



ALGORITMO A*

Ingeniería del Conocimiento

UNIVERSIDAD
COMPLUTENSE
MADRID

Guion de la asignatura de Ingeniería del Conocimiento de cuarto de carrera del grado de
Ingeniería del Software de la Universidad Complutense de Madrid

Iker Burgoa
4ºE INGENIERÍA DEL SOFTWARE

INDICE

INTRODUCCIÓN	2
DETALLES DEL FUNCIONAMIENTO	3
DETALLES DE LA IMPLEMENTACIÓN	3
MANUAL DE USUARIO	4
CASOS DE PRUEBA	6
BIBLIOGRAFÍA	10



UNIVERSIDAD
COMPLUTENSE
MADRID

INTRODUCCIÓN

El algoritmo A* fue presentado por Peter E. Hart, Nils J. Nilsson y Bertram Raphael en el año 1968. A* es un algoritmo de búsqueda en anchura que se emplea para calcular el camino mínimo en una red. Se trata de un algoritmo heurístico, ya que una de sus principales características es el uso de una función de evaluación heurística, donde etiquetará a los distintos nodos de la red y servirá para determinar la probabilidad para conseguir el camino óptimo siempre que sea posible.

La función se representa como:

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

Donde:

- **G(n)** = La distancia del camino desde el nodo origen hasta el nodo n. En este caso los nodos son las diagonales y arriba, abajo, derecha e izquierda.

Donde G(n) puede tomar dos valores:

- 1) $G(n) = 1$, si mi dirección no es una diagonal.
- 2) $G(n) = \sqrt{2}$ si mi dirección es una diagonal.

- **H(n)** = La distancia estimada desde el nodo n hasta el nodo destino.

Donde H(n) toma el siguiente valor:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- X1 = fila del nodo actual (n)
- X2 = fila del nodo destino
- Y1 = columna del nodo actual (n)
- Y2 = columna del nodo destino

H(n) es una función heurística, donde se expresa la idea de cuánto queda para alcanzar el nodo destino.

Hay que tener en cuenta que si todo nodo de la red cumple que **G(n) = 0**, entonces estamos ante una **búsqueda voraz**, y en el caso de que **H(n) = 0**, entonces estamos ante una **búsqueda de coste uniforme no informada**.

La complejidad computacional del algoritmo está relacionada con la calidad de la heurística del problema. En el peor caso, será exponencial **O(2^n)** en el mejor de los casos será en tiempo lineal **O(n)**.

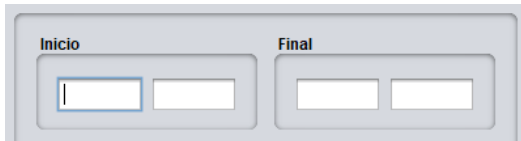
El problema del algoritmo es el espacio que requiere para su ejecución. Dado que tiene que almacenar todos los posibles nodos de cada estado, eso requiere que la cantidad de memoria sea exponencial con respecto al tamaño del problema.

DETALLES DEL FUNCIONAMIENTO

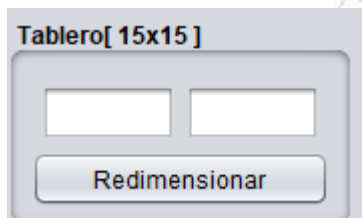
Tenemos una interfaz gráfica donde se puede comprobar tenemos que seleccionar el Inicio y final del camino introduciendo los datos mediante teclado, como también la redimensión del propio tablero.

Para poner obstáculos, con un click del ratón sobre las celdas del mapa es suficiente.

Inserción datos de Inicio y Final:

A screenshot of a graphical user interface showing two input sections. The first section is labeled 'Inicio' and contains two adjacent text input boxes. The second section is labeled 'Final' and also contains two adjacent text input boxes. The interface has a light gray background and rounded corners.

Inserción del nuevo tamaño del tablero

A screenshot of a dialog box titled 'Tablero[15x15]'. Inside the dialog, there are two text input boxes side-by-side. Below these boxes is a button labeled 'Redimensionar'. The dialog box has a light gray border and a standard Windows-style title bar.

DETALLES DE LA IMPLEMENTACIÓN

El lenguaje utilizado para la práctica ha sido Java, lo he utilizado debido al uso que le doy, es de los lenguajes que más a mano tengo y donde puedo defenderme mejor, también para el tema de la interfaz gráfica que he usado para su desarrollo Java Swing.

El tablero tiene dos thread, uno para pintar el camino y otro para su dinamismo.

