

# Trabajo Práctico Especial

## Nro. 1

---

Juego: 0h n0!

### **INTEGRANTES**

Maximiliano J. Valverde (Legajo: 51158)

Ma. Florencia Besteiro (Legajo: 51117)

Julieta Sal-lari (Legajo: 49629)

### **PROFESORES**

Maria Cristina Parpaglione

Ignacio Luciani

## Indice

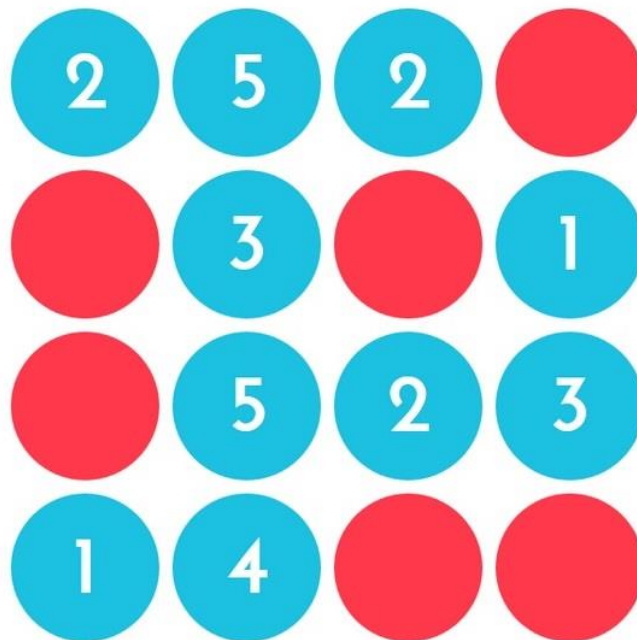
Reglas y objetivos del juego: Oh n0!	2
Heurísticas.....	3
Heurística 1 : Escenario ALL BLUE .....	3
Heurística 2: Escenario ALL RED.....	4
Costo .....	5
Búsquedas y Resultados.....	5
Resultados: ALL BLUE .....	5
Resultados: ALL RED.....	6
Análisis y Conclusiones .....	6
Tiempo de ejecución.....	6
Cantidad de nodos expandidos.....	6
General.....	6
Nota.....	6

## Reglas y objetivos del juego: Oh n0!

Se cuenta con un tablero de ciertas dimensiones, estas pueden ser 5x5, 6x6, 7x7 u 8x8. El mismo contiene celdas que pueden ser de color blanco, rojo o azul.

Dentro de las celdas azules, están aquellas que poseen un número. El mismo, indica la cantidad de celdas de dicho color, que deben ser visibles (en la misma fila y columna). Por ejemplo: si poseo una celda azul con el número 3, esto indica que la misma debe ver SOLAMENTE otras 3 celdas del mismo color en la misma fila y columna. Lo que bloquea la “visión” de una celda, es un bloque de color rojo.

Para ganar el juego, todas las celdas que poseen número, deben ver SOLAMENTE la cantidad de celdas azules que su valor indica y no deben quedar celdas sin colorear (color blanco).



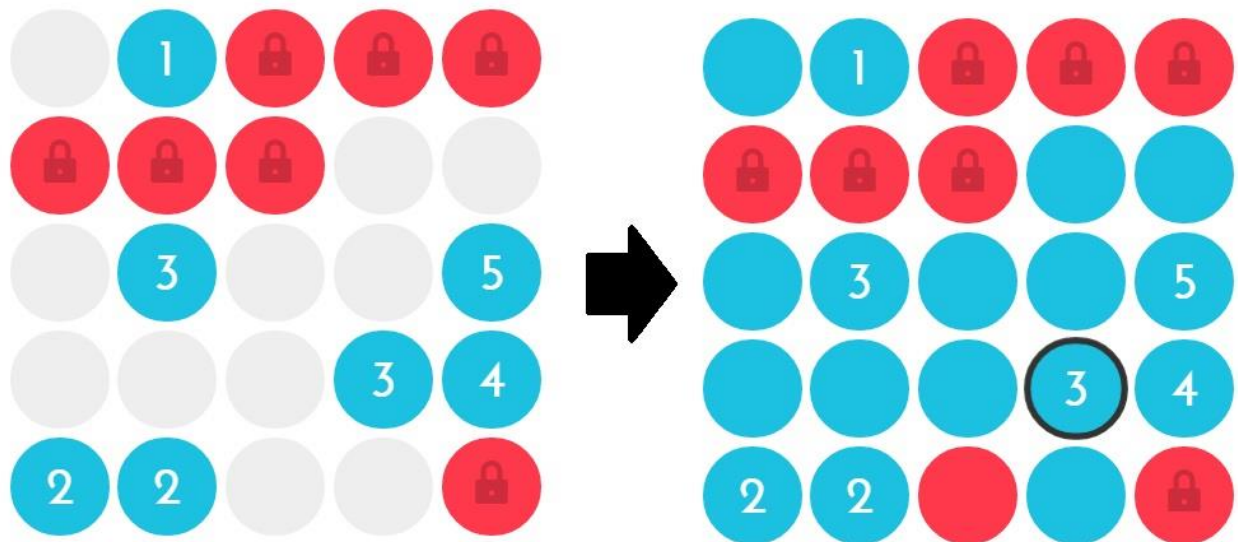
## Heurísticas

Se implementaron 2 heurísticas admisibles. La primera, **Heurística 1**, se utiliza en un escenario donde se pintan todas las celdas iniciales blancas con azul para comenzar con las búsquedas correspondientes mientras que la segunda, **Heurística 2**, se desarrolla en una situación donde se prepara el tablero pintando todas las celdas blancas de color rojo.

