

# Agente de Navegación Autónoma

*Optimizando Rutas con Inteligencia Artificial*

**Grupo: IA India de Hevia**

Integrantes:

Benjamín Camus

Nicolás Armijo

Francisco González

Giovanni Mealla

2 de diciembre 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).

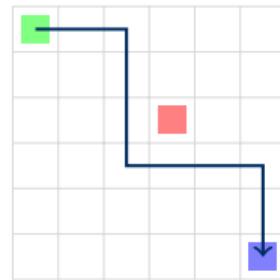


Figura: Esquema de Navegación

## Fase 2. Proceso Empresarial (Videojuegos e IA)

### Objetivo Estratégico

¿Qué busca el negocio con IA dentro del videojuego?

- ▶ Crear un **comportamiento inteligente competitivo** que mejore la experiencia del jugador.
- ▶ Reducir costos de diseño manual de niveles mediante **automatización del comportamiento enemigo**.
- ▶ Incorporar IA que diferencie el producto frente a la competencia (mayor **engagement**).

## Objetivo Operativo

¿Cómo se implementará la IA dentro del proceso de producción?

- ▶ El agente REINFORCE decide la navegación óptima en un **mapa estilo dungeon / arena**.
- ▶ Integración con el motor de juego (Unity/Unreal) mediante una capa de **estado–acción**.
- ▶ Entrenamiento autónomo con más de **10.000 simulaciones internas** por episodio.

## Relación con el entorno

El entorno de 9x9 simula:

- ▶ Toma de decisiones en mapas con **obstáculos**, rutas alternativas y objetos.
- ▶ Un sistema de recompensas comparable a la **economía interna de un videojuego**.
- ▶ Un jugador/enemigo que prioriza eficiencia y evita situaciones de riesgo.

# 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.