El algoritmo A estrella, para la lista abierta, lista cerrada y su camino final he usado arrays dinámicos como `ArrayList<Nodo>` donde `Nodo` tiene las coordenadas de las celdas del tablero, el tipo y demás.

La clase del algoritmo tiene toda la lógica de la aplicación, listas y la heurística.

He usado un cronómetro para su inicio y su fin para los resultados.

MANUAL DE USUARIO

Se ha hecho entrega de un archivo .jar en la carpeta, el cual ejecuta nuestro algoritmo.

Se abre un tablero con las opciones de : Iniciar, Redimensionar y Reiniciar:

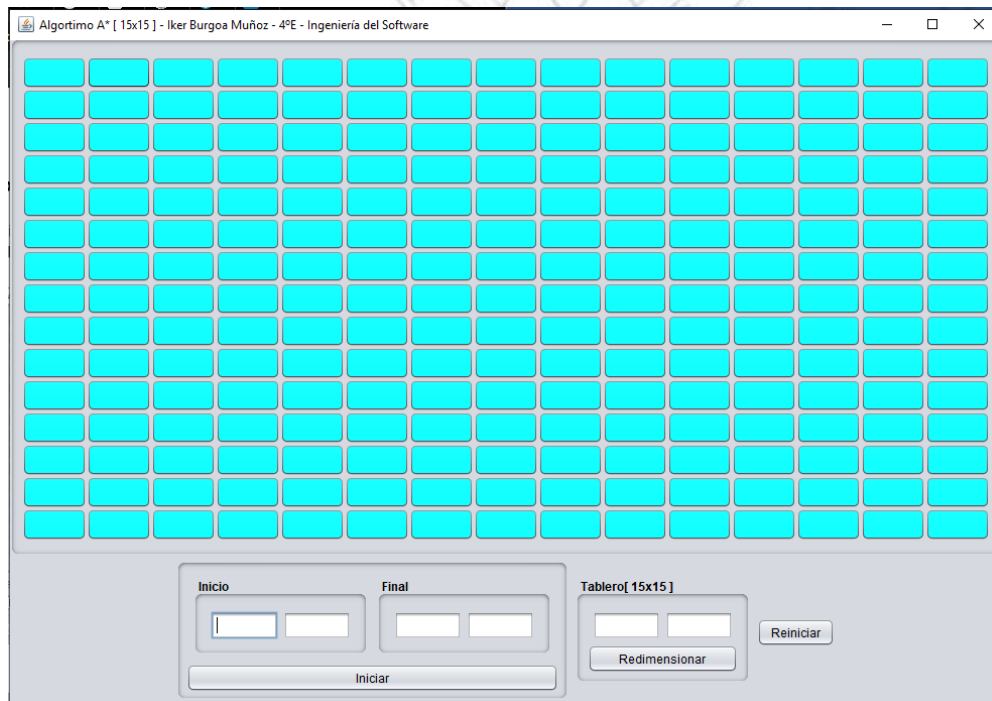
INICIAR: Al haber introducido los datos de inicio y final , que son los datos mínimos necesarios para completar el algoritmo, al clicar se realiza el camino óptimo como queda reflejada en la segunda captura.

REDIMENSIONAR: Es importante recalcar que no hay un límite de tamaño para el algoritmo, para un mejor uso recomendable no superar el 20x20.

El uso mínimo está puesto en 3x3.

