

Análisis de Algoritmos A^* y $A^*\varepsilon$

Jaime Hernández

Curso 2024/2025

1. Introducción

Esta práctica implementa y analiza los algoritmos A^* y $A^*\varepsilon$, ambos empleados para la búsqueda de caminos óptimos. A lo largo de esta práctica, se han probado distintas heurísticas y valores de ε para evaluar su comportamiento en varios tipos de terreno. La documentación abordará los siguientes elementos:

- Explicación detallada del algoritmo A^* y traza de un caso de ejemplo.
- Análisis comparativo de las distintas heurísticas implementadas, evaluando el número de nodos explorados.
- Pruebas de $A^*\varepsilon$ con distintos valores de ε .
- Comparación entre los algoritmos A^* y $A^*\varepsilon$.
- Varios mapas de prueba con diseños específicos para mostrar distintas situaciones del problema: problemas sin solución, varios caminos posibles, etc.

2. Descripción de los Algoritmos

2.1. Algoritmo A^*

El algoritmo A^* es un método de búsqueda de caminos que encuentra el camino óptimo entre un origen y un destino, evaluando cada nodo mediante la función:

$$f(n) = g(n) + h(n) \quad (1)$$

donde:

- $g(n)$ representa el coste real desde el origen hasta el nodo n .
- $h(n)$ es una estimación del coste restante desde n hasta el destino.

Para organizar la búsqueda, A^* mantiene dos listas:

- **Lista abierta:** contiene nodos por explorar, ordenados por el menor valor de $f(n)$.
- **Lista cerrada:** contiene nodos ya explorados.

2.2. Algoritmo $A^*\varepsilon$

$A^*\varepsilon$ es una variante de A^* que permite encontrar soluciones subóptimas pero con una mayor eficiencia. Este algoritmo usa un factor ε para controlar el balance entre optimalidad y eficiencia:

$$f_{threshold} = (1 + \varepsilon) \cdot f_{min} \quad (2)$$

Este factor permite considerar soluciones con un coste hasta $(1 + \varepsilon)$ veces el óptimo, lo cual reduce el tiempo de búsqueda.

3. Heurísticas Implementadas

Se implementaron distintas heurísticas para evaluar el rendimiento de A^* y $A^*\varepsilon$.

- **Heurística Nula:** $h(n) = 0$, similar a Dijkstra.
- **Manhattan:** Suma de diferencias absolutas:

$$h(n) = |x - x_{goal}| + |y - y_{goal}| \quad (3)$$

- **Euclídea:** Distancia en línea recta:

$$h(n) = \sqrt{(x - x_{goal})^2 + (y - y_{goal})^2} \quad (4)$$

- **Chebyshev:** Máximo de las diferencias absolutas:

$$h(n) = \max(|x - x_{goal}|, |y - y_{goal}|) \quad (5)$$

- **Diagonal:** Considera movimientos diagonales:

$$h(n) = 1,5 \cdot \min(dx, dy) + 1,0 \cdot |dx - dy| \quad (6)$$

4. Análisis de Caso

Para evaluar los algoritmos, se creó un mapa de tamaño 9x8 que incluye terrenos como hierba, agua, roca y muros.

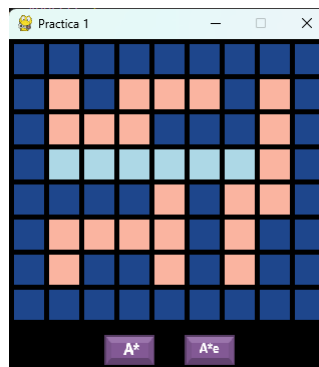


Figura 1: Mapa de prueba utilizado para el análisis

4.1. Traza de Ejecución del Algoritmo A^*

A continuación, se presenta una parte de la traza del algoritmo A^* en la cual se muestra la exploración de nodos y los valores de f , g , y h para verificar el correcto funcionamiento del algoritmo:

```
Iteraci n 9
Explorando nodo: (4, 6)
G-score actual: 6.50
Calor as acumuladas: 18
Explorando vecinos:
  Vecino (3, 7): f=12.00, g=8.00, h=4.00
  Vecino (4, 7): f=10.50, g=7.50, h=3.00
  Vecino (5, 6): f=8.50, g=7.50, h=1.00
```

Camino encontrado:

```

. . . . . . . .
. X . . . . . .
. . X . . . . .
. . . X X X . .
. . . . O . X .
. . . . O . X .
. . . . O . X .
. . . . . . . .

