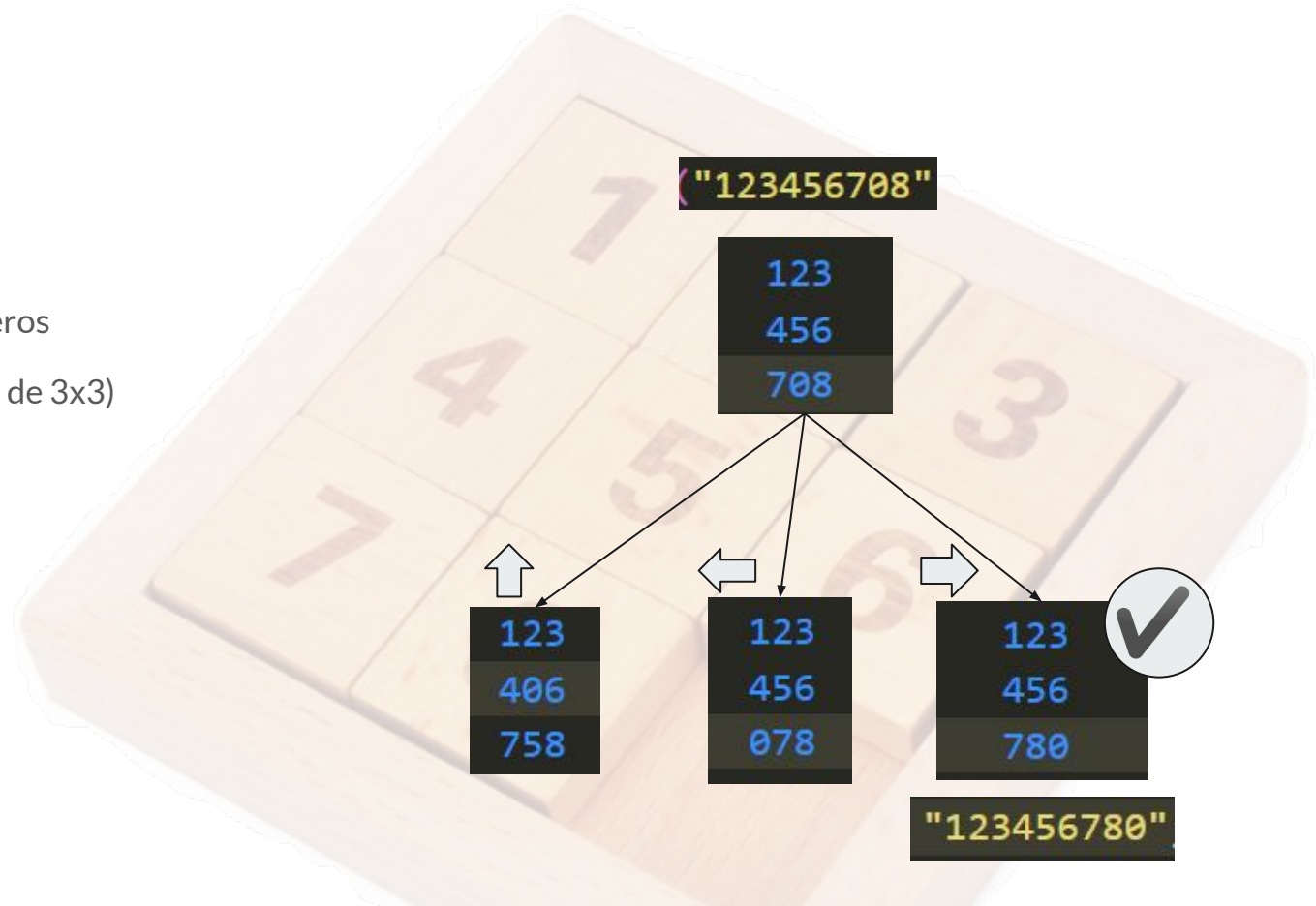
Integrantes:

- Lucas Catolino
- Matias Ricarte

Problema

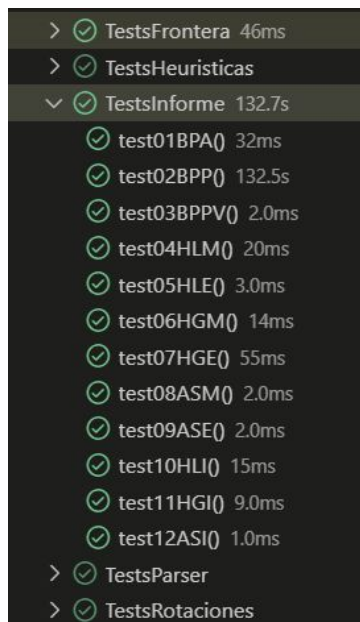
Rompecabezas de números

(8 números en una grilla de 3x3)



