



# Algoritmo A\*

MEMORIA DE LA PRÁCTICA 1, INGENIERÍA DEL  
CONOCIMIENTO

JAIME AMER OLIVER

## Contenido

1. Introducción .....	2
2. Características .....	2
3. Manual de Usuario .....	2
3.1. Modo Automático .....	2
3.2. Modo Interactivo.....	3
4. Ejemplo de ejecución .....	<b>¡Error! Marcador no definido.</b>
5. Problemas encontrados .....	6

## 1. Introducción

El objetivo de la práctica 1 de la asignatura de Ingeniería del Conocimiento consiste en conocer y entender el funcionamiento del Algoritmo A\* y realizar una implementación en código de forma libre.

Soy Jaime Amer Oliver, estudiante de Ingeniería del Software y autor del programa que se presentará a continuación.

Debo recalcar que, por desgracia, el programa no funciona del todo correctamente. A última hora se han detectado fallos en la ejecución y, a veces, toma caminos que no son del todo favorables para la solución óptima que se busca.

## 2. Características

La aplicación de este Algoritmo A\* se ha programado en C++ orientado a objetos en Visual Studio 2017. Se trata de una aplicación muy simple con interfaz de consola en el que cualquier dato se introduce por el teclado y consta de unos menús que van guiando al usuario durante su ejecución. Los requisitos para su funcionamiento son:

- Sistema Operativo: Windows.
- Arquitectura: 32/64 bits.
- RAM: Menos de 1MB.

El programa debería ejecutarse en cualquier versión de Windows, tanto antiguo como moderno y en 32 o 64 bits. No debería tener incompatibilidades.

He decidido programar la práctica en C++ al tener mucho más conocimiento y experiencia que con Java al tener bastante dominio del lenguaje y de cómo se gestiona la memoria con la programación de punteros.

El código del proyecto, por si alguien quiere verlo, se encuentra en GitHub accediendo a través de este link: <https://github.com/JaimeAmer/Algoritmo-A->

## 3. Manual de Usuario

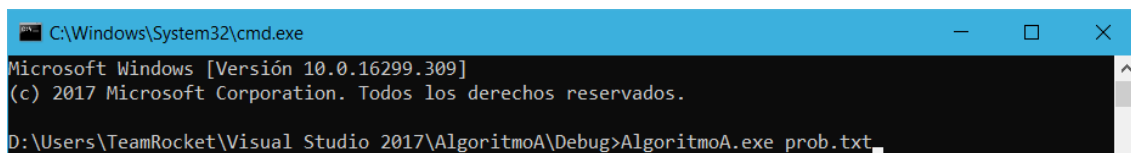
La aplicación consta de dos métodos de ejecución:

- Modo Automático
- Modo Interactivo

### 3.1. Modo Automático

El Modo Automático lee un fichero de entrada y ejecuta el Algoritmo A\* automáticamente y muestra los resultados de forma interactiva.

Se puede ejecutar insertando como parámetro el nombre o la ruta del fichero, como se puede ver en la siguiente imagen:

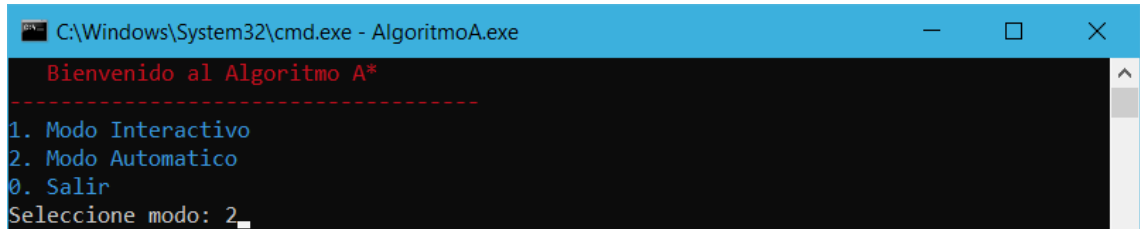


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.16299.309]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.
D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug>AlgoritmoA.exe prob.txt_
```

**prompt> AlgoritmoA.exe [nombre o ruta del fichero]**

Si no se indica la ruta del fichero, es importante que lo esté en la misma carpeta que el ejecutable.

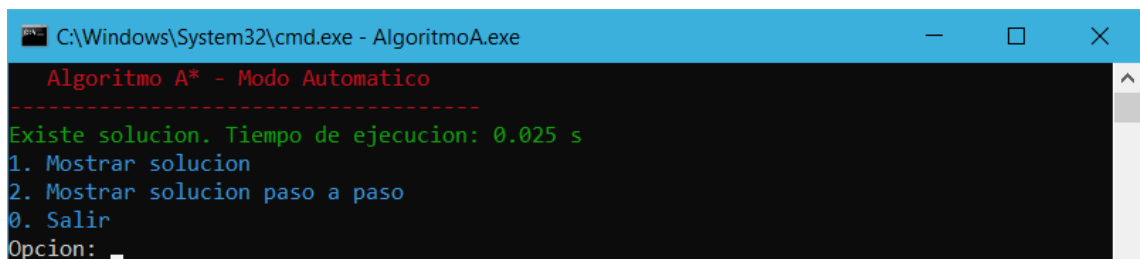
También se puede iniciar el Modo Automático ejecutando el programa haciendo doble clic sobre el icono y seleccionando la Opción 2 del menú principal, como podemos ver en la imagen:



```
C:\Windows\System32\cmd.exe - AlgoritmoA.exe

Bienvenido al Algoritmo A*
-----
1. Modo Interactivo
2. Modo Automatico
0. Salir
Seleccione modo: 2_
```

Una vez el programa haya leído el fichero correctamente y exista una solución para el problema a resolver, se mostrará la siguiente pantalla con unos menús para mostrar la solución. En caso contrario, nos saldrá que no existe una solución alcanzable.



```
C:\Windows\System32\cmd.exe - AlgoritmoA.exe

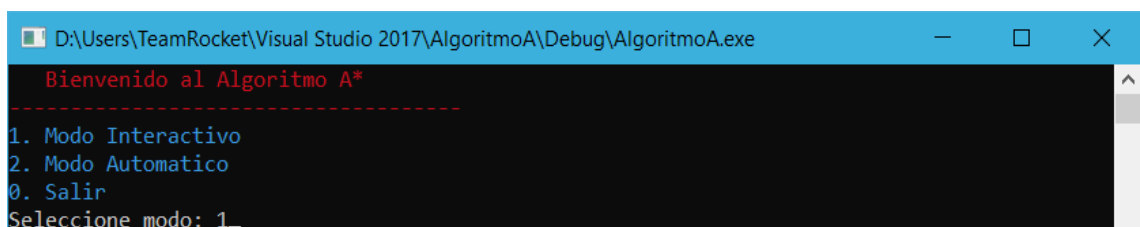
Algoritmo A* - Modo Automatico
-----
Existe solucion. Tiempo de ejecucion: 0.025 s
1. Mostrar solucion
2. Mostrar solucion paso a paso
0. Salir
Opcion: _
```

- Opción 1: Mostrar solución
  - Nos mostrará directamente el camino que se debe tomar desde el origen al destino.
- Opción 2: Mostrar solución paso a paso
  - Nos mostrará la solución indicándonos en cada paso el camino que se debe tomar. Además, nos mostrará el número de Nodos por los que pasará.
- Opción 0: Salir
  - Dependiendo de cómo hayamos iniciado el programa, saldrá al menú principal de la aplicación o se terminará la ejecución.

### 3.2. Modo Interactivo

El Modo Interactivo, sin embargo, guía al usuario durante toda la ejecución del programa.

Para empezar, ejecutamos el programa haciendo doble clic sobre el icono o invocándolo por consola de comandos sin parámetros adicionales. Cuando se abra, elegimos la opción 1 del menú.



```
D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe

Bienvenido al Algoritmo A*
-----
1. Modo Interactivo
2. Modo Automatico
0. Salir
Seleccione modo: 1_
```

Ahora, debemos introducir los datos que manejará el algoritmo:

- Dimensiones de la matriz
- Posición de origen (Aparecerá en la tabla como 'A')
- Posición de destino (Aparecerá en la tabla como 'B')
- Posición de los obstáculos (Aparecerán como 'X')

```

D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Interactivo
-----
Inserte dimensiones de la tabla (Ax B): 8x8
  
```

Insertión de la dimensión de la matriz, 8x8 por ejemplo

```

D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Interactivo
-----
0
1
2
3
4
5
6
7
0 1 2 3 4 5 6 7
Inserte posición de origen (A B): 0 0_
  
```

Insertión del nodo origen, en la posición 0x0 por ejemplo

