

INTELIGENCIA ARTIFICIAL - 2025

INF-404-11145  
Trabajo Práctico 2

Módulo 2

Alumno: DIEGO DANIEL LUCCHELLI

Titular experto: PABLO ALEJANDRO VIRGOLINI

Titular disciplinar: MARIA PAULA GONZALEZ  
Fecha de entrega o fecha límite: 12/05/2025

Modalidad: Individual

Contenido

[Desarrollo de la actividad 2](#_Toc197879358)

[Contexto de la situación problemática 2](#_Toc197879359)

[Consignas 2](#_Toc197879360)

[Respuestas 2](#_Toc197879361)

[Anexo 5](#_Toc197879362)

[Bibliografía 5](#_Toc197879363)

# Desarrollo de la actividad

## Contexto de la situación problemática

Se encuentra en “Situación problemática” en canvas.

## Consignas

A partir de la descripción realizada del problema y sus implicancias, se deben cumplir las siguientes actividades, de forma individual.

1.    Implementar un proceso de búsqueda exhaustiva. Describir sus características, las ventajas y limitaciones frente al problema abordado y presentar un ejemplo.

2.    Implementar un proceso de búsqueda heurística. Describir sus características, las ventajas y limitaciones frente al problema abordado y presentar un ejemplo.

3.    Hacer un análisis comparativo de ambos enfoques destacando las ventajas y dificultades de los métodos adoptados en cada caso.

4.    Elaborar una conclusión que resuma el trabajo realizado, la opción considerada más apropiada y la conveniencia de resolver el problema planteado a partir de la búsqueda en el espacio de estados.

## Respuestas

#### *Implementar un proceso de búsqueda exhaustiva. Describir sus características, las ventajas y limitaciones frente al problema abordado y presentar un ejemplo.*

Cuando nos referimos a búsqueda exhaustiva, estamos hablando de un método de búsqueda utilizado en la inteligencia artificial principalmente para la resolución de problemas, se basa en la exploración del espacio de estados pasando por todos los estados posibles hasta que se encuentre una solución, en caso de existir. En este método no se utiliza información adicional y por ese motivo se lo conoce también como búsqueda a ciegas o no informados.

Existen varios métodos para implementar estas búsquedas entre los que se encuentran:

* Primero en anchura (BFS)
* Primero en profundidad (DFS)
* Descenso Iterativo (Variante de DFS)

Algunas de sus Ventajas son:

* Es determinista
* Puede ser fácil de implementar en situaciones con pocos nodos
* Permite encontrar una solución, siempre que ésta existe

Algunas de sus desventajas son:

* La alta complejidad de los árboles ramificados a menudo impide recorrer completamente el espacio de estados.
* No es eficiente, ya que muchos de los caminos recorridos pueden no llevar a la solución

El código para esta solución está realizado en Python, con ayuda de IA y modificaciones propias. Se utilizó el método primero anchura debido a que al tener solo 7 nodos resulta simple de ver e implementar.

Se simula una línea con 7 nodos como si fuera una cinta transportadora, arrancando en el nodo B=3 y finalizando en el nodo A=5. De esta manera explora cada uno de los nodos yendo de lado a lado y aumentando la distancia hasta encontrar la solución.

En el código se muestra como salida el camino recorrido, 3 → 4 → 2 → 5 y la barra posee los siguientes colores:

* 🟧 Naranja = posición inicial B
* 🟢 Verde = punto de montaje real A
* 🔷 Azul claro = pasos explorados

El código se encuentra en el archivo adjunto (probado y funcionando):

“Diego\_Lucchelli\_TP2\_Busqueda\_exhaustiva.py”

#### *Implementar un proceso de búsqueda heurística. Describir sus características, las ventajas y limitaciones frente al problema abordado y presentar un ejemplo.*

Una búsqueda heurística (hallar, descubrir), se utiliza para resolver problemas teniendo cierta información adicional que nos guía hacia una solución posible, esto hace que se deban implementar reglas de control heurístico en el proceso y trata de resolver los problemas de la búsqueda exhaustiva, a diferencia de ésta, a cada nodo se lo asocia con un valor que es una estimación de la distancia real de su cercanía a la meta. Estos valores se incorporan mediante funciones que evalúan los caminos como los más prometedores, al centrar la búsqueda en los nodos más prometedores, reduce el tamaño del espacio de estados a explorar.

