



## **Reto de Robótica Móvil y Aprendizaje Reforzado**

### **Persecución Autónoma en el Planeta Colonizado**

#### **Trama:**

Han pasado varios años desde la colonización del planeta, y la tecnología ha evolucionado rápidamente. Los vehículos autónomos ahora dominan las ciudades, y la seguridad es gestionada por robots inteligentes. En este reto, se enfrentan al desafío de programar dos robots con roles opuestos: un **robot policía** (representado por un círculo) y un **robot ladrón** (representado por un triángulo).

El objetivo del robot policía es capturar al ladrón lo más rápido posible, mientras que el ladrón debe evadirlo moviéndose estratégicamente. El entorno sigue siendo un mapa con casillas transitables de **roca** y casillas peligrosas de **lava**. Es de aclarar que los roles de los robots pueden intercambiarse en cualquier momento, es decir, el policía se convierte en ladrón y viceversa.

---

#### **Objetivo del Reto:**

Diseñar e implementar un sistema autónomo que permita a los robots:

1. Navegar estratégicamente por el mapa, considerando las restricciones y costos asociados a los diferentes tipos de terreno.
2. Diseñar una **tabla de estados y acciones DE MANERA ÓPTIMO** para resolver el problema desde la perspectiva de ambos roles (policía y ladrón).
3. Implementar los algoritmos de **Q-Learning** y **SARSA** para optimizar las estrategias de movimiento de ambos robots.
4. Comparar el desempeño de ambos algoritmos en términos de convergencia, tiempo de ejecución y efectividad en cumplir los objetivos de cada rol.

---

#### **Instrucciones:**

##### **1. Mapa y Movimientos:**

- El mapa está representado como una matriz que contiene casillas de roca y lava.
- Los robots pueden moverse en las direcciones: **arriba, abajo, izquierda, derecha**.
- Cada casilla tiene un costo asociado:
  - Moverse en roca: -1 punto.
  - Moverse a una casilla de lava: -20 puntos (termina el juego).
  - El policía captura al ladrón: +50 puntos.



## 2. Diseño óptimo de la tabla de Estados y Acciones

## 3. Implementación de Algoritmos de Aprendizaje por Refuerzo:

- Implementen **Q-Learning** para aprender las mejores estrategias de movimiento para ambos roles.
- Programen **SARSA** como alternativa y comparen su desempeño.
- Asegúrense de que ambos algoritmos puedan adaptarse al cambio de roles en tiempo real.

## 4. Comparación y Optimización:

- Analicen los resultados obtenidos con cada algoritmo en términos de:
  - Velocidad de convergencia.
  - Tiempo de ejecución en simulaciones.
  - Efectividad para cumplir los objetivos de cada rol.
- Optimicen las estrategias ajustando parámetros como la tasa de aprendizaje ( $\alpha$ ) y el factor de descuento ( $\gamma$ ).

---

### Criterios de Evaluación:

1. Correctitud y optimalidad de la tabla de estados y acciones diseñada.
2. Precisión y eficiencia de los algoritmos implementados para ambos roles.
3. Eficacia del sistema al adaptarse al cambio de roles en tiempo real.
4. Calidad de la comparación entre **Q-Learning** y **SARSA**.
5. Innovación en las estrategias utilizadas para optimizar el desempeño de los robots.

---

### Entrega Esperada:

- Código funcional que incluya el reconocimiento del mapa, la tabla de estados y acciones, y la implementación de **Q-Learning** y **SARSA**.
- Gráficos o reportes que muestren la evolución de las recompensas y las estrategias óptimas aprendidas.
- Ejecución en tiempo haciendo uso de los robots donde muestre el funcionamiento del sistema en diferentes escenarios y roles.