

# Sistemas de Inteligencia Artificial

## TPE 1: Métodos de Búsqueda No Informados e Informados

Grupo 2

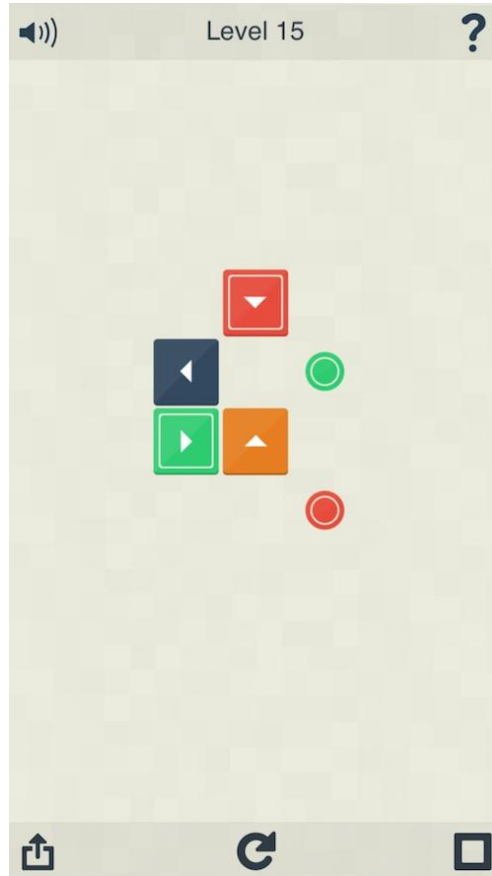
José Torreguitar - 57519

Tomas Soracco - 56002

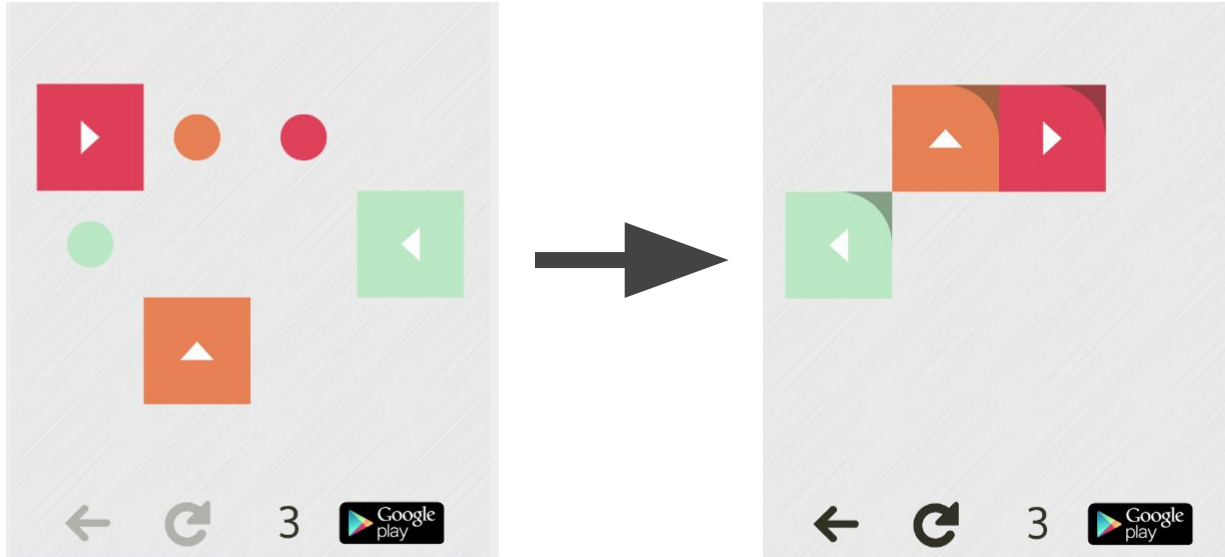
Sofía Picasso - 57700

# Simple Squares

Objetivo: Empujar  
los cuadrados  
hasta el punto de  
su mismo color



# Estado inicial y estado solución



# Representación del Problema

- Estados: Matriz  $N \times N$  que representa el tablero, array de cuadrados con su posición en el momento, su dirección y su posición meta, y la posición de los puntos de cambio de dirección.
- Acciones posibles: mover algunos de los cuadrados en la dirección de su flecha, siempre y cuando ese movimiento no los saque de la matriz.
- Condición de terminación: lograr que cada cuadrado termine en su posición meta.