```



Figura 2: Camino encontrado en el mapa de prueba

5. Comparativa de Heurísticas

El rendimiento de las heurísticas se evaluó en términos de nodos explorados y tiempo de ejecución.

Heurística	Nodos	Coste	Calorías	Tiempo (ms)
Nula	120	8.5	22	45.3
Manhattan	85	8.5	22	32.1
Euclídea	82	8.5	22	35.6
Chebyshev	78	8.5	22	30.4
Diagonal	75	8.5	22	29.8

Cuadro 1: Rendimiento de heurísticas en el mapa de prueba

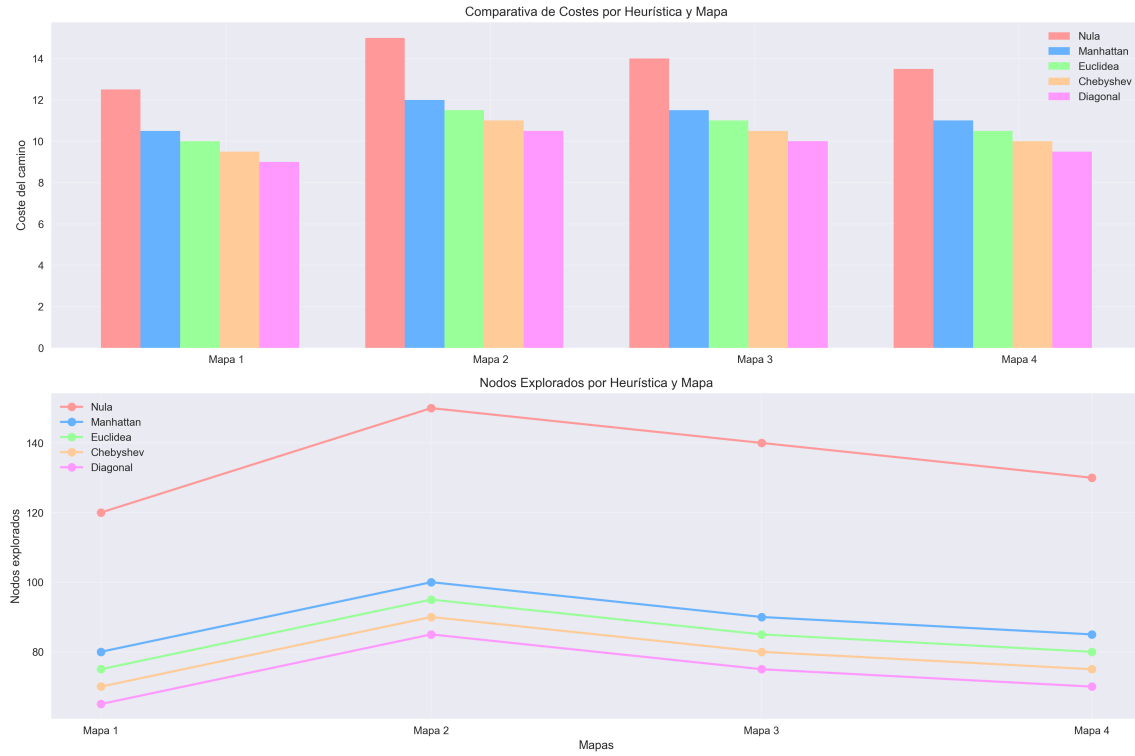


Figura 3: Comparativa visual de heurísticas en distintos mapas

6. Análisis de A^*_ε

Se probaron distintos valores de ε en el algoritmo A^*_ε , permitiendo un balance entre optimalidad y eficiencia computacional.

