2020/1
André Gomes Heringer
Pedro Nascimento Costa
Introdução a Inteligência Artificial
Ciência da Computação
Universidade Federal de Minas Gerais

Relatório IIA - Trabalho Prático 2 - Aprendizado por Reforço

Introdução

Neste segundo trabalho prático para a matéria de Introdução à Inteligência Artificial demos continuidade aos nossos estudos através de uma implementação do algoritmo de aprendizado por reforço usando o algoritmo Q-Learning apresentado durante as aulas. O cenário do nosso problema é o mesmo do primeiro trabalho prático, envolvendo um agente que precisa navegar por uma fábrica possuindo uma restrição de movimentos, ou seja, o agente precisa recarregar depois de realizar uma certa quantidade de movimentos.

Baseado na especificação do trabalho foram definidas algumas estruturas de dados que nos auxiliaram na modelagem e execução do trabalho. Segundo as instruções na especificação o mundo foi modelado seguindo um MDP seguindo as seguintes características:

1. S' é um espaço de estados em que todos os possíveis estados podem estar, seja umespa¸co vazio, um ponto de localização ou o próprio objetivo.
2. A é o espaço de ações que contém todas as possıveis ações que um agente pode realizar: BAIXO, CIMA, ESQUERDA, DIREITA.
3. r:SxAxS→R, é a função de recompensa. Essa função foi definida uma forma que permanecer no mundo (dar um passo para um ponto vazio) a recompensa é −1. Em contrapartida, dar um passo para um ponto de localização, a recompensa é +1. Contudo, bater em uma parede ou se perder no mundo a recompensa é −10. Chegarao objetivo a recompensa é +10. Formalmente, r(s,a,s′) é a recompensa associada a uma transição de estado dada uma determinada ação. A função recompensa pode ser alterada pensando em um parâmetro de aprendizado.

4. λ define o fator de desconto.

5. T:SxA→∆(S) por fim é função de transição. No contexto deste trabalho, iremos considerar a transição determinıstica. Em outras palavras, o agente irá realizar a ação desejada sem uma distribuição de probabilidade associada aquela. Formalmente,T(s′|s,a) = 1 que significa que a probabilidade de ir para um estados′dado um estadoatual se uma ação é igual a 1.

Detalhes de Implementação

Dadas estas características foram definidas duas estruturas de dados auxiliares, uma delas foi uma matriz com os valores de todas as recompensas presentes no mapa de fábrica, a outra é um dicionário que guarda os valores de aprendizado de todos os estados visitados pelo agente. O dicionário de estados e construído conforme o agente navega pela fábrica.

Uma decisão de projeto importante foi colocar como terminais estados fora do mapa da fábrica e estados definidos como parede, dessa forma a qualquer momento que o agente sai do mapa ou bate em uma parede ele encerra um episódio e recebe uma recompensa de -10.

Para executar o programa, o comando deve ser da seguinte forma:
./qlearning.sh main.py ENTRY_FILE ALPHA EPSILON LAMBDA N

Sendo que:

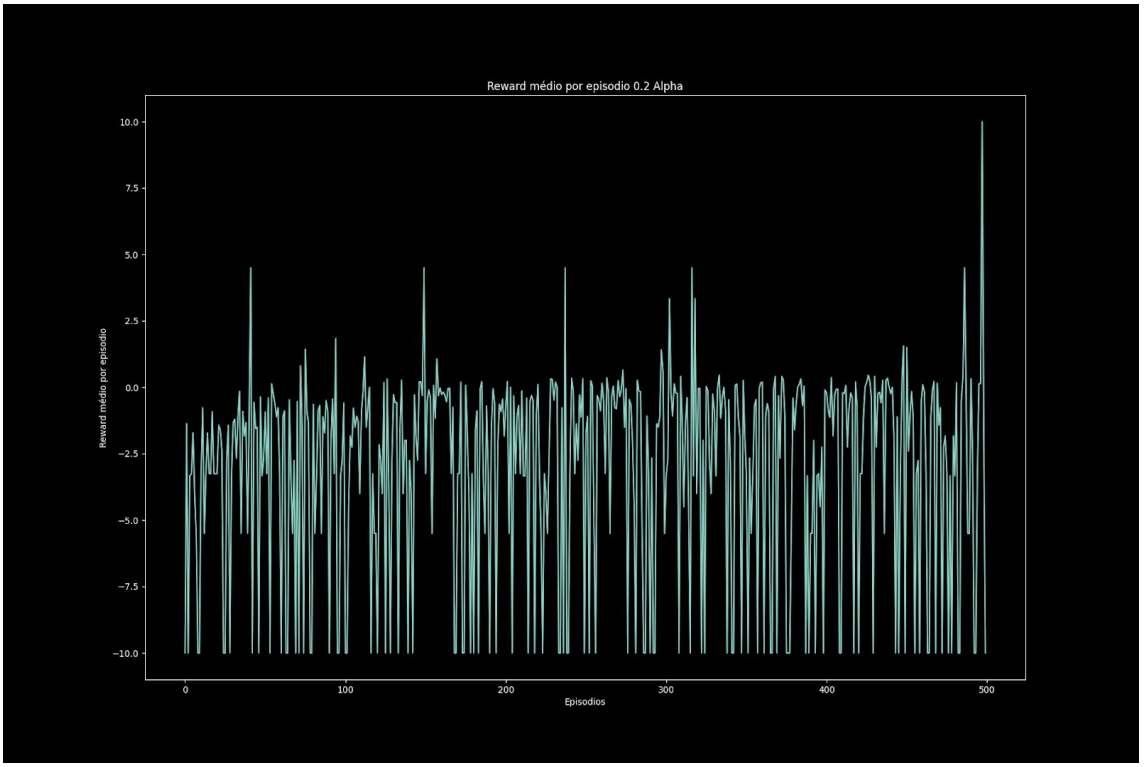
- **ENTRY_FILE :** Caminho para arquivo de entrada que contém o mapa
- **ALPHA :** Fator de aprendizado entre 0 e 1
- **EPSILON :** Fator da política Epsilon greedy, varia de 0 a 1
- **LAMBDA :** Fator de desconto, varia de 0 a 1
- **N :** Número de episódios a serem rodados

A saída será salva no arquivo “pi.txt”