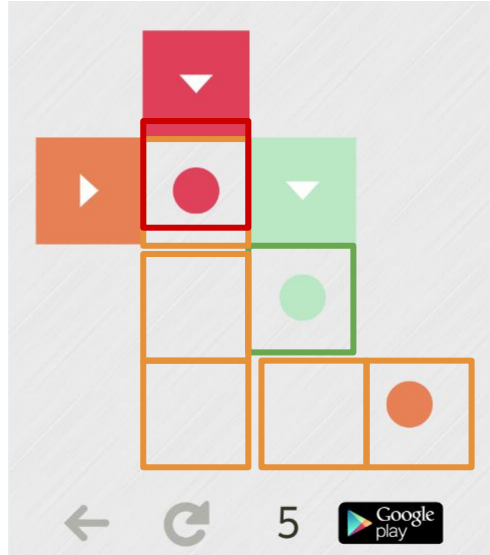
# Función de Costo

- Se decidió utilizar una única función de costo.
- El desplazamiento de un cuadrado en la dirección de su flecha tiene un costo de 1.
- Los cuadrados no pueden desplazarse fuera del tablero.
- En el caso de que un cuadrado empuje a uno o más cuadrados al ser movido, este movimiento también tendrá un costo de 1, ya que solo se tiene en cuenta el movimiento del primer cuadrado y no el del resto que se mueven en consecuencia.



# Heurística 1: HeuristicDistance

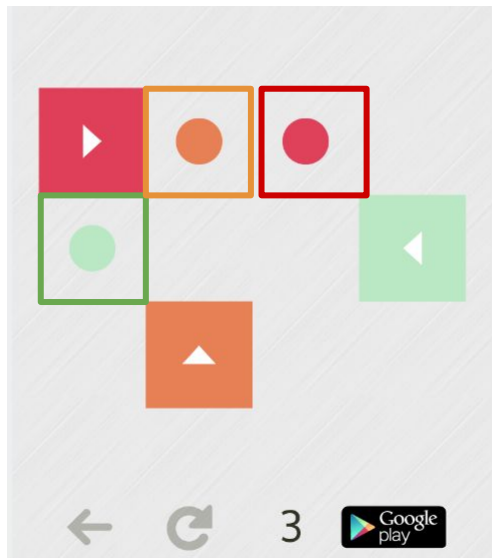
Sumatoria de las distancias a la cual cada cuadrado se encuentra de su posición meta.



En este ejemplo, la heurística sería igual a 7

## Heurística 2: HeuristicInPlace

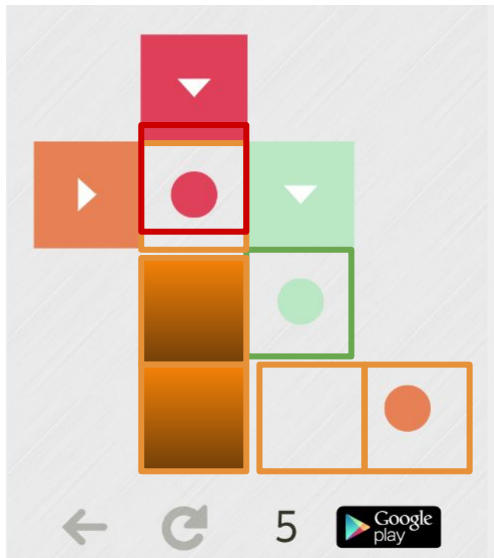
Cuenta la cantidad de cuadrados que **no** se encuentran en su posición meta.



En este ejemplo, la heurística sería igual a 3

# Heurística 3: HeuristicWeightedDistance

Sumatoria de las distancias a la cual cada cuadrado se encuentra de su posición meta, pero en esta heurística, se suma 1 en el caso de que esa distancia sea en la misma dirección que la flecha del cuadrado y 3 de lo contrario.



