



BATTLESHIP

PRESENTADO POR :
CAYETANO MOLINA
ESTEFANIA ELVIRA
PRISCILLA GONZÁLEZ





DESCRIPCIÓN DEL PROBLEMA

Consiste en derribar los barcos del oponente. Cada jugador cuenta con 2 cuadrículas, una para poner sus barcos y otra donde intenta adivinar dónde se encuentran los barcos de su oponente. El juego se divide en 2 fases, la fase de posicionamiento y la fase de ataque.



ANÁLISIS

Dentro de las consideraciones a tomar en cuenta hay que aclarar que el propósito del agente es que aprenda a reconocer patrones en cuanto a sus tiros. Que entienda que si tiene un acierto, lo más posible es que haya un cuadro cercano que también sea un acierto. Incluso, se debe tomar en cuenta que dado que no hay otro oponente como tal, cuando el atacante hunda algún barco no va a existir una respuesta de que en verdad hundió un barco. Es decir, cada vez que el agente tire un torpedo, lo único que podrá saber es si fue un acierto o bien si fue un fallo.



SOLUCIÓN

MEMORIA

se pueden utilizar modelos o algoritmos que tomen eso en cuenta como lo son los algoritmos de aproximación de funciones.

AMBIENTE

En este ambiente se podrían definir tanto el tamaño de la cuadrícula de juego como el tamaño y cantidad de los barcos utilizados.

FUNCIÓN

Se tendrán 3 valores, -1 para representar espacios vacíos, 0 para representar fallos y 1 para representar aciertos.

RECOMPENSA

- Aciertos = 1 punto
- Primer fallo = 0 puntos
- Volver a disparar a una casilla ya antes disparada = -0.2 puntos
- Fallo pero tiene aciertos anteriores cerca = 0.2 puntos por cada otro acierto a menos de 2 casillas.
- Ganar la partida = 10 puntos

COMPROBAR

Se llevará un conteo de los aciertos a cada barco y se comparará con el tamaño total que ocupan los barcos.

HERRAMIENTAS APLICADAS

Para la solución del problema se utilizaron 2 librerías específicamente para llevar a cabo el entrenamiento del agente y creación del ambiente. Además de algoritmos.

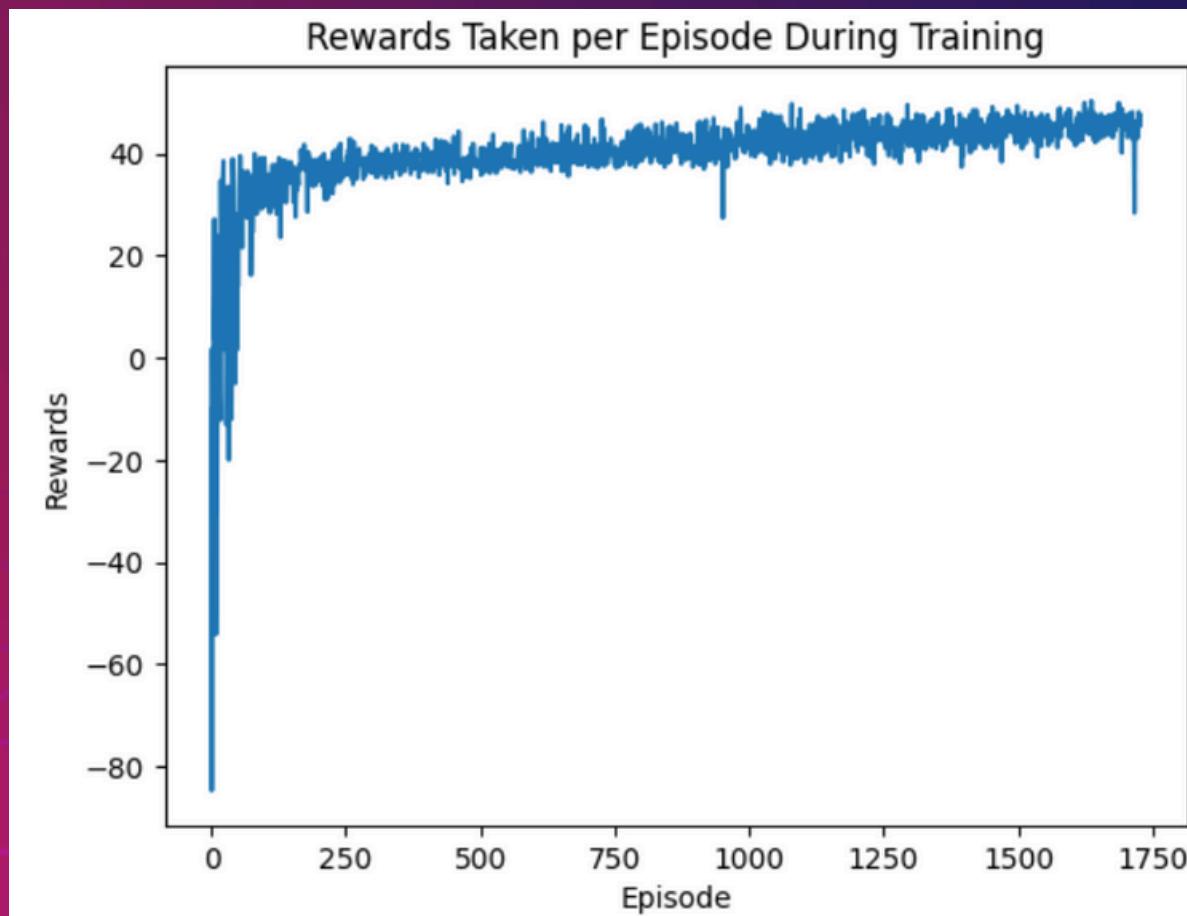
- ▶ GYMNASIUM
- ▶ STABLE BASELINE 3
- ▶ PPO
- ▶ DQN
- ▶ A2C



