# Desafío Nº 1 – Aprendizaje por Refuerzo I (FIUBA 2025)

## 1. Introducción

En este trabajo se aborda la resolución de un problema simple mediante Aprendizaje por Refuerzo, utilizando el algoritmo Q-Learning. Se implementa un entorno personalizado estilo FrozenLake de 5x5 sin estocasticidad (acciones deterministas), sin usar la librería Gym obsoleta. El objetivo es aprender una política óptima que maximice la recompensa llegando al objetivo sin caer en pozos.

## 2. Definición del Problema

Se diseña un entorno cuadrado 5x5 en el cual un agente inicia en la posición (0,0) y debe llegar a la posición (4,4). El entorno contiene casillas seguras (F), pozos (H) y una meta (G). El agente puede moverse en cuatro direcciones y recibe +1 al alcanzar la meta, y 0 en otros casos. El episodio termina si cae en un pozo o si llega al objetivo. La política se aprende mediante Q-Learning.

## 3. Implementación del Entorno y Algoritmo

Se desarrolló el entorno herendando de gymnasium.Env, utilizando espacios de observación y acción discretos. La tabla Q se inicializa en ceros y se actualiza mediante la ecuación clásica del algoritmo Q-Learning con política ε-greedy. Los parámetros utilizados fueron: tasa de aprendizaje 0.1, factor de descuento 0.99, ε inicial de 1.0 con decaimiento hasta 0.1.

## 4. Resultados y Gráfico de Convergencia

Se entrenó el agente durante 1000 episodios. Se observó convergencia en la política aprendida, reflejada en un aumento progresivo de la recompensa media. A continuación se presenta el gráfico de convergencia (Reward vs Episodios).

\*\*[Insertar gráfico aquí]\*\*

## 5. Política Aprendida

Una vez entrenado el agente, se imprime la política aprendida representada con flechas en la grilla, indicando la mejor acción en cada estado según la tabla Q. Las casillas con pozos se indican como 'H', la meta como 'G', y el resto con las direcciones ← ↓ → ↑.

## 6. Conclusiones

El experimento demostró la efectividad del algoritmo Q-Learning en un entorno personalizado. El agente fue capaz de aprender una política óptima y generalizable sin el uso de librerías obsoletas. El diseño modular permite extender el entorno y probar otros algoritmos como SARSA o Monte Carlo fácilmente.

## 7. Repositorio del código

https://github.com/usuario/repo-desafio-ar1 (a completar por el alumno)