

Fundamentos de Inteligencia Artificial

Práctica 2 Búsqueda heurística. A*

Autor: Víctor Guzmán Pedrajas
DNI: 48717414-X
Turno: Miércoles, 8:30 - 10

Índice

Estructura del código.....	3
Heurísticas.....	4
Heurística trivial.....	4
Distancia Manhattan.....	5
Distancia euclídea (y distancia euclídea al cuadrado).....	5
Aproximación a la distancia euclídea.....	6
Traza de un ejemplo.....	7
Iteración 1.....	7
Iteración 2.....	8
Iteración 3.....	8
Iteración 4.....	8
Iteración 5.....	9
Iteración 6.....	9
Iteración 6.....	9
Iteración 7.....	10
Iteración 8.....	10
Iteración 9.....	10
Iteración 10.....	11
Iteración 11.....	11
Iteración 12.....	11
Iteración 13.....	12
Iteración 14.....	12
Iteración 15.....	12
Iteración 16.....	13
Iteración 17.....	13
Iteración 18.....	13
Iteración 19.....	14
Iteración 20.....	14
Iteración 21.....	14
Resultado final.....	15
Experimentación.....	15
Mapa original.....	15
Mapa 1.....	16
Mapa 2.....	17
Mapa 3.....	18
Mapa 4 (utilizado para la traza).....	18
Mapa 5.....	19
Mapa 6.....	20
Mapa 7.....	21
Mapa 8.....	22
Mapa 9.....	23
Mapa 10.....	24
Resultados de la experimentación.....	24

Estructura del código

Para implementar el algoritmo A*, se ha utilizado como estructuras de datos una lista para la lista interior, y un heap mínimo para la lista frontera. El heap está ordenado según el valor de la f , de forma que siempre devolverá el nodo de menor f .

Además, se ha utilizado una lista auxiliar para la lista frontera, a la que se accederá en el caso de que la lista interior ya contenga el elemento a insertar, para actualizar la g si es necesario. Esta lista auxiliar se ha omitido del pseudocódigo, ya que tiene siempre los mismos elementos que el heap.

Algoritmo

El pseudocódigo del algoritmo es el siguiente:

```
AEstrella
    contadorDeNodos = 0

    interior = vacia
    frontera = inicio

    array_de_expandidos[inicio] = 0

    mientras frontera no esté vacío
        n = obtener y borrar cima del heap // sabemos que es el de
        menor f
        interior.añadir(n)

        si n es meta
            iterador = n
            camino[iterador] = 'X'
            mientras iterador no tenga padre
                iterador = su padre
                camino[iterador] = 'X'
            fmientras

            devolver
        fsi

    array_de_expandidos[inicio] = contadorDeNodos
    contadorDeNodos ++

    generar hijos de n

    para cada hijo de n que no esté en la lista interior
        g2 = n.g + 1

        si lista_frontera contiene al hijo
            hijo_anterior = lista_frontera.obtener(hijo)
            si g2 < hijo_anterior.g
                hijo.padre = n
```

```

        hijo.g = g2
        hijo.f = hijo.g + hijo.h
    fsi
sino
    lista_frontera.añadir(hijo)
fsi
fpara
fmientras

    devolver
fAEstrella

```

Nodos

Los nodos almacenan su posición, que los identifica; los tres valores necesarios para la heurística (f,g y h); una referencia al nodo padre (null en el nodo inicial) y una lista de hijos.

A la hora de generar los hijos, se genera primero el hijo de la derecha, después el de arriba, después el de abajo y después el de la izquierda. La eficiencia de esta generación depende del mapa concreto en el que se esté, aunque dado que el robot siempre empieza en la izquierda y la meta está a la derecha, el hijo derecho debería generarse antes que el izquierdo, independientemente del mapa.

La función de comparación necesaria para el heap ordena según la f del nodo y, si la f es igual, da prioridad a los nodos que estén más a la derecha.

Al generar los nodos podemos encontrarnos los siguientes casos:

- Es una pared o un obstáculo. El nodo no se genera y se pasa a los siguientes.
- Es un nodo ya recorrido. El nodo se genera y se añade a la lista de hijos del nodo actual, aunque al ir a añadirlo a la lista frontera, vemos que ya está en la interior, por lo que no se añade.
- Es un nodo no recorrido. El nodo se genera y se añade a la lista de hijos del nodo actual. Al ir a añadirlo a la lista frontera, pueden darse dos casos:
 - Es hijo de otro nodo ya recorrido. Se compara la longitud del camino recorrido para llegar al nodo. Si el nuevo camino es menor que el antiguo, se modifica y se actualiza f. Se actualiza el heap para mantener la ordenación.
 - No es hijo de otro nodo ya recorrido. Se añade a la lista frontera. La función de comparación utilizada por el heap asegura que se ordenarán correctamente según la f.

Heurísticas

El mejor nodo a elegir de entre los de la lista frontera se elige mediante el valor f, que corresponde a la suma del camino recorrido hasta el nodo (g) más una heurística que estima la distancia a la meta.

Conforme la estimación se va acercando al valor real, el algoritmo se encamina más rápidamente al nodo destino, teniendo que explorar menos nodos.

Heurística trivial

La heurística más trivial que existe es presuponer que la distancia a la meta es 0. De esta forma, se ordenaría los nodos según el camino recorrido, buscando simplemente el camino más corto, pero sin conocer el destino, encontrando el destino mediante una búsqueda completa.

Como ejemplo, en el mapa original, esta heurística generaría 179 nodos. Se puede usar el valor devuelto por esta heurística como línea base para comparar la calidad de otras heurísticas.

Distancia Manhattan

La primera heurística utilizada es la distancia Manhattan entre el nodo que se está analizando y el destino. Esta distancia consiste en calcular la distancia horizontal sumada a la distancia vertical o, matemáticamente:

$$h = |(Y_B - Y_A)| + |(X_B - X_A)|$$

En este ejemplo, la distancia Manhattan entre A y B sería de 14: 11 en horizontal y 3 en vertical.

A _ _ _ _ _
_ _ _ _ _
_ _ _ _ _
_ _ _ _ _ B

Utilizando esta heurística se generan 47 nodos en el mapa original. En la sección de experimentación se pueden ver los resultados de esta heurística con más mapas.

