

Entrega 1

1. DESARROLLO DE UN AGENTE DE BÚSQUEDA

Introducción

Durante este curso hemos abordado los siguientes temas fundamentales:

- **Agentes de Inteligencia Artificial:** Sistemas capaces de percibir su entorno y actuar sobre él con base en objetivos. Su arquitectura básica involucra sensores, actuadores, y una función de agente que define el comportamiento.
- **Agentes de búsqueda:** En entornos donde el conocimiento es incompleto o no reactivo, los agentes deben deliberar. Estos agentes pueden clasificarse en:
 - **Agentes de búsqueda ciega (no informada):** No utilizan conocimiento adicional del problema, como BFS o DFS.
 - **Agentes de búsqueda informada:** Hacen uso de una heurística para estimar la calidad de los estados y tomar decisiones más eficientes (ej. algoritmo A*, Greedy Search).

Objetivo del Taller

Diseñar e implementar un agente de búsqueda para un problema definido por ustedes, si es el caso, pueden usar una heurística propia y justificar el algoritmo elegido según propiedades teóricas.

Desarrollo del Agente

Paso 1: Selección del problema

Cada grupo de estudiantes debe elegir un problema original donde se aplique un algoritmo de búsqueda. Algunos ejemplos:

- Navegación de robots
- Planeación de rutas (GPS)
- Juegos de estrategia
- Solución de rompecabezas (como el 8-puzzle)
- Planeación de tareas en producción o logística

Paso 2: Justificación del algoritmo

Deben seleccionar un **algoritmo de búsqueda ciega o informada, según el problema a resolver** (Best First Search, Breadth First Search, Deep Search, A*, Greedy Best First Search, etc.) y justificar su elección con base en:

Complejidad	¿El algoritmo garantiza encontrar una solución si existe?
Optimalidad	¿La solución encontrada es la mejor posible?
Complejidad en memoria	¿Cuánta memoria consume en el peor de los casos?
Complejidad temporal	¿Cuánto tiempo puede tardar en ejecutarse?

Deben comparar al menos dos algoritmos antes de seleccionar uno.

Paso 3: Definición del espacio de estados

Deberán estructurar el problema en términos de:

- **States (estados):** ¿Cómo se representa el estado del agente?
- **Actions (acciones):** ¿Qué puede hacer el agente en cada estado?
- **Result (modelo de transición):** ¿A qué nuevo estado lleva cada acción?
- **Action cost (costo):** ¿Cuánto cuesta realizar cada acción?

Esta parte debe quedar explicada en la wiki del repositorio con gráficos o pseudocódigo si es necesario. La mejor forma de hacerlo es usando un esquema detallado del espacio de estados

Paso 4: Diseño de la heurística

Si usan una heurística, deben:

- Definirla matemáticamente
- Explicar de dónde surge (ejemplo: distancia en el espacio, diferencia de posiciones, número de piezas fuera de lugar, etc.)
- Indicar si es admisible y/o consistente

Paso 5: Desarrollo del código

Implementar el agente de búsqueda informada en un notebook o script de Python.

La solución debe:

- Representar correctamente el espacio de búsqueda
- Encontrar el camino desde el estado inicial al estado meta
- Mostrar paso a paso (si es posible) la exploración de nodos

Paso 6: Presentación de resultados

Deben presentar:

- El camino encontrado
- El número de nodos explorados

- Tiempo de ejecución
- Comparación con otro algoritmo (si lo desean)

2. PROBLEMA DE ALGORITMOS EVOLUTIVOS

Establecer una situación de tu vida diaria o entorno cercano que implique tomar decisiones bajo restricciones. Por ejemplo:

- Organizar horarios semanales
- Elegir qué materias cursar y cuándo
- Optimizar tu entrenamiento o tiempo libre
- Armar equipos o turnos en un grupo de trabajo
- Planificar un viaje con múltiples destinos

La tarea consiste en:

- Describir el problema en lenguaje claro (1 párrafo).
- Indicar si se trata de un problema de planificación (qué hacer y cuándo) o de programación (ordenar tareas u operaciones con recursos).
- Explicar por qué sería adecuado usar un algoritmo genético para resolverlo.
- Describir cómo representar una posible solución (cromosoma) y cómo evaluar su calidad (fitness).
- Describir los elementos del GA en tu problema: selección, cruce, mutación y criterio de parada.

ENTREGABLE

Repositorio de GitHub que incluya:

Carpeta / Sección	Contenido requerido
/Agent/	Notebook o script con el agente funcional
/GA/	Notebook o script con el GA funcional
/imgs/	Diagramas de estados, visualización de búsqueda, etc.
README.md	Breve resumen del problema, participantes y tecnologías usadas
Wiki de GitHub	Explicación completa del proceso: introducción teórica, definición del problema, desarrollo del agente, heurística utilizada, análisis y conclusiones.