Existen varios métodos para implementar estas búsquedas entre los que se destacan:

* Primero el mejor
* A\*

Algunas de sus Ventajas son:

* Puede tener mejor rendimiento que las búsquedas exhaustivas, sobre todo en espacios de estados grandes, ya que permite orientar la búsqueda hacia los nodos más prometedores
* Puede encontrar soluciones aceptables a problemas complejos
* Usa conocimiento del entorno

Algunas de sus desventajas son:

* Es más complejo de implementar
* No se garantiza encontrar la solución óptima
* La eficiencia depende de la calidad de la función heurística y puede llevar a malos resultados en caso de que la misma no se encuentre correctamente diseñada

El código para esta solución está realizado en Python, con ayuda de IA y modificaciones propias. Se utilizó el método A\* con 8 nodos. Se utilizó “heapq” que es un módulo de Python que sirve para mantener ordenados parcialmente los elementos en una cola y con cierta prioridad, muy útil para los valores de este caso para seleccionar el mejor nodo de f(n).

Como en el caso anterior, solo se analiza el eje horizontal, para encontrar el camino más eficiente y al final muestra un grafo con el camino recorrido y una secuencia de los nodos en texto.” 2 → 3 → 4 → 5 → 6”

El código se encuentra en el archivo adjunto (probado y funcionando):

“Diego\_Lucchelli\_TP2\_Busqueda\_A.py”

1. Hacer un análisis comparativo de ambos enfoques destacando las ventajas y dificultades de los métodos adoptados en cada caso.

Como ya se mencionó anteriormente la búsqueda exhaustiva es más simple de implementar en espacios de estados pequeños y brinda la posibilidad de encontrar la solución en caso de que esta exista, en cambio la búsqueda heurística puede ser más eficiente que la exhaustiva en grandes espacios de estados y encontrar soluciones prometedoras más rápidamente, en cuanto a las dificultades, me resultó un poco más complicado pensar en la función heurística y el concepto de la distancia real al objetivo, no tanto en este caso que analizamos solo el eje horizontal, sino más bien pensando en como sería en una situación en la que se incluyan los tres ejes.

Respecto al código, me encontré con problemas con algunas sentencia de librerías que estaban “obsoletas” y tuve que colocar unas líneas para ignorar el warning porque no encontré una manera de corregirlo, de todos modos el código funciona correctamente.

#### *Elaborar una conclusión que resuma el trabajo realizado, la opción considerada más apropiada y la conveniencia de resolver el problema planteado a partir de la búsqueda en el espacio de estados.*

En mi caso en particular me pareció más simple la búsqueda exhaustiva que la heurística, aunque en ambos casos hay pocos nodos queda muy claro como se va saltando hacia un lado y hacia el otro hasta llegar a la solución, además de que el código es más simple, aunque hay que estar atentos al cambio de sentido. Claramente el hecho de que solo se haya analizado para el eje horizontal simplifica mucho las cosas y resulta una buena aproximación para poder afrontar casos más complejos en el fututo. Para casos más complejos como al momento de analizar desplazamientos los tres ejes, usaría la búsqueda heurística.

# Anexo

Los archivos de Python se encuentran en el siguiente repositorio:

<https://github.com/lucca86/UE21_Inteligencia_Artificial_2025>

# Bibliografía

* **Russell, S. y Norvig, P.**(2004). *Inteligencia artificial. Un enfoque moderno.* España: Pearson.
* **Nilsson, J. N. (2001**). *Inteligencia artificial. Una nueva síntesis.* Madrid: Mc GrawHill
* **García Serrano, A.**(2012).*Inteligencia artificial. Fundamentos, práctica y aplicaciones.* España Alfaomega.