Distancia euclídea (y distancia euclídea al cuadrado)

Otra posible heurística es la distancia euclídea entre el nodo que se está analizando y el destino. Esta distancia consiste en calcular la distancia en línea recta, sin tener en cuenta casillas ni direcciones. Matemáticamente, esta distancia se puede calcular de la siguiente forma:

$$\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

El mayor problema de esta heurística es que obliga a utilizar números reales para poder representarla correctamente, con la consiguiente pérdida de eficiencia. Para evitar esto, se puede considerar la distancia euclídea al cuadrado, que se puede representar con números enteros.

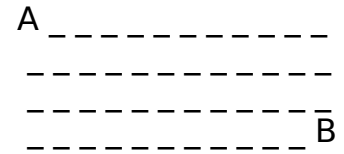
Utilizando la distancia euclídea al cuadrado no obtenemos la distancia real, pero para el propósito que se utiliza en el algoritmo es perfectamente válida. A cambio obtenemos un mayor riesgo de *overflow* en nuestros registros, por lo que se recomienda utilizar enteros en formato long. Sin embargo, esto hace que el valor de

g quede prácticamente despreciado, pudiendo llevar a resultados erróneos al no tenerse en cuenta la longitud del camino recorrido.

El valor de f quedaría como:

$$f = g + ((x_B - x_A)^2 + (y_B - y_A)^2)$$

Siguiendo el ejemplo anterior, la distancia euclídea entre los puntos A y B sería 11'4018, mientras que la distancia euclídea al cuadrado sería 130.



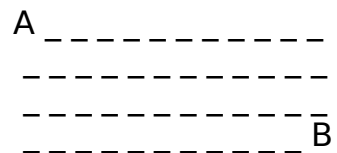
Utilizando esta heurística se generan 171 nodos en el mapa original. En la sección de experimentación se pueden ver los resultados de esta heurística con más mapas.

Aproximación a la distancia euclídea

Una tercera posible heurística es una distancia que aproxime la distancia euclídea. Esta aproximación se basa en dar un peso a cada movimiento en el plano. De esta forma, los cálculos de la distancia horizontal o vertical, tienen un peso de 10, mientras que los pesos de la distancia diagonal, tienen un peso de 14 (estos números vienen de asignar 1 y $\sqrt{2}$ respectivamente, es decir, las dimensiones de un triángulo rectángulo de catetos = 1, y multiplicar por 10 para aproximar los decimales).

Así, la fórmula de f sería $f = g + h$ siendo h:

$$h = 14 * \min(\text{distanciaX}, \text{distanciaY}) + 10 * |(distanciaX - distanciaY)|$$

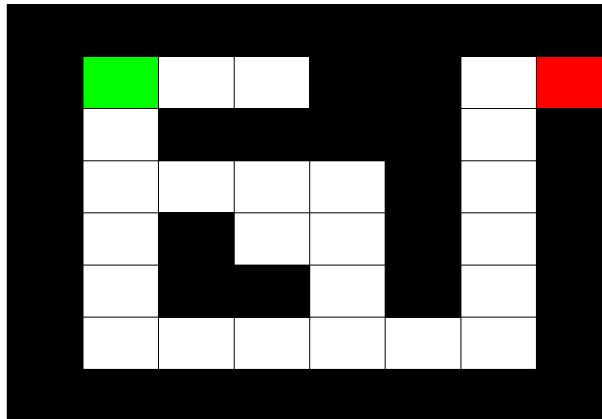


Siguiendo el ejemplo, la aproximación a la distancia euclídea entre los puntos A y B sería de 122.

Utilizando esta heurística se generan 32 nodos en el mapa original. En la sección de experimentación se pueden ver los resultados de esta heurística con más mapas.

Traza de un ejemplo

Para ilustrar el ejemplo se va a utilizar el siguiente mapa:



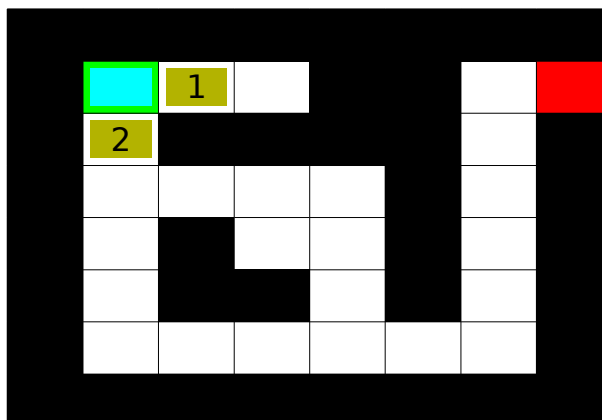
La heurística utilizada para la traza ha sido la distancia Manhattan.
La leyenda del mapa es la siguiente:

■	Muro / obstáculo
■	Inicio
■	Meta
■	Lista interior
■	Lista frontera
■	Nodo en evaluación
■	Nodo hijo ya en la lista interior
■	Nodo hijo ya en la lista frontera

A continuación se encuentra la traza del mapa de ejemplo:

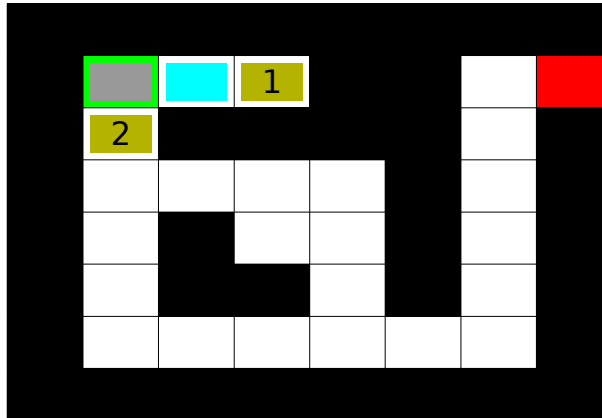
Iteración 1

El nodo inicial genera 2 hijos, que se añaden a la lista frontera, primero el derecho y después el de abajo.



