Universidad de Concepción
Facultad de Ingeniería
Departamento de Informática y Ciencias de la Computación

Tarea 1

Aprendizaje por Refuerzo - GridWorld

Integrante: Roberto Felipe Artigues Escobar

Profesor: Julio Godoy

Fecha de entrega: 12 de Noviembre de 2023
Concepción, Chile

Instrucciones de ejecución del código

Todo el código correspondiente a la tarea se encuentra en la carpeta **GridWorldEnvs**. Paso por paso se tiene que hacer lo siguiente:

1. Abrir una terminal en la carpeta **GridWorldEnvs**.
2. Ejecutar el comando "**pip install -e .**" para instalar las dependencias.
3. Ejecutar el comando "**python tarea1.py**" para ejecutar el código.

La razón por la cual se incluye la carpeta completa de **GridWorldEnvs** es porque se modificaron varios archivos de la librería original.

Al final del código hay varias secciones comentadas, las cuales se pueden descomentar para ejecutar los experimentos requeridos para responder cada pregunta.

También se incluye el archivo **plotter.py** para graficar los resultados obtenidos. Para ejecutarlo, se debe hacer lo siguiente:

1. Descargar matplotlib con el comando "**pip install matplotlib**".
2. Ejecutar el comando "**python plotter.py**" para obtener los gráficos.

Los gráficos se crean usando los archivos de rewards en la carpeta **GridWorldEnvs/Rewards**. Se tiene que especificar el archivo a usar en el código del programa.

Pregunta 1

A continuación se muestran los resultados obtenidos con SARSA y Q-learning para el Map1.

Resultados con Map1:

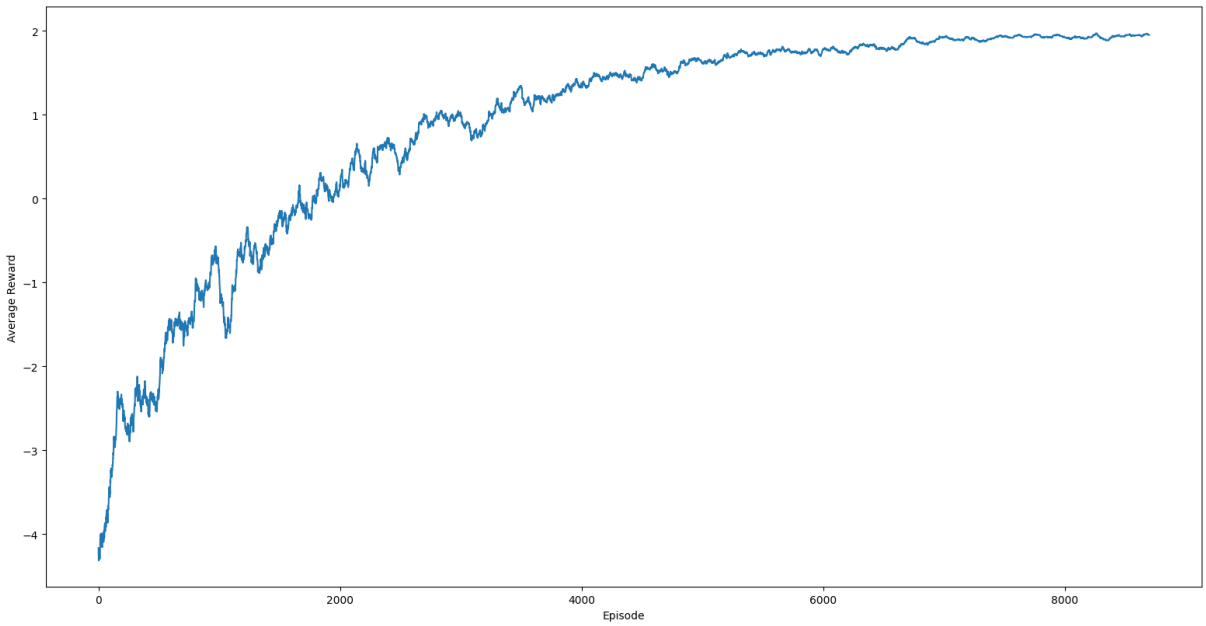

Figura 1: Resultados con SARSA para Map1

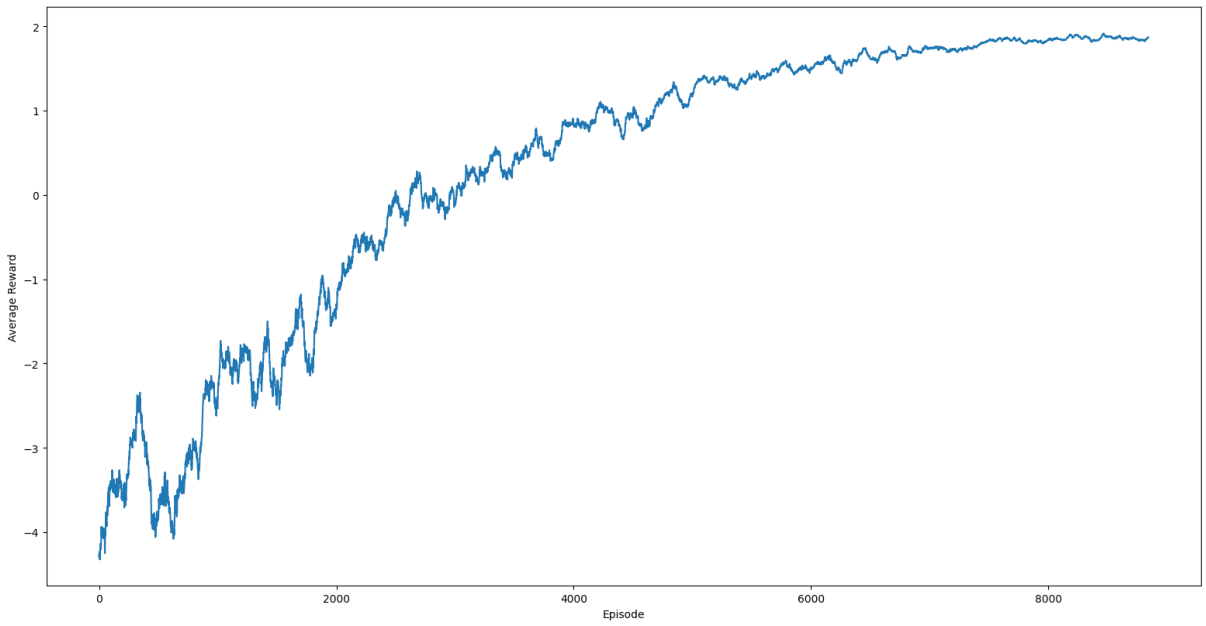

Figura 2: Resultados con Q-learning para Map1

Resultados con Map2:

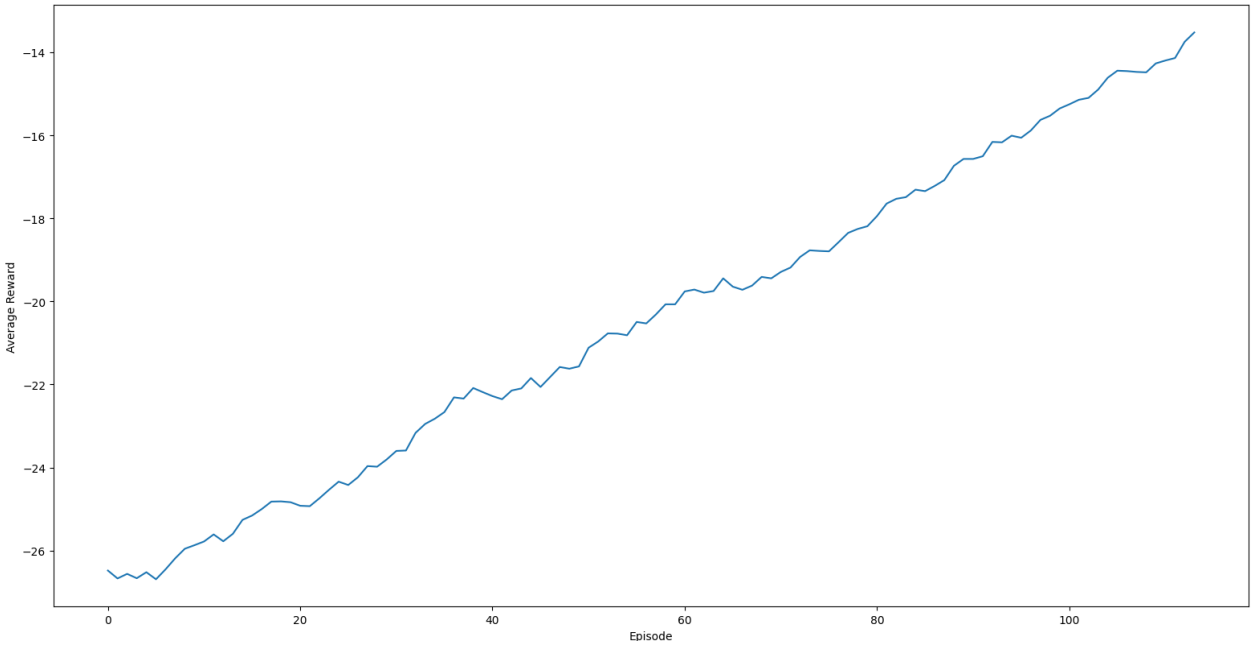

Figura 3: Resultados con SARSA para Map2

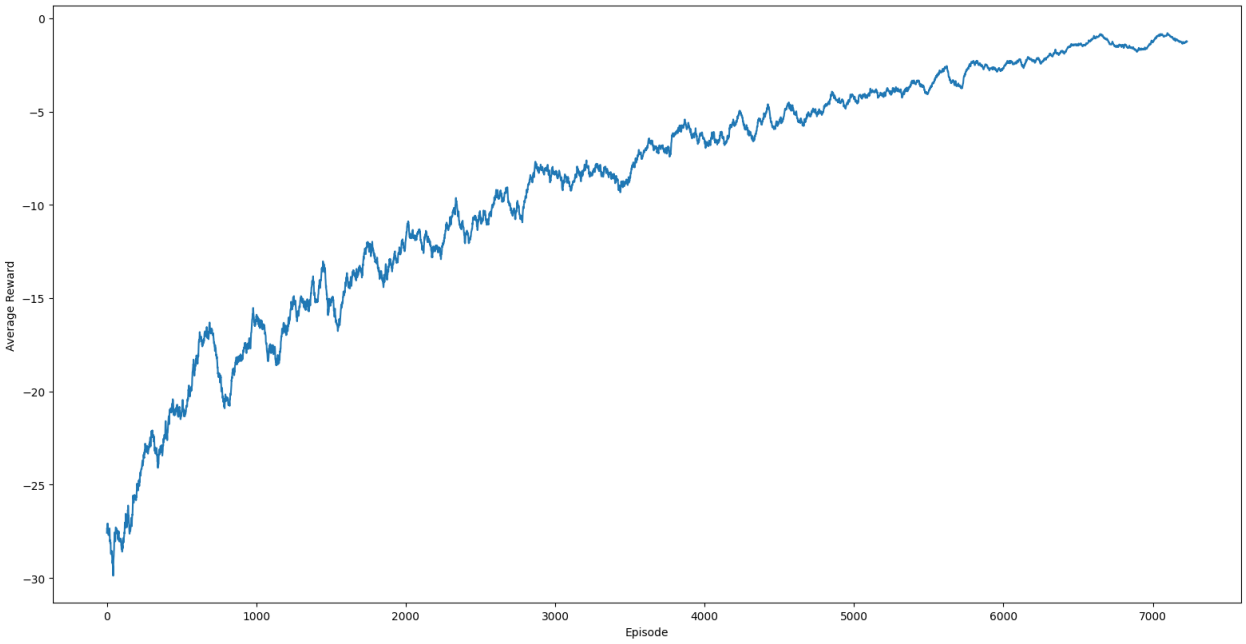

Figura 4: Resultados con Q-learning para Map2

Con los parámetros originales no es posible que el agente aprenda a llegar al objetivo en Map2, ya que el agente se queda dando vueltas en el mapa sin llegar al objetivo.

Resultados con Map2 y parametros modificados:

Los valores que se utilizaron en los parámetros para los resultados de la Figura 5 y la Figura 6 se muestran en la Tabla 1.

Parámetro	Original	Modificado	
MAX_STEPS	100	200	
LEARNING_RATE	0.2	0.1	
GAMMA	0.9	0.95	1
EPSILON	1	1	