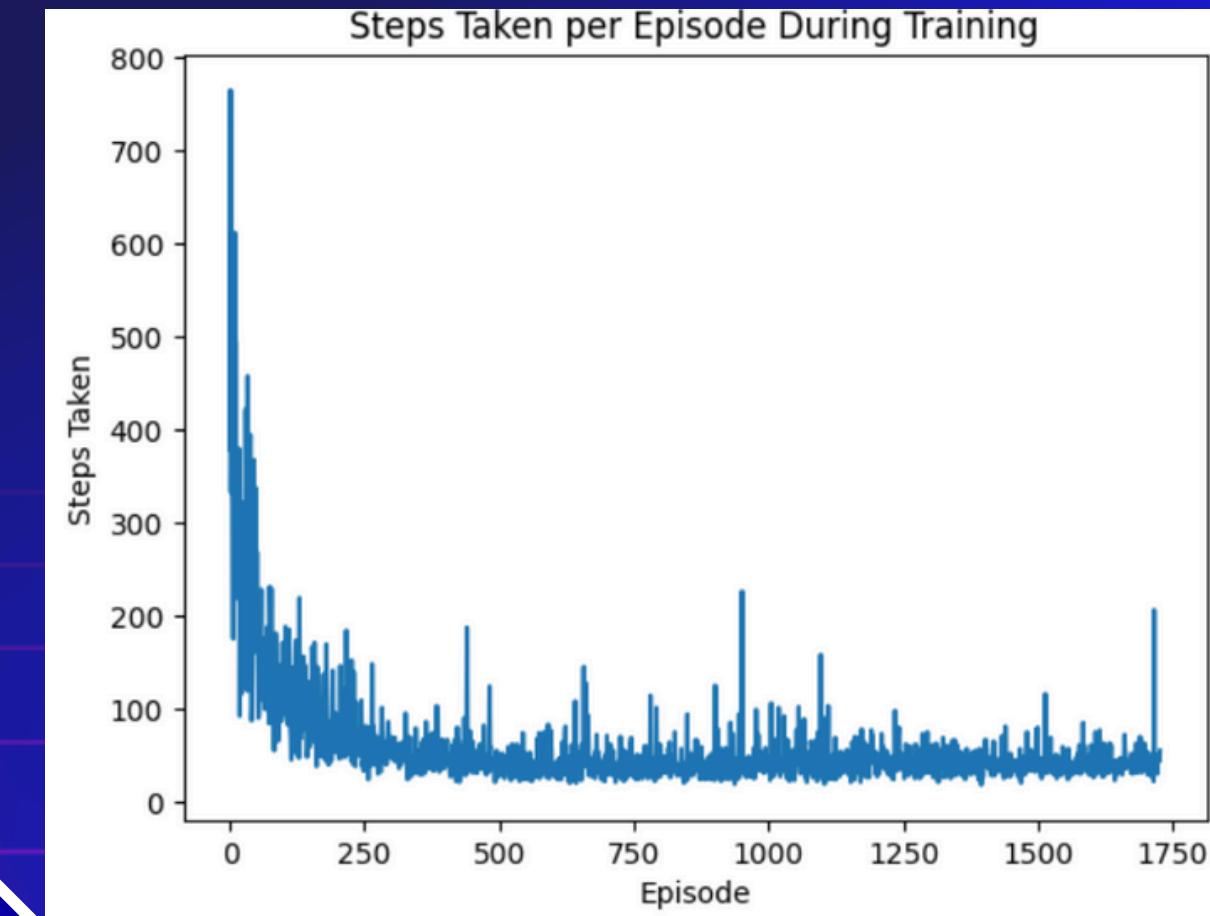
RESULTADOS PPO



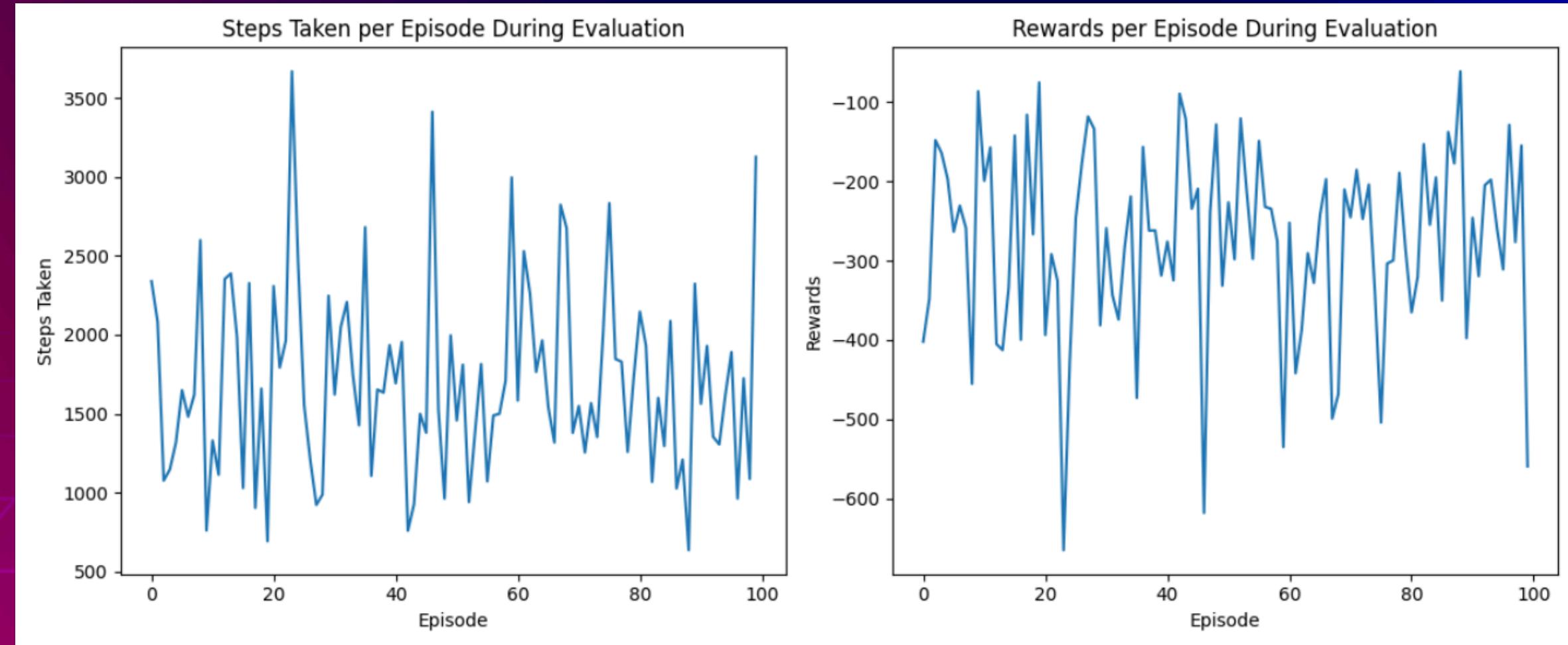
RECOMPENSA OBTENIDA POR CADA EPISODIO PARA ALGORITMO PPO



PASOS TOMADOS POR CADA EPISODIO PARA ALGORITMO PPO