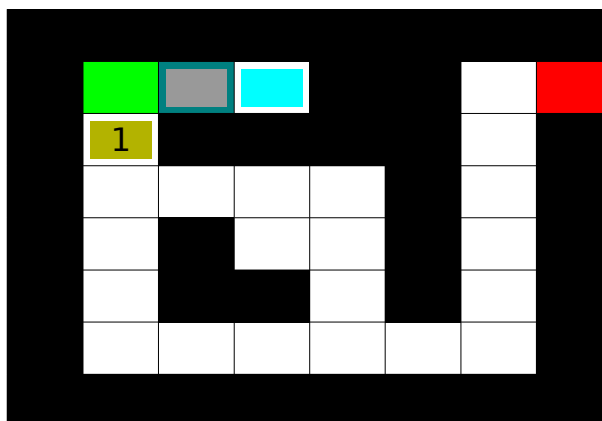
Iteración 2

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera. Debido a la f del nodo añadido, se pone por delante del que ya estaba en la lista.



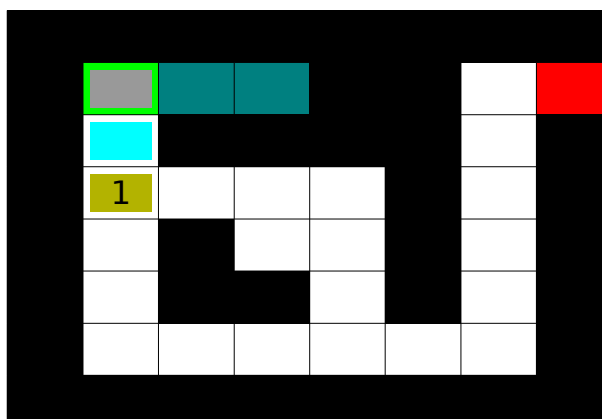
Iteración 3

El nodo que se evalúa genera 1 hijo, que ya está en la lista interior.



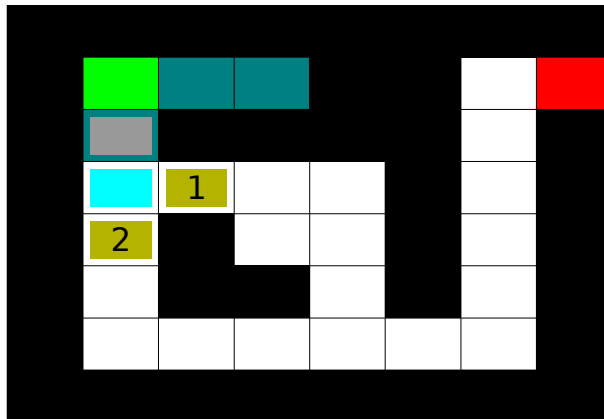
Iteración 4

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



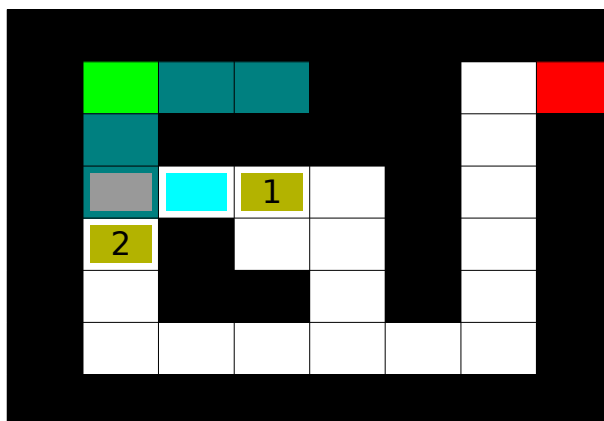
Iteración 5

El nodo que se evalúa genera 3 hijos. Uno está ya en la lista interior, los otros dos se añaden a la lista frontera.



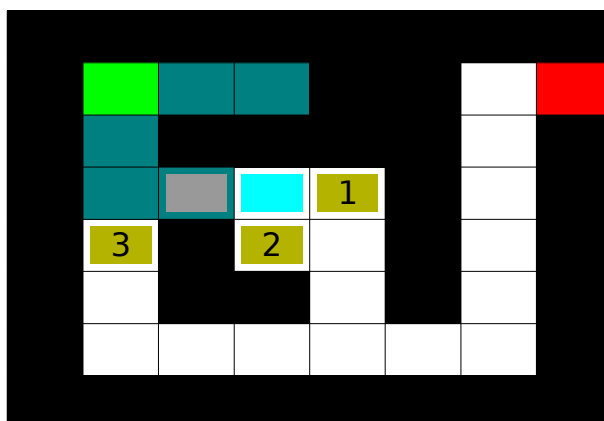
Iteración 6

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera. Debido a la f del nodo añadido, se pone por delante del que ya estaba en la lista.



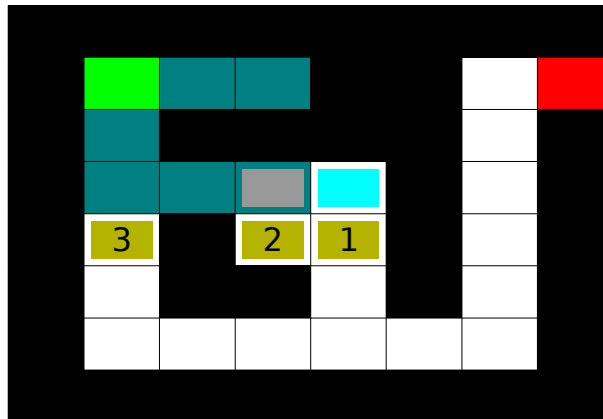
Iteración 6

El nodo que se evalúa genera 3 hijos. Uno está ya en la lista interior, los otros dos se añaden a la lista frontera.