Tabla 1: Valores de parámetros utilizados

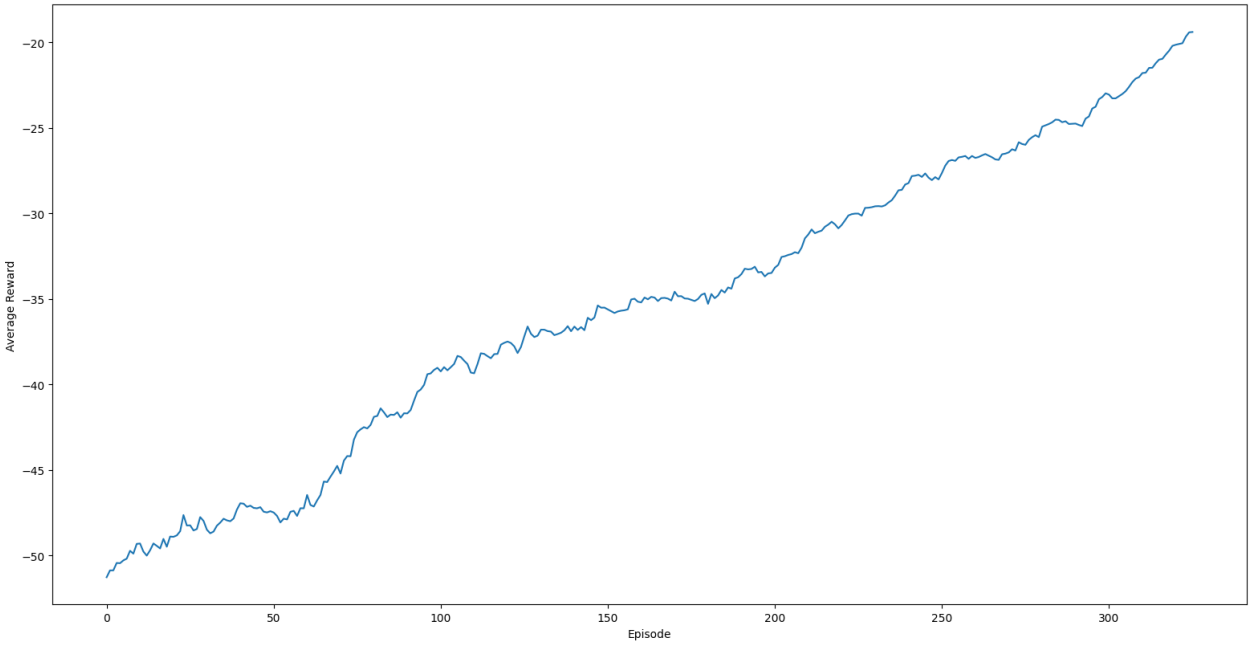

Figura 5: Resultados con SARSA para Map2 con gamma = 0.95

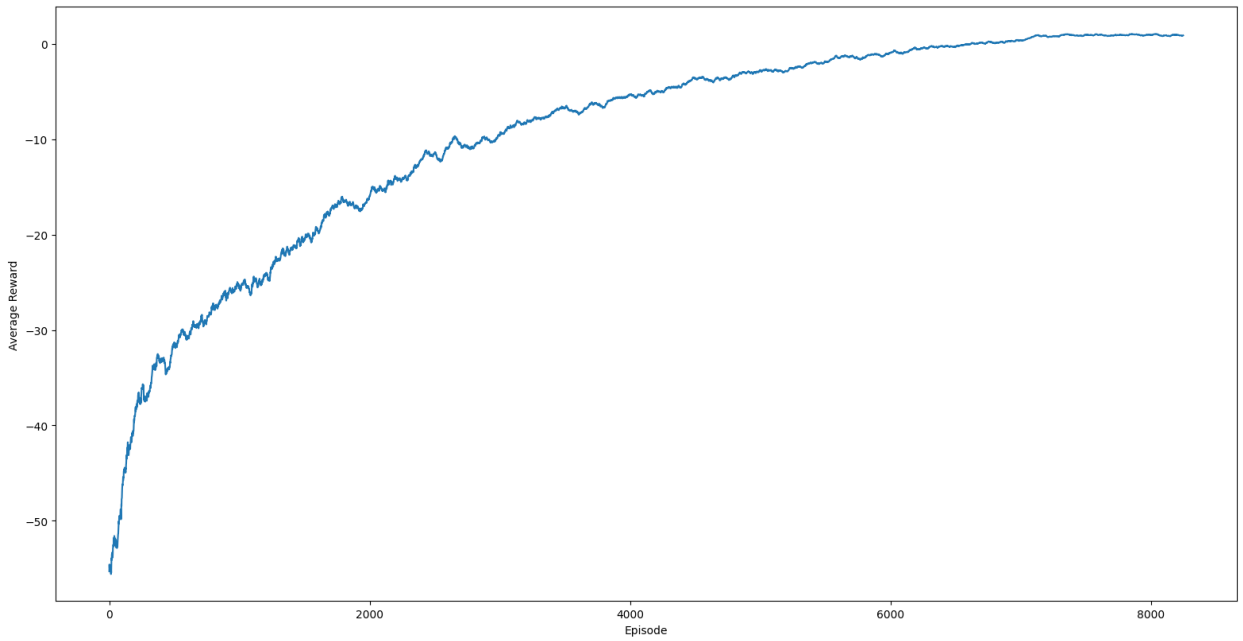

Figura 6: Resultados con Q-learning para Map2 con gamma = 0.95

Al menos SARSA no siempre encontraba la solución óptima. En cambio, Q-learning siempre encontraba la solución óptima usando los parámetros modificados.

