

## Hoja de Trabajo: 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 Semáforo Inteligente (Reactivo)

Crea un agente que controle un semáforo, cambiando su estado (verde, amarillo, rojo) según el flujo del tráfico. El agente debe ajustar la duración de cada estado según el número de vehículos detectados.

#### Requisitos:

- Usar condiciones para cambiar el estado del semáforo.
- Simular la detección de vehículos y ajustar el tiempo de espera.
- Imprimir el estado del semáforo en cada cambio.

### 🤖 Problema 2: Agente Buscador de Objetos (Basado en Objetivos)

Desarrolla un agente que se desplace en una cuadrícula (**5x5**) buscando un objeto colocado aleatoriamente, avanzando paso a paso hasta encontrarlo.

#### Requisitos:

- Representar el entorno con una matriz.
- Usar un bucle para mover al agente hacia el objeto.
- Mostrar la cuadrícula en cada movimiento.

### **Problema 3: Sistema Experto para Diagnóstico Simple (Basado en Conocimiento)**

Crea un sistema experto que realice diagnósticos básicos según los síntomas ingresados por el usuario.

#### **Requisitos:**

- Usar condicionales para definir reglas de diagnóstico.
- Solicitar al usuario que ingrese síntomas.
- Imprimir un diagnóstico según las reglas.

### **Problema 4: Agente de Recomendación de Películas (Basado en Aprendizaje)**

Diseña un agente que recomiende películas en función del género favorito del usuario.

#### **Requisitos:**

- Definir un diccionario de películas organizadas por género (acción, comedia, drama, etc.).
  - Pedir al usuario que indique su género favorito.
  - Mostrar una recomendación aleatoria.
- 

#### **Entrega:**

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



#### **Evaluación (2 puntos):**

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