

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025



Asignatura

Razonamiento y Planificación Automática

Nombre del Estudiante

Jonnier Moreno Bertel

Profesor

Félix Plúas

Actividad

Búsqueda de rutas en empresa de paquetería

Fecha

07/12/2025

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025

Actividad: búsqueda de rutas en empresa de paquetería

Para esta actividad se ha proporcionado el software necesario para llevarla a cabo, se pide interpretar los resultados, basado en la teoría aprendida en clases y los datos observados arrojados por el software. El caso problema a trabajar en la actividad se ilustra en la figura 1.

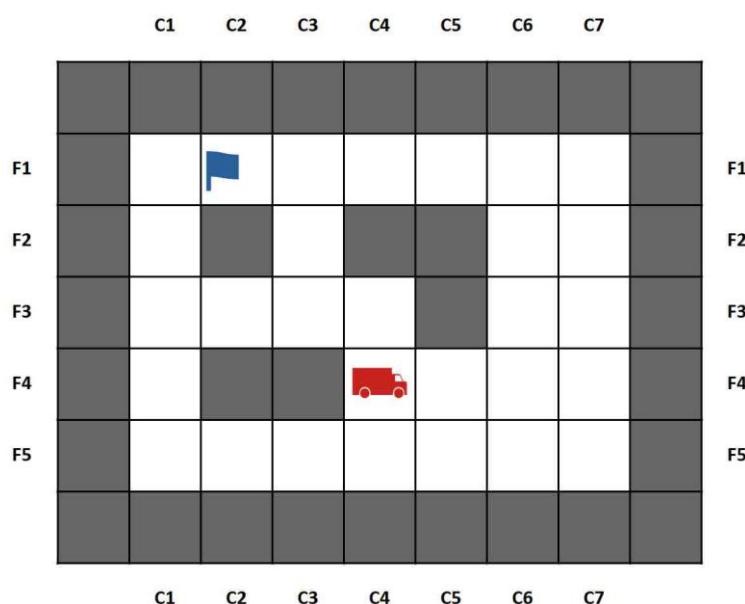


Figura 1: -Situación de inicio del problema. El camión representa la posición inicial del agente y la bandera un lugar al que debe ir a recoger un paquete.

Partiendo de las herramientas suministradas, se pide resolver cada uno de los casos en los que fueron implementados los algoritmos. Se notifica en este punto que la solución de la actividad no implementó IA en ningún punto para su resolución, se destaca que el análisis realizado es de autoría propia del estudiante, partiendo de los conocimientos y conceptos aportados en clase.

A continuación, se realiza la resolución de cada uno de los casos.

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025

Caso 1. Probar y comparar los algoritmos de búsqueda en amplitud y búsqueda en profundidad.

1. Muestre la tabla con los datos de ambas ejecuciones.

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por amplitud	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	5	25	7

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por profundidad	C4F4 -> C4F5 -> C5F5 -> C6F5 -> C6F4 -> C6F3 -> C6F2 -> C6F1 -> C5F1 -> C4F1 -> C3F1 -> C2F1	12	11	22	10

2. ¿Obtiene amplitud un resultado óptimo? ¿Por qué?

Por simple inspección del problema y análisis de los resultados capturados del algoritmo de **búsqueda por amplitud**, se puede determinar que se obtuvo un resultado óptimo, partiendo de la premisa de que se recorrió un camino que le permitió llegar a la meta objetivo en el menor costo posible.

3. ¿Obtiene profundidad un resultado óptimo? ¿Por qué?

Para este ejercicio en particular, el algoritmo de **búsqueda por profundidad** solo podría ser óptimo en el caso de que tome inicialmente (y de forma aleatoria) la ruta más corta y se expanda en ese sentido hasta llegar a la meta objetivo, sin embargo, y tal como se puede apreciar en su solución, seleccionar una ruta inequívoca, puede desembocarse en un número de iteraciones mucho más grande.

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025

4. ¿Cuál de los dos algoritmos es más eficiente en este caso particular? Justifique su respuesta.

Dado que ambos métodos consisten en barrer un algoritmo genérico, sin tener en cuenta el costo de cada una de sus acciones y en vista de que el coste de solución fue menor para el de **búsqueda por amplitud**, Se puede inferir que el algoritmo por amplitud fue mucho más optimo en costo, de menor iteración para alcanzar la solución y por ende más eficiente entre ambos, claro, expande mas nodos que el método de **búsqueda por profundidad** pero inclusive encuentra una solución en un nivel más bajo y menos profundo que el algoritmo por profundidad.