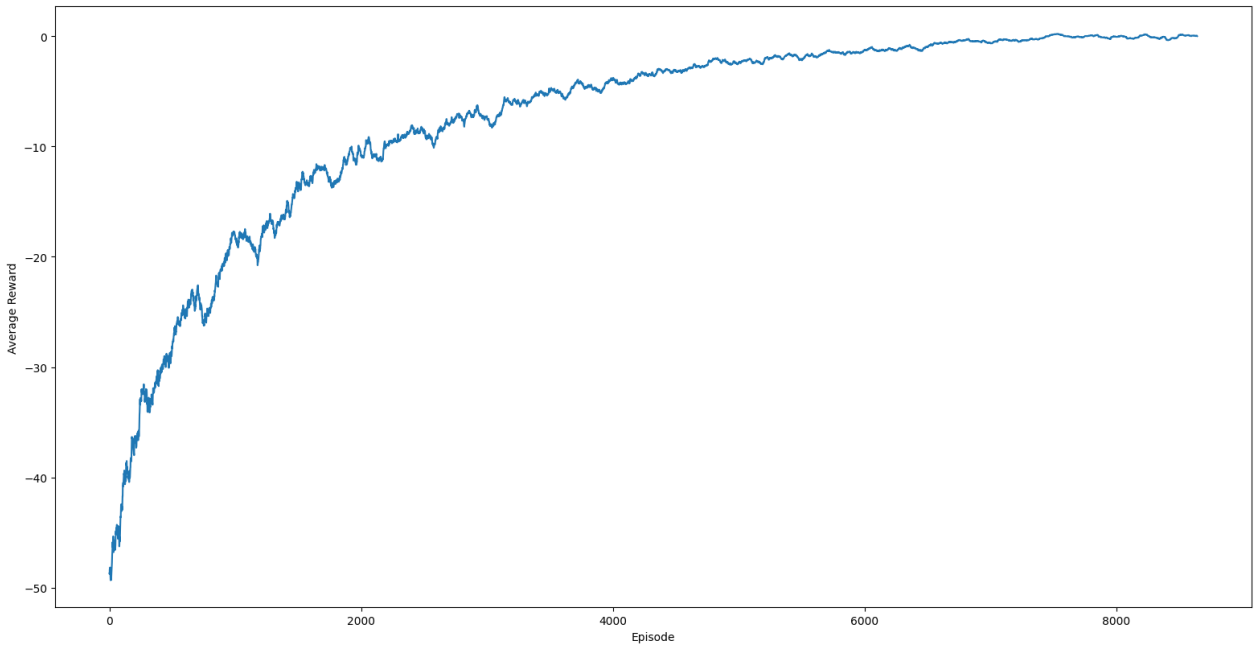

Figura 7: Resultados con SARSA para Map2 con gamma = 1

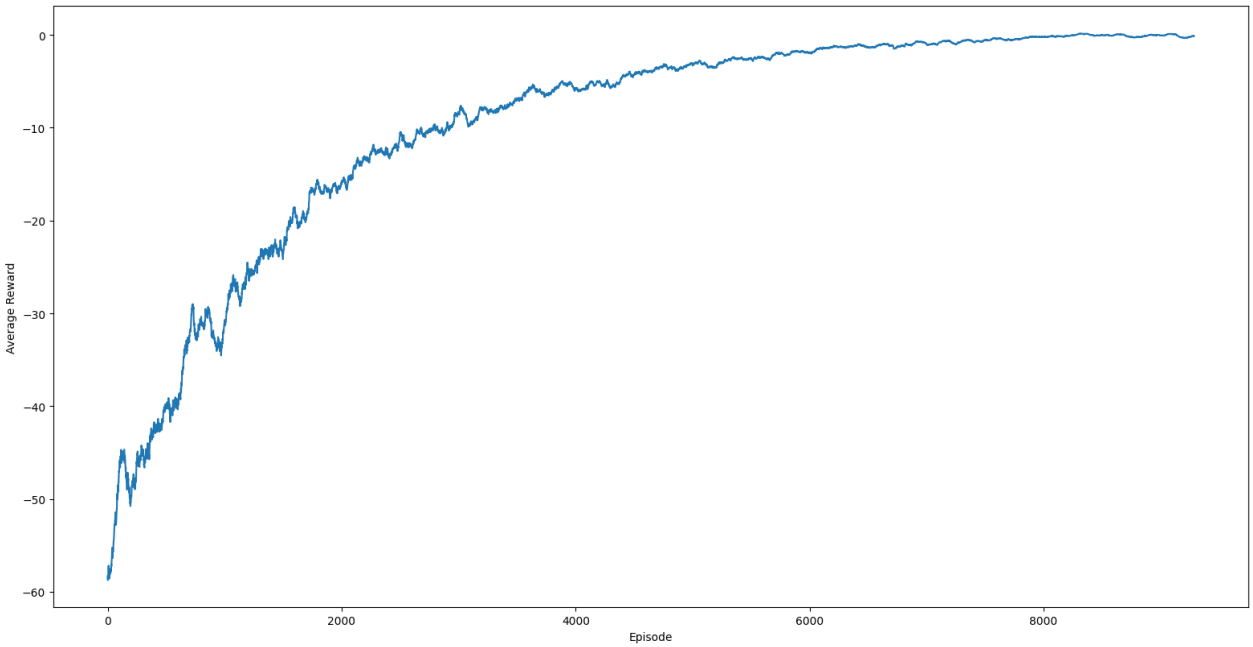

Figura 8: Resultados con Q-learning para Map2 con gamma = 1

La modificación de los parámetros tuvo un impacto significativo en el desempeño de los algoritmos SARSA y Q-learning en el Map2. Con un valor de **gamma** igual a 0.95, se observó que SARSA no siempre encontraba la solución óptima. Por otro lado, Q-learning mostró una mejora notable, encontrando consistentemente la solución óptima.

Al incrementar el valor de **gamma** a 1, se intensifica la importancia de las recompensas futuras. Por lo que ambos algoritmos deciden tomar el camino más corto hacia el objetivo, sin importar las recompensas que se obtengan en el camino.

Pregunta 2

Para el entrenamiento, se volvieron a utilizar los parámetros modificados, específicamente utilizando **gamma** = 1.