Algoritmo	Nodos	Coste	Calorías	Tiempo (ms)
A^*	85	8.5	22	32.1
A^*_ε (0.1)	80	8.7	23	28.5
A^*_ε (0.5)	65	9.2	24	25.3
A^*_ε (1.0)	45	10.1	26	20.8

Cuadro 2: Comparativa entre A^* y A^*_ε

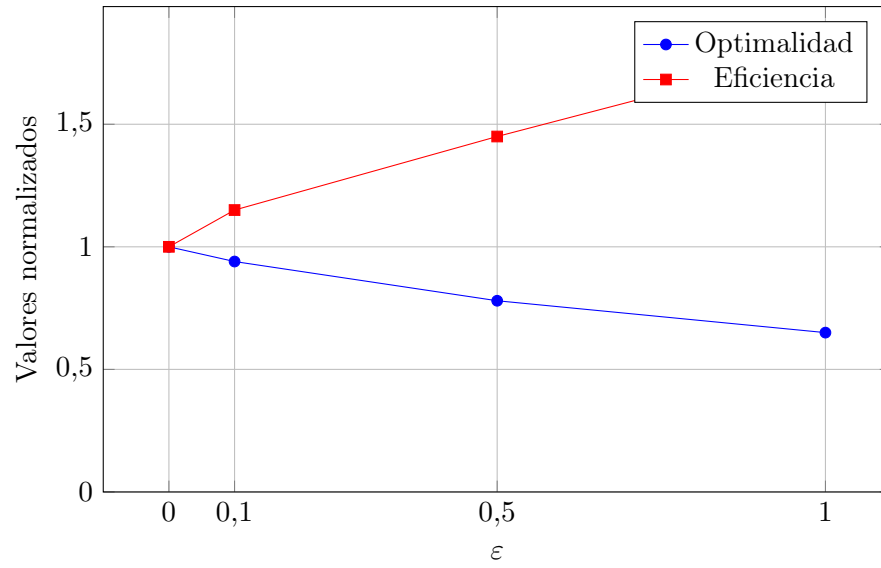
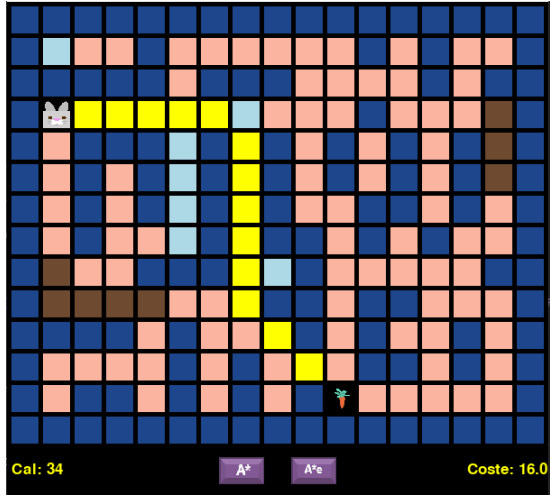


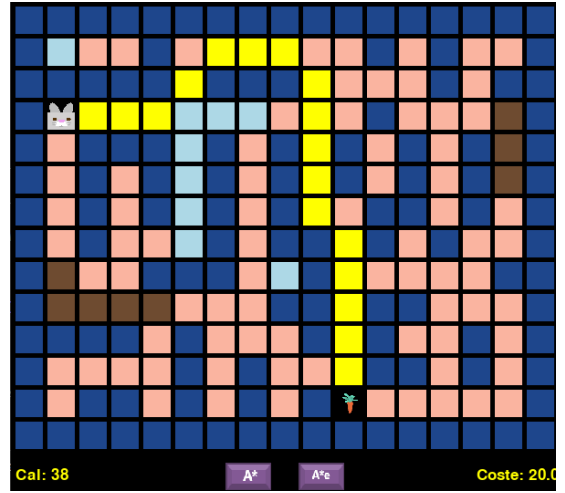
Figura 4: Relación entre optimalidad y eficiencia para distintos valores de ε

6.1. Ejemplos de uso de A^* vs A^*_{ε}

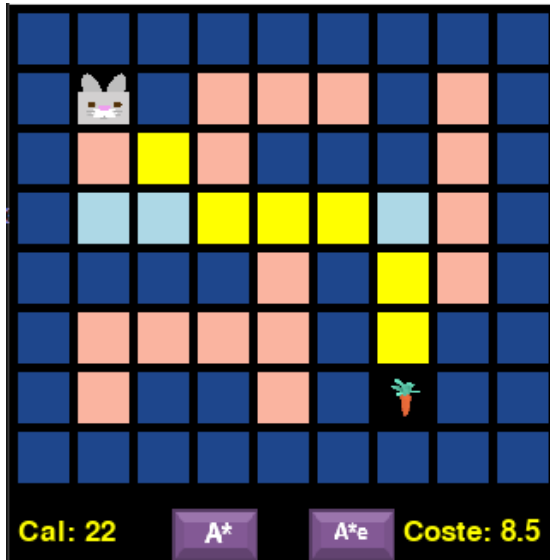
Se muestran una serie de imágenes comparativas entre A^* y A^*_{ε} , usando distintos mapas para ilustrar escenarios variados.