Al dar a redimensionar se reinicia el tablero pero con el nuevo tamaño

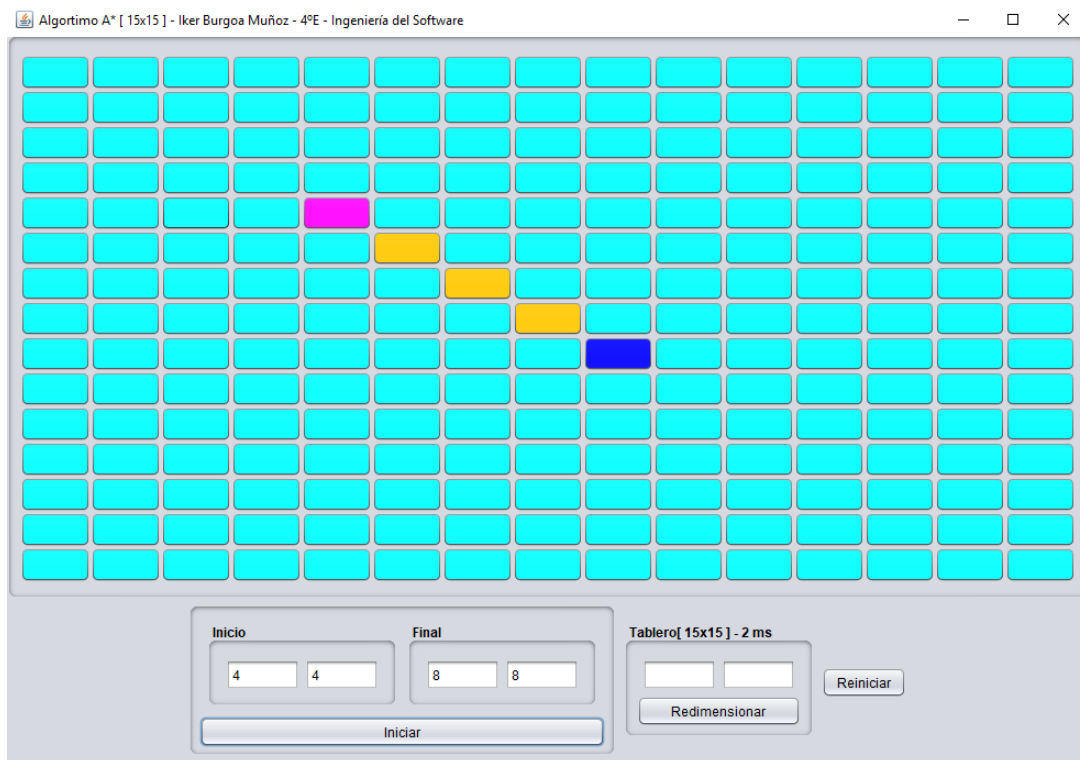
REINICIAR: Reinicia el programa desde cero eliminando todo lo pintado en el tablero.



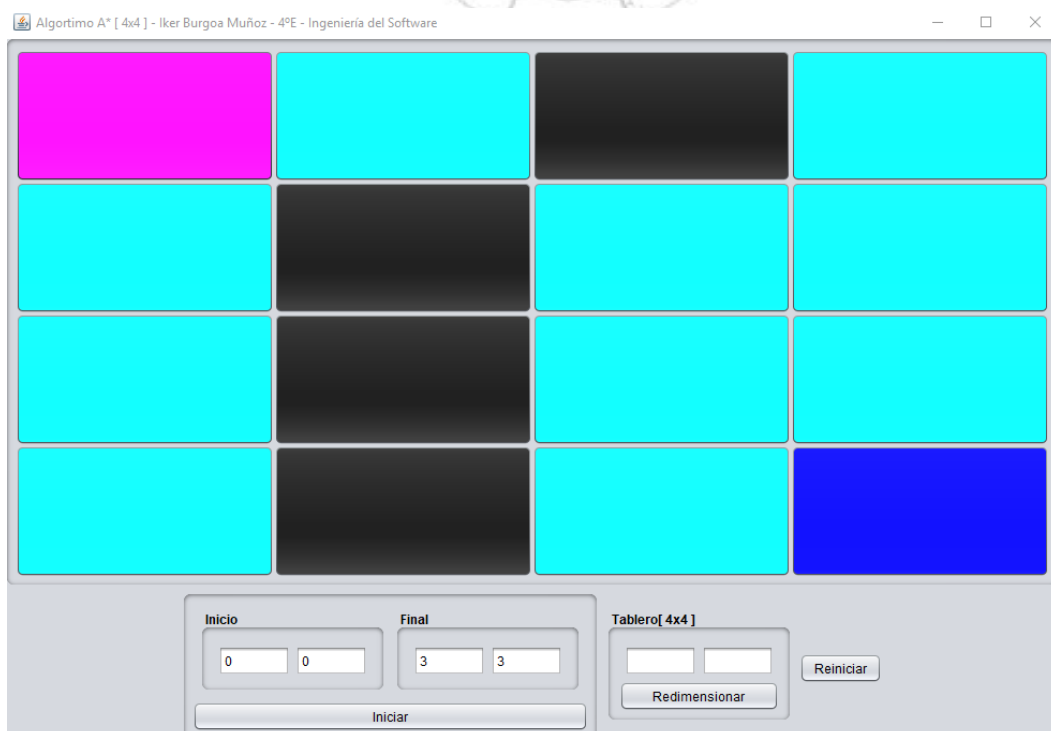
El tablero se inicia en un principio en 15x15.

Los valores de inicio y final se deben meter por teclado.