```

D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Interactivo
-----
0 A
1
2
3
4
5
6
7
0 1 2 3 4 5 6 7
Inserte posición de destino (A B): 7 7_
  
```

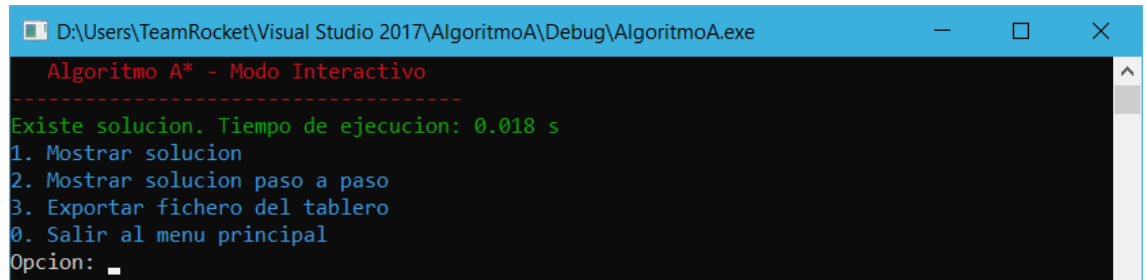
Insertión del nodo destino, en la posición 7x7 por ejemplo

```

D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Interactivo
-----
0 A
1
2 X X X X X
3       X X
4  X
5  X X
6   X X X X X X
7       B
0 1 2 3 4 5 6 7
Inserte posición del obstáculo (7 7 para terminar): 4 5_
  
```

Insertión de los obstáculos

Para terminar de insertar los obstáculos, hay que insertar la posición de destino (7 7 en el caso del ejemplo) y el programa enseguida procederá a resolver el problema.



```

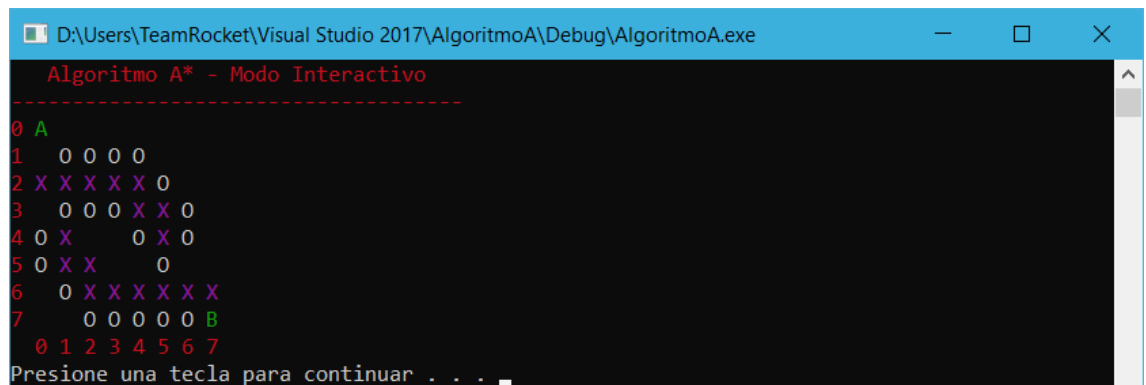
D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Interactivo
-----
Existe solución. Tiempo de ejecución: 0.018 s
1. Mostrar solución
2. Mostrar solución paso a paso
3. Exportar fichero del tablero
0. Salir al menú principal
Opcion: 

```

Si el algoritmo A\* ha encontrado una solución, nos mostrará el menú de la imagen (con funciones parecidas al del modo automático). En caso contrario, nos dirá que no se ha encontrado solución y saldrá al menú principal.

- Opción 1: Mostrar solución
  - Nos mostrará directamente el camino que se debe tomar desde el origen al destino.
- Opción 2: Mostrar solución paso a paso
  - Nos mostrará la solución indicándonos en cada paso el camino que se debe tomar. Además, nos mostrará el número de Nodos por los que pasará.
- Opción 3: Exportar fichero del tablero
  - Esta opción nos permite crear un fichero con los datos del tablero a resolver, por si queremos ejecutarlo de nuevo en el modo automático.
- Opción 0: Salir
  - Saldrá al menú principal de la aplicación.

Si elegimos la opción 1, veremos el tablero directamente con la solución óptima. El camino elegido se mostrará con caracteres 'O'.