En este ejemplo, la heurística sería igual a 11



# Problemas con las heurísticas

## Heurística 1:

La distancia puede ser una medida engañosa ya que un cuadrado a una posición de su meta puede estar moviéndose en dirección distinta a donde se encuentra tal meta. No es admisible.

## Heurística 2:

Dado que el caso de llevar un cuadrado a su meta no suele suceder muy a menudo, la heurística no suele cambiar, y en la mayoría de los casos el algoritmo deberá operar eligiendo el próximo estado aleatoriamente.

## Heurística 3:

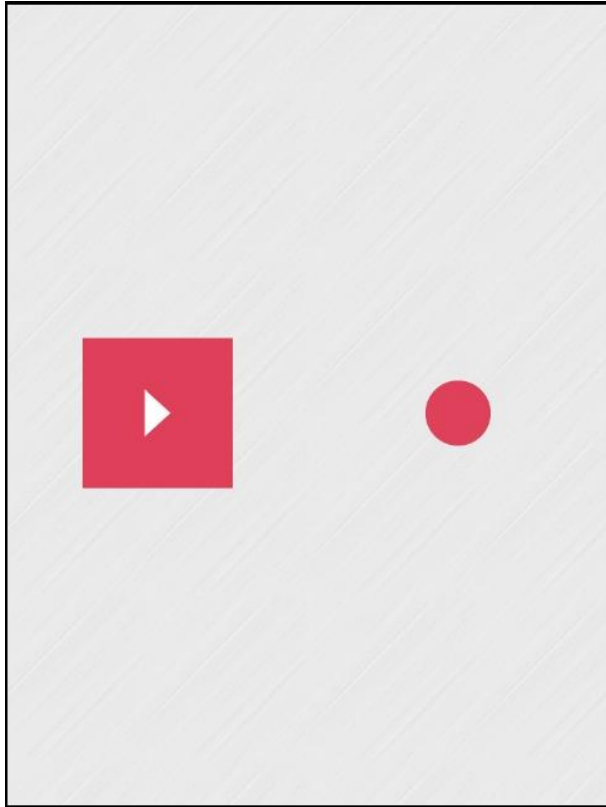
No tiene en cuenta la posición de los puntos que cambian la dirección de la flecha del cuadrado, ni de los otros cuadrados que pueden empujarse entre sí. No es admisible.



# RESULTADOS

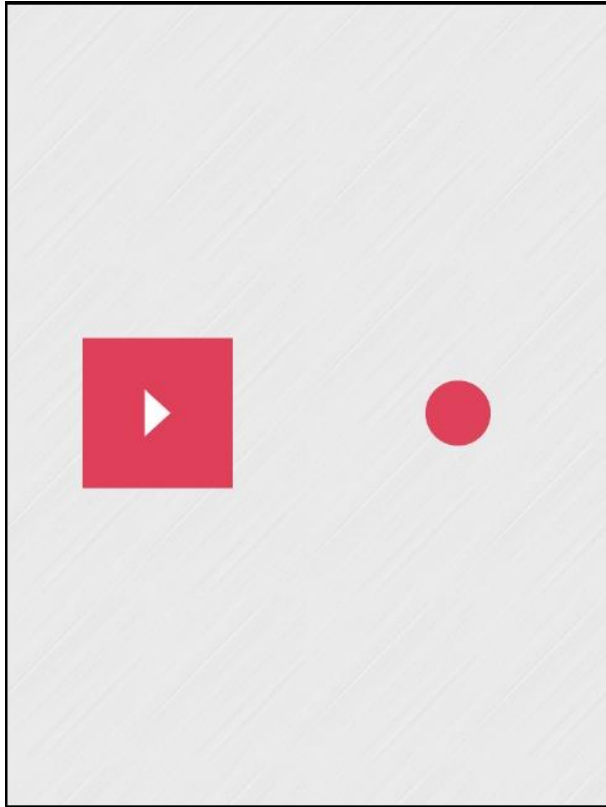


# Tablero 1



	Altura	Nodos Frontera	Nodos Expandidos	Tiempo	Costo
BFS	2	1	2	0,013	2
DFS	2	1	2	0,001	2
IDDFS	2	1	4	0	2
A* HD	2	1	2	0,004	2
a* HIP	2	1	2	0	2
a* HWD	2	1	2	0	2
G HD	2	1	2	0,001	2
G HIP	2	1	2	0	2
G HWD	2	1	2	0	2