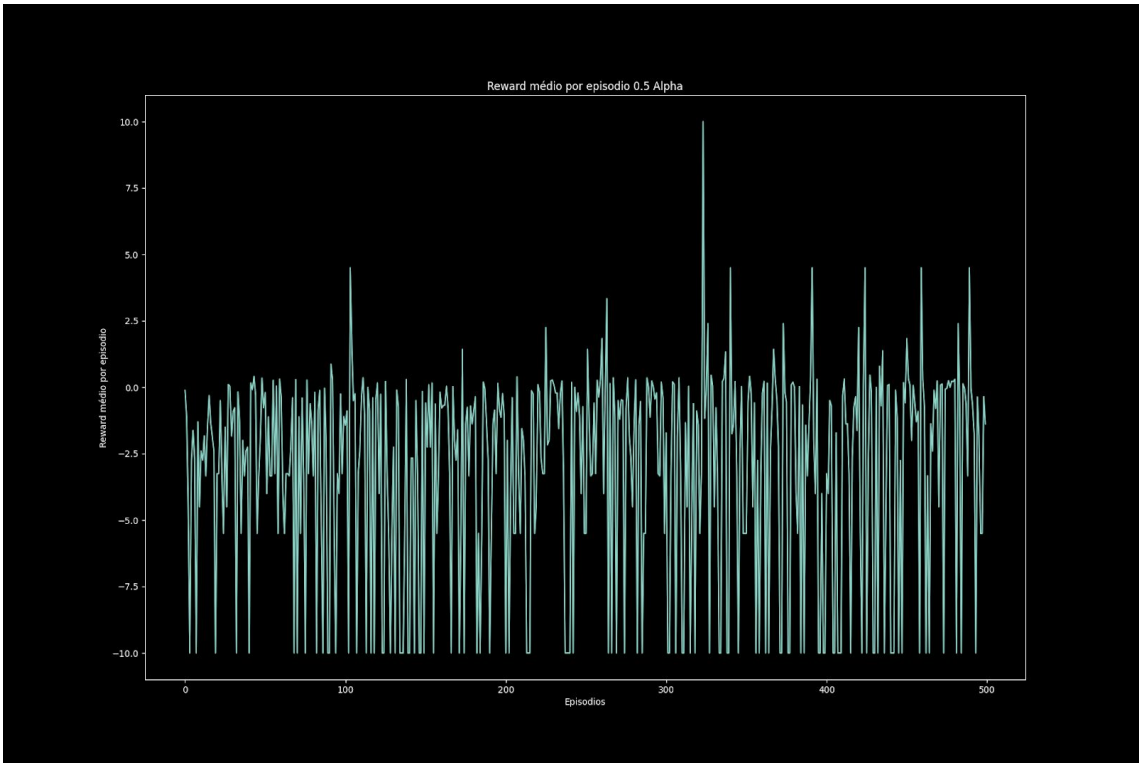
Demonstração de Resultados

Nos gŕaficos a seguir, temos uma análise da variação de Alpha lamba e epsilon, variando um deles e mantendo os demais em 0.5.