Q-learning determinístico vs estocástico

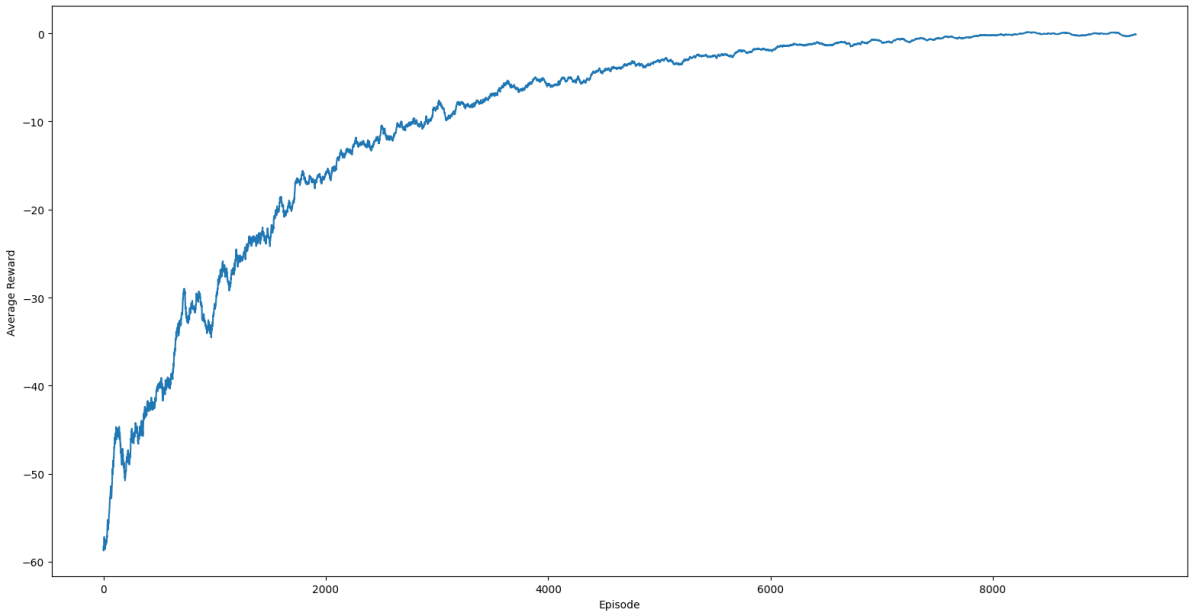

Figura 9: Resultados Q-learning determinístico para Map2 con gamma = 1

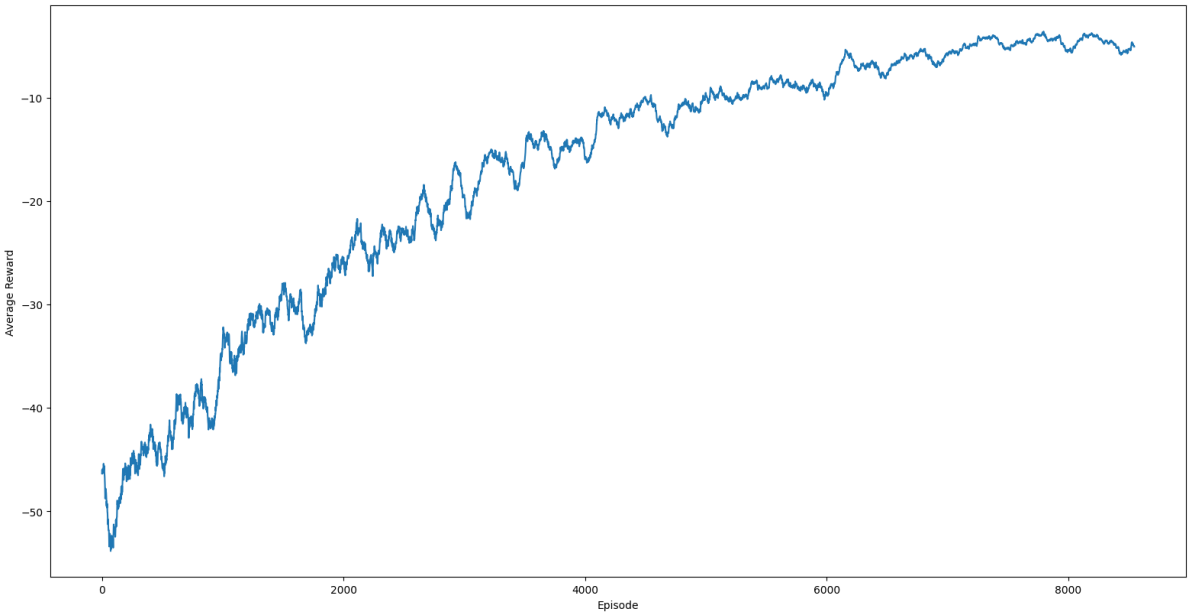

Figura 10: Resultados Q-learning estocástico para Map2 con gamma = 1

SARSA determinístico vs estocástico

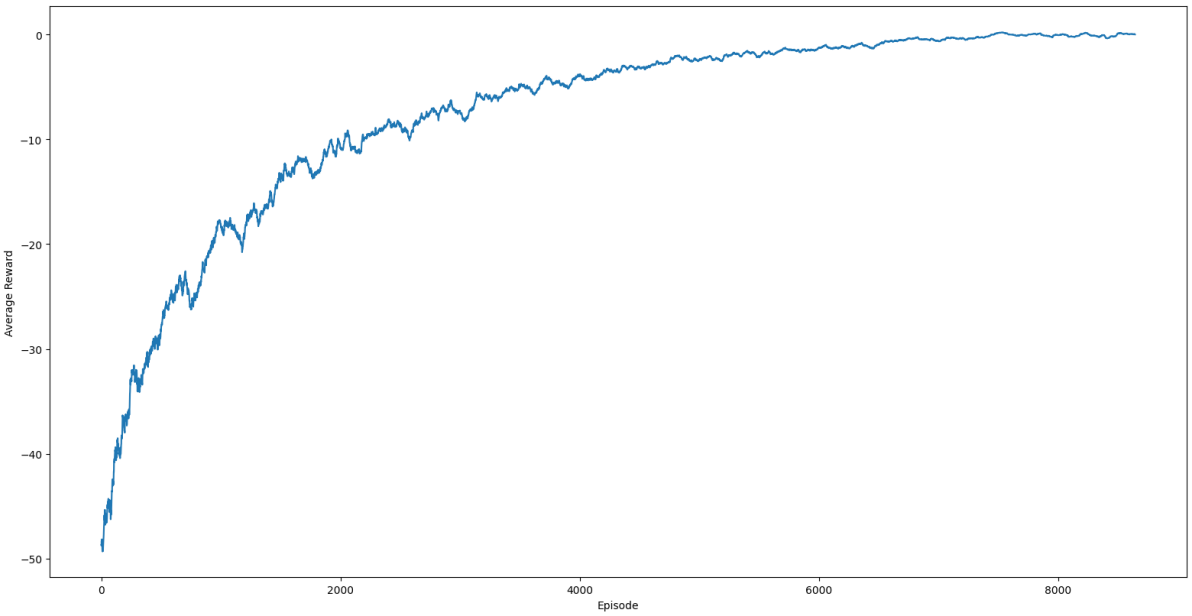

Figura 11: Resultados SARSA determinístico para Map2 con gamma = 1

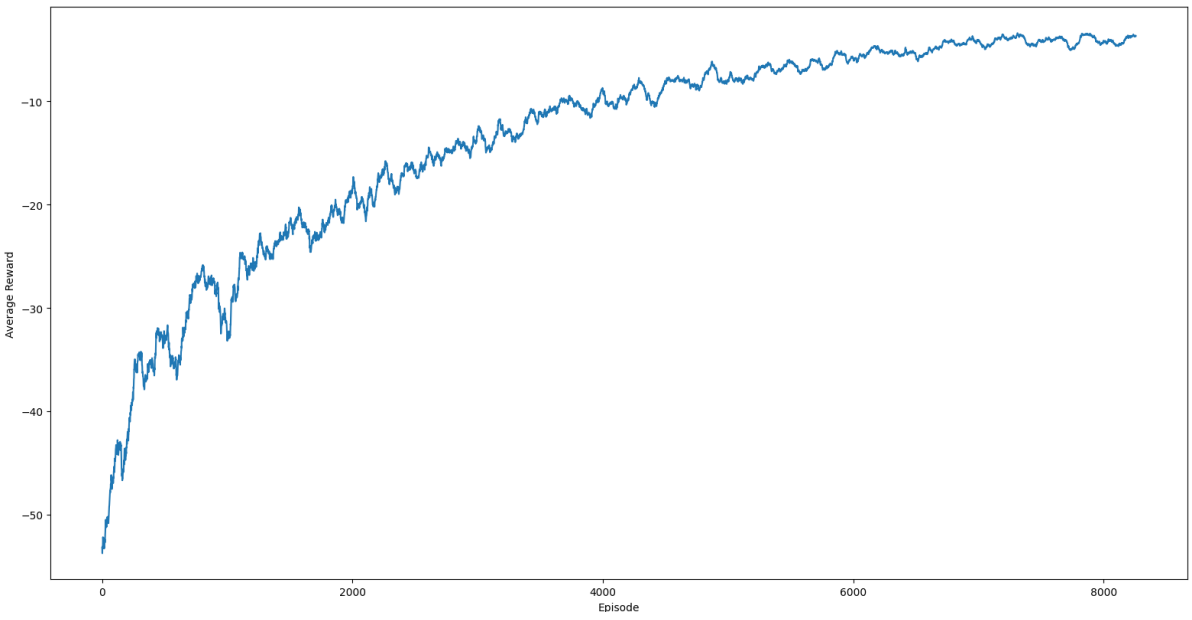

Figura 12: Resultados SARSA estocástico para Map2 con gamma = 1

Ambos algoritmos, SARSA y Q-learning, lograron encontrar soluciones efectivas en una cantidad de episodios similar, aunque la aleatoriedad inherente a los entornos estocásticos impactó claramente sus curvas de aprendizaje. Esta aleatoriedad introduce una variabilidad en las recompensas y trayectorias que los algoritmos deben aprender a navegar.

En particular, Q-learning, que es un algoritmo off-policy, tiende a verse más afectado por esta aleatoriedad. Al aprender una política óptima mientras explora otras opciones, la variabilidad en los resultados de las acciones puede llevar a una adaptación más lenta. Por otro lado, SARSA, siendo un algoritmo on-policy, aprende directamente de la política actual que está siguiendo, lo que le permite adaptarse más rápidamente a los cambios y variaciones del entorno. Esto significa que SARSA puede responder de manera más efectiva a la aleatoriedad en los movimientos, ajustando su política de aprendizaje de forma continua y coherente con la experiencia directa.

Esta diferencia en la adaptabilidad ante la aleatoriedad subraya la importancia de elegir el algoritmo adecuado en función de las características específicas del entorno en el que se va a operar. Mientras que Q-learning busca la solución óptima sin considerar la política actual, SARSA integra la experiencia real en su proceso de aprendizaje, lo que puede ser ventajoso en entornos donde la previsibilidad no está garantizada.