# Tablero 1

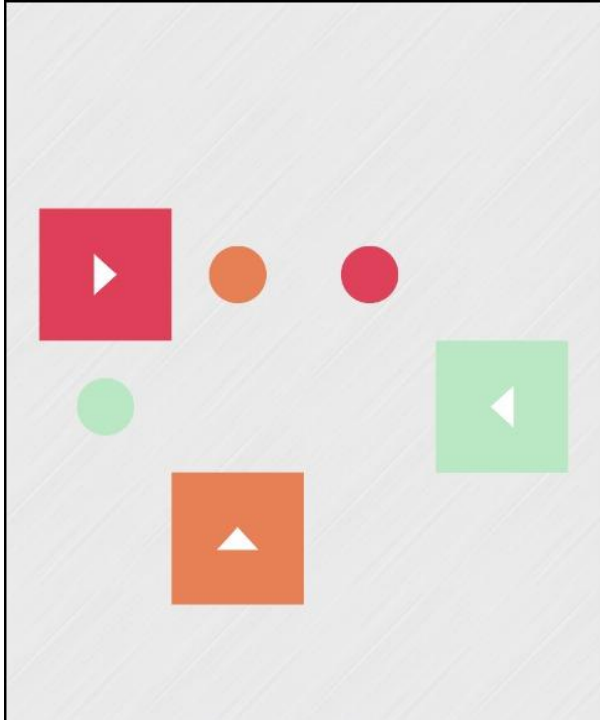


## Conclusiones:

Dado la simpleza del problema (solo hay un movimiento disponible), todos los algoritmos tienen resultados similares con la excepción del algoritmo IDDFS, el cual se reinicia en cada nivel de profundidad y por esta razón tiene más explosiones que el resto.

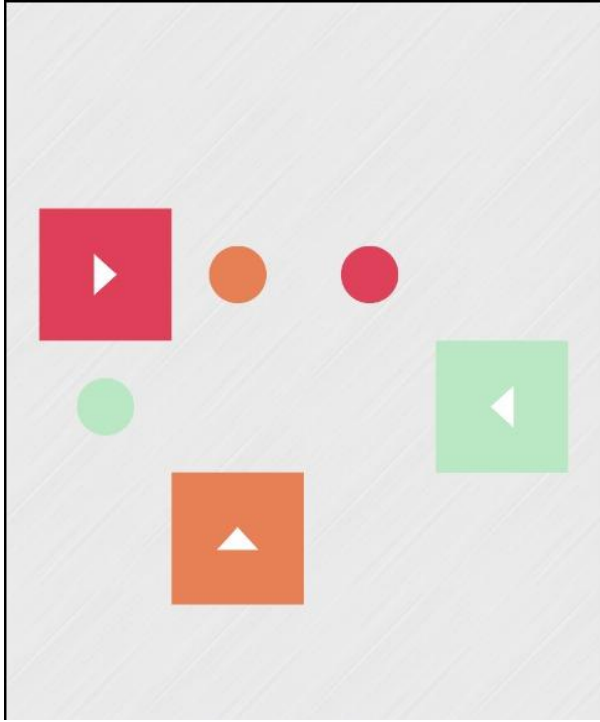


# Tablero 3



	Altura	Nodos Frontera	Nodos Expandidos	Tiempo	Costo
BFS	7	783	724	0,11	7
DFS	7	7	155	0,002	7
IDDFS	7	7	1221	0,116	7
A* HD	7	15	7	0	7
a* HIP	7	186	111	0,034	7
a* HWD	7	289	262	0,028	7
G HD	7	15	7	0	7
G HIP	7	15	7	0,001	7
G HWD	7	18	20	0	7