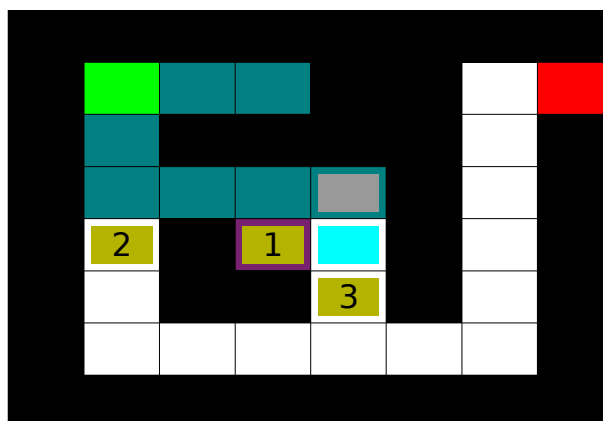
Iteración 7

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



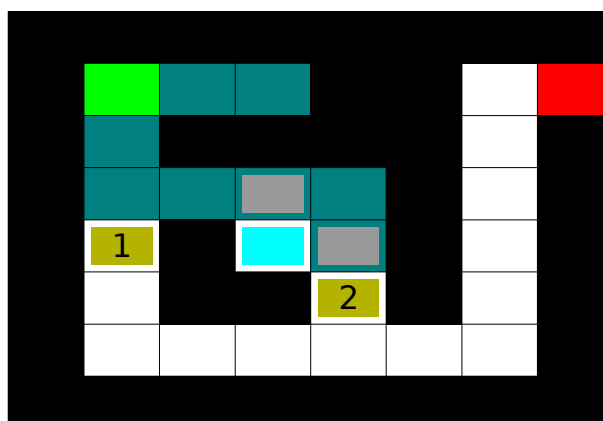
Iteración 8

El nodo que se evalúa genera 3 hijos. Uno ya se encuentra en la lista interior, el otro ya se encuentra en la lista frontera pero, como tiene una g menor, no se actualiza; el otro no se encuentra en ninguna lista, por lo que se añade a la lista frontera.



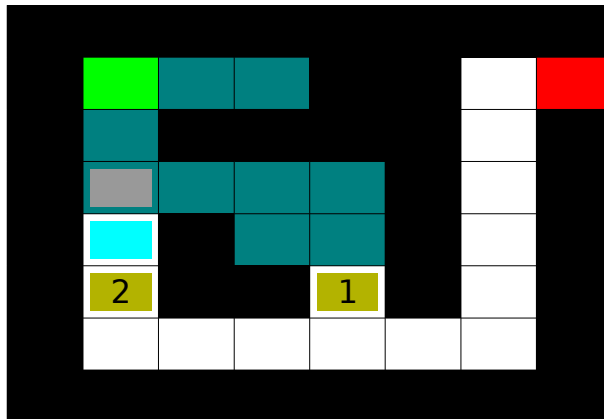
Iteración 9

El nodo que se evalúa genera 2 hijos. Ambos están ya en la lista interior, por lo que no se hace nada con ellos.



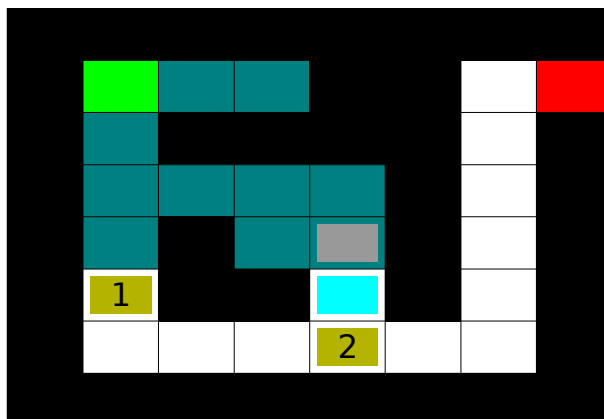
Iteración 10

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



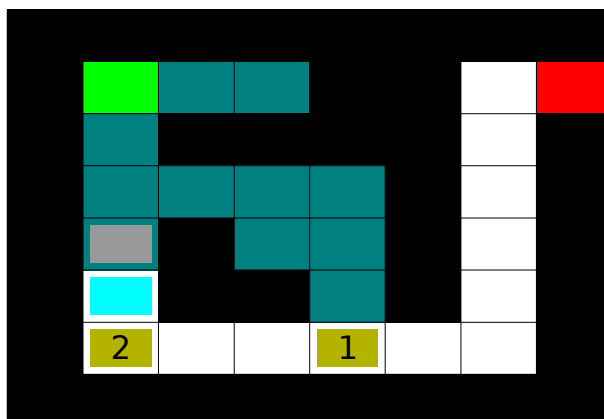
Iteración 11

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



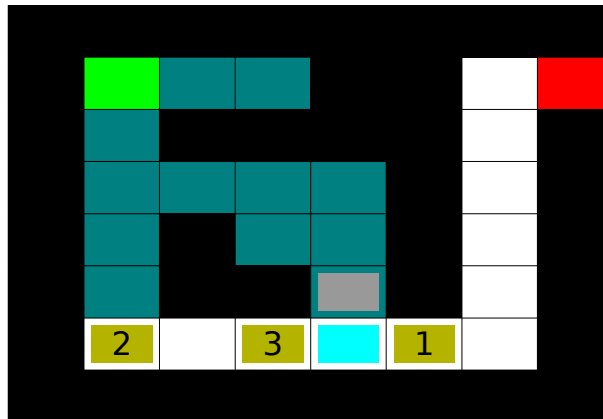
Iteración 12

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



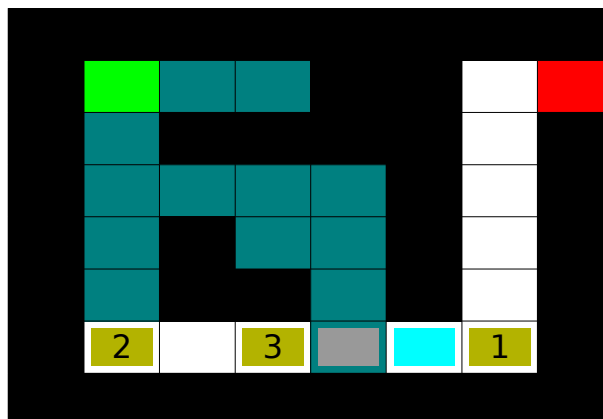
Iteración 13

El nodo que se evalúa genera 3 hijos. Uno está ya en la lista interior, los otros dos se añaden a la lista frontera.