Pregunta 3

Se mostraran los resultados entre Q-learning y Double Q-learning para Map2, usando **gamma** = 1. Primero en un ambiente determinístico y luego en un ambiente estocástico.

Q-learning determinístico vs Double Q-learning determinístico

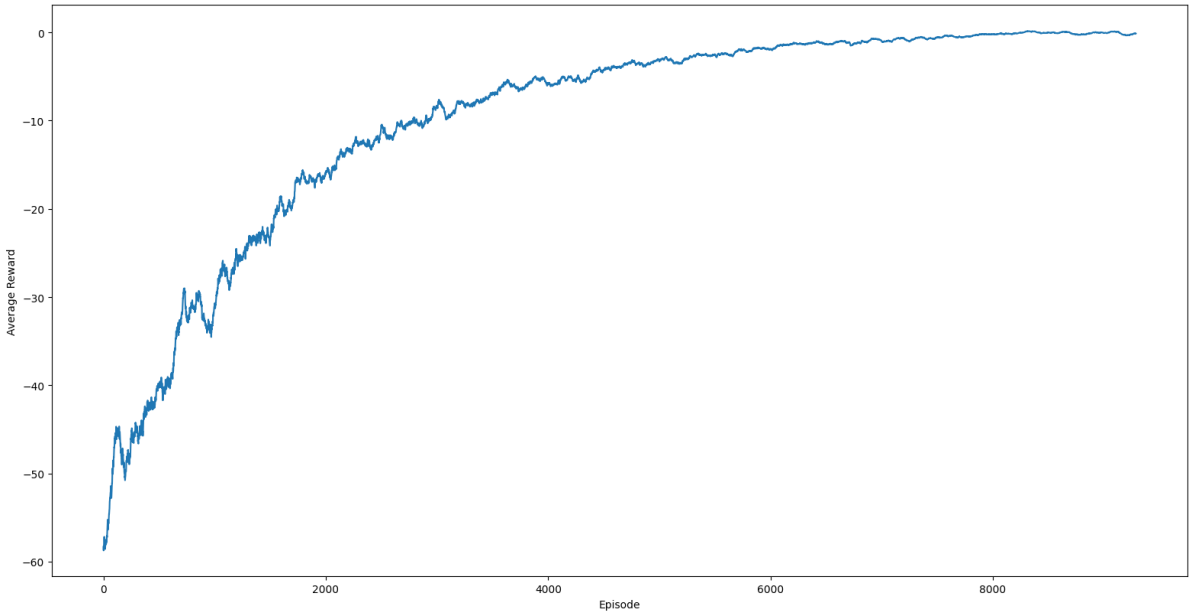

Figura 13: Resultados Q-learning determinístico

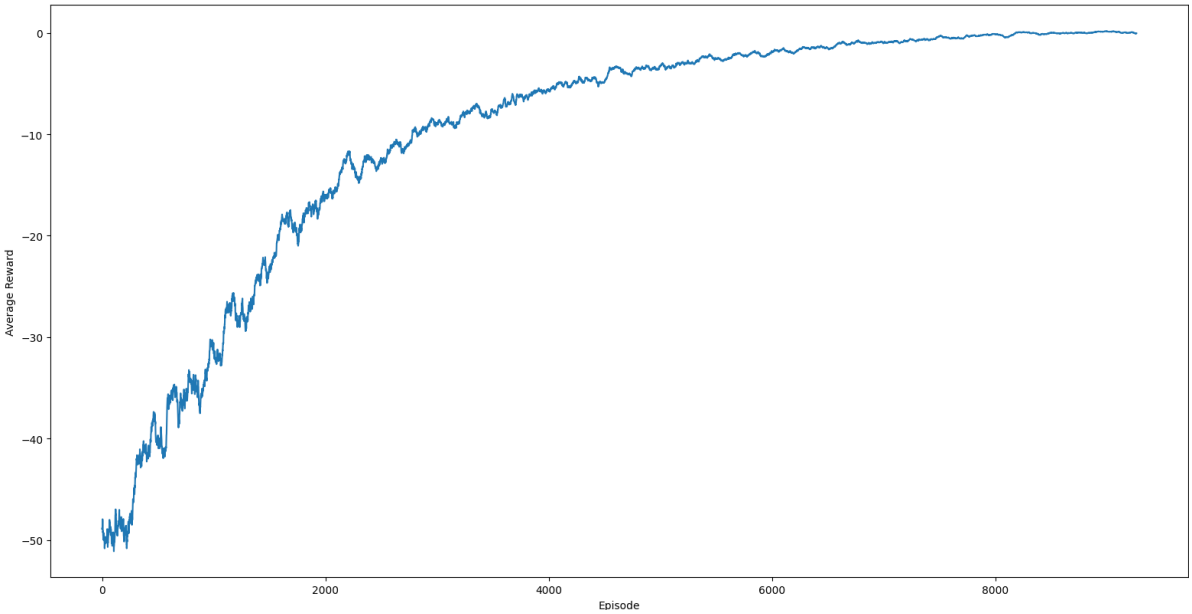

Figura 14: Resultados Double Q-learning determinístico

Q-learning estocástico vs Double Q-learning estocástico

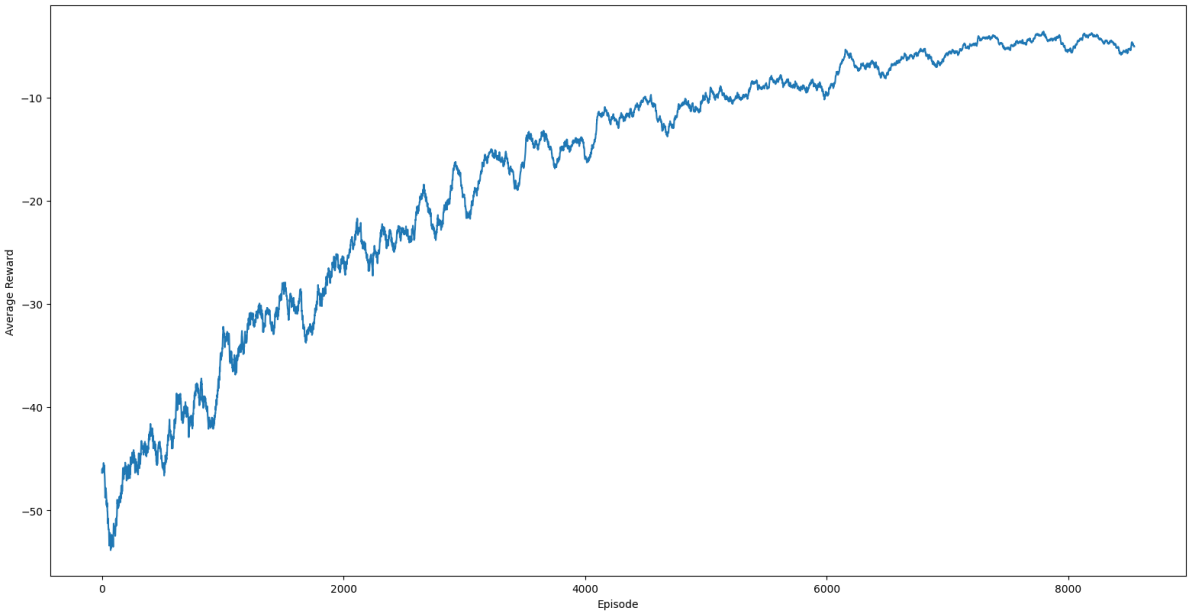

Figura 15: Resultados Q-learning estocástico

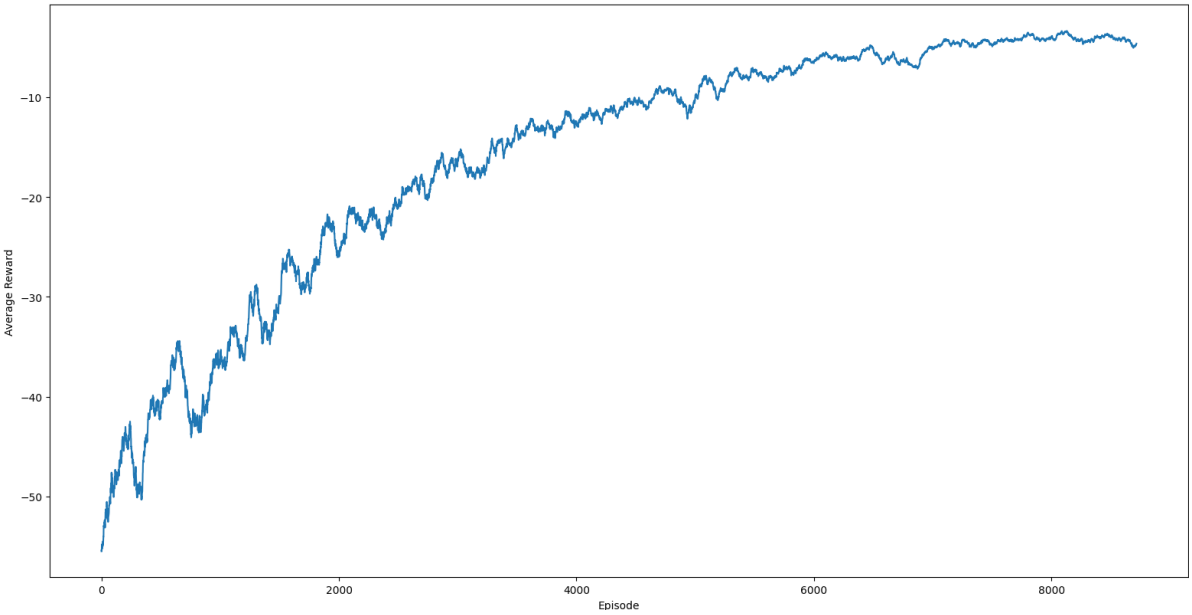

Figura 16: Resultados Double Q-learning estocástico

En un entorno determinístico, Double Q-learning no mostró una mejora significativa en comparación con Q-learning. Esto se debe a que en un entorno determinístico, Q-learning ya puede aprender la política óptima sin sobrestimar los valores de acción.

Por otro lado, en un entorno estocástico, Double Q-learning mostró una mejora significativa en comparación con Q-learning. Esto se debe a que en un entorno estocástico, Q-learning sobrestima los valores de acción, lo que puede llevar a una política subóptima. Double Q-learning puede evitar este problema, ya que utiliza dos funciones de valor para estimar y reducir la sobreestimación.

Pregunta 4

Se comparara el desempeño de SARSA y Q-learning originales a la version con **eligibility traces** para Map2, usando **gamma** = 1 y **lambda** = 0.5. En un ambiente estocástico.