Para su correcto funcionamiento debido a un pequeño fallo interno, es recomendable al introducir un valor, teclear el tabulador o con el teclado clicar en la siguiente celda, para su visualización.

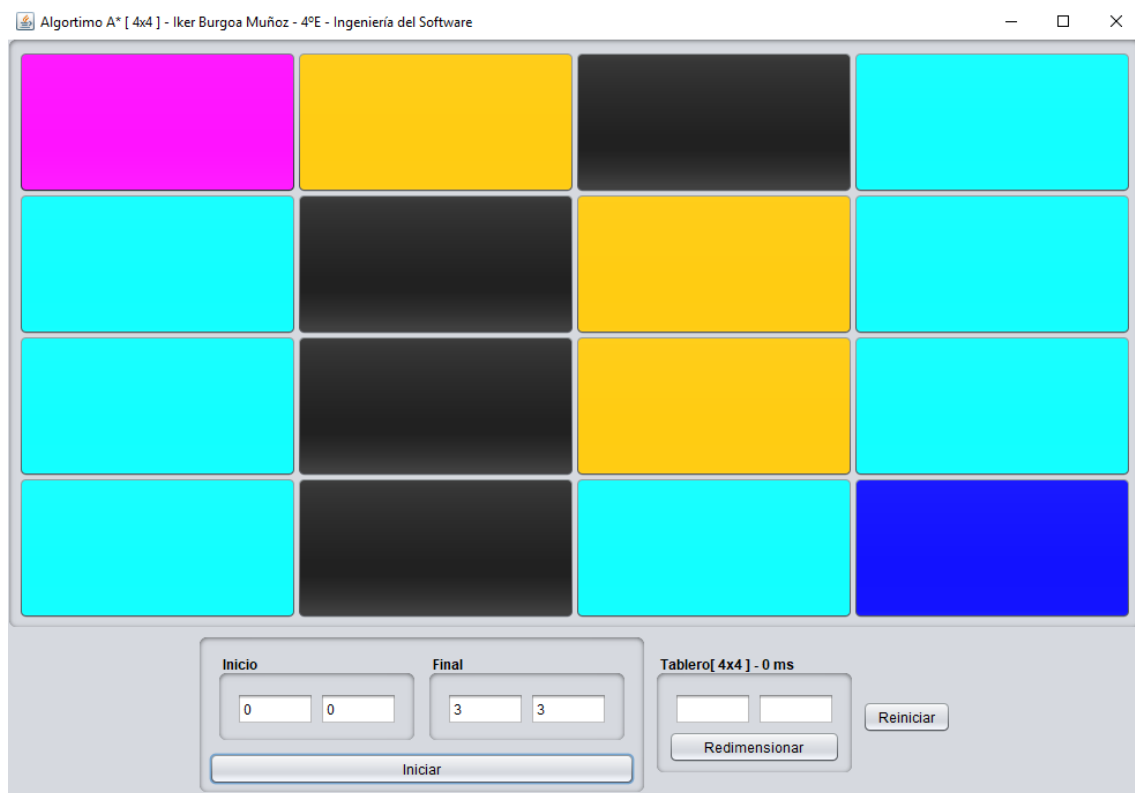


En este ejemplo se puede observar como tenemos un inicio(4,4) y un final (8,8), lo naranja es el camino óptimo.



En este ejemplo redimensioné el tablero inicial a un 4x4.

Los cuadrados negros son los obstáculos por los que el algoritmo no puede pasar. Para su inicio le damos a iniciar y obtenemos el siguiente resultado:



CASOS DE PRUEBA

Caso fácil:

Sin obstáculos. Inicio y fin en extremos:



Inicio: (1,2)

Fin: (12,10)