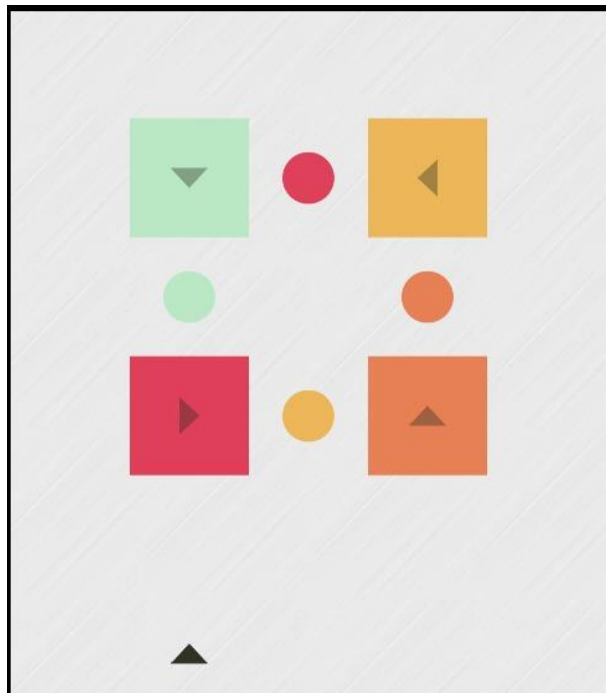
# Tablero 3



## Conclusiones:

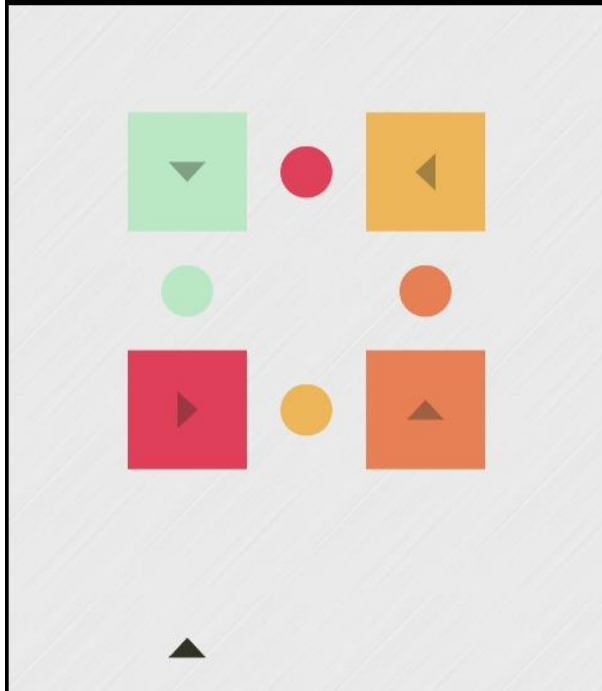
En este caso pueden notarse las diferencias entre los algoritmos. IDDFS continúa siendo el algoritmo con más nodos expandidos. Los algoritmos informados lograron llegar a la solución expandiendo la menor cantidad de nodos en general, especialmente A\*

# Tablero 21



	Altura	Nodos Frontera	Nodos Expandidos	Tiempo	Costo
BFS	23	15443	89582	0,785	23
DFS	2252	2674	190761	1,872	2252
IDDFS	23	6452	77848	4,171	23
A* HD	24	6452	8707	1,170	24
a* HIP	23	13406	40253	9,306	23
a* HWD	24	15114	31808	7,535	24
G HD	50	9846	15933	2,378	50
G HIP	47	10856	22059	1,870	47
G HWD	28	6555	39907	6,907	28

