

# GUÍA PRÁCTICA

1. DATOS GENERALES	
<b>Asignatura:</b> SISTEMASINTELIGENTES	<b>Código de la Asignatura:</b> SIS-341
<b>Carrera:</b> Ingeniería de Sistemas	
<b>Curso:</b> A	<b>Semestre:</b> SÉPTIMO
<b>Contenido Analítico:</b> <ul style="list-style-type: none"> <li>•Introducción</li> <li>•Fundamentos de Inteligencia Artificial</li> </ul>	<b>Unidad Temática:</b> Introducción a la Inteligencia Artificial <b>Email:</b> vrodriqueze@doc.emi.edu.bo
<b>Docente:</b> Msc.Victor Rodríguez Estévez	
<b>Bibliografía a seguir:</b> <ul style="list-style-type: none"> <li>• Inteligencia Artificial un Enfoque Moderno</li> </ul>	
<b>Práctica:</b> 1	<b>Título:</b> Introducción a la Representación de Estados
<b>Material de Apoyo:</b> Diapositivas, lecturas	<b>Carga horaria:</b> 6

2. OBJETIVO
<ul style="list-style-type: none"> <li>• Representar problema</li> <li>• Implementar algoritmo de búsqueda en la representación</li> <li>• Establecer heurísticas admisibles para mejorar la eficiencia de la búsqueda</li> <li>• Describir rendimiento y los resultados del programa agente propuesto.</li> <li>• Construir el agente que resuelve el problema de del rompecabezas N</li> </ul>

3. SOFTWARE, SIMULADORES Y/O EQUIPOS	
<b>Detalle</b> Interprete Python IDE Pycharm	<b>Cantidad</b> 27 27

## 1 Introducción

En esta práctica debes realizar la implementación de tres heurísticas admisibles para resolver el juego **Rompe Cabezas N**. El Rompe Cabezas N es un clásico problema de un tablero matricial de  $N \times N$  en cual hay  $(N \times N - 1)$  posiciones ocupadas por fichas numeradas (en forma aleatoria) además de un espacio vacío (sin número asignado), la idea es resolverlo para ello debemos mover este espacio de forma horizontal (h) o vertical (v), con la ficha adyacente. El objetivo es llegar de un estado inicial (desordenado) a un estado final (ordenado). El Agente debe resolver el problema guiado por una heurística admisible.

## 2 Metodología

1. Representar en una estructura de datos, el problema, identificando Estado inicial y estado meta.
2. Definir las funciones sucesores que permitan cambiar un estado a otro, respetando las restricciones del problema.

3. Definir tres funciones Heurísticas.
4. Implementar los algoritmos: Búsqueda codiciosa, A\*. utilizar las tres heurísticas planteadas.
5. Generar aleatoriamente 1000 posibles estados iniciales (verificando que sean válidas) y encontrar las soluciones mediante búsqueda codiciosa, A estrella, y las tres heurísticas.
6. Encontrar en promedio el número de pasos en la resolución.
7. Encontrar el tiempo promedio en la resolución.
8. Verificar completitud (Se encontraron siempre las soluciones?)
9. Seleccionar la mejor heurística, sustentar con los resultados obtenidos.
10. Integrar al Agente en el juego para poder dar soporte al jugador, permitir que el usuario pueda jugar y ejecutar al agente en cualquier momento.(pygame)

## 3 Entregable

1. Documento: en pdf, el cual deberá contener:
  - Título del trabajo, autor, nombre del docente de la asignatura. (iniciar en la primera hoja, no precisa carátula)
  - Resumen: Debajo de los datos anteriores, resumen del trabajo (no mas de 4 líneas, en cursiva)
  - Introducción: Describir el problema y que se planea hacer en el trabajo.
  - Fundamentación teórica: Revisar bibliografía sobre heurísticas y teoría los algoritmos codicioso y A estrella (debidamente referenciados (normas APA 7.0). no mas de dos planas).
  - Metodología: Describir los pasos que se realizó, (no escribir código fuente), mejoras al código, si es necesario diagramas o esquemas pueden ser utilizados. Herramientas utilizadas. Patrones, etc.
  - Resultados: Mostrar los resultados. Resultados de la experimentación (Tiempo de ejecución, Nodos expandidos/generados, Longitud de la solución, Factor de ramificación efectivo) Para comparar las heurísticas: Prueba t de Student (si los datos son normales) Prueba de Wilcoxon (no paramétrica, recomendada).
  - Discusión: Interpretar y comparar los resultados, limitaciones y posibles mejoras con otras técnicas.
  - Conclusiones: Contrastar lo estudiado en la teoría y lo experimentado en la práctica.
2. Código Fuente: Lenguaje de programación Python, la librería pygame **código ordenado** y documentado.

**Fecha Entrega:** Impostergablemente 12/09/2025.

**Evaluando el proceso :** Estrategia de pensamiento metacognitiva: "Identificar y corregir errores".

Piensa si la práctica te ha emocionado y crees que tiene utilidad para tu aprendizaje. Para ello califica de 1 a 5 la práctica:

1. Nada 2. Regular 3. Ni mucho ni poco 4. Mucho 5. Bastante

"Reconocer lo que no se ha hecho bien o lo que falta". Si has puntuado 3 o menos, indícame qué crees que tengo que cambiar en el diseño de la práctica para mejorar.