

# Taller-Quiz 01: Algoritmos de Búsqueda en Inteligencia Artificial

Prof. Jorge Iván Padilla Buriticá, Ph.D.

Febrero 2026

**Duración:** 60 minutos    **Valor total:** 5 %

**Entrega:** Enviar archivo .ipynb a: jipadillab@eafit.edu.co

## Instrucciones

El taller consta de dos partes. La primera parte (50 %) se responde indicando únicamente el literal de la respuesta (ej. 1-a, 2-b). La segunda parte (50 %) requiere la elección de **una** de las dos opciones propuestas para ser resuelta mediante código Python.

---

## 1. Parte 1: Preguntas de Opción Múltiple (50 %)

*Seleccione la respuesta correcta para cada enunciado.*

1. ¿Cuál es la principal diferencia entre búsqueda informada y no informada?
  - a) La búsqueda informada utiliza una función heurística.
  - b) La búsqueda no informada es siempre más rápida.
  - c) Solo la búsqueda informada garantiza encontrar una solución.
2. En el algoritmo BFS (Breadth-First Search), ¿qué estructura de datos se utiliza típicamente para la frontera?
  - a) Pila (Stack / LIFO).
  - b) Cola (Queue / FIFO).
  - c) Árbol binario de búsqueda.
3. ¿Cuál es la complejidad temporal de BFS en términos del factor de ramificación ( $b$ ) y la profundidad ( $d$ )?
  - a)  $O(d^b)$
  - b)  $O(b \cdot d)$
  - c)  $O(b^d)$
4. El algoritmo A\* se define por la función de evaluación  $f(n) = g(n) + h(n)$ . ¿Qué representa  $g(n)$ ?

- a) El costo estimado desde el nodo  $n$  hasta el objetivo.
  - b) El costo real acumulado desde el inicio hasta el nodo  $n$ .
  - c) La distancia en línea recta al objetivo.
5. Para que el algoritmo  $A^*$  sea óptimo, la heurística  $h(n)$  debe ser:
- a) Consistente y admisible.
  - b) Mayor que el costo real siempre.
  - c) Un valor aleatorio entre 0 y 1.
6. ¿Cuál de estos algoritmos es de búsqueda no informada?
- a) Búsqueda Voraz (Greedy Best-First Search).
  - b) Algoritmo  $A^*$ .
  - c) Búsqueda en Profundidad (DFS).
7. ¿Qué sucede si la heurística  $h(n)$  en  $A^*$  es siempre 0?
- a) El algoritmo se convierte en BFS.
  - b) El algoritmo se comporta como el de Dijkstra.
  - c) El algoritmo no encuentra la solución.
8. La búsqueda de costo uniforme (Uniform Cost Search) es óptima si:
- a) Los costos de los arcos son mayores a cero.
  - b) Se usa una heurística potente.
  - c) El grafo es pequeño.
9. ¿Qué algoritmo expande siempre el nodo que parece estar más cerca del objetivo según la heurística, ignorando el costo acumulado?
- a)  $A^*$
  - b) Búsqueda Voraz (Greedy).
  - c) BFS.
10. ¿Cuál es una desventaja notable de BFS frente a DFS?
- a) No es completo.
  - b) El consumo de memoria puede ser muy alto.
  - c) Siempre encuentra el camino más largo.

## 2. Parte 2: Aplicación en Python (50 %)

Elija **SOLO UNA** de las siguientes opciones y desarrolle la solución en un Jupyter Notebook.

### Opción A: Navegación de Agente en Rejilla (Grid World)

Implemente un algoritmo de búsqueda ( $A^*$  o BFS) para un agente que debe moverse en un plano 2D.

- **Inicio:**  $(0, 0)$
- **Meta:**  $(9, 9)$
- **Obstáculos:** Celdas  $(2, 2)$ ,  $(2, 3)$ ,  $(2, 4)$ ,  $(5, 5)$ ,  $(5, 6)$ ,  $(6, 5)$ ,  $(7, 5)$ .
- **Movimientos:** Arriba, abajo, izquierda, derecha (costo 1).
- **Requerimiento:** Visualizar la ruta final y el costo total.

### Opción B: Ruta Óptima entre Capitales de Colombia

Utilice búsqueda informada ( $A^*$ ) para encontrar la mejor ruta terrestre entre **Cali** y **Bucaramanga**. Use la siguiente tabla de distancias aproximadas (km) para construir su grafo:

Origen	Destino	Distancia (km)
Cali	Armenia	180
Cali	Pasto	380
Armenia	Ibagué	80
Armenia	Medellín	260
Ibagué	Bogotá	200
Bogotá	Tunja	140
Tunja	Bucaramanga	280
Medellín	Bucaramanga	390

**Heurística sugerida (Distancia en línea recta a Bucaramanga):**

- Cali: 560, Armenia: 400, Ibagué: 350, Medellín: 290, Bogotá: 300, Tunja: 210, Bucaramanga: 0.

**Requerimiento:** El código debe generar el grafo, aplicar la búsqueda y mostrar la secuencia de ciudades de la mejor ruta con su kilometraje total.