Henrique de Andrade Assme

Market Making with Deep Reinforcement Learning Based on Signals and Order Book Tick Data from Brazilian Market

São Paulo, SP

2025

Henrique de Andrade Assme

Market Making with Deep Reinforcement Learning Based on Signals and Order Book Tick Data from Brazilian Market

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Orientador: Prof. Dr. Fabio Gagliardi Cozman

Coorientador: Me. Renan de Luca Avila

São Paulo, SP

2025

Catalogação-na-publicação

Assme, Henrique
Market Making with Deep Reinforcement Learning Based on Signals and Order Book Tick Data from Brazilian Market / H. Assme -- São Paulo, 2025.
10 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Deep Reinforcement Learning 2.Market Making I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

Dedico esse trabalho, primeiramente à minha família. Meus pais, Rodrigo e Lisandra, e meu irmão, Felipe, e por meio deles, aos meus avós, por todo o alicerce que me permitiu chegar até esse momento. Dedico também à minha namorada, Bruna, por todo o apoio e companheirismo ao longo dos anos de faculdade. Por fim, dedico aos meus amigos e meus irmãos da Ordem DeMolay e a cada um que, de alguma forma, contribuiu em minha vida nesses últimos cinco anos.

Agradecimentos

Os agradecimentos principais são direcionados ao meu orientador, Prof. Dr. Fabio Gagliardi Cozman, ao meu coorientador, Me. Renan de Luca Ávila, e ao BTG Pactual, pelo apoio, pela orientação e pela disponibilidade das ferramentas necessárias que possibilitaram o desenvolvimento deste projeto.

Resumo

O presente trabalho de conclusão de curso apresenta o desenvolvimento e avaliação de um agente de *market making* baseado em *Deep Reinforcement Learning* aplicado ao mercado brasileiro. O modelo foi implementado em um ambiente de simulação que reproduz a dinâmica do melhor *bid* e *ask* a partir de dados *tick-by-tick* reais da B3, reconstruídos através do BTG Pactual, permitindo a execução de ordens passivas e o cálculo contínuo do *mark-to-market*, do inventário e do *Profit and Loss*. O agente foi treinado com o algoritmo *Proximal Policy Optimization*, em arquitetura *actor-critic* rasa customizada, e recompensado por uma função híbrida centrada em lucro e penalização de risco.

Os resultados dos experimentos mostram que o agente aprendeu a manter comportamento estável e não especulativo, equilibrando ações de *quote bid, quote ask e quote both*, com PnL médio positivo e baixo risco de inventário em ativos de alta e média liquidez. Em cenários de menor liquidez, observou-se um comportamento menos estável, refletindo a sensibilidade do modelo à ativos líquidos e a condições de mercado mais complexas. As métricas agregadas por ativo e categoria de liquidez indicam consistência entre os dias de teste e demonstram a viabilidade da aplicação de DRL ao MM com dados brasileiros, abrindo caminho para futuras extensões com a inclusão de sinais externos e espaço de ação contínuo.

Keywords: *deep reinforcement learning*, *market making*, PPO; B3

Abstract

This final course project presents the development and evaluation of a market-making agent based on Deep Reinforcement Learning applied to the Brazilian market. The model was implemented in a simulation environment that reproduces the dynamics of the best bid and ask using real tick-by-tick data from B3, reconstructed through BTG Pactual, allowing the execution of passive orders and the continuous calculation of mark-to-market, inventory, and profit and loss. The agent was trained with the Proximal Policy Optimization algorithm, in a customized shallow actor-critic architecture, and rewarded by a hybrid function centered on profit and risk penalty.

The results of the experiments show that the agent learned to maintain stable and non-speculative behavior, balancing quote bid, quote ask, and quote both actions, with a positive average PnL and low inventory risk in high and medium liquidity assets. In scenarios of lower liquidity, less stable behavior was observed, reflecting the model's sensitivity to liquid assets and more complex market conditions. The aggregated metrics by asset and liquidity category indicate consistency between the test days and demonstrate the feasibility of applying DRL to the MM with Brazilian data, paving the way for future extensions with the inclusion of external signals and continuous action space.

Palavras-chave: *deep reinforcement learning*, *market making*, PPO; B3

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

DRL	*Deep Reinforcement Learning*
MM	*Market Making*
API	*Application Programming Interface*
RL	*Reinforcement Learning*
LOB	*Limit Order Book*
Attn	*Attention*
PnL	*Profit and Loss*
B3	Brasil, Bolsa, Balcão
BTG	*Banking and Trading Group*
PPO	*Proximal Policy Optimization*
MDP	*Markov Decision Process*
TTL	*Time to Live*
FIFO	*First In First Out*
MTM	*Mark to Market*
qty	*Quantity*
CPU	*Central Processing Unit*
GPU	*Graphics Processing Unit*
MLP	*Multi Layer Perceptron*
ReLU	*Rectfied Linear Unit*
SB3	*Stable Baselins 3*
UTC	*Coordinated Universal Time*
MAP	*Medium Absolute Portfolio*
MDD	*Maximum Drawdown*