```

D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Interactivo
-----
0 A
1 0 0 0 0
2 X X X X X 0
3 0 0 0 X X 0
4 0 X 0 X 0
5 0 X X 0
6 0 X X X X X
7 0 0 0 0 0 B
  0 1 2 3 4 5 6 7
Presione una tecla para continuar . . . 

```

Si elegimos la opción 2, veremos construirse el camino cada vez que pulsemos una tecla.

```

Algoritmo A* - Modo Interactivo
-----
0 A
1 0
2 X X X X X
3   X X
4  X   X
5  X X
6   X X X X X
7           B
  0 1 2 3 4 5 6 7
Num. Iteracion: 0
Presione una tecla para continuar . . .

```

```

Algoritmo A* - Modo Interactivo
-----
0 A
1 0 0
2 X X X X X
3   X X
4  X   X
5  X X
6   X X X X X
7           B
  0 1 2 3 4 5 6 7
Num. Iteracion: 1
Presione una tecla para continuar . . .

```

```

Algoritmo A* - Modo Interactivo
-----
0 A
1 0 0 0 0
2 X X X X X 0
3 0 0 0 X X 0
4 0 X   0 X 0
5 0 X X   0
6   X X X X X
7           B
  0 1 2 3 4 5 6 7
Num. Iteracion: 13
Presione una tecla para continuar . . .

```

```

Algoritmo A* - Modo Interactivo
-----
0 A
1 0 0 0 0
2 X X X X X 0
3 0 0 0 X X 0
4 0 X   0 X 0
5 0 X X   0
6 0 X X X X X
7   0 0 0 0 0 B
  0 1 2 3 4 5 6 7
Num. Iteracion: 19
Presione una tecla para continuar . . .

```

Si queremos exportar el fichero del tablero, simplemente habrá que elegir la opción 3 e indicar el nombre o la ruta del fichero.

```

D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Interactivo
-----
Introduce nombre o ruta del fichero: ejemplo1.txt
Fichero exportado correctamente.
Presione una tecla para continuar . . .

```

Si no ha habido problemas o conflictos con los ficheros, el programa nos confirmará que se ha exportado correctamente.

#### 4. Problemas encontrados

El código de la práctica se ha desarrollado sin problemas, aunque la ejecución del Algoritmo A\* no ha sido del todo satisfactorio.

Horas previas de la entrega, durante la sesión de laboratorio, hemos podido comprobar que a veces el algoritmo toma un camino que no es del todo óptimo. Por ejemplo, para este tablero:

```

D:\Users\TeamRocket\Visual Studio 2017\AlgoritmoA\Debug\AlgoritmoA.exe
Algoritmo A* - Modo Automatico
-----
0 A
1
2
3
4   X
5   X
6   X
7   B
  0 1 2 3 4 5 6 7
Num. Iteracion: 0
Presione una tecla para continuar . . .

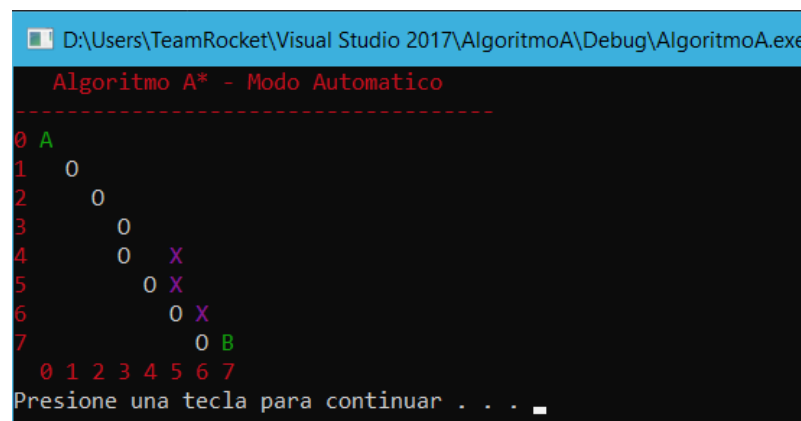
```

Vemos que en la Iteración 6, en la posición (6,5) el mejor adyacente es el que se sitúa en la posición (7,6). Sin embargo, por problemas o por errores de programación en el cálculo de la distancia euclídea entre dos puntos, a veces tomaba valores anómalos y el Algoritmo veía como siguiente Nodo óptimo la posición (5,6).



ACTUALIZACIÓN DE LAS 22:57.

Tras mi persistencia en la búsqueda del fallo, ahora está arreglado. El ejemplo problemático funciona bien:



Resulta que la función de la raíz cuadrada daba valores anómalos al hacer cast entre números enteros y float.