### Heurística 1 : Escenario ALL BLUE

Para ejecutar las búsquedas solicitadas, se prepara el tablero inicial pintando TODAS las celdas blancas de azul. Los movimientos (reglas) a aplicar, son colocar bloques (celdas de color rojo) en distintas posiciones del tablero, bloqueando la visión de las fichas con número.

**Una regla no es válida, si al aplicarla sucede que una ficha con número vea una MENOR CANTIDAD DE CELDAS de las que debe ó en dicha posición, ya haya un bloqueo.**

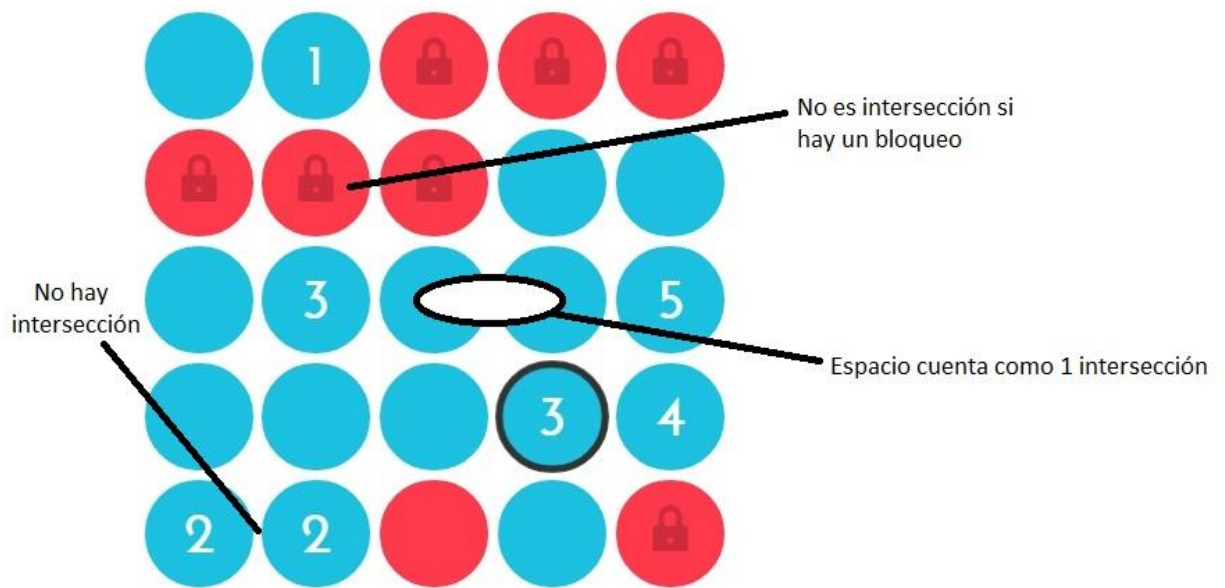


Como se puede ver, una celda puede ser vista por una o varias, por lo que tenemos que tenerlas en cuenta al momento de calcular el valor para nuestra heurística.

El cálculo de la misma viene dado por la siguiente ecuación:

$$h = n - i - 1$$

Donde  $n$  es la cantidad de celdas insatisfechas (no ven la cantidad de celdas necesarias) e  $i$  la cantidad de intersecciones. Contamos intersección como espacio entre dos fichas con valor. Ese espacio solo debe contener fichas azules SIN valor.

