

Hoja de Trabajo 3: Agentes Inteligentes en Python

Curso: Inteligencia Artificial

Tema: Agentes Inteligentes

Valor: 2 puntos

Entrega: Versionado en GitHub



Instrucciones:

- Desarrolla cada uno de los siguientes problemas de agentes inteligentes en **Python**.
 - Crea un repositorio en **GitHub** y sube cada solución en un archivo separado.
 - El nombre del repositorio debe ser: **agentes-inteligentes-{tu_nombre}**.
 - En el **README.md**, explica brevemente el funcionamiento de cada agente.
 - Realiza al menos **2 commits** por cada problema (inicio y finalización).
-

Problema 1: Agente de Patrullaje (Reflejo Simple)

Crea un agente reflejo que patrulle una ruta predefinida. El agente debe seguir un camino fijo y reaccionar si detecta un obstáculo, cambiando su dirección de manera aleatoria.

Requisitos:

- Definir un entorno con una ruta de patrullaje.
- Implementar detección de obstáculos y cambio de dirección.
- Mostrar los movimientos del agente en la consola.

Problema 2: Agente Explorador de Mapas (Con Estado Interno)

Desarrolla un agente que explore un entorno representado como una cuadrícula. El agente debe recordar las áreas visitadas y evitar repetir caminos ya explorados.

Requisitos:

- Representar el entorno con una matriz de celdas.
- Almacenar en memoria las posiciones ya visitadas.
- El agente debe moverse explorando nuevas zonas.

Problema 3: Agente de Navegación Autónoma (Basado en Metas)

Crea un agente que tenga como meta encontrar la salida en un laberinto de tamaño 5x5. Debe buscar la ruta más corta para alcanzar la meta.

Requisitos:

- Definir un laberinto con paredes y una meta.
- Implementar un algoritmo de búsqueda para encontrar la salida.
- Mostrar la ruta seguida por el agente.

Problema 4: Agente de Selección de Rutas (Basado en Utilidad)

Diseña un agente que seleccione la mejor ruta en un entorno con múltiples caminos y valores de recompensa. El agente debe elegir el camino con mayor utilidad.

Requisitos:

- Representar el entorno con valores de recompensa en cada celda.
- Implementar una función de utilidad para evaluar los caminos.
- Imprimir el recorrido óptimo seleccionado por el agente.

Entrega:

- Enviar el enlace al repositorio de GitHub.
- Cada problema debe estar en un archivo `.py` separado.
- Incluir un `README.md` explicativo.

Punteo (2 puntos):

- 0.5 puntos por cada problema completado.
- 0.5 puntos por el versionado correcto en Git (nombre, commits y README).