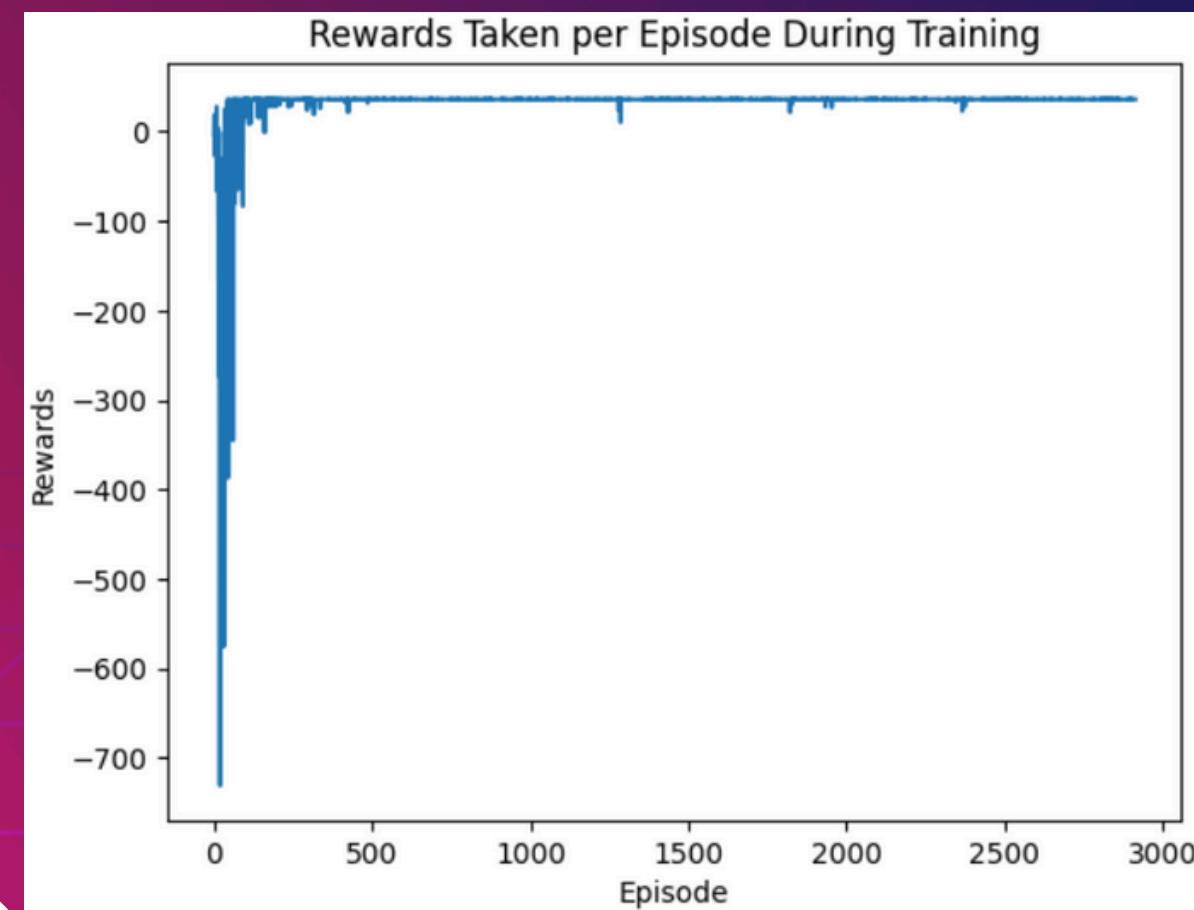
MÉTRICAS DEL ALGORITMO PPO PARA AMBIENTE DE EVALUACIÓN



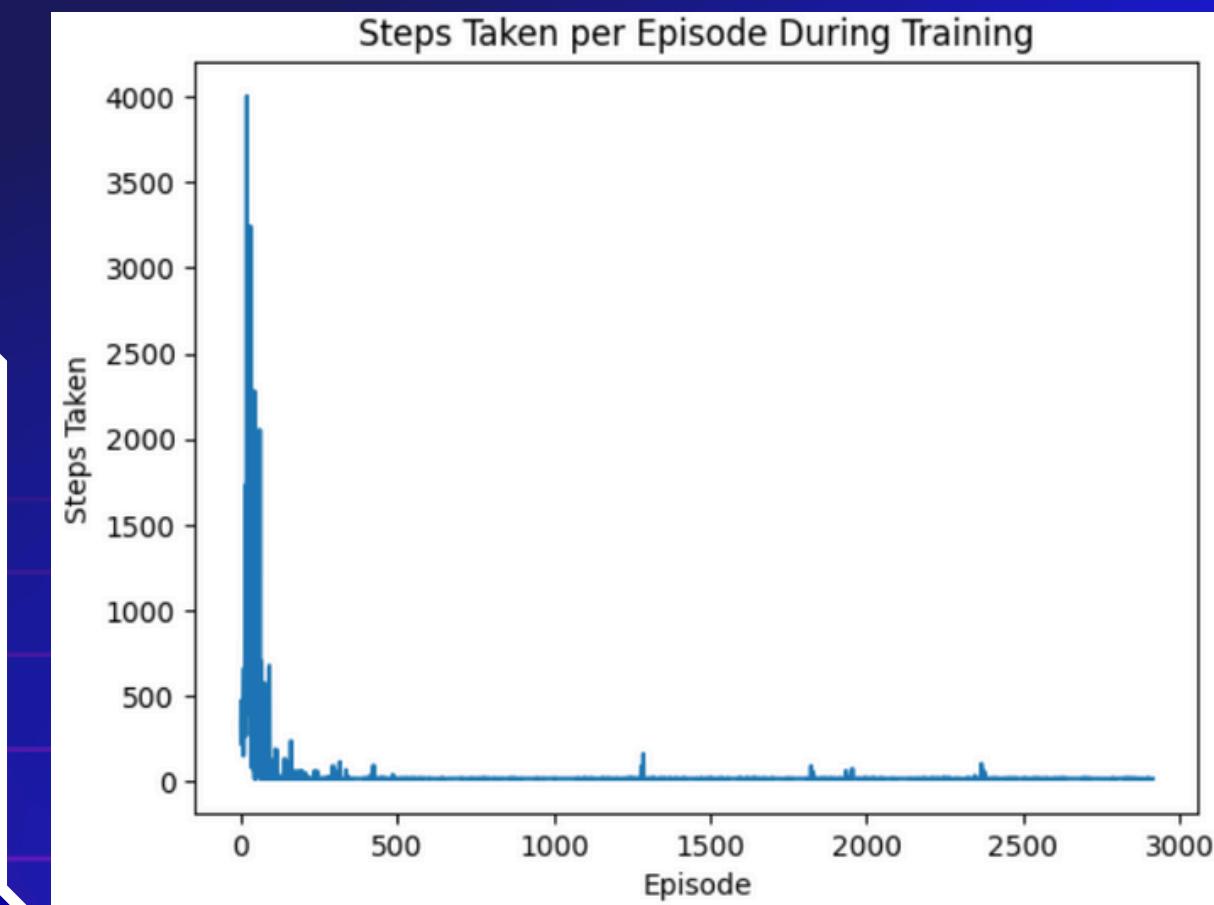
RESULTADOS DQN



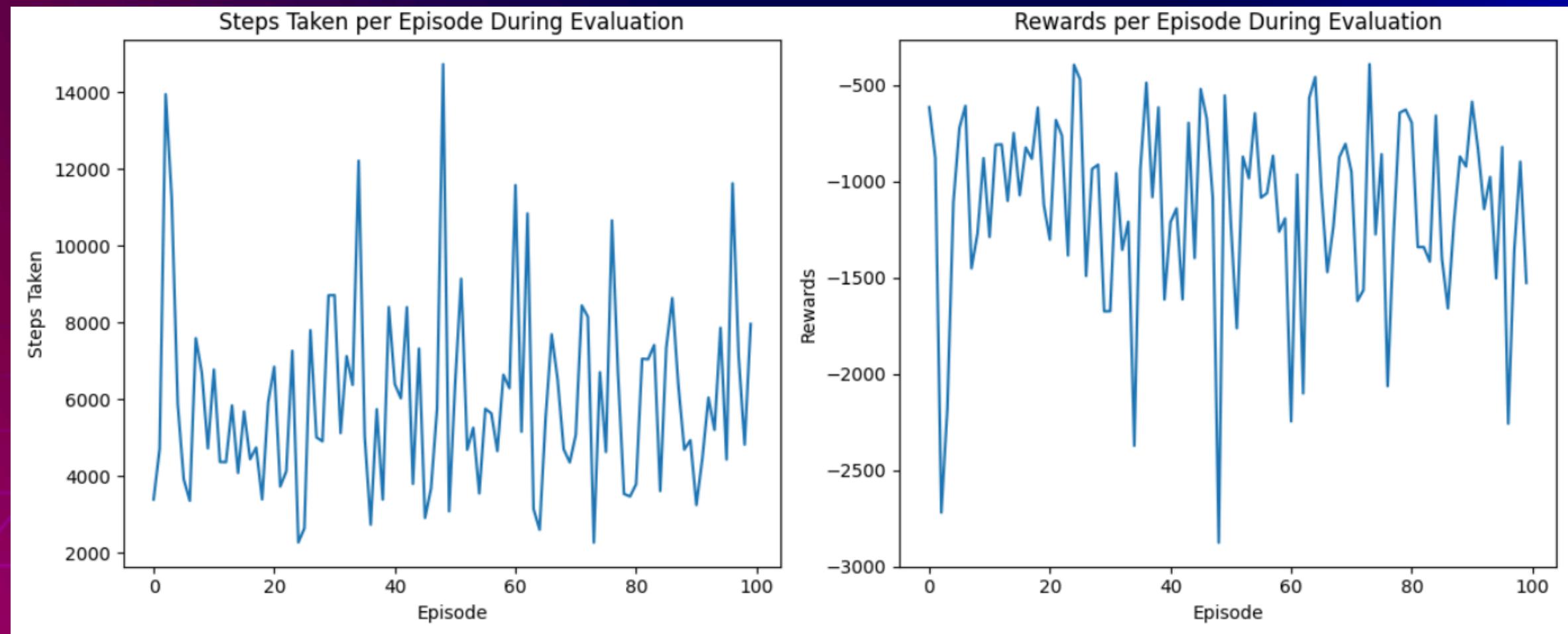