# Tablero 21



## Conclusiones:

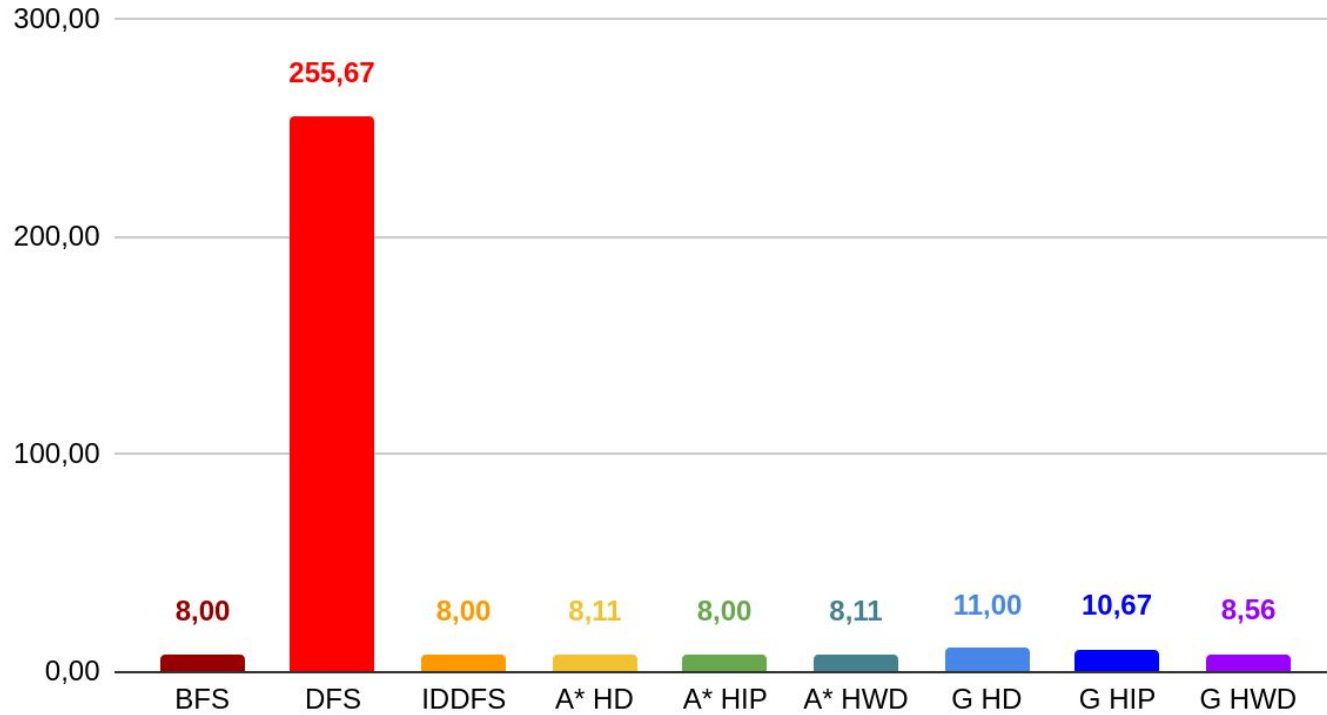
Mientras aumenta la complejidad del problema, más notorias se hacen las diferencias entre los distintos algoritmos. Se denota más la diferencia de nodos expandidos entre los métodos informados y los no informados. También se puede ver que el algoritmo IDDFS tiene menos nodos en frontera ya que tiene su profundidad restringida. A\*, BFS y IDDFS tienen los mejores costos.