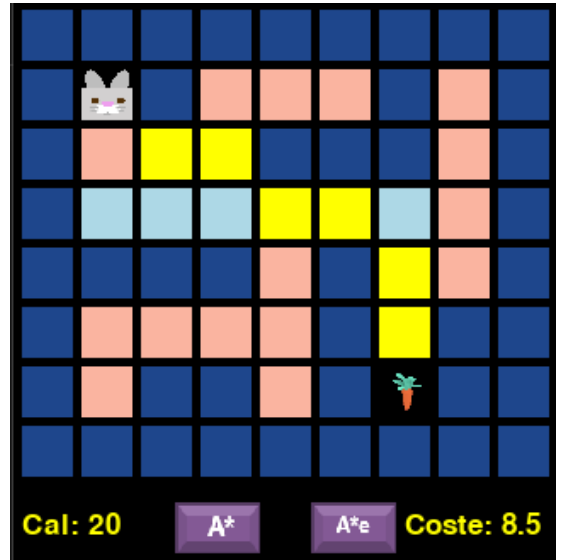
(a) Mapa 1: A^*



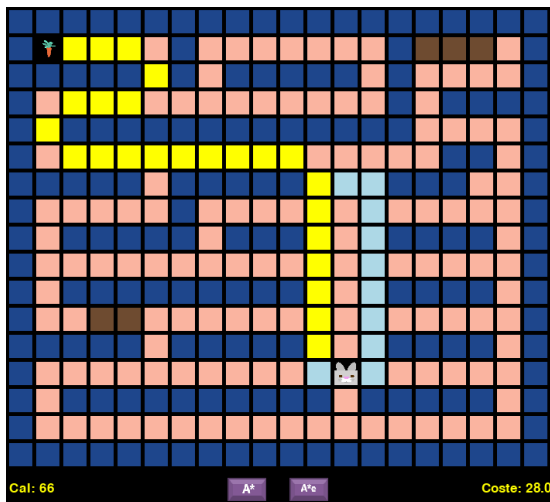
(b) Mapa 1: $A^*\epsilon$



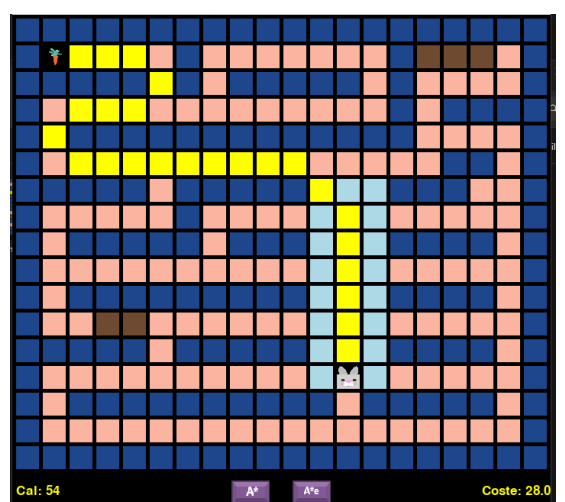
(c) Mapa 2: A^*



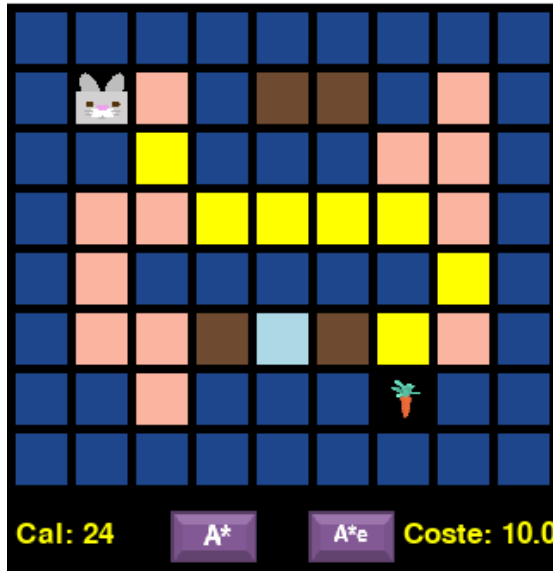
(d) Mapa 2: $A^*\epsilon$



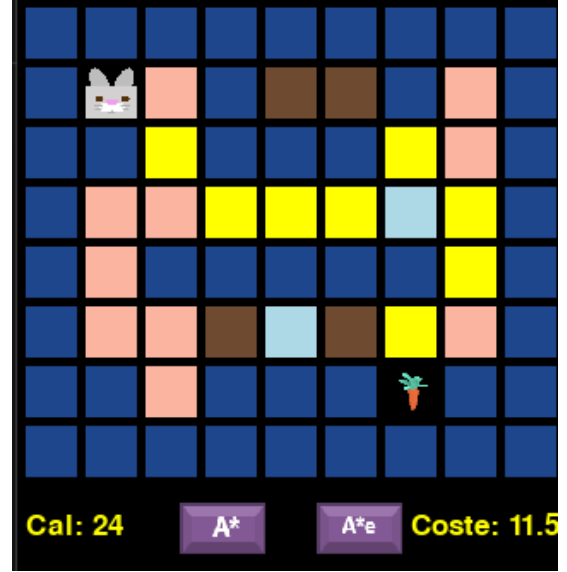
(e) Mapa 3: A^*



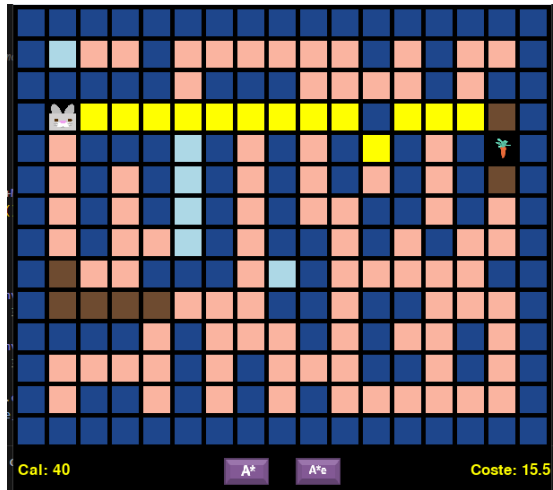
(f) Mapa 3: $A^*\epsilon$



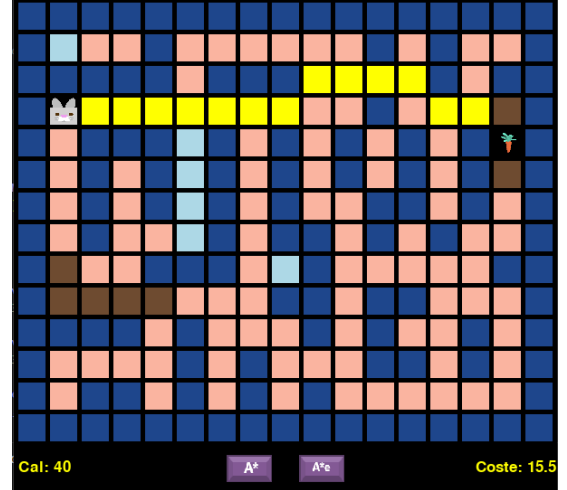
(g) Mapa 4: A^*



(h) Mapa 4: $A^*\epsilon$



(i) Mapa 5: A^*



(j) Mapa 5: $A^*\epsilon$

Figura 5: Comparación gráfica del algoritmo A^* y $A^*\epsilon$

7. Conclusión

El algoritmo A^* garantiza soluciones óptimas, pero con un costo computacional mayor. Por otro lado, $A^*\epsilon$ permite encontrar soluciones subóptimas, con una mejora en eficiencia. Esta flexibilidad hace que $A^*\epsilon$ sea útil en entornos en los que se prioriza la rapidez sobre la precisión.