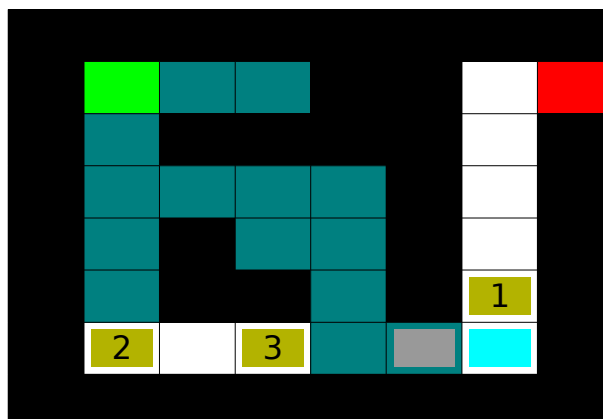
Iteración 14

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



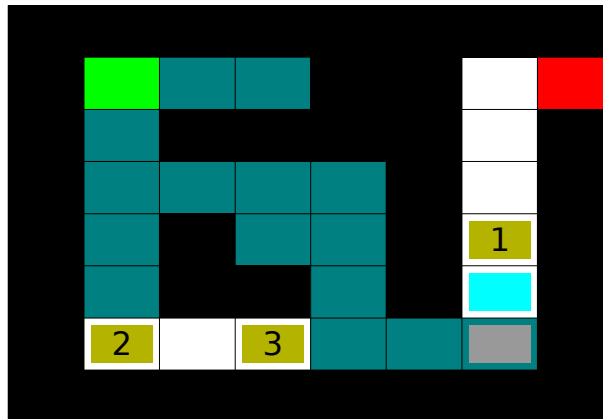
Iteración 15

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



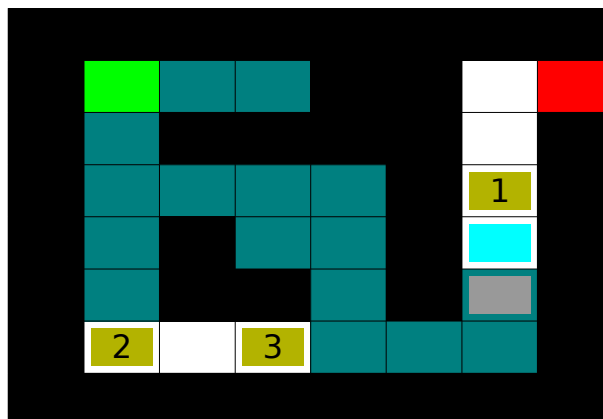
Iteración 16

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



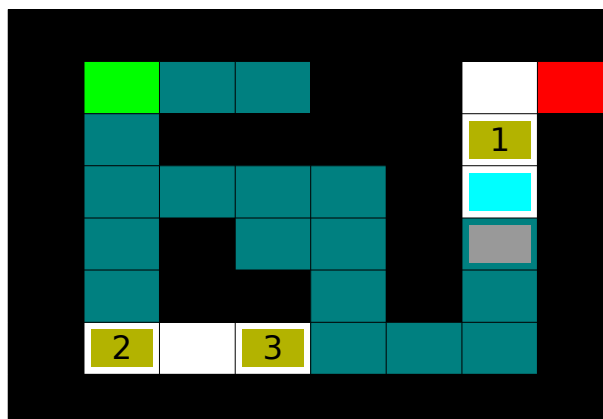
Iteración 17

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



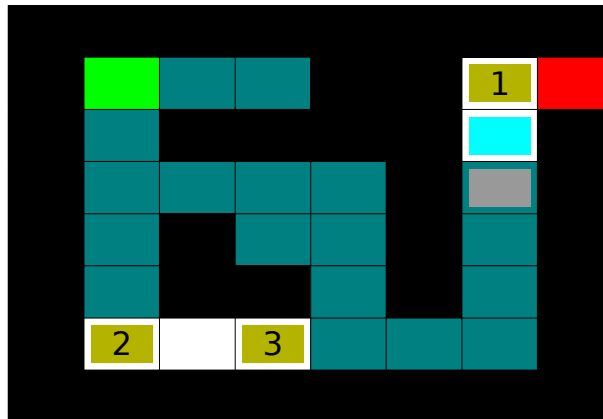
Iteración 18

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



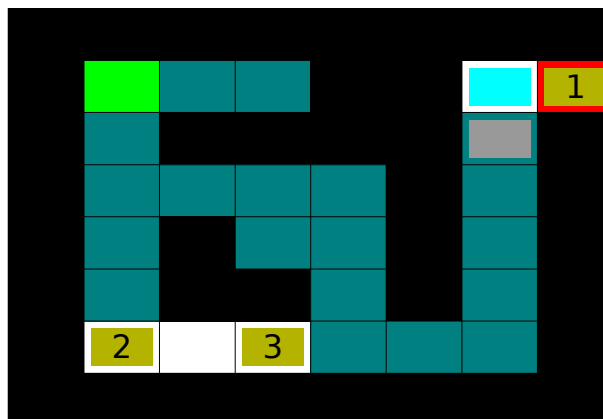
Iteración 19

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



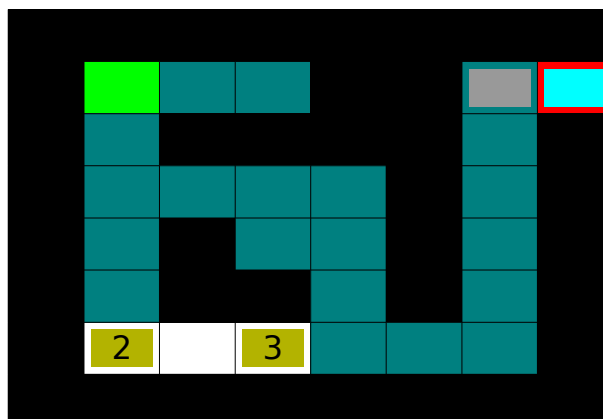
Iteración 20

El nodo que se evalúa genera 2 hijos. Uno está ya en la lista interior, el otro se añade a la lista frontera.



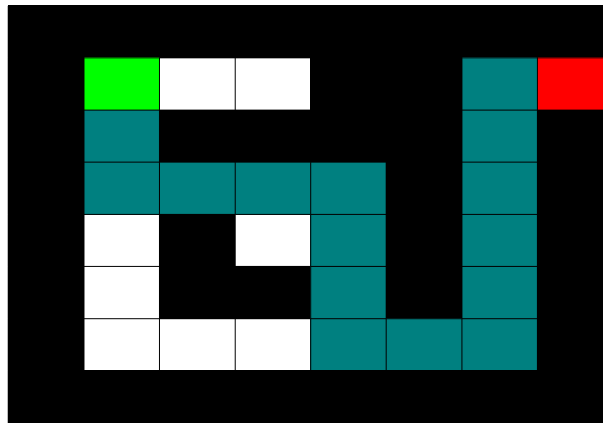
Iteración 21

El nodo que se evalúa es el nodo destino. Se termina el algoritmo y se recorre el camino generado a través de los padres de cada nodo.