Obstáculos : la diagonal



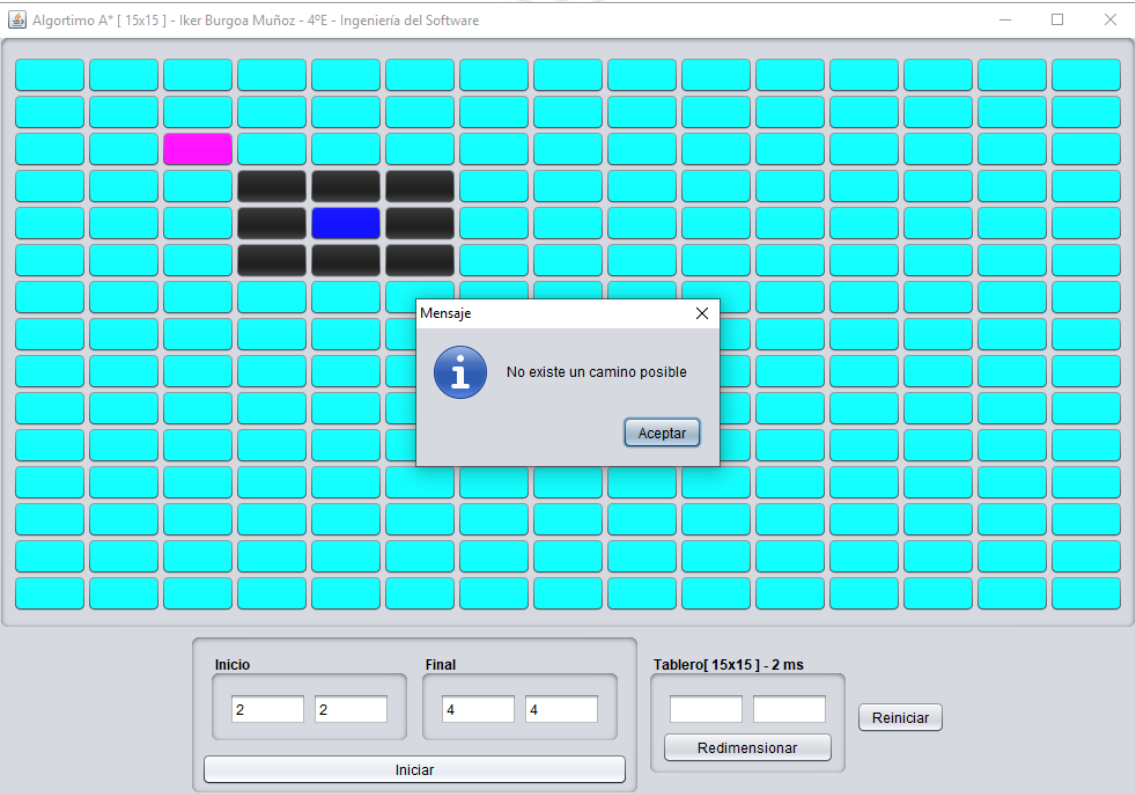
Inicio: (4,5)

Fin : (7,9)