Lista de símbolos

Δ	Letra grega delta maiúscula
Σ	Letra grega sigma maiúscula
λ	Letra grega lambda minúscula
τ	Letra grega tau minúscula

Sumário

1 Introdução

O *Market Making* (MM) consiste em cotar simultaneamente ordens limitadas de compra e venda no livro de ofertas para capturar o *spread*, controlando o risco de inventário ao longo do tempo. Em ambientes eletrônicos de alta frequência, essa prática é fundamental para promover liquidez e contribuir com a formação de preços. No Brasil, a estratégia enfrenta desafios adicionais, como janelas de leilão, custos de transação relativamente elevados e variações significativas de liquidez entre ativos, fatores que tornam a tarefa especialmente sensível ao desenho do agente e à qualidade da simulação.

Abordagens analíticas (e.g., *Avellaneda-Stoikov*) produzem cotações ótimas sob hipóteses simplificadoras que limitam a sua aplicação a dados reais (independência entre fluxo de ordens, dinâmica difusa sem tendência, intensidades parametrizadas). Por essa razão, cresce o interesse por abordagens de *Deep Reinforcement Learning* (DRL), que aprendem políticas diretamente a partir de dados de mercado ou simulações, sem a necessidade de modelagem explícita. Revisões recentes como em (GAPEROV et al., 2021) apontam que técnicas de DRL superam estratégias tradicionais em termos de retorno ajustado ao risco.

Avanços adicionais incluem a incorporação de sinais externos de tendência e volatilidade no espaço de estados e o uso de espaços de ação contínuos (GAPEROV; KOSTANJAR, 2021), o que aumenta a expressividade das políticas. Também foram propostas arquiteturas modernas, como a Attn-LOB, capazes de lidar com a complexidade dos dados de livros de ofertas em alta frequência (GUO; LIN; HUANG, 2023). Apesar desses progressos, ainda há escassez de estudos aplicados ao mercado brasileiro, que possui particularidades relevantes.

O presente trabalho tem como objetivo desenvolver e avaliar um agente de MM baseado em DRL treinado com dados *tick-by-tick* da B3. O ambiente de simulação reproduz a dinâmica do melhor *bid* e *ask* e permite ao agente executar cotações de forma passiva. O treinamento utiliza o algoritmo *Proximal Policy Optimization* (PPO), com espaço de ações discreto e função de recompensa centrada em PnL penalizado por risco de inventário. Busca-se, dessa forma, avaliar a viabilidade da aplicação de DRL ao *market making* em um mercado emergente, pavimentando extensões futuras previstas na literatura, como a inclusão de sinais externos e ações contínuas.

1.1 Objetivo

Desenvolver e avaliar agente de *market making* baseado em *Deep Reinforcement Learning* que opere no topo do livro de ofertas (nível 1) com dados *tick-by-tick* de *trades* do BTG, que captura diretamente os dados da B3 em tempo real. O objetivo é maximizar o PnL e controlar o risco de inventário em um ambiente de simulação fiel ao fluxo de melhor *bid/ask* e às execuções de negócio efetivos. O trabalho busca preencher uma lacuna na literatura ao aplicar técnicas de DRL ao mercado brasileiro, cujas particularidades de liquidez e microestrutura ainda são pouco explorada em estudos acadêmicos.

1.2 Objetivos específicos

- Processar e preparar os dados, incluindo o nível 1 do livro de ofertas e informações de *trades*, aplicando filtros de leilão, deduplicação de estados e limpeza de registros inválidos.
- Projetar e implementar um ambiente de simulação que replique a dinâmica conjunta de cotações e *trades*, permitindo execução de ordens passivas e contabilização de PnL, caixa e inventário.
- Desenvolver e treinar um agente baseado no algoritmo PPO, utilizando espaço de ações discreto e recompensas centradas no PnL penalizado por risco de inventário, em linha com abordagens da literatura como (GUO; LIN; HUANG, 2023).
- Instrumentar métricas de avaliação (distribuição de ações, PnL e inventário), permitindo análise interpretativa da política aprendida.
- Documentar limitações do modelo atual e propor extensões futuras, como inclusão de sinais externos ou o uso de ações contínuas.

2 Fundamentação Teórica

2.1 Aspectos Conceituais

Market Making

O *market making* consiste em cotar simultaneamente ordens limitadas de compra e venda em um ativo, buscando capturar o *spread* e prover liquidez ao mercado. O *market maker* desempenha papel central na formação de preços e na redução da volatilidade, mas precisa gerenciar o risco de inventário, que surge da execução assimétrica das ordens.

Aprendizado por Reforço Profundo (DRL)

O problema de MM pode ser formulado como um processo de decisão sequencial, no qual um agente observa o estado do mercado e decide como cotar suas ordens. Nesse contexto, o *Reinforcement Learning* (RL) busca miximizar o retorno acumulado, e o *Deep Reinforcement Learning* (DRL) utiliza redes neurais profundas para representar políticas e funções de valor. Aplicações recentes, como em (GAPEROV et al., 2021), demonstram que DRL supera modelos analíticos clássicos e heurísticas em termos de retorno ajustado ao risco.