RECOMPENSA OBTENIDA POR CADA EPISODIO PARA ALGORITMO DQN



PASOS TOMADOS POR CADA EPISODIO PARA ALGORITMO DQN



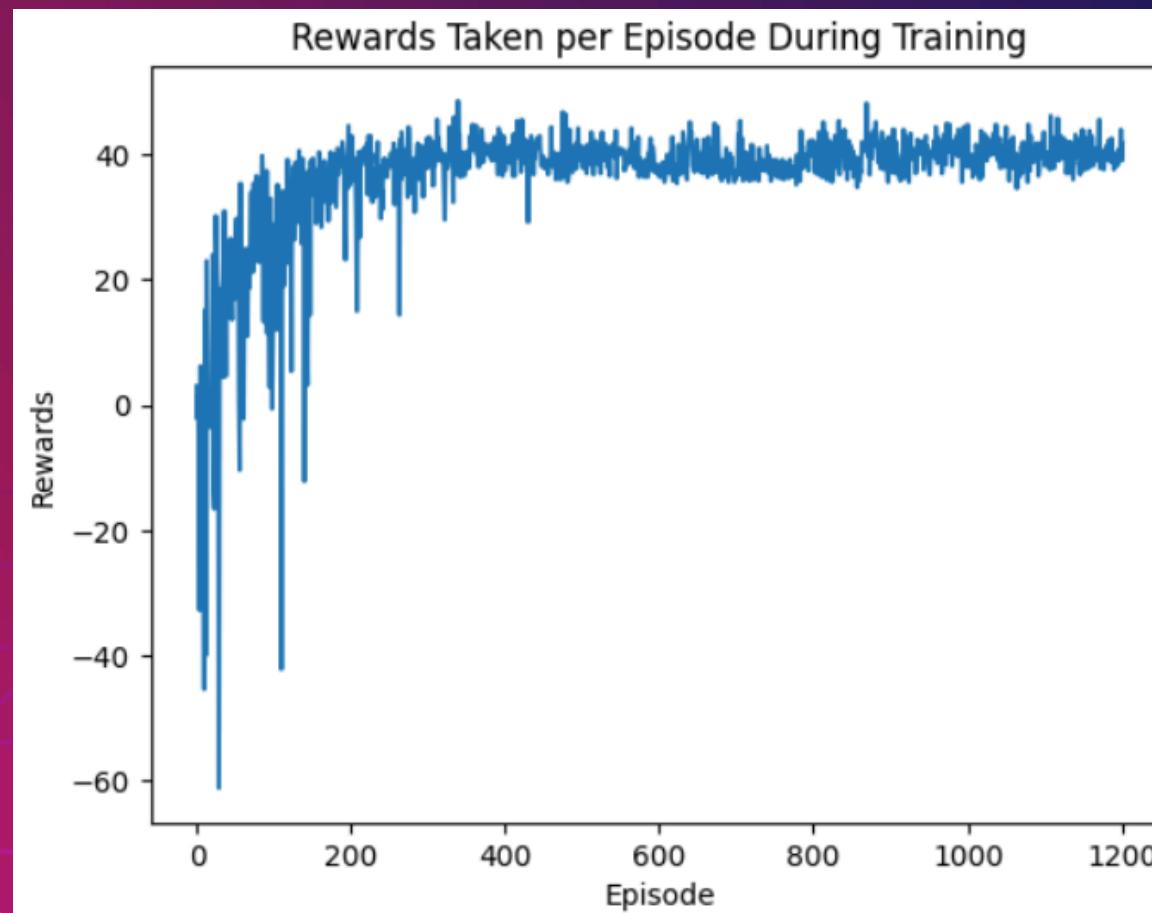
MÉTRICAS DEL ALGORITMO DQN PARA AMBIENTE DE EVALUACIÓN