Todas esas características ya mencionadas del algoritmo de **búsqueda por amplitud** para este caso permiten pensar que consume menos recursos que el de **algoritmo por profundidad** y por ende este primero será más eficiente.

Caso 2. En este caso se deben probar y **comparar los algoritmos de búsqueda amplitud, Dijkstra y A* cuando hay costes distintos de 1**. Vea que los ficheros contienen costes modificados para este caso. Para A*, observe que el código proporciona una función heurística basada en la distancia de Manhattan.

Costos asociados al caso #2

Izquierda = 3; Derecha = 1; Arriba = 2; Abajo = 3

1. Muestre una tabla con los datos de todas las ejecuciones.

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por amplitud	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	9	25	7

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025

Búsqueda por Dijkstra	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	9	23	7
-----------------------	--	---	---	----	---

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por A*	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	9	11	9

2. ¿Obtiene UCS (Dijkstra) el camino de coste óptimo?

Sí, obtiene el camino cuya sumatoria de pesos lo guía por el camino más óptimo en costos.

3. ¿Obtiene A* el camino de coste óptimo?

Sí, al igual que Dijkstra, bajo el entendimiento de optimización de costos.

4. ¿Cuál de los dos algoritmos (UCS o A*) es más eficiente en el caso planteado?

A* resulta ser más eficiente en este caso, aunque ambos llegan a la solución óptima de menor costo, nótese que el algoritmo analiza la mitad aproximadamente de los nodos analizados por algoritmo UCS en tanto, para este caso se estaría hablando de una eficiencia en tiempo y de recursos.

5. ¿Se puede afirmar que las respuestas 2 y 3 siempre serán de este modo, aunque se varíe el mapa?

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025

```

Experimento con algoritmo <function uniform_cost at 0x0000011565EDB130>:
#####
#.P   #
#. # # #
#.  ## #
#.##T #
#.##. #
#####
Total length of solution: 10
Total cost of solution: 17.0
max fringe size: 5
visited nodes: 26
iterations: 26

Experimento con algoritmo <function astar at 0x0000011565EDB250>:
#####
#.P   #
#. # # #
#.  ## #
#.##T #
#.##. #
#####
Total length of solution: 10
Total cost of solution: 17.0
max fringe size: 5
visited nodes: 24
iterations: 24

```

Figura 2: - Contra ejemplo con mapa modificado.

No. Si el mapa varia, se podría tal vez garantizar que se halle la solución más optima en ambos casos, sin embargo, debido a que la función heurística (en este caso la **distancia de Manhattan**) no puede garantizar que el algoritmo **A*** sea superior al análisis en bruto de **USC** esta afirmación no se puede realizar, como contra ejemplo se presenta la figura 2. En resumen, no se cumple para esta función heurística que no tiene en cuenta las paredes.

6. ¿Se puede afirmar que las respuestas 2 y 3 siempre serán de este modo, aunque se varíen los costes? Justifique su respuesta.

Se podría interpretar que sí, de modo que lo que garantizaría la eficiencia de uno sobre el otro (en este caso **A*** sobre **UCS**), es que **A*** cuenta con la **función Heurística**, que constantemente le estima la distancia de la posición actual con respecto a la meta objetivo, pudiendo así, obtener la mejor ruta para llegar a la **meta objetivo**, en tanto **UCS** llegaría a la ruta de optimo valor, sin embargo, emplearía mas recursos para ello.

Caso 3. En este caso abordamos la comparación de **diferentes funciones heurísticas** para el **mismo algoritmo** (**A***), teniendo además como referencia el resultado del algoritmo **UCS** (o Dijkstra).

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel	07/12/2025
	Nombre: Jonnier	

1. Muestre una tabla con los datos de todas las ejecuciones.

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por Dijkstra	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	9	23	7

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por A* heurística 1	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	9	11	9

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por A* heurística 2	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	9	18	9

Algoritmo o caso	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por A* heurística 3	C4F4 -> C4F3 -> C3F3 -> C3F2 -> C3F1 -> C2F1	6	9	6	5

2. Muestre la captura de las soluciones obtenidas para cada ejecución.

Función A* - H1	##### # P. # # .## # # ..# # # ##T # # # # ##### Total length of solution: 6 Total cost of solution: 9.0 max fringe size: 9 visited nodes: 11 iterations: 11	Función A* - H2	##### # P. # # .## # # ..# # # ##T # # # # ##### Total length of solution: 6 Total cost of solution: 9.0 max fringe size: 9 visited nodes: 18 iterations: 18
Función A* - H3	##### # P. # # .## # # ..# # # ##T # # # # ##### Total length of solution: 6 Total cost of solution: 9.0 max fringe size: 5 visited nodes: 6 iterations: 6		

Figura 3: - Algoritmo A* ante diferentes funciones heurísticas.