Resultado final

Según el ejemplo, el camino encontrado es:



Experimentación

Para realizar la experimentación se han comparado la cantidad de nodos generados por las tres heurísticas utilizadas con los de la heurística trivial, obteniendo así una mejora con respecto a la línea base.

Las ejecuciones se han realizado con una serie de mapas inventados para este propósito, intentando probar el algoritmo de la forma más exhaustiva posible.

Mapa original

Se trata del mapa proporcionado en el enunciado de la práctica:

[illegible]

En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 179 nodos.
- Distancia Manhattan: 47 nodos.

$$Mejora = \left(1 - \frac{47}{179}\right) * 100 = 73,74 \%$$

- Distancia euclídea: 28 nodos.

$$Mejora = \left(1 - \frac{28}{179}\right) * 100 = 84,36 \%$$

- Aproximación euclídea: 32 nodos.

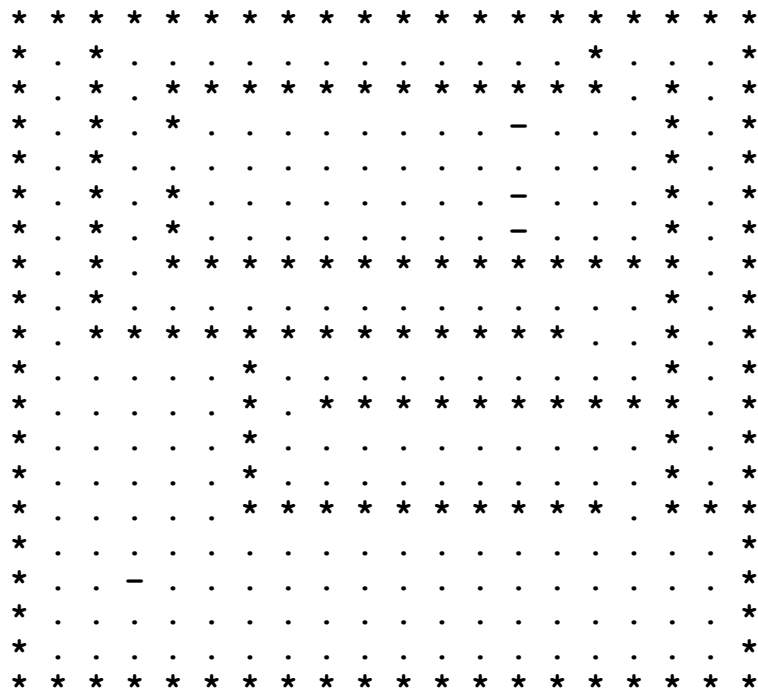
$$Mejora = \left(1 - \frac{32}{179}\right) * 100 = 82,12 \%$$

Mapa 1

20

1

10



En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 230 nodos.
- Distancia Manhattan: 218 nodos.

$$Mejora = \left(1 - \frac{218}{230}\right) * 100 = 5,21 \%$$

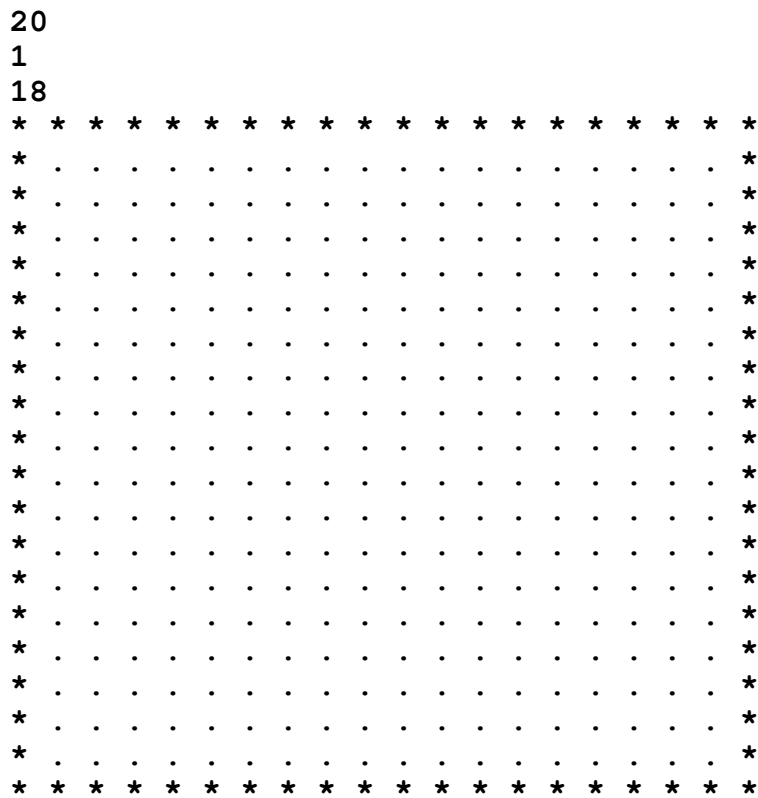
- Distancia euclídea: 191 nodos.

$$Mejora = \left(1 - \frac{191}{230}\right) * 100 = 16,96 \%$$

- Aproximación euclídea: 200 nodos.

$$Mejora = \left(1 - \frac{200}{230}\right) * 100 = 13,04 \%$$

Mapa 2



En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 324 nodos.
- Distancia Manhattan: 35 nodos.

$$Mejora = \left(1 - \frac{35}{324}\right) * 100 = 89,2 \%$$

- Distancia euclídea: 35 nodos.

$$Mejora = \left(1 - \frac{35}{324}\right) * 100 = 89,2 \%$$

- Aproximación euclídea: 35 nodos.

$$Mejora = \left(1 - \frac{35}{324}\right) * 100 = 89,2 \%$$

Mapa 3

```
10
1
1
* * * * * * * * *
* . . . . . * . *
* . * * * * * * . *
* . . . * . * * . *
* . * . * . . * . *
* . * . * . * * . *
* . * . . . . * . *
* . * * * * * * . *
* . * * * * * * . *
* . . . . . . . *
* * * * * * * * *
```

En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 39 nodos.
- Distancia Manhattan: 39 nodos.

$$Mejora = \left(1 - \frac{39}{39}\right) * 100 = 0 \%$$

- Distancia euclídea: 39 nodos.

$$Mejora = \left(1 - \frac{39}{39}\right) * 100 = 0 \%$$

- Aproximación euclídea: 39 nodos.