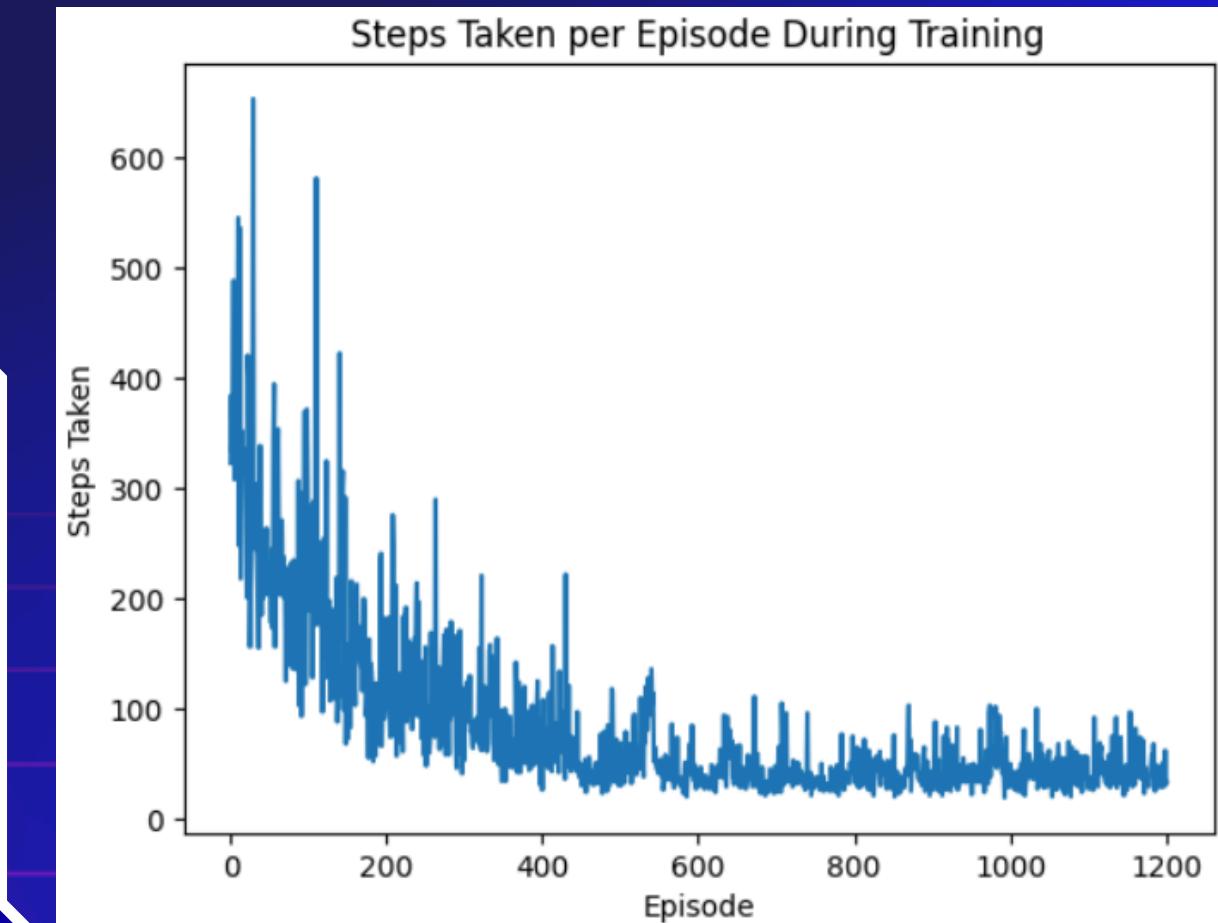
RESULTADOS A2C



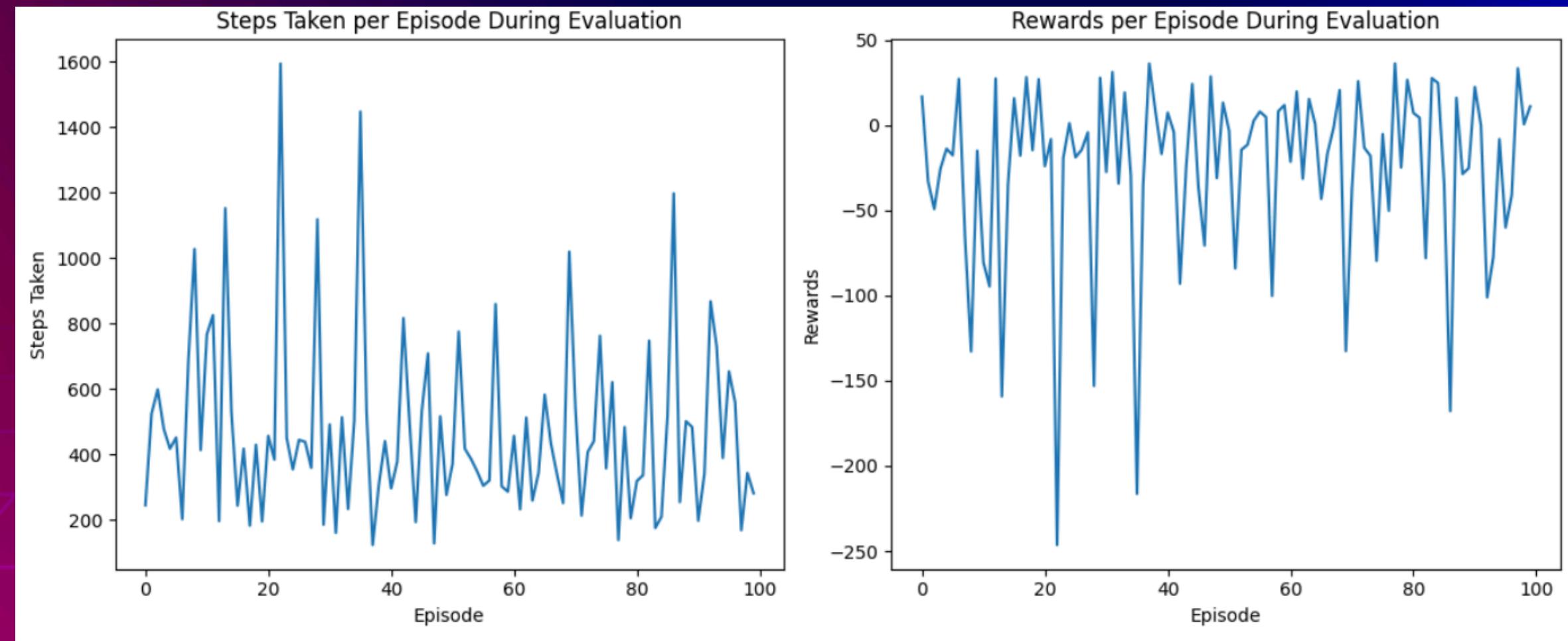
RECOMPENSA OBTENIDA POR CADA EPISODIO PARA ALGORITMO A2C



PASOS TOMADOS POR CADA EPISODIO PARA ALGORITMO A2C



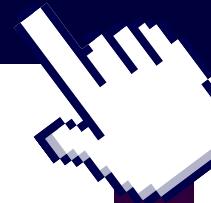
MÉTRICAS DEL ALGORITMO A2C PARA AMBIENTE DE EVALUACIÓN



CONCLUSIONES

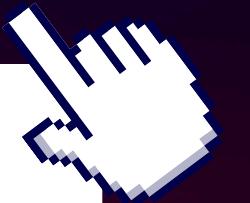
01

El algoritmo de A2C fue el que mejor logró reconocer los patrones para una solución de BattleShip.



02

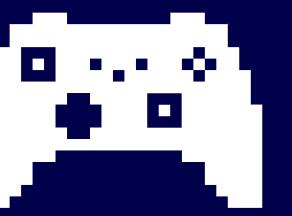
El algoritmo de DQN ofrece una estabilización rápida durante el entrenamiento (la más rápida de todos los modelos) pero puede tender a sobreajustarse fácilmente.



03

Es necesario implementar alguna medida que evite el sobreajuste y que pueda adquirir más información de otras configuraciones de barcos no antes vistas.





**¡MUCHAS
GRACIAS!**