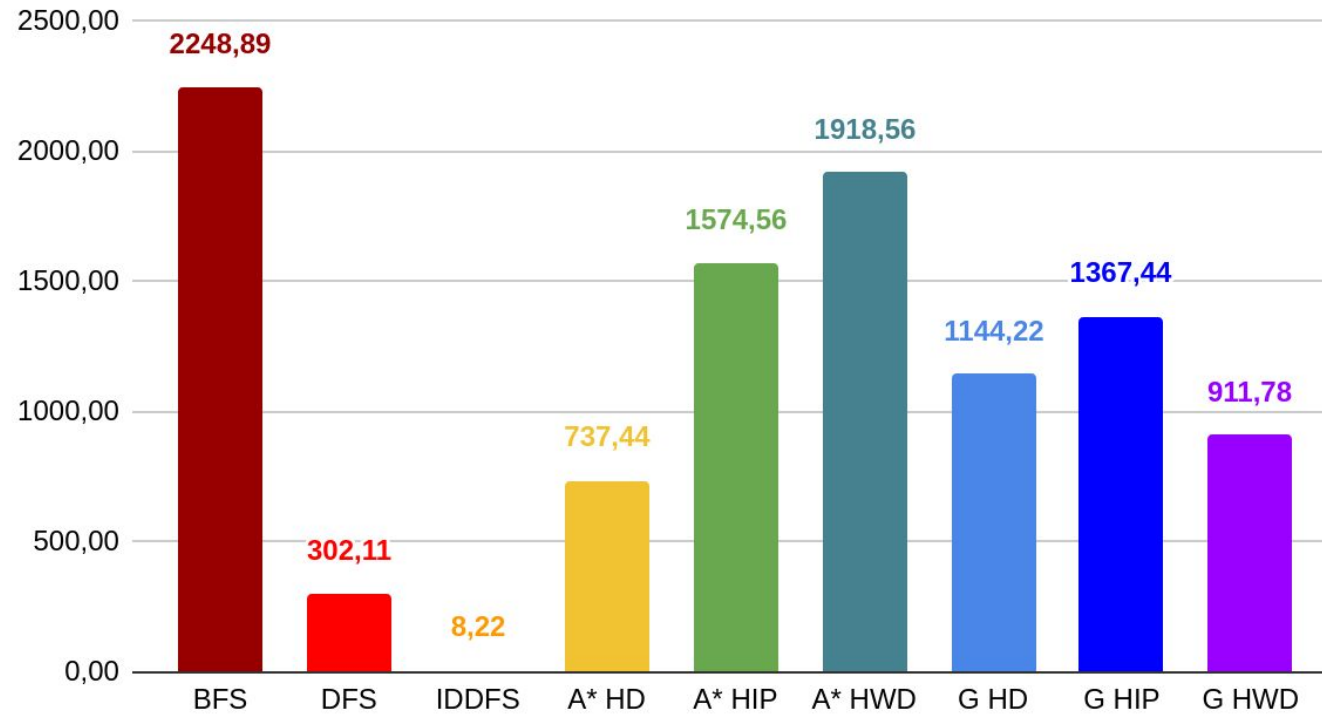
# GRÁFICOS



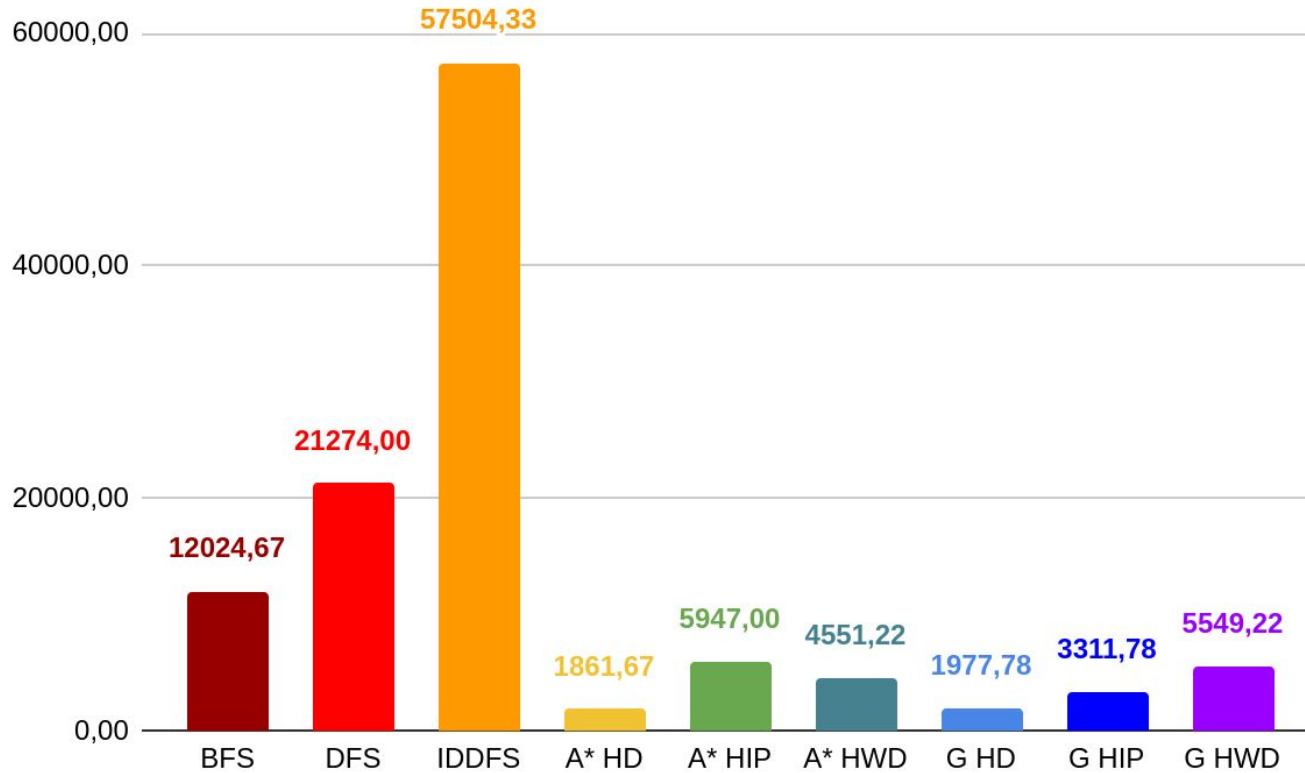
## Altura Promedio