Dados do Livro de Ofertas e Trades

Estratégias de MM modernas dependem de dados de alta frequência (*tick-by-tick*) provenientes do livro de ofertas (*quotes*) e das transações efetivamente realizadas (*trades*). Os *quotes* capturam a dinâmica do melhor *bid* e *ask*, enquanto os trades refletem a execução real de ordens, ambos compondo a microestrutura do mercado. Trabalhos como (GUO; LIN; HUANG, 2023) utilizam esses dados para treinar agentes capazes de aprender políticas mais realistas e adaptáveis.

2.2 Revisão Bibliográfica

O MM é uma estratégia fundamental em mercados financeiros eletrônicos, cujo objetivo é prover liquides por meio da oferta simultânea de ordens de compra e venda, capturando o *spread* e gerenciando os riscos associados, especialmente o de inventário. O problema pode ser formulado como um processo de decisão sequencial, tipicamente modelado via *Markov Decision Process* (MDP), o que motiva a aplicação de técnicas de aprendizado por reforço.

Abordagens tradicionais, como o modelo de Avellaneda-Stoikov, permitem derivar cotações ótimas por meio de equações diferenciais estocásticas, mas dependem de hipóteses simplificadoras, como difusão sem tendência para o preço de referência e independência entre fluxos de ordens, que não se confirmam em dados reais (GAPEROV et al., 2021). Essa limitação impulsiona o desenvolvimento de métodos baseados em aprendizado profundo por reforço, que aprendem diretamente a partir de dados do mercado, dispensando a necessidade de modelagem explícita.

Nos últimos anos, o uso de DRL em MM ganhou relevância. (GAPEROV; KOSTANJAR, 2021) propõem um arcabouço que integra sinais externos de previsão de tendência e volatilidade ao espaço de estados, além de utilizar um espaço de ações contínuo baseado em *ticks*. O modelo combina ideias de neuroevolução e aprendizado por reforço adversarial, alcançando desempenho superior em termos de retorno ajustado ao risco quando comparado a *benchmarks* tradicionais como *fixed offset*.

Outros trabalhos reforçam esse avanço. A revisão sistemática conduzida por (GAPEROV et al., 2021) aponta que as abordagens de DRL superam estratégias clássicas e heurísticas em termos de retorno ajustado ao risco. O estudo classifica os modelos em categorias, baseados em informação, em modelos analíticos não profundos e profundos, e demonstra a tendência de utilização de arquiteturas relativamente rasas, recompensas centradas em *PnL* penalizado por inventário e espaços de ação discretos ou contínuos com pequenas dimensionalidades.

Por fim, (GUO; LIN; HUANG, 2023) aplicam DRL ao problema de MM com dados do LOB em alta frequência. Os autores introduzem a arquitetura de Attn-LOB, baseado em mecanismos de atenção, e uma função de recompensa híbrida que combina lucro e risco. Além disso, exploram um espaço de ações contínuo que permite maior flexibilidade nas cotações. Os resultados mostram ganhos de desempenho relevantes frente a agentes baseados em *deep* Q-*learning* ou em políticas discretizadas, indicando o potencial das arquiteturas modernas para lidar com a complexidade dos mercados financeiros.

Assim, os três trabalhos convergem para a constatação de que abordagens de DRL oferecem vantagens significativas em relação a modelos analíticos tradicionais, sobretudo na capacidade de lidar com ambientes não estacionários, de alta dimensionalidade e múltiplos riscos associados.

3 Método do trabalho

O presente trabalho implementa um agente de MM inspirado nos conceitos discutidos na literatura recente.

O algoritmo escolhido foi o **Proximal Policy Optimization** (PPO), amplamente utilizado em problemas de controle contínuo e com bons resultados em aplicações financeiras (GAPEROV et al., 2021). A política é do tipo *actor-critic* customizada, com um extrator de características simples baseado em redes *feedforward*, opção alinhada com a tendência identificada na literatura de empregar arquiteturas relativamente rasas para lidar com a baixa razão sinal-ruído dos dados de alta frequência.

A arquitetura da política foi adaptada para o problema do MM brasileiro. O modelo define uma política do tipo *actor-critic* customizada, com um extrator de características simples (*feedforward neural network*) de profundidade configurável. Essa escolha está em linha com as observações de (GAPEROV et al., 2021), que destacam a predominância de arquiteturas relativamente rasas em aplicações financeiros, devido ao risco de *overfitting* e à baixa razão sinal-ruído dos dados.

O ambiente de simulação (*TradeEngineEnv*) foi desenvolvido para reproduzir a dinâmica do livro de ofertas a partir de dados de alta frequência do BTG. Em consonância com a abordagem de (GUO; LIN; HUANG, 2023), que utilizam dados de LOB par treinar agentes de DRL, este trabalho considera exclusivamente o nível 1 do livro (melhor bid/ask), de forma a simplificar o espaço de estados e permitir treinos mais estáveis.