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025

3. ¿Obtiene UCS (Dijkstra) el camino de coste óptimo?

Sí, el algoritmo **UCS (Dijkstra)** obtiene el costo más óptimo, pero lo hace después de expandir múltiples nodos para llegar a la meta objetivo por el mejor camino e ignorando las funciones heurísticas (pues no las tiene en cuenta).

4. ¿Obtiene A* el camino de coste óptimo con todas las heurísticas?

Sí, se evidencia en la ejecución del script (figura 3) y en los resultados, obteniendo el camino de coste óptimo para todas las diferentes heurísticas aplicadas al problema bajo el algoritmo **A***.

5. ¿Es el algoritmo A* igualmente eficiente en todos los casos planteados? Intente explicar la razón de las diferencias observadas.

No, la eficiencia variará según la **función heurística** utilizada, en este caso, se evidencian resoluciones del problema más eficientes que otras, por ejemplo, en el caso **A* - H3** (figura 3) se presenta el algoritmo con mejor función heurística para este sistema, no solo obtiene el costo optimo, sino también el más eficiente en cuanto tiempo y utilización de menos recursos, seguido de **A* - H1** el cual en los ejercicios del caso dos utiliza la distancia de Manhattan como función heurística y por ultimo y menos eficiente el **A* - H2** que aunque llega al coste optimo, emplea más recursos y tiempo para llegar a la respuesta.

6. ¿Se puede afirmar que la respuesta 4 no variaría al cambiar las paredes del mapa y las posiciones de inicio y fin que aparecen en el mapa para ninguna de las heurísticas presentadas?

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Moreno Bertel Nombre: Jonnier	07/12/2025

No se puede afirmar lo anterior, nuevamente, la ruta que tome el algoritmo dependerá de la función heurística y del costo asociado a cada acción que se ejecute, con lo cual, al variar la función heurística esta puede beneficiar un camino u otro, llegando así a la meta objetivo desde movimientos distintos y muy probablemente sin preservar el coste óptimo.

7. Si la respuesta a la pregunta 6 es no en algún caso, ¿puede probar esta afirmación diseñando un mapa (variable MAP) que compruebe este hecho? Añada este mapa y sus resultados a la tabla original.

Este hecho se comprueba a continuación (figura 4), solo bastó con añadir dos paredes para contraargumentar dicha afirmación.

```

Experimento con algoritmo <function astar>
#####
# P   #
# # # #
#.  ## #
# ##T #
#.... #
######
Total length of solution: 10
Total cost of solution: 17.0
max fringe size: 5
visited nodes: 24
iterations: 24

Experimento con algoritmo <function astar>
#####
# P   #
# # # .
#   ## #
# ##T #
#.... #
######
Total length of solution: 10
Total cost of solution: 17.0
max fringe size: 5
visited nodes: 24
iterations: 24

Experimento con algoritmo <function astar>
#####
# P...#
# # # .
#   ## #
# ##T...
#.... #
######
Total length of solution: 12
Total cost of solution: 21.0
max fringe size: 5
visited nodes: 18
iterations: 18

```

Figura 4: - Algoritmo A* ante diferentes funciones heurísticas y un distinto mapa.

Algoritmo o caso Mapa 2	Solución	Longitud	Coste solución	Nodos expandidos	Tamaño máximo de lista
Búsqueda por A* heurística 1	C4F4 -> C4F5 -> C3F5 -> C2F5 -> C1F5 -> C1F4 -> C1F3 -> C1F2 C1F1 -> C2F1	10	17	24	5
Búsqueda por A* heurística 2	C4F4 -> C4F5 -> C3F5 -> C2F5 -> C1F5 -> C1F4 -> C1F3 -> C1F2 C1F1 -> C2F1	10	17	24	5
Búsqueda por A* heurística 3	C4F4 -> C5F4 -> C6F4 -> C7F4 -> C7F3 -> C7F2 -> C7F1 -> C6F1 C5F1 -> C4F1 -> C3F1 -> C2F1	12	21	18	5