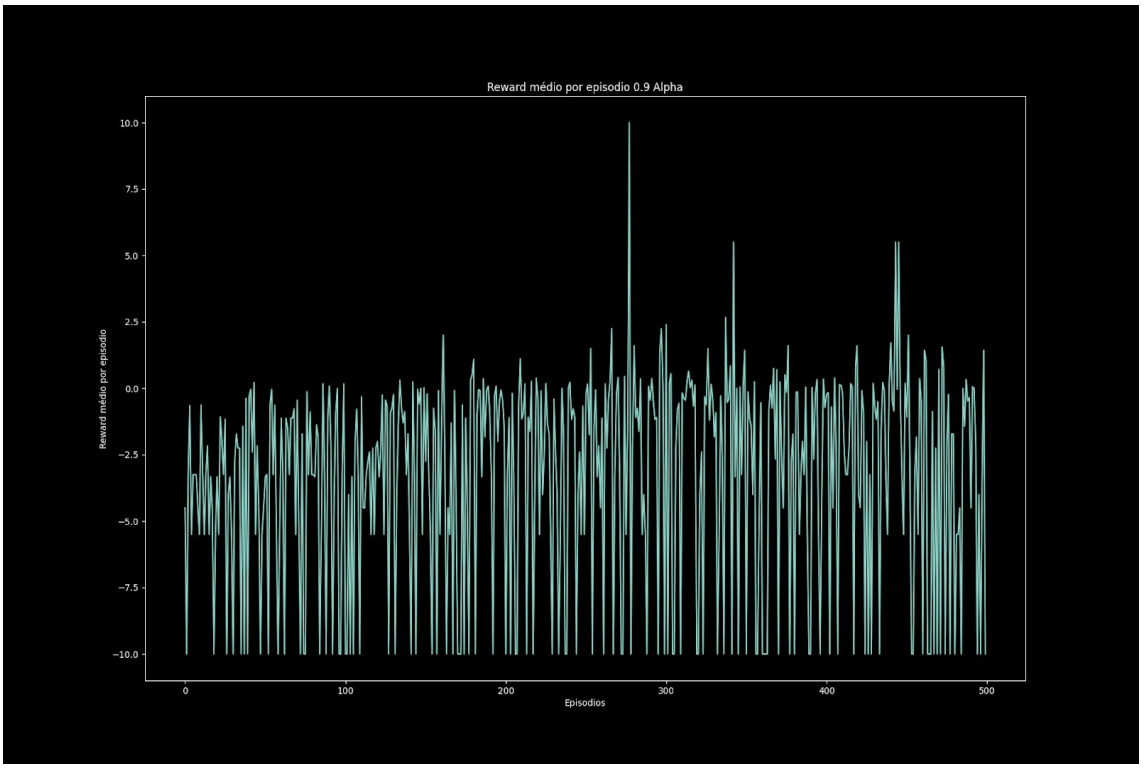
- **Alpha 0.2**



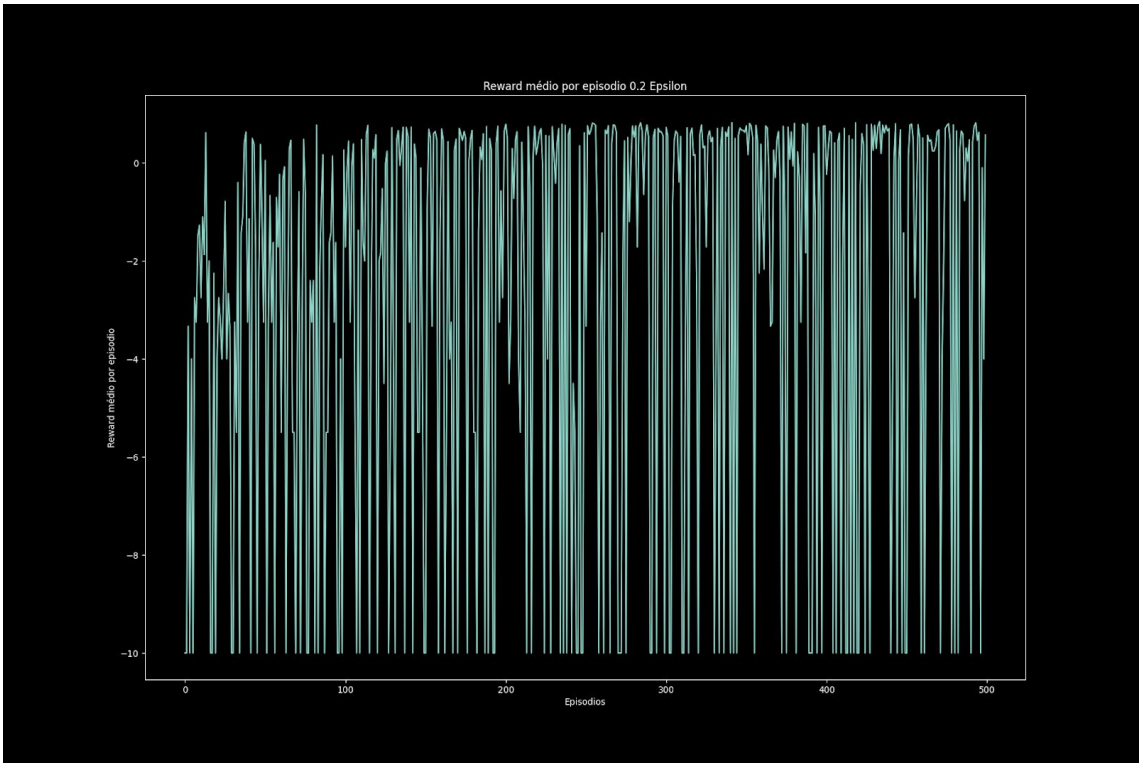
- **Alpha, Epsilon e Lambda 0.5**



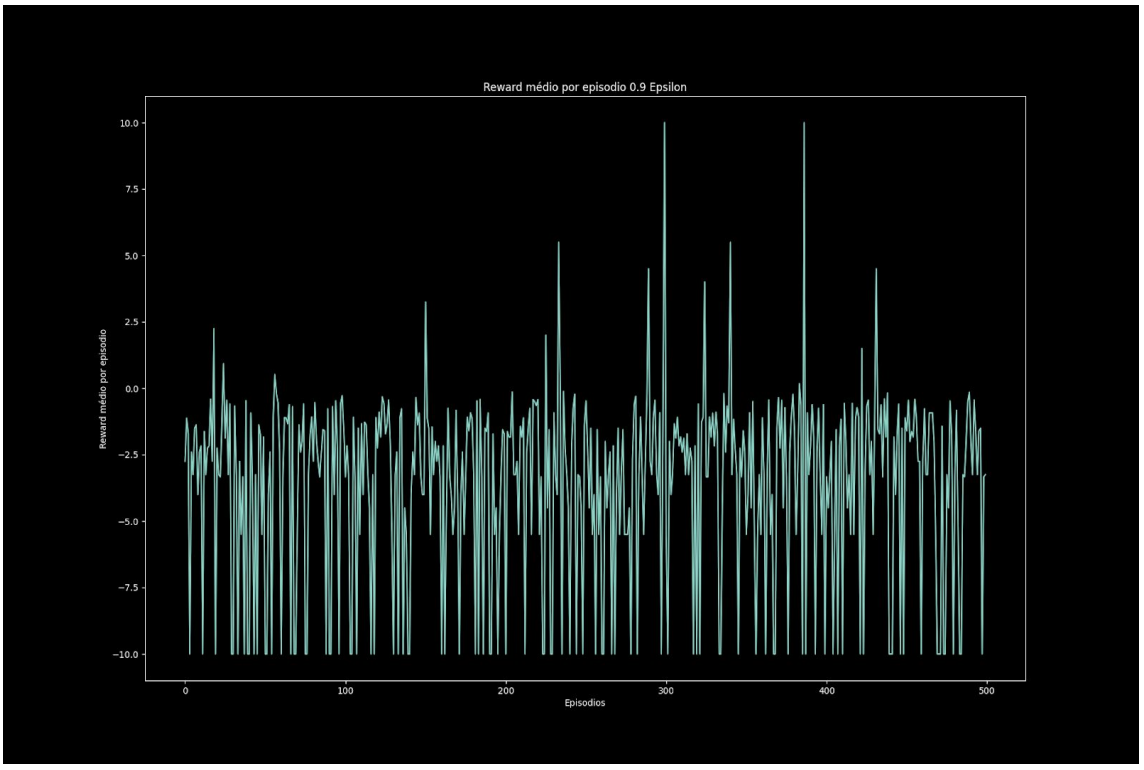
- **Alpha 0.9**



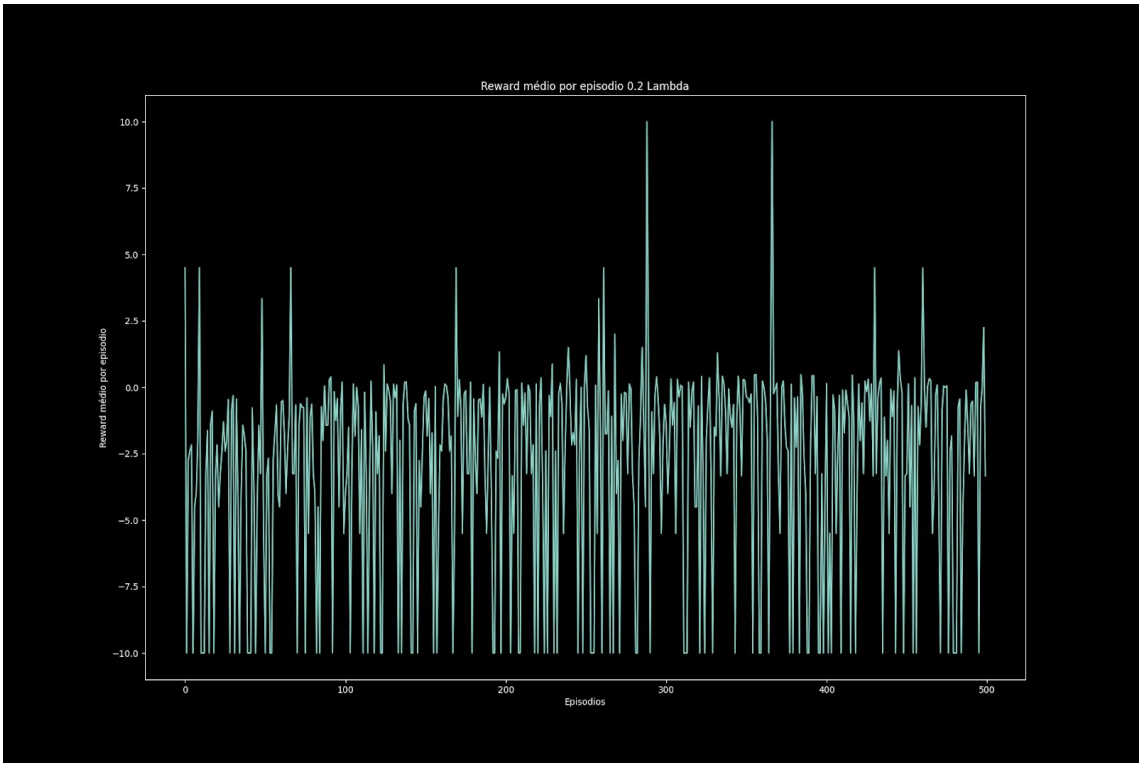
- **Epsilon 0.2**



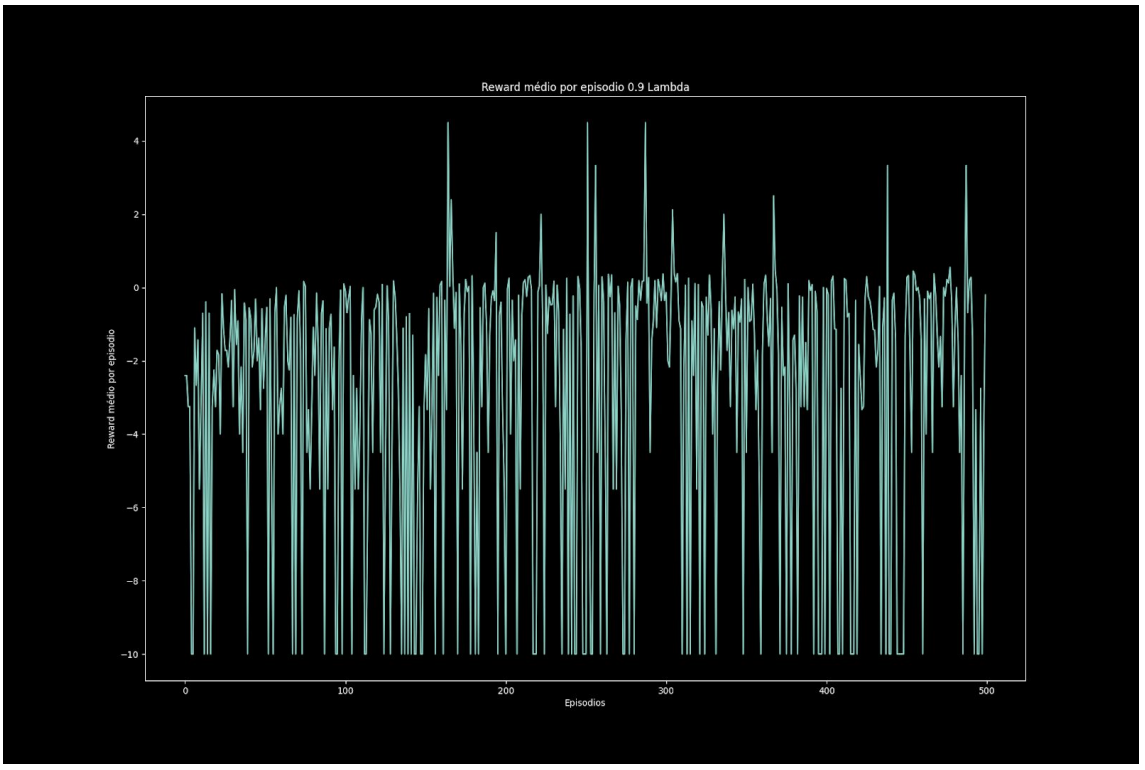
- **Epsilon 0.9**



- **Lambda 0.2**

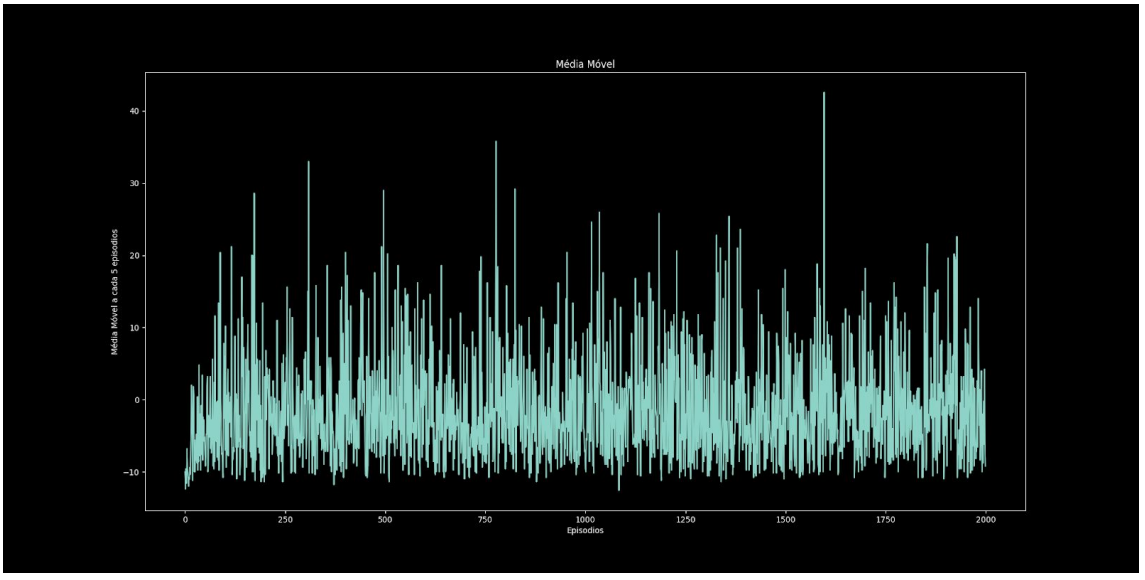


- **Lambda 0.9**

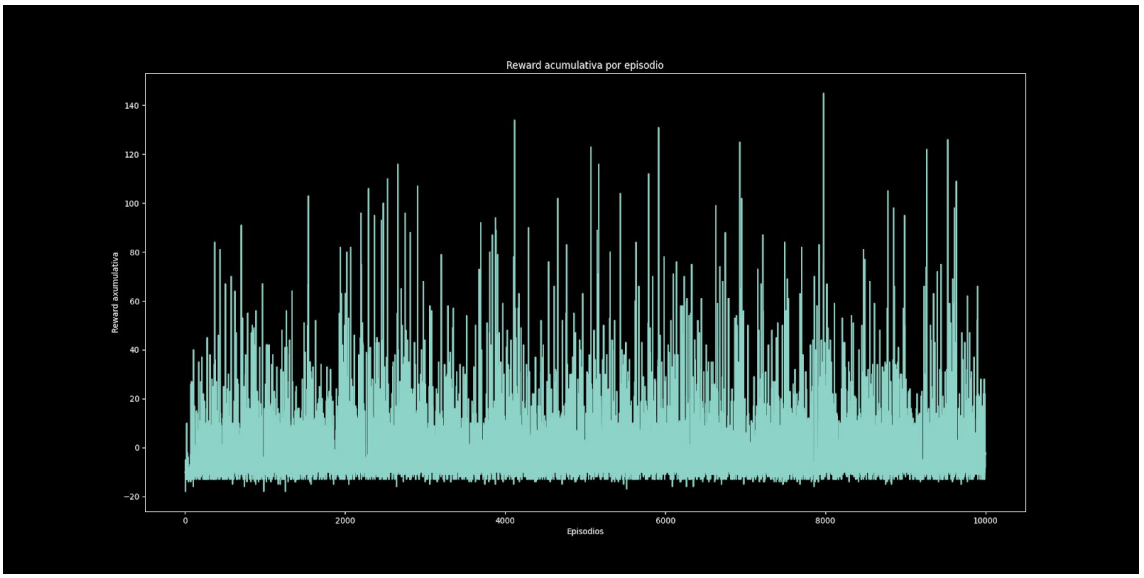


Nos gráficos a seguir apresentamos uma média móvel da recompensa a cada 5 episódios e o valor acumulativo por episódio da recompensa.

Média Móvel



Reward Acumulado



Conclusão

Após a extração dos resultados encontrados, vimos que o algoritmo depende bastante da política para tomar sua decisão, notavelmente com valores baixos de Epsilon vimos uma qualidade bem pior nas recompensas.

Olhando para a recompensa acumulada por episódio, vimos que os resultados tendem a ficar na faixa de -10 a 20, com alguns outliers de grande recompensas, que chegam a ultrapassar 100. Acreditamos que isso ocorra devido a natureza da política de escolha do Epsilon Greedy, que em alguns casos acaba sendo guloso e fica andando um tempo considerável entre dois caracteres “#”, devido a sua recompensa de +1.

Bibliografia:

https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/

https://www.geeksforgeeks.org/what-is-reinforcement-learning/

https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56

https://towardsdatascience.com/a-beginners-guide-to-q-learning-c3e2a30a653c

https://medium.com/analytics-vidhya/the-epsilon-greedy-algorithm-for-reinforcement-learning-5fe6f96dc870

https://www.geeksforgeeks.org/epsilon-greedy-algorithm-in-reinforcement-learning/