O espaço de ações é discreto, composto pelas opções: cotar no bid, cotar no ask, cotar em ambos ou não enviar cotações. Apesar de não implementar um espaço contínuo como (GAPEROV; KOSTANJAR, 2021) ou (GUO; LIN; HUANG, 2023), esta simplificação inicial é coerente com o objetivo de explorar a viabilidade do uso de DRL em um cenário realista de mercado brasileiro.

A função de recompensa adotada é baseada no PnL do agente, penalizado indiretamente pelo risco de inventário. Essa formulação segue o padrão identificado na revisão de (GAPEROV et al., 2021), em que recompensas densas e centradas no PnL são majoritárias. Também foram adicionados à essa recompensa alguns fatores de bônus ou penalidade com relação à ação que o modelo decidiu, além de normalizar as recompensas para maior estabilização. Futuramente, o modelo pode ser estendido para incorporar sinais adicionais , tal como sugerido por (GAPEROV; KOSTANJAR, 2021), que demonstram ganhos ao incluir previsões de tendência e volatilidade no espaço de estados.

Por fim, foi implementado um *callback* de monitoramento das ações escolhidas

durante o treinamento, permitindo análise interpretativa semelhante às técnicas de interpretabilidade discutidas por (GAPEROV; KOSTANJAR, 2021). Esse recurso possibilita avaliar o equilíbrio entre as diferentes ações (compra, venda, ambas, nenhuma), facilitando a identificação de comportamentos enviesados, como a predominância do *NO_QUOTE*.

3.1 Funcionamento do Modelo e Estratégia de Treinamento

3.1.1 Ciclo de um passo do ambiente

Nesta seção descreveremos, em nível operacional, o que acontece em um passo de decisão do ambiente *TradeEngineEnv*, desde o recebimento da ação até a atualização do estado e da recompensa. A formulação segue a prática de MM por RL, na qual o agente observa o livro, decide se (e onde) ofertar compra e/ou venda, aguarda execução passiva (espera ser agredido) e atualiza seu PnL marcado à mercado com controle de inventário. A estrutura geral é consistente com o procedimento de MM apresentado por (GAPEROV; KOSTANJAR, 2021) e com o desenho de recompensa híbrida discutido por (GUO; LIN; HUANG, 2023).

1. **Leitura do estado em t**

   No início do passo t, o ambiente lê do *dataframe* a observação corrente, composta pelo topo do livro contendo o melhor preço de compra bid_t e o melhor preço de venda ask_t (e, opcionalmente, seus tamanhos), garantindo valores válidos a cada instante por preenchimento à frente (*forward-fill*) quando necessário. Define-se o *mid-price*:

   $$mid_t = \frac{bid_t + ask_t}{2}.$$

   A representação do estado por níveis do topo do LOB e/ou sinais auxiliares é padrão em MM com RL, vide (GAPEROV; KOSTANJAR, 2021).

2. **Precificação das cotações do modelo**

   Com base no *mid*, o ambiente calcula os preços candidatos de cotação, sempre respeitando o *grid* de *tick*:

   $$model\_bid_t = round\_down\_to\_tick(mid_t - \Delta)$$
   $$model\_ask_t = round\_up\_to\_tick(mid_t + \Delta)$$

   em que Δ é um deslocamento mínimo. No caso do modelo criado, $\Delta = 0.01$ em reais, compatível com o *tick* do ativo, de 0.01 para ações com preço acima de R$1,00.

3. **Interpretação da ação do agente**

   A ação discreta $a_t \in \{QUOTE\_BID, QUOTE\_ASK, QUOTE\_BOTH, NO\_QUOTE\}$ determina se e em quais lados o ambiente coloca ordens no instante t:

   - Se o lado incluir *bid*, abre (ou renova) uma ordem de compra no preço $model\_bid_t$ com quantidade fixa $quote\_size$.
   - Se o lado incluir *ask*, abre (ou renova) uma ordem de compra no preço $model\_ask_t$ com a mesma quantidade $quote\_size$.

   Há dois controles operacionais para as ordens: (i) apenas uma ordem viva por lado, estabelecendo um limite de profundidade do inventário de ordens do agente, e (ii) TTL (*time-to-live* em ticks), ordens mais antigas expiram para evitar *staleness* (ficarem antigas com relação ao mercado) e para reforçar *requotes* periódicos.

4. **Avanço no tempo para** $t + 1$

   O ambiente avança uma linha no *dataframe*. Como o modelo é passivo, não há execução imediata em t, ele espera para ser agredido em $t + 1$.

5. **Expiração de ordens (TTL) e manutenção da fila**

   Ao entrar em $t + 1$, o ambiente envelhece as ordens abertas em t e remove as que atingiram o TTL configurado, utilizando uma fila do FIFO por lado (*first-in-first-out*). Assim, só permanecem ativas as ordens mais recentes até o momento de verificar execução.