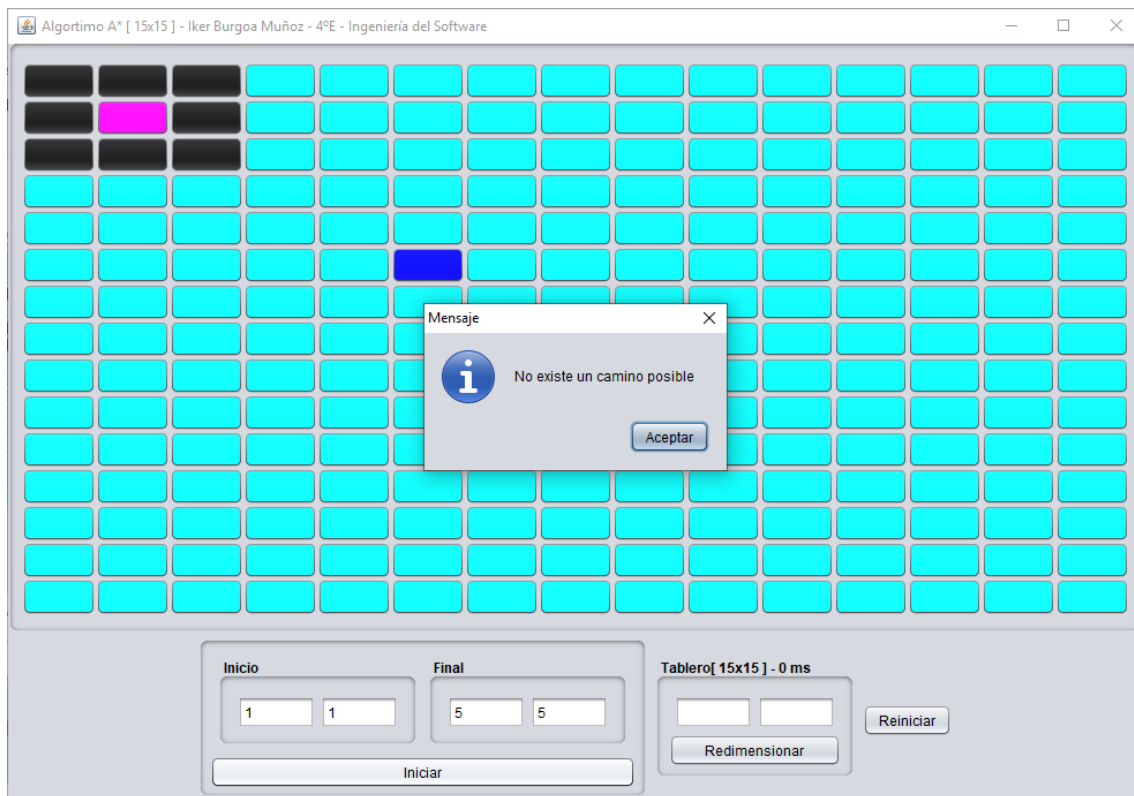


Casos imposibles:

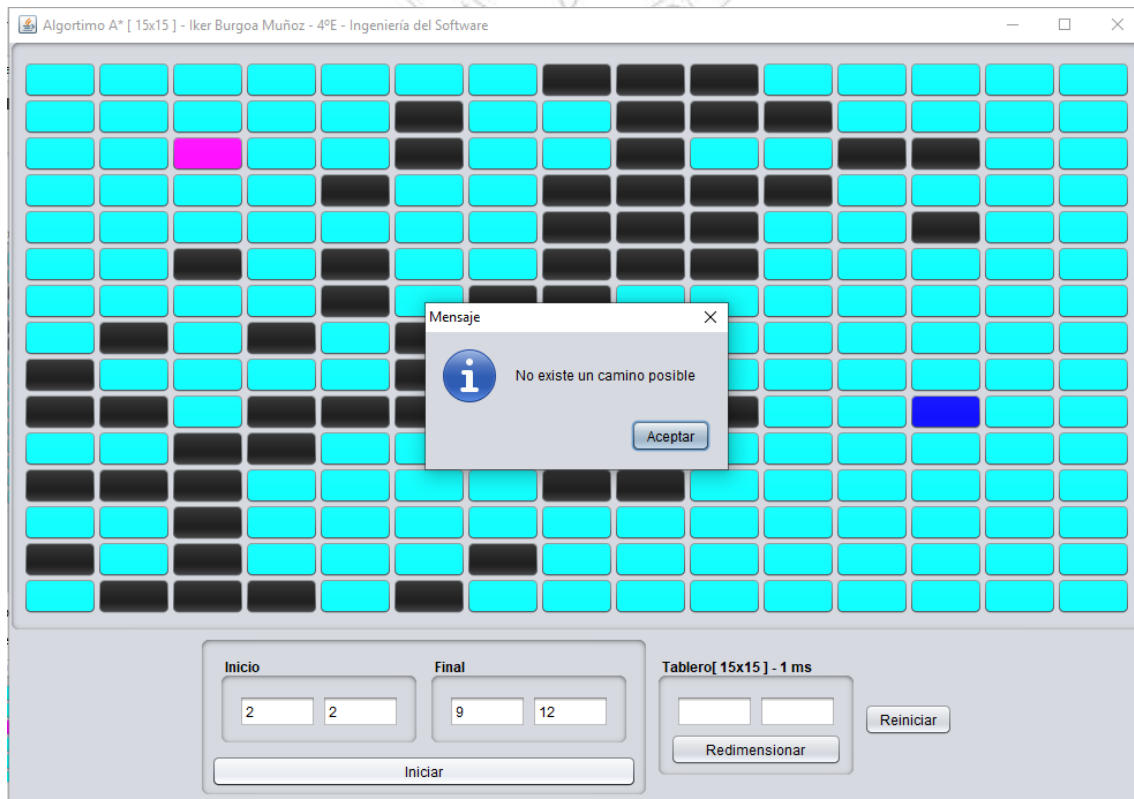
Final tapado entero:



Inicio tapado entero:



Camino donde Inicio y Fin no están cubiertos, pero los obstáculos bloquean todos los posibles caminos.



BIBLIOGRAFÍA

- <http://idelab.uva.es/algoritmo>
- https://www.ecured.cu/Algoritmo_de_B%C3%BAsqueda_Heur%C3%ADstica_A*
- https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A*
- <http://www.cristalab.com/tutoriales/algoritmo-a-para-encontrar-el-camino-mas-corto-en-ia-c91591/>
- <https://advanceintelligence.wordpress.com/2014/10/07/algoritmo-a-estrella/>
- Material del Campus Virtual.



UNIVERSIDAD
COMPLUTENSE
MADRID