$$Mejora = \left(1 - \frac{39}{39}\right) * 100 = 0 \%$$

Mapa 4 (utilizado para la traza)

```
8
1
1
* * * * * * *
* . . . * * . *
* . * * * * . *
* . . . . * . *
* . * . . * . *
* . * * . * . *
* . . . . . *
* * * * * * *
```

En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 24 nodos.
- Distancia Manhattan: 21 nodos.

$$Mejora = \left(1 - \frac{21}{24}\right) * 100 = 12,5 \%$$

- Distancia euclídea: 19 nodos.

$$Mejora = \left(1 - \frac{19}{24}\right) * 100 = 20,83 \%$$

- Aproximación euclídea: 19 nodos.

$$Mejora = \left(1 - \frac{19}{24}\right) * 100 = 20,83 \%$$

Mapa 5

```
5
1
3
* * * * *
* . . . *
* . - . *
* . . . *
* * * * *
```

En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 8 nodos.
- Distancia Manhattan: 5 nodos.

$$Mejora = \left(1 - \frac{5}{8}\right) * 100 = 37,5 \%$$

- Distancia euclídea: 5 nodos.

$$Mejora = \left(1 - \frac{5}{8}\right) * 100 = 37,5 \%$$

- Aproximación euclídea: 5 nodos.

$$Mejora = \left(1 - \frac{5}{8}\right) * 100 = 37,5 \%$$

Mapa 6

Este mapa contiene dos caminos posibles, uno muchísimo más largo que el otro.

20

1

18

```
* * * * * * * * * * * * * * * * * *
* . . . * . . . * . . . * . . . *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . . . * *
* . * . * . * . * . * . * . . . * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
* . * . * . * . * . * . * . * * * *
```

En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 68 nodos.
- Distancia Manhattan: 54 nodos.

$$Mejora = \left(1 - \frac{54}{68}\right) * 100 = 20,59 \%$$

- Distancia euclídea: 149 nodos. Esta heurística, dado que a efectos prácticos desprecia el valor de g, ha tomado el camino más largo.

$$Mejora = \left(1 - \frac{149}{68}\right) * 100 = -119,12 \%$$

- Aproximación euclídea: 67 nodos.

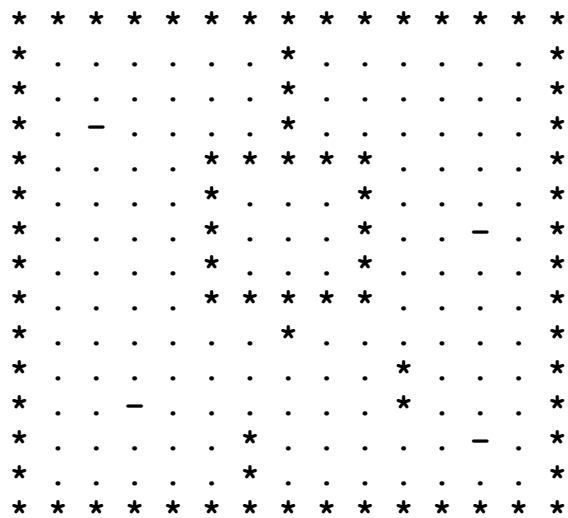
$$Mejora = \left(1 - \frac{67}{68}\right) * 100 = 1,47 \%$$

Mapa 7

15

3

7



En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 104 nodos.
- Distancia Manhattan: 54 nodos.

$$Mejora = \left(1 - \frac{54}{104}\right) * 100 = 48,08 \%$$

- Distancia euclídea: 23 nodos.

$$Mejora = \left(1 - \frac{23}{104}\right) * 100 = 77,88 \%$$

- Aproximación euclídea: 23 nodos.

$$Mejora = \left(1 - \frac{23}{104}\right) * 100 = 77,88 \%$$

Mapa 8

20

1

18

```

* * * * * * * * * * * * * * * * * *
* . * . . . . . . . . . . . . . * . *
* . * * * . . . . . . . . . * * * . *
* . . . * . . . . . . . . . * . . . *
* . . . * * * . . . . . . . * * * . *
* . . . . . * . . . . . . . * . . . *
* . . . . . * * * . . . * * * . . . *
* . . . . . . * . . . * . . . . . . *
* . . . . . . * . . . * . . . . . . *
* - . . . - . . . * * . . . - . . . *
* . - . - - . . . * * . . . . - . . *
* - . . . - . . . * * . - . - . . - *
* . . . . . . . . . . . . . - . . . *
* . . . . . . . . . * * . - . - . . - *
* . . . . . . . . . * * . . . . - . . *
* . . . . . . . . . * * . - . - . . - *
* . . . . . . . . . . . . . - . . . *
* . . . . . . . . . * * . . . - . . . *
* . . . . . . . . . * * . . . . . . . *
* * * * * * * * * * * * * * * * * *

```

En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 193 nodos.
- Distancia Manhattan: 36 nodos.

$$Mejora = \left(1 - \frac{36}{193}\right) * 100 = 81,35 \%$$

- Distancia euclídea: 35 nodos.

$$Mejora = \left(1 - \frac{35}{193}\right) * 100 = 81,86 \%$$

- Aproximación euclídea: 35 nodos.