6. **Verificação de execução passiva em** $t + 1$

   A execução ocorre sempre ao preço da sua própria ordem, característica que define o *maker*, nunca ao preço do topo nem do trade. Isso reforça a natureza passiva do agente. Há duas vias para execução, que são testadas nessa ordem:

   a) **Execução por *trade* no seu preço, com agressor do lado oposto**

      Se em $t+1$ ocorre um *trade* exatamente no preço determinado pela ordem, ela pode ser do agressor vendador ou do agressor comprador.

      O agressor vendedor pode executar as ordens de *bid* no preço determinado pelas ordens. O agressor comprador pode executar as ordens de *ask* também no preço determinado por essas ordens.

      Se o volume do *trade* está disponível, o preenchimento é limitado pelo tamanho negociado. Após preencher a $quote\_size$, nada mais é executado naquele lado.

   b) **Execução por "*book-cross*"**

Se não houve *trade* no preço das ordens vivas, verifica-se o cruzamento no topo do livro com a condição de que o preço do *bid* deve ser maior que o preço do *ask*.

Se $ask_{t+1} \leq seu\_bid$, executa o *bid* no preço.
Se $bid_{t+1} \geq seu\_ask$, executa o *ask* no preço.

Se a cotação não executar por nenhuma via, a ordem permanece viva até morrer pelo TTL no próximo passo.

7. **Marcação a mercado (MTM) em** $t+1$

   Com base no inventário I após eventuais *fills*, o ambiente computa a riqueza realizável se a posição fosse zerada nesse instante de tempo.

   Se $I > 0$ (*long*), zera o inventário com o *bid*: $I \cdot bid_{t+1}$.

   Se $I < 0$ (*short*), zera o inventário com o *ask*: $I \cdot ask_{t+1}$.

   Essa "*zeragem*" é somada ao caixa do agente, resultando no MTM_{t+1}

8. **Recompensa** r_t

   A recompensa combina três termos principais, escolhidos para remunerar o *spread*, evitar especulação e controlar o inventário. Para complementar a recompensa, foram criados mais quatro termos, sendo dois bônus e duas penalizações, baseados na ação e no inventário do agente.

   a) **Variação do MTM (ΔMTM com amortecimento em ganhos)**

   $$\Delta MTM_t = MTM_{t+1} - MTM_t$$

   Aplicamos um fator de amortecimento apenas quando $\Delta MTM > 0$, reduzindo incentivo ao carregar inventário apenas porque o MTM está subindo, que caracteriza um comportamento especulativo por parte do agente. Para $\Delta MTM < 0$, não há amortecimento, dessa forma as perdas prejudicam de maneira integralmente, incentivando o agente a reduzir os riscos.

   $$dumped\_MTM_t = \begin{cases} \Delta MTM_t & \Delta MTM < 0 \\ (1 - dumped\_factor) \cdot \Delta MTM_t & \Delta MTM \geq 0 \end{cases}$$

   b) ***Realized this step*** **(lucro de execução por passo)**

   Remunera o *spread* efetivamente capturado nesse passo:

   $$r_t^{\text{exec}} = \sum_{i \in \text{sells}} (price_i^{\text{fill}} - \text{mid}_{t+1}) \cdot qty \; + \sum_{j \in \text{buys}} (\text{mid}_{t+1} - price_j^{\text{fill}}) \cdot qty$$

   Esse termo remunera baseado na ação corrente (executar limites favoráveis ao *mid*)

c) **Penalidade de inventário (risco)**

Penalidade quadrática $\lambda \cdot I^2$ para desincentivar posições grandes e prolongadas.

d) ***Inventory Reduction* (Bônus de execução para diminuição de inventário)**

Esse termo busca incentivar o modelo a reduzir sua posição absoluta, com um controle com o inventário relativo do agente.

$$r_t^{inv\_red} = inv\_red\_weight \cdot abs(I_t)/max\_I$$

Com inventário positivo, o agente recebe esse bônus ao fazer ações de venda ou cotação em ambos os lados. Com inventário negativo, o agente recebe ao fazer ações de compra ou cotação em ambos os lados. O bônus não depende da execução da ação.

e) ***Near Miss* (Bônus pela ação correta não executada)**

Recompensa para publicação de uma cotação ativa e, mesmo não sendo executado, o movimento subsequente do *mid price* confirma que a oferta estava bem posicionada, ou seja, o preço de mercado se moveu na direção da cotação do agente.

$$mid_{t+1} > mid_t + tick\_size, \text{ para venda}$$
$$mid_{t+1} < mid_t - tick\_size, \text{ para compra}$$

f) ***No quote inventory* (penalização por inércia com inventário)**

Penalização pelo agente não tomar uma decisão que busque reduzir seu inventário. Também normalizado com a razão do inventário atual pelo inventário máximo.

$$p_t^{noq\_inv} = no\_quote\_inv\_weight \cdot abs(I_t)/max\_I$$

g) ***No quote streak* (penalização por sequências de *NO_QUOTE*)**

Penalização pelo agente escolher a ação de NO_QUOTE de forma seguida. A recompensa acumula com a quantidade de ações seguidas.

$$p_t^{noq\_streak} = no\_quote\_streak\_penalty \cdot streak\_count_t$$

(GUO; LIN; HUANG, 2023) destacam que o PnL puro incentiva o comportamento especulativo do agente. Uma função de recompensas híbrida que combina o lucro de execução/riqueza com controle de risco de inventário tende a reduzir esse viés.