SARSA vs SARSA con eligibility traces

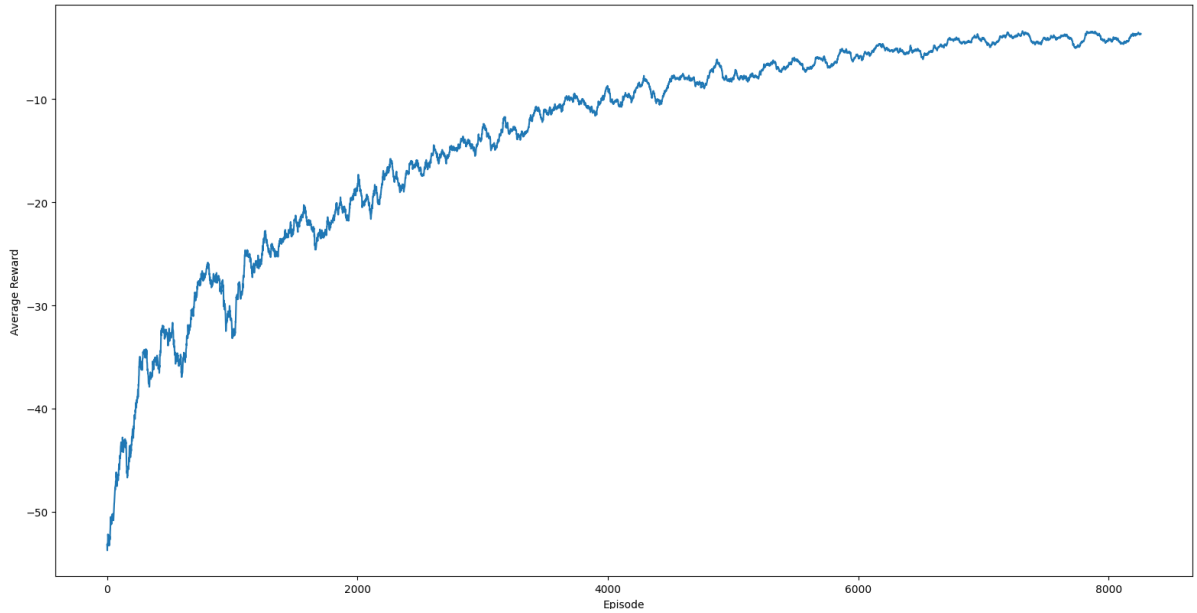

Figura 17: Resultados SARSA estocástico

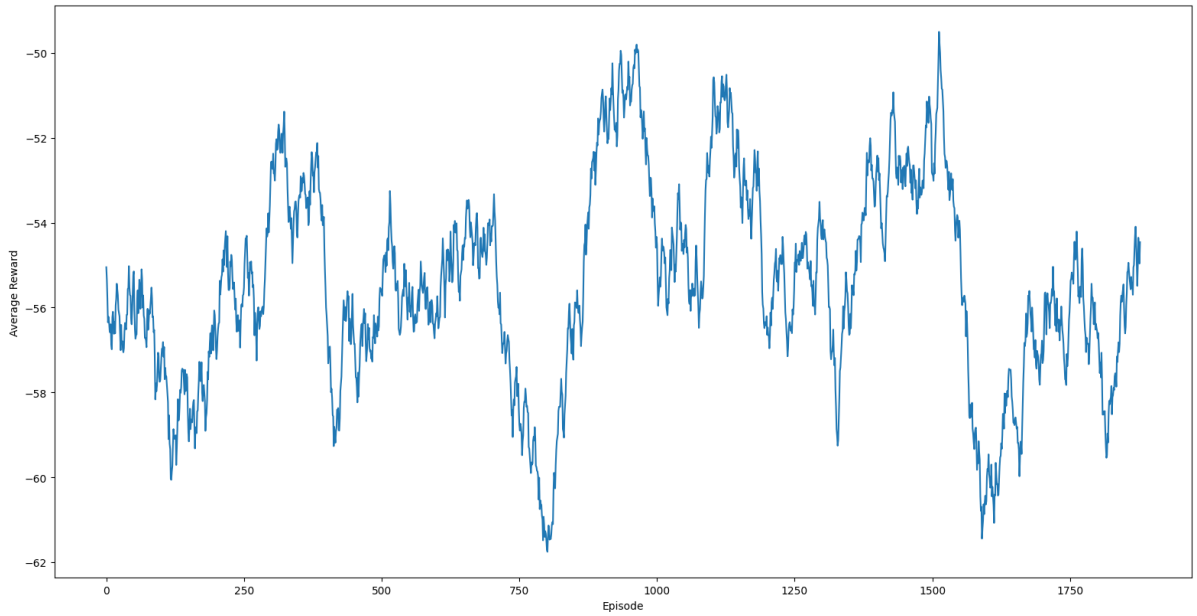

Figura 18: Resultados SARSA con eligibility traces estocástico

Q-learning vs Q-learning con eligibility traces

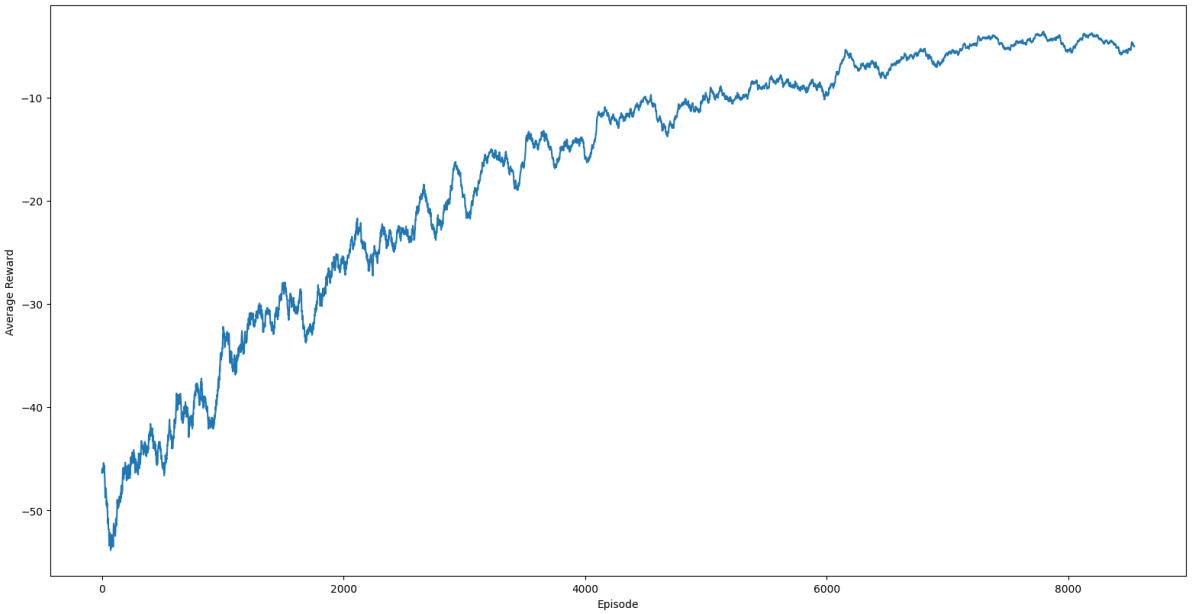

Figura 19: Resultados Q-learning estocástico

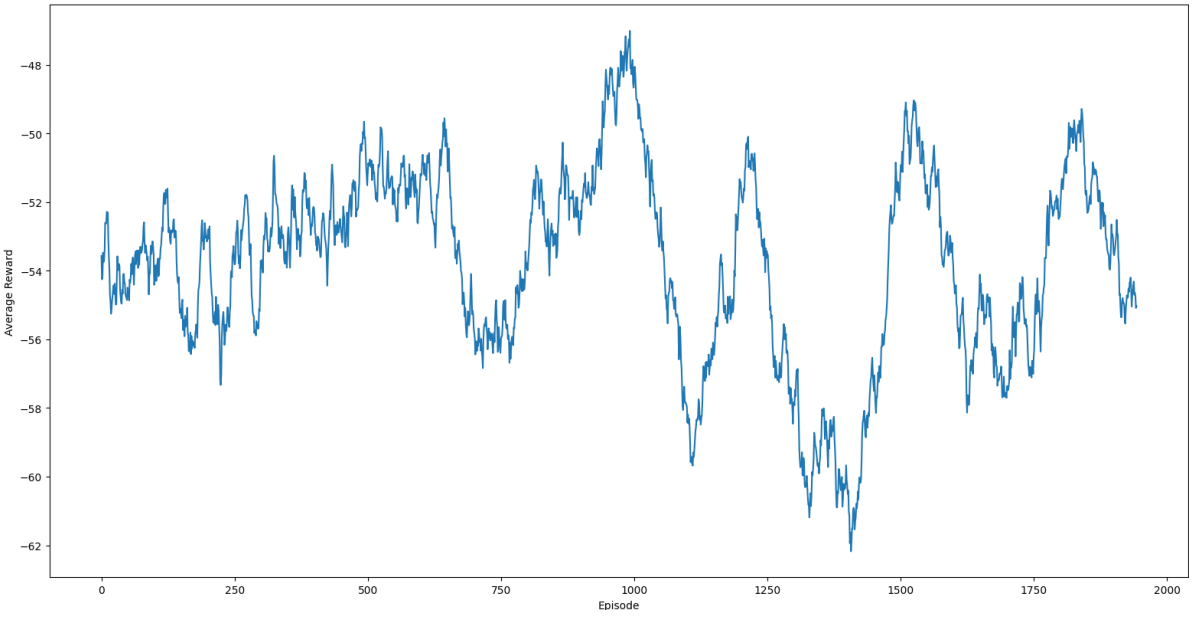

Figura 20: Resultados Q-learning con eligibility traces estocástico

En ambos casos, el rendimiento de los algoritmos SARSA y Q-learning se vio afectado negativamente con la introducción de trazas de elegibilidad. En el caso de SARSA, a pesar del impacto en el rendimiento general, el algoritmo aún logra alcanzar el objetivo de manera relativamente consistente y rápida. Esta capacidad podría atribuirse a su naturaleza on-policy, que le permite ajustar mejor su política a la experiencia directa y reciente, incluso con la complejidad añadida de las trazas de elegibilidad.

Por otro lado, Q-learning, siendo un algoritmo off-policy, parece tener dificultades significativas en este escenario. La combinación de aprendizaje off-policy con trazas de elegibilidad podría estar llevando al algoritmo a una especie de paradoja de decisión, donde el deseo de optimizar recompensas a largo plazo lo aleja del objetivo.