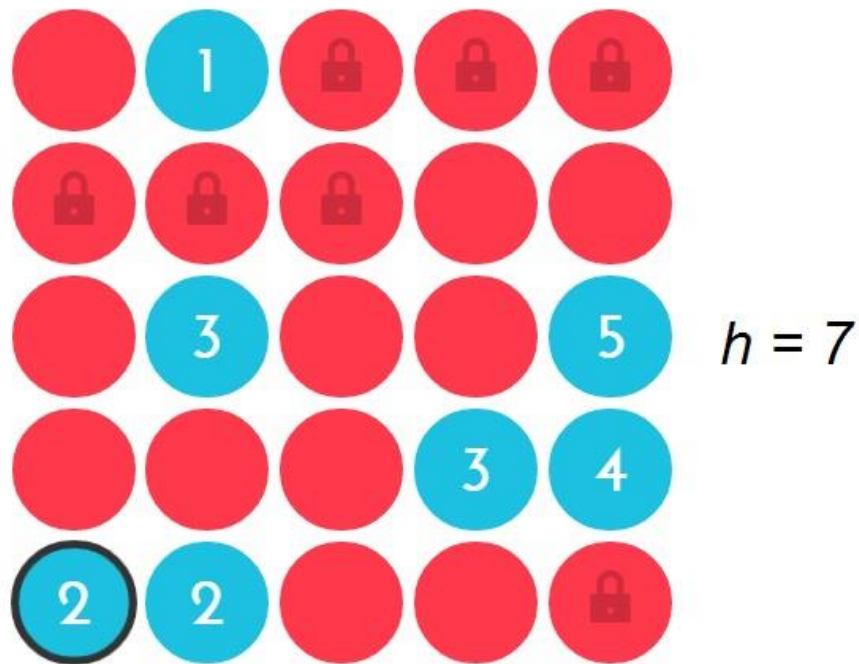


Se le resta uno como valor de un paso, que es colocar un bloqueo.

## Heurística 2: Escenario ALL RED

En este escenario, se comienza preparando el tablero reemplazando todas las celdas vacías (color blanco), con bloqueos. Las reglas para esta ejecución, es pintar celdas de color azul hasta llegar a una solución. Las mismas son válidas, siempre que las fichas con número no vean un número MAYOR DE CELDAS al requerido, y que no haya una celda azul en dicha posición.

**El valor de la heurística para esta situación es la cantidad de celdas insatisfechas, ya que siempre será menor o igual, a la cantidad de celdas que necesitan “desbloquearse” para expandir la visión y así poder ver la cantidad necesaria.**



## Costo

Definimos como costo de aplicar una regla el valor 1, ya que la diferencia entre un tablero a otro es el pintar una celda de color azul o rojo, no genera mayor esfuerzo.

## Búsquedas y Resultados

Se debieron implementar 5 tipos de búsquedas:

- **NO INFORMADAS : BFS, DFS, ITERATIVE**
- **INFORMADAS: GREEDY, A\***

## Resultados: ALL BLUE

### BFS

Depth: 6  
Execution time: 603 milliseconds  
Generated nodes: 792  
Expanded nodes: 735  
Border nodes: 56  
OK! solution found!

### DFS

Depth: 7  
Execution time: 25 milliseconds  
Generated nodes: 47  
Expanded nodes: 8  
Border nodes: 38  
OK! solution found!

### ITERATIVE

Depth: 6  
Execution time: 736 milliseconds  
Generated nodes: 792  
Expanded nodes: 1019  
Border nodes: 68  
OK! solution found!

### GREEDY

Depth: 7  
Execution time: 40 milliseconds  
Generated nodes: 51  
Expanded nodes: 20  
Border nodes: 30  
OK! solution found!

### A\*

Depth: 6  
Execution time: 204 milliseconds  
Generated nodes: 275  
Expanded nodes: 169  
Border nodes: 105  
OK! solution found!

## Resultados: ALL RED

BFS	DFS	ITERATIVE	GREEDY	A*
Depth: 8 Execution time: 8629 milliseconds Generated nodes: 1520 Expanded nodes: 1515 Border nodes: 4 OK! solution found!	Depth: 9 Execution time: 2821 milliseconds Generated nodes: 426 Expanded nodes: 383 Border nodes: 42 OK! solution found!	Depth: 8 Execution time: 11245 milliseconds Generated nodes: 1520 Expanded nodes: 1825 Border nodes: 4 OK! solution found!	Depth: 9 Execution time: 581 milliseconds Generated nodes: 122 Expanded nodes: 86 Border nodes: 35 OK! solution found!	Depth: 8 Execution time: 784 milliseconds Generated nodes: 184 Expanded nodes: 105 Border nodes: 78 OK! solution found!

## Análisis y Conclusiones

### Tiempo de ejecución

Como se puede apreciar por los resultados, el valor es **MENOR** en los algoritmos de búsquedas informadas en comparación con los de las **NO** informadas, a excepción de la estrategia **DFS** en el escenario **ALL BLUE**. Esto se dio suponemos, debido a que la solución se encontraba en la primer rama del árbol. Llegamos a esta conclusión, comparando la profundidad con la cantidad de nodos explotados.

### Cantidad de nodos expandidos

En ambos escenarios, se puede ver que la cantidad de nodos expandidos es **NOTABLEMENTE MENOR**, en las búsquedas informadas. Esto se debe al criterio que toman para elegir el camino a seguir y a la heurística elegida. La salvedad esta con el algoritmo de **DFS** de **ALL BLUE** por lo mencionado anteriormente.

### General

Para nuestro problema , el algoritmo **GREEDY** fue más eficiente para encontrar la solución.

### Nota

Para nuestro problema, nos solicitaron hacer el análisis con un tablero de 7x7. Este llegaba al rango de las horas en tiempo de ejecución, por lo que no llegamos a resultados para comparar. Los presentados en este informe, son en base a un análisis en un tablero de 5x5. Si bien sabemos que contamos con un error o una validación que no estamos teniendo en cuenta, los resultados obtenidos fueron coherentes con lo visto en clase.