9. **Atualização de variáveis e montagem do próximo estado**

Por fim, o ambiente atualiza o *cash*, *inventory*, MTM e guarda o MTM_t para calcular o próximo ΔMTM. Também incrementa o passo temporal e aplica, se configurado, o limite de passos do episódio. Monta a próxima observação do livro e prossegue para o próximo passo.

3.1.2 Justificativas das simplificações e definições operacionais

Espaço de ações discreto

Adota-se um conjunto discreto de ações que decide se e em quais lados as ordens serão postadas. A simplificação reduz a dimensionalidade do controle, melhora a estabilidade de treino inicial e preserva a essência do papel do *market maker*, de prover liquidez e gerir inventário. A estrutura do ciclo observar-cotar-aguardar-executar-recotar corresponde ao procedimento de MM com MDP apresentado em (GAPEROV; KOSTANJAR, 2021). Embora os autores empreguem deslocamentos contínuos em *ticks* relativos ao livro, a presente versão discreta mantém o princípio de cotar relativo ao estado do LOB, servindo como etapa inicial antes de estender para ações contínuas.

Ausência de sinais externos no estado (formulação do nível 1

O estado exposto ao agente contém apenas o topo do livro (melhor *bid/ask* e tamanhos opcionais), sem sinais auxiliares de tendência/volatilidade/fluxo de ordens. Essa decisão isola o efeito das regras de execução e do desenho da recompensa sobre o comportamento do agente, evitando viés especulativo induzido por *features* mal calibradas.

Esse risco é discutido ao criticar recompensas baseadas exclusivamente em PnL, propondo-se recompensas híbridas, como (GUO; LIN; HUANG, 2023). A inclusão de sinais é prevista como uma extensão do modelo, conforme a organização do estado em (GAPEROV; KOSTANJAR, 2021).

Forward fill do topo do livro

Forward fill é um procedimento de pré-processamento que propaga para frente o último valor conhecido das colunas de topo do livro, tipicamente o melhor *bid* e o melhor *ask*, para linhas do *dataframe* que não trazem atualização explícita do *book*. Por exemplo, linhas que registram *trade*. Dessa forma, todo passo temporal do ambiente tem um estado bem definido de LOB nível 1.

Esse procedimento é necessário pois o agente precisa enxergar a cada ciclo de *maker* (observar-cotar-aguardar-executar-recotar) o bid_t e ask_t a cada decisão. Na prática, os dados *tick-by-tick* chegam intercalando mensagens de *book* e de *trade*. Nas linhas de *trade*, o topo do livro pode não vir repetido, então o mecanismo de *forward fill* garante consistência do estado sem precisar criar ou inventar preços novos, apenas mantendo o último topo observado até surgir uma nova atualização de *book*.

Grid de *tick*, arredondamentos e métricas básicas de preço

Seja bid_t o melhor preço de compra e ask_t o melhor preço de venda no topo do livro no instante t. Define-se o *spread* como:

$$spread_t = ask_t - bid_t,$$

e o preço médio (*mid-price*) como:

$$mid_t = \frac{bid_t + ask_t}{2}.$$

O tamanho do *tick* (τ) é o menor incremento permitido de preço para o ativo. Todo preço válido deve ser múltiplo de τ. Nas formulações de MM por RL, é comum parametrizar cotações relativas ao estado do LOB em unidades de *ticks* (deslocamentos em relação ao *mid* ou aos melhores preços) e, ao final, quantizá-las para *grid* de *tick*.

Ao construir as cotações do modelo a partir do mid_t com um deslocamento Δ (em reais) ou δ (em *ticks*, $\Delta = \delta\tau$), aplicam-se:

$$model\_bid_t = round\_down\_to\_tick(mid_t - \Delta)$$
$$model\_ask_t = round\_up\_to\_tick(mid_t + \Delta)$$

O uso de *round-down* para *bid* e *round-up* para *ask* evite que erros numéricos coloquem as ordens dentro do *spread* ou em preços inválidos, assegurando consistência com o *grid*. Esse procedimento é a contraparte prática da ideia de "deslocamentos relativos em ticks"do espaço de ações de (GAPEROV; KOSTANJAR, 2021).

Uma ordem viva por lado e TTL curto

Restringe-se a uma ordem ativa por lado (*bid*/*ask*) e aplica-se TTL curto em passos discretos. A política limita inflação de ordens, modela *staleness* e obriga *requotes* frequentes, aproximando a rotina operacional do maker no ciclo do algoritmo 1 descrito por (GAPEROV; KOSTANJAR, 2021). O TTL também reduz a complexidade do estado com idades múltiplas de ordens e melhora a estabilidade de treino.