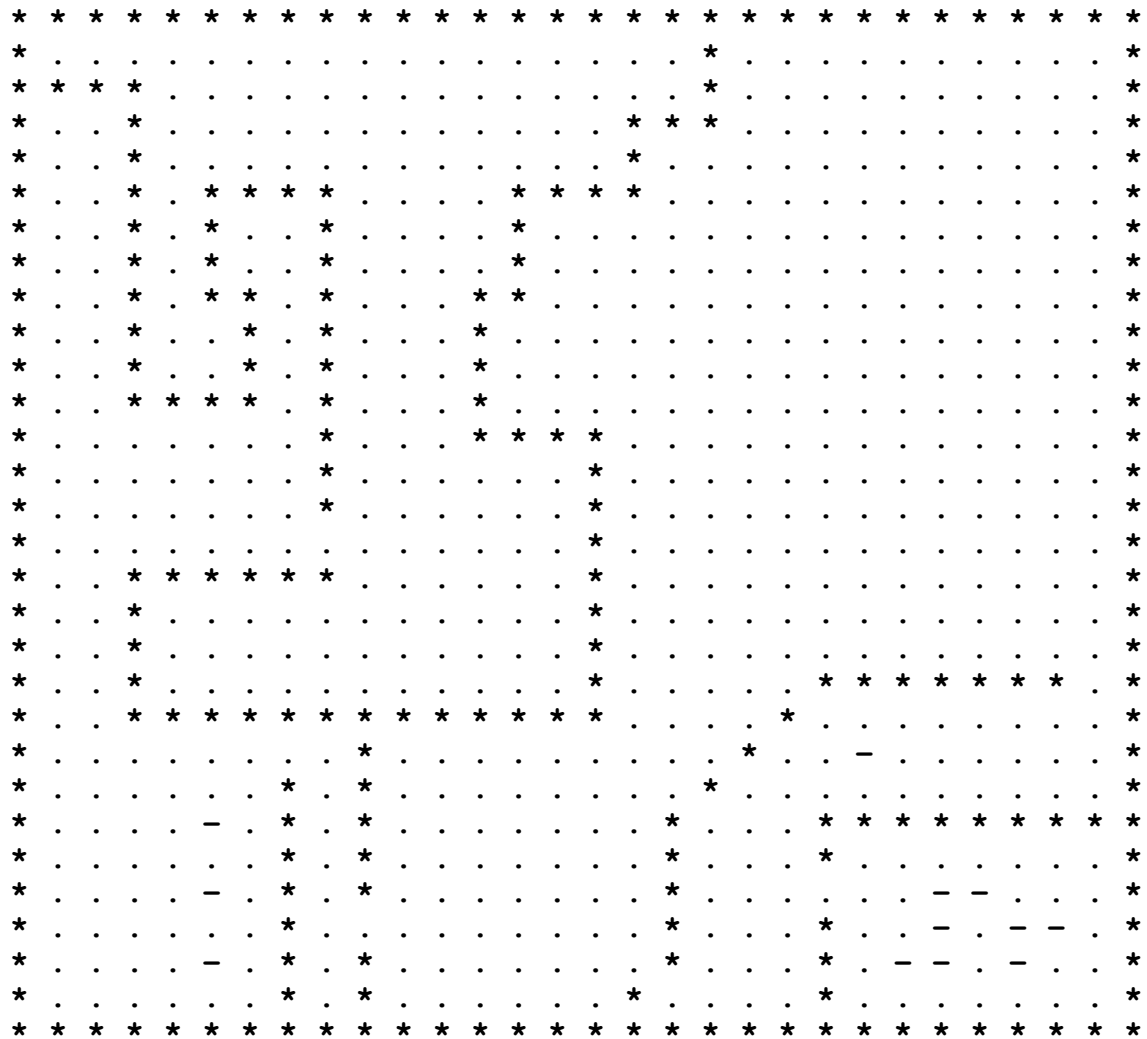
$$Mejora = \left(1 - \frac{35}{193}\right) * 100 = 81,86 \%$$

Mapa 9

30

1

28



En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 644 nodos.
- Distancia Manhattan: 489 nodos.

$$Mejora = \left(1 - \frac{489}{644}\right) * 100 = 24,07 \%$$

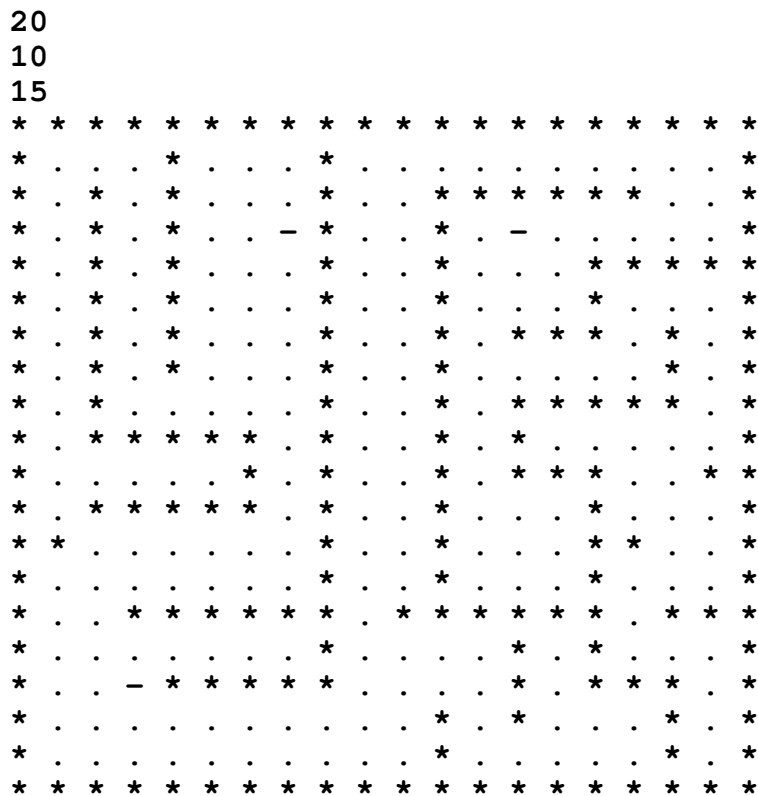
- Distancia euclídea: 228 nodos.

$$Mejora = \left(1 - \frac{228}{644}\right) * 100 = 64,6 \%$$

- Aproximación euclídea: 264 nodos.

$$Mejora = \left(1 - \frac{264}{644}\right) * 100 = 59,01 \%$$

Mapa 10



En este mapa se han obtenido los siguientes resultados:

- Heurística trivial: 207 nodos.
- Distancia Manhattan: 205 nodos.

$$Mejora = \left(1 - \frac{205}{207}\right) * 100 = 0,97 \%$$

- Distancia euclídea: 181 nodos.

$$Mejora = \left(1 - \frac{181}{207}\right) * 100 = 12,56 \%$$

- Aproximación euclídea: 197 nodos.

$$Mejora = \left(1 - \frac{197}{207}\right) * 100 = 4,83 \%$$

Resultados de la experimentación

La distancia Manhattan ha obtenido una mejora media del 35,75%.

La distancia euclídea ha obtenido una mejora media del 33,33%.

La distancia euclídea aproximada ha obtenido una mejora media del 42,52%.

La heurística que mejores resultados obtiene es la 3ª, la aproximación a la distancia euclídea. Además, en ninguno de los casos probados ha obtenido una solución no óptima, a diferencia de la distancia euclídea al cuadrado, que ha obtenido una solución totalmente errónea.