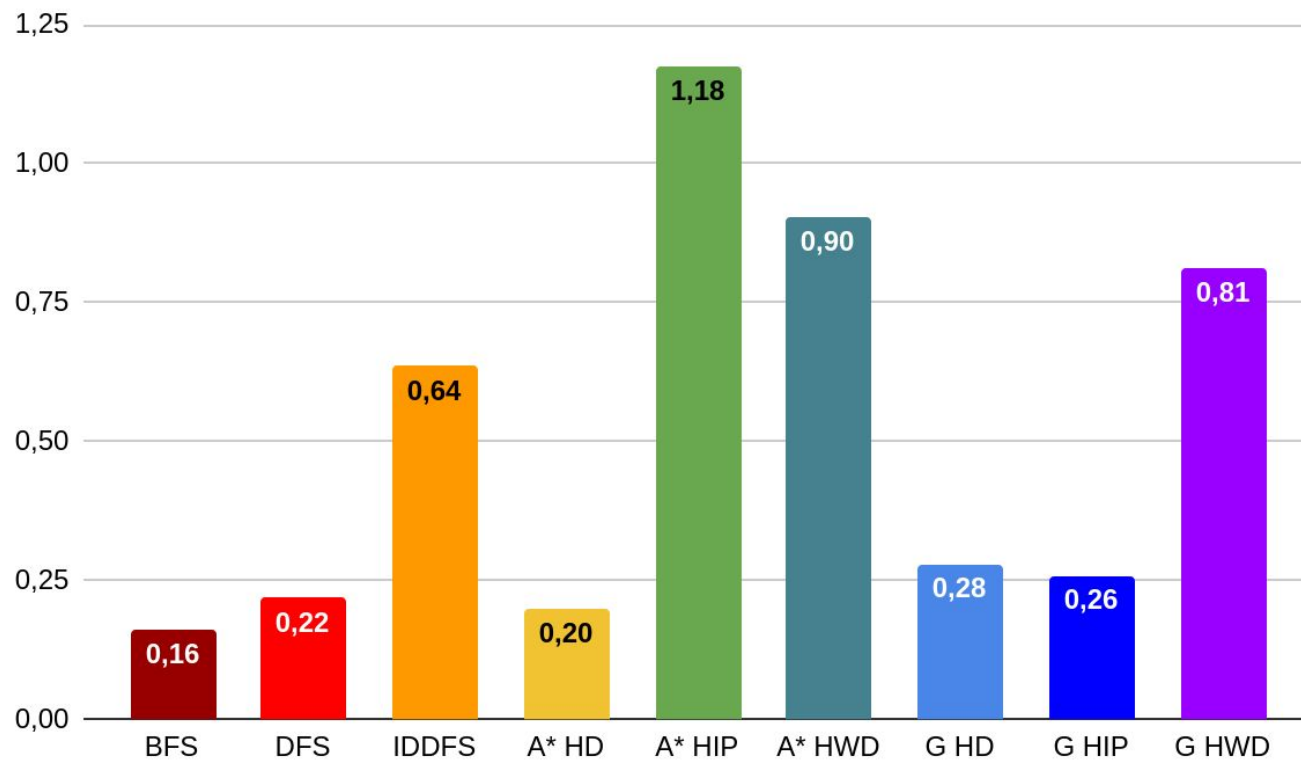
## Nodos Frontera



## Nodos Expandidos



## Tiempo Promedio



## Costo Promedio