Staleness refere-se a perda de atualidade de uma ordem que permanece ativa enquanto mercado se desloca, tornando-a desalinhada ao valor justo corrente e mais exposta à seleção adversa (tente a executar justamente quando o movimento de preço é desfavorável). O TTL curto atua como mecanismo anti-*staleness* pois força *requotes* frequentes, mantendo as cotações coerentes com o *mid* e com o inventário do agente.

Execução passiva (*maker*) vs. agressiva (*taker*)

Adota-se distinção operacional entre ordem limite (não cruza o *book*) e ordem a mercado (cruz o *book*). Na execução passiva (*maker*), o agente posta ordens limite e espera ser agredido por um *taker*. Quando há execução, ela ocorre no próprio preço cotado pelo agente, não pelo topo do livro.

O objetivo é capturar o *spread* comprando abaixo do *mid* e vendendo acima, ao custo de risco de inventário, para evitar o acúmulo de posição, e seleção adversa, onde ordens passivas tendem a executar quando o preço está prestes a andar contra o agente. Na execução agressiva (*taker*), envia-se ordem que cruza o *spread* para executar imediatamente no preço do lado oposto, de forma a pagar o *spread*.

No ambiente proposto, o agente atua apenas como *maker*: a execução é verificada em $t + 1$ por trade exato no preço da ordem (com agressor coerente) ou por *book-cross* (o topo cruza o nível da ordem), sempre liquidando ao preço da própria cotação. Essa modelagem segue o procedimento de MM como MDP e mantém o foco em capturar o *spread* com auxílio do controle de inventário, como recomendado nas discussões sobre recompensa híbrida (GUO; LIN; HUANG, 2023).

Regra de execução por "trade exato"e "*book-cross*"

A execução é verificada em $t + 1$ por duas vias complementares:

(i) ocorrência de *trade* exatamente no preço da ordem do agente, com agressor do lado oposto;

(ii) cruzamento do topo do livro, quando $ask_{t+1} \leq bid\_do\_agente$ (para *bids*) ou $bid_{t+1} \geq ask\_do\_agente$ (para *asks*).

A verificação de igual de igualdade usa tolerância numérica na ordem de 10^{-12} para evitar falsos negativos por ponto flutuante. Essa modelagem operacionaliza a execução passiva descrita no procedimento do *maker*.

Marcação a mercado (*mark-to-market*) e ΔMTM

Chama-se *long* a posição comprada (inventário $I > 0$), que se beneficia de altas de preço. *Short* é a posição vendida (inventário $I < 0$, que se beneficia de quedas. Em microestrutura, a liquidação imediata da posição usa os preços do topo do livro, dessa forma posições *long* são zeradas ao *bis* e posições *short* ao *ask*, refletindo o custo de sair imediatamente da posição como tomador de liquidez (*taker*).

A marcação a mercado é a riqueza realizável se o agente zerasse a posição dele em algum momento t. Seja C_t o caixa acumulado e (bit_t, ask_t) o topo do livro no instante t. Define-se:

$$MTM_t = \begin{cases} C_t + |I_t| \cdot bid_t & I_t \geq 0(\text{long}) \\ C_t + |I_t| \cdot ask_t & I_t < 0(\text{short}) \end{cases}$$

Essa conversão é compatível com a prática de MM com zeragem do lado oposto do livro e é a base para medir riqueza e risco de inventário ao longo do tempo segundo (GAPEROV; KOSTANJAR, 2021).

A variação de marcação a mercado entre os passos captura o efeito conjunto de novos negócios realizados e a re-precificação do inventário remanescente aos preços de saída imediata. Em ambiente de RL para MM, ΔMTM costuma integrar a recompensa como componente de riqueza/risco, tipicamente combinado com penalidade de inventário para desincentivar alavancagem especulativa.

Ao longo do treinamento e teste, percebeu-se que o ΔMTM muitas vezes dominava a recompensa, gerando recompensas grandes tanto positivas quanto negativas e que ofuscavam a utilidade dos demais termos. Para controlar esse comportamento, um peso para o ΔMTM final foi adicionado, *mtm_weight*, resultando na formulação final do termo de MTM para a recompensa como:

$$MTM\_reward_t = mtm\_weight \cdot dumped\_MTM_t$$

"*Realized this step*" (lucro de execução no passo)

Para isolar o mérito da execução no passo corrente, considera-se um termo de lucro realizado contra o *mid* futuro:

$$r_{t+1}^{\text{exec}} = \sum_{i \in \text{sells}} (price_i^{\text{fill}} - \text{mid}_{t+1}) \cdot qty \; + \sum_{j \in \text{buys}} (\text{mid}_{t+1} - price_j^{\text{fill}}) \cdot qty$$

análogo ao componente de recompensa que compara preço de execução e *mid* futuro em (GAPEROV; KOSTANJAR, 2021). Esse termo remunera a captura de *spread* da ação atual, enquanto que o ΔMTM reflete o efeito de marcação do inventário.

