

# Agente de Navegación Autónoma

*Optimizando Rutas con Inteligencia Artificial*

30 de noviembre de 2025

# El Desafío (Inteligencia)

## Objetivo Principal

Desarrollar un agente capaz de navegar autónomamente en un entorno complejo (Grid 9x9) desde un punto **A** hasta un punto **B**.

### ⚠ Dificultades del Entorno:

- ▶ **Obstáculos ("Hoyos"):** Terminan el episodio inmediatamente (Castigo -1.0).
- ▶ **Recompensas:** Meta principal (+1.0) y Objeto opcional (+0.7).

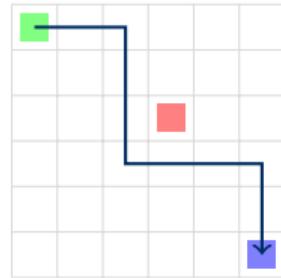


Figura: Esquema de Navegación

# La Estrategia (Proceso Empresarial)

## Cambio de Paradigma

Abandonamos la programación imperativa ("Si X, haz Y") por un enfoque de **Aprendizaje Automático**.

- ➊ **Aprendizaje Empírico:** El agente no conoce las reglas físicas; las descubre interactuando (Prueba y Error).
- ➋ **Simulación Masiva:** Ejecución de **10,000 episodios** para refinar la política de toma de decisiones.
- ➌ **Optimización:** El sistema mejora gradualmente buscando maximizar la recompensa total acumulada.

# Solución Técnica (Tecnología)

## Algoritmo: Policy Gradient (REINFORCE) El

"cerebro" del agente es una tabla de probabilidades (Softmax) que se ajusta dinámicamente.

### Mecánica de Aprendizaje

1. El agente actúa aleatoriamente al inicio.
2. Si gana → **Refuerza** las acciones.
3. Si pierde → **Debilita** las acciones.

### Stack Tecnológico

-  Python 3.x: Núcleo.
-  NumPy: Matemática.
-  Matplotlib: Gráficos.

# Resultados y Riesgos (Proceso de Cambio)

## Hallazgo Clave: Aversión al Riesgo

El agente desarrolló un comportamiento **Conservador**.

- ▶ **Observación:** El agente ignora sistemáticamente el objeto de bonificación (+0.7).
- ▶ **Causa:** El objeto está rodeado de "hoyos". El riesgo de muerte (-1.0) supera la ganancia potencial.
- ▶ **Conclusión:** El algoritmo prioriza la seguridad y el cumplimiento de la misión principal.