Implementación y consideraciones

Implementación y consideraciones



>	✓	TestsFrontera	46ms
>	✓	TestsHeurísticas	
✓	✓	TestsInforme	132.7s
	✓	test01BPA()	32ms
	✓	test02BPP()	132.5s
	✓	test03BPPV()	2.0ms
	✓	test04HLM()	20ms
	✓	test05HLE()	3.0ms
	✓	test06HGM()	14ms
	✓	test07HGE()	55ms
	✓	test08ASM()	2.0ms
	✓	test09ASE()	2.0ms
	✓	test10HLI()	15ms
	✓	test11HGI()	9.0ms
	✓	test12ASI()	1.0ms
>	✓	TestsParser	
>	✓	TestsRotaciones	

La metodología de desarrollo implementada fue TDD:

1. Creación de test con cierta funcionalidad inexistente
2. El test falla, se implementa la funcionalidad
3. El test pasa

```
ordenado = new Tablero("123456780");  
2 = new Tablero("123456708");  
random = new Tablero("013425786");  
random2 = new Tablero("023451768");
```



Implementación y consideraciones

Los tableros fueron considerados strings
Dependiendo de la posición del cero se habilitan las
“rotaciones” disponibles

```
Tablero tableroOrdenado = new Tablero("123456780");  
Tablero tablero32 = new Tablero("123456708");  
Tablero tableroRandom = new Tablero("013425786");  
Tablero tableroRandom2 = new Tablero("023451768");
```



Implementación y consideraciones

```
public boolean isSolvable() {  
    // Si esta mal armada la matriz no es resoluble  
    if (!this.wellConfigured()) {  
        return false;  
    }  
  
    int inv = 0;  
    // Elimino el espacio vacío  
    String estadoAux = this.estado.replace("0", "");  
  
    // Por cada elemento, veo si los que vienen después son menores, aumentando el  
    // inv  
    for (int i = 0; i < estadoAux.length(); i++) {  
        int pre = (int) estadoAux.charAt(i) - ASCII;  
        for (int j = i; j < estadoAux.length(); j++) {  
            int post = (int) estadoAux.charAt(j) - ASCII;  
            if (pre > post) {  
                inv++;  
            }  
        }  
    }  
  
    // Si inv es par es resoluble, sino no  
    return (inv % 2 == 0) ? true : false;  
}
```

Un tablero es válido si:

1. Tiene 9 valores (del 0 al 8 sin repetir)
2. El tablero tiene una cantidad par de “inversiones”
 - Una inversión es cuántos elementos menores tiene a la derecha cada elemento
 - La paridad se mantiene ya que cada rotación aumenta o disminuye en 2 la inversión
 - El estado final 123456780 tiene inversión 0

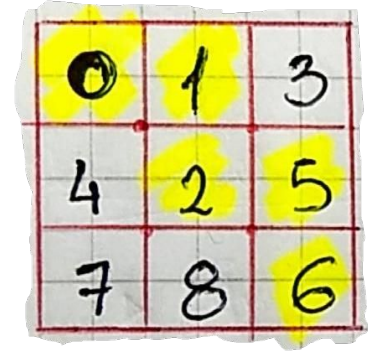
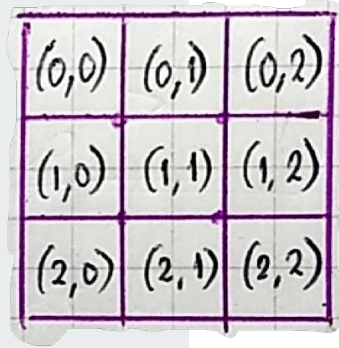
Cálculo de heurísticas

Heurística 1: Manhattan

Definimos la distancia manhattan como:

$$h_m(s) = \frac{\sum_{p \in \text{Piezas fuera de lugar}} m(p)}{2}$$

siendo $m(p) = (|p_{x \text{ ideal}} - p_{x \text{ actual}}| + |p_{y \text{ ideal}} - p_{y \text{ actual}}|)$



Ejemplo:

Pieza 0: $(|2 - 0| + |2 - 0|) = 4$

Pieza 1: $(|0 - 1| + |0 - 0|) = 1$

Pieza 2: $(|1 - 1| + |0 - 1|) = 1$

Pieza 5: $(|1 - 2| + |1 - 1|) = 1$

Pieza 6: $(|2 - 2| + |1 - 2|) = 1$

$$h_m(s) = 4$$

Heurística 2: Euclides

De forma análoga a Manhattan, definimos la distancia de Euclides como la suma de las hipotenusas de los catetos formados por la distancia de Manhattan dividido por 2.

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

0	1	3
4	2	5
7	8	6

Ejemplo:

$$\text{Pieza 0: } \sqrt{(|2 - 0|^2 + |2 - 0|^2)} = 2$$

$$\text{Pieza 1: } \sqrt{(|0 - 1|^2 + |0 - 0|^2)} = 1$$

$$\text{Pieza 2: } \sqrt{(|1 - 1|^2 + |0 - 1|^2)} = 1$$

$$\text{Pieza 5: } \sqrt{(|1 - 2|^2 + |1 - 1|^2)} = 1$$

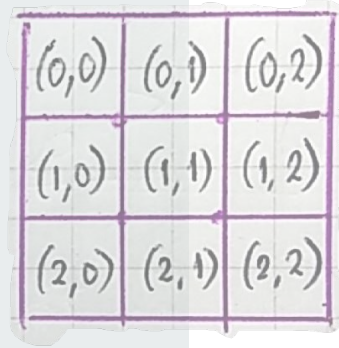
$$\text{Pieza 6: } \sqrt{(|2 - 2|^2 + |1 - 2|^2)} = 1$$

$$h_e(s) = 3$$

Heurística 3: No admisible

Definimos la heurística no admisible como la cantidad de números fuera de lugar para el estado actual.

Interesante: un estado de menor heurística significa un estado más ordenado.



(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

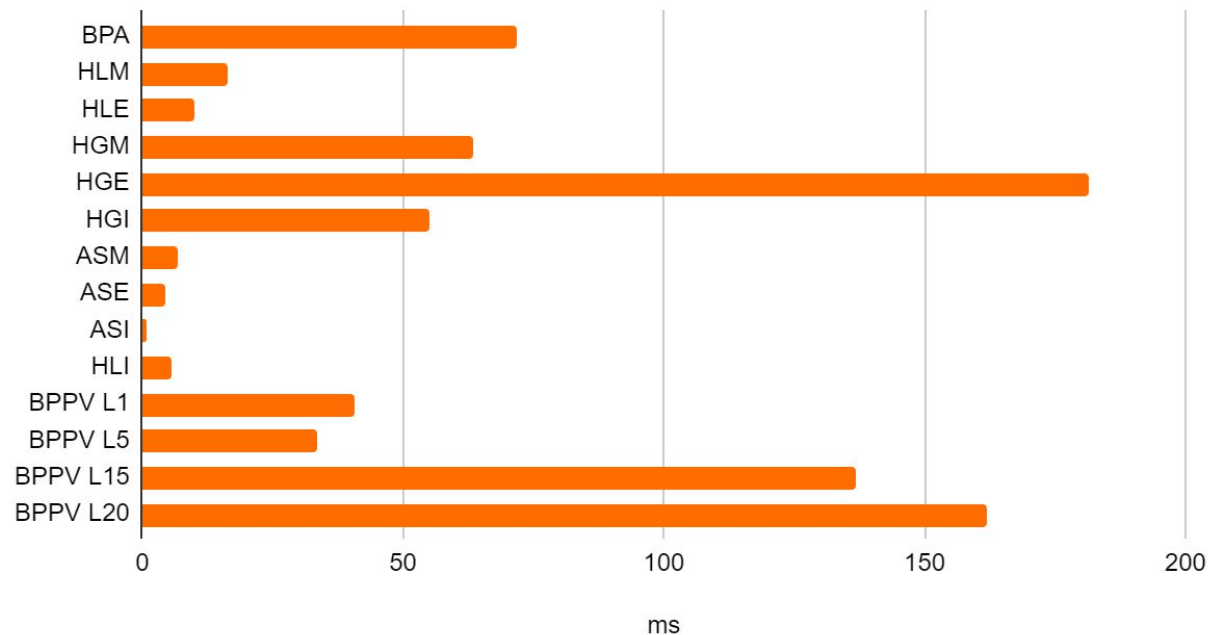
Para este estado la heurística resulta 3, pero en 2 movimientos se resuelve

1	2	3
4	5	6
0	7	8

$$h_{na}(s) = 3$$

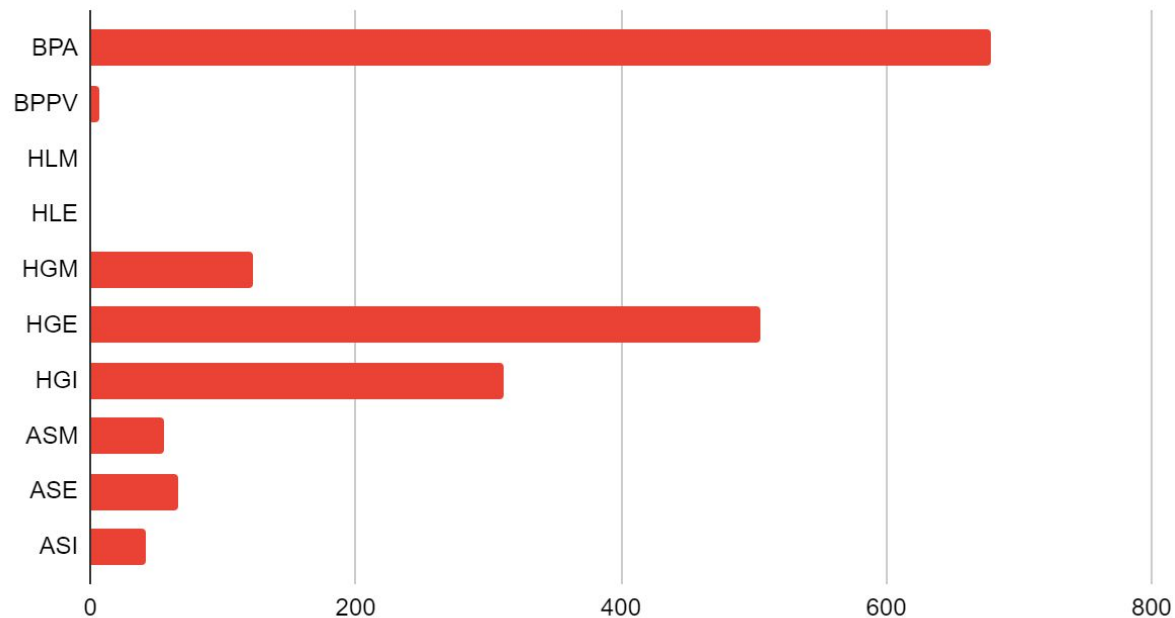
Resultados y conclusiones

Tiempo promedio de ejecución (ms)



Descripción	Tiempo
BPA	72±11
BPP	199607±54390
HLM	16.5±7.3
HLE	9.8±3.1
HGM	63.3±22.1
HGE	181.7±68.7
ASM	6.8±3.6
ASE	4.3±2.1
HLI	5.5±3.7
HGI	55.0±34.7
ASI	41.3±7.3
BPPV L1	40.7±14.8
BPPV L5	33.5±7.0
BPPV L15	136.7±30.5
BPPV L20	162.0±37.0

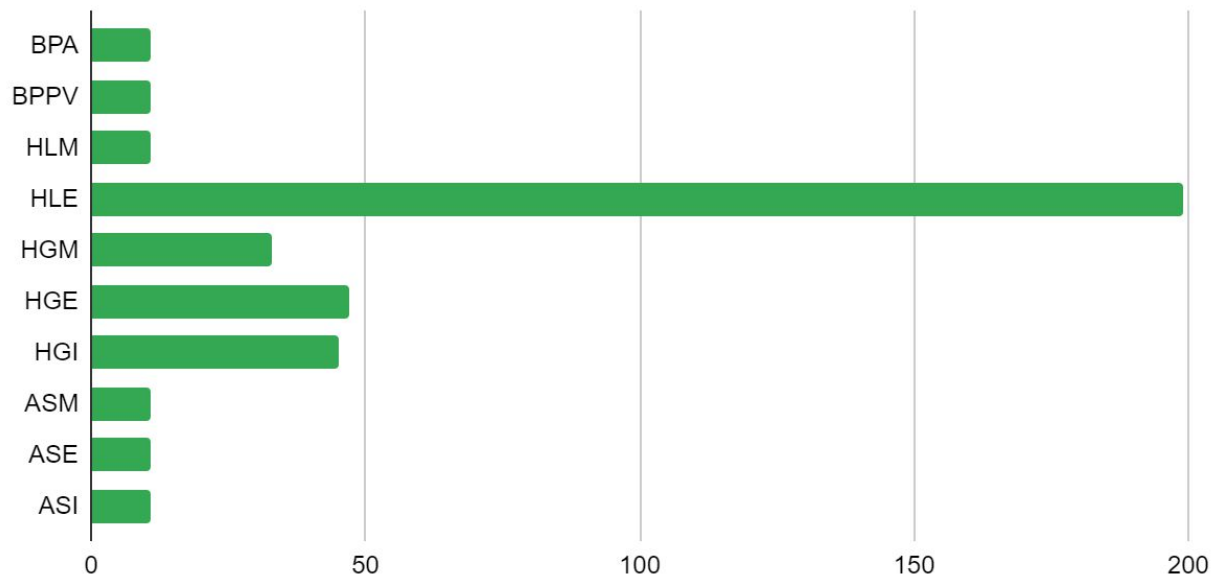
Nodos frontera



Descripción	Cantidad
BPP	70533
BPA	679
BPPV	7
HLM	0
HLE	0
HGM	123
HGE	505
HGI	312
ASM	56
ASE	66
ASI	42

Descripción	Cantidad
BPP	92103
BPA	11
BPPV	11
HLM	11
HLE	199
HGM	33
HGE	47
HGI	45
ASM	11
ASE	11
ASI	11

Profundidad/Costo

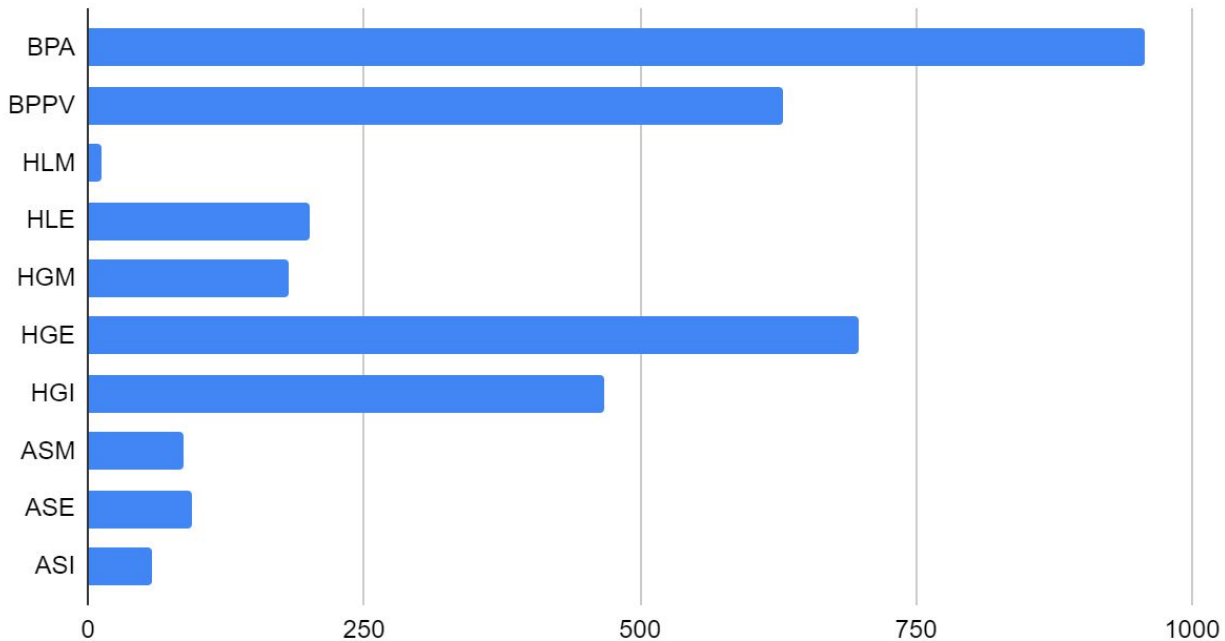


Movimientos



Descripción	Cantidad
BPP	133323
BPA	957
BPPV	630
HLM	12
HLE	200
HGM	181
HGE	698
HGI	467
ASM	86
ASE	94
ASI	58

Nodos expandidos



¿Preguntas?



Muchas gracias

Lucas Catolino
Matias Ricarte