Esse é um dos termos principais relacionados à recompensa direta da ação realizada em cada passo. Durante os treinamentos e testes, observou-se que os valores negativos eram, em magnitude, significativamente superiores aos valores positivos.

Essa assimetria ocorre porque quando o agente é agredido em uma posição passiva contrária ao movimento subsequente do preço, a perda é contabilizada de forma concentrada em um único passo, refletindo a diferença entre o preço de execução e o novo *mid price* do mercado. Já os ganhos tendem a ser menores e mais distribuídos, pois o agente frequentemente realiza pequenos *spreads* positivos entre a compra e a venda.

Essa desproporção entre ganhos e perdas fazia com que a política aprendida se tornasse excessivamente conservadora, favorecendo ações de *NO_QUOTE*. Para corrigir esse viés, introduziu-se uma ponderação assimétrica de pesos no termo de execução, ampliando o impacto de ganhos e reduzindo o peso das perdas.

Esse ajuste preserva o sinal econômico correto da recompensa, mas reduz o desequilíbrio estatístico entre amostras positivas e negativas, o que favorece a estabilidade do aprendizado e incentiva o agente a explorar cotações lucrativas sem ser penalizado de forma desproporcional por oscilações adversas.

$$realized_t = \begin{cases} weight\_pos \cdot r_t^{exec} & r_t^{exec} > 0 \\ weight\_neg \cdot r_t^{exec} & r_t^{exec} < 0 \end{cases}$$

Penalidade de inventário

Aplica-se penalidade quadrática λI^2 na recompensa para conter exposições prolongadas e grandes posições. A penalidade de inventário é canônica em MM, mas a escolha por I^2 reforça a aversão a caudas de inventário, adequada ao presente agente com ações discretas e TTL curto.

Bônus e Penalidades

Os quatro termos auxiliares da função de recompensa foram concebidos para induzir comportamentos desejáveis de MM e atenuar estados indesejados no aprendizado, sem distorcer o objetivo econômico central baseado em lucro e risco. Cada termo atua sobre aspectos distintos do controle de posição, atividade de cotação e participação de mercado.

1. *Inventory Reduction* (Bônus de execução para diminuição de inventário)

   Esse termo busca incentivar o modelo a reduzir posições extremas e manter o inventário próximo de 0, condição ideal para um MM passivo que busca capturar o *spread* sem exposição direcional ao preço.

   O bônus é proporcional ao inventário relativo e é ativado quando o agente toma decisões que reduzem o valor absoluto de sua posição, independente da ordem ser ou não executada. Assim, se o inventário é positivo (posição comprada), ações de venda ou *both* são recompensadas; se negativo, ações de compra ou *both* recebem o bônus.

   Essa abordagem visa internalizar o controle de risco na função de recompensa encorajando decisões que corrigem desequilíbrios de estoque mesmo antes de ocorrer uma execução.

2. ***Near Miss* (Bônus pela ação correta não executada)**

   Esse termo atua como um reforço positivo para decisões de cotação que, embora não resultem em execução imediata, demonstram uma leitura correta da direção local do mercado.

   O bônus é concedido quando o agente publica uma cotação de compra (ou venda) e, no próximo *tick*, o *mid price* move-se na direção dessa cotação, mostrando que o mercado confirma a intenção do agente.

   Esse comportamento é desejável porque reflete posicionamento informativo e competitivo, típico de MM eficientes.

   Do ponto de vista do aprendizado, esse termo densifica o sinal de recompensa em ambientes com execuções raras, permitindo que o agente receba reforço parcial mesmo sem *trade*, o que acelera a convergência e melhora a estabilidade do PPO.

$$mid_{t+1} > mid_t + tick\_size, \text{ para venda}$$
$$mid_{t+1} < mid_t - tick\_size, \text{ para compra}$$

3. ***No quote inventory* (penalização por inércia com inventário)**

   Essa penalização é aplicada quando o agente mantém uma posição aberta (inventário $\neq 0$), mas opta por não cotar em nenhum dos lados do livro.

   O objetivo é desincentivar comportamentos de inércia que aumentam o risco de exposição direcional, especialmente em momentos de volatilidade, nos quais o agente deveria estar buscando reduzir o inventário.

   A penalização é proporcional à fração do inventário atual em relação ao limite máximo permitido, refletindo a ideia de que quanto maior a exposição, maior o custo de permanecer passivo.

$$p_t^{noq\_inv} = no\_quote\_inv\_weight \cdot abs(I_t)/max\_I$$

4. ***No quote streak* (penalização por sequências de inatividade)**

   Esse termo visa penalizar o agente por sequências consecutivas da ação NO_QUOTE, comportamento que pode indicar colapso exploratório, quando a política evita cotar devido a punições a punições passadas, levando a uma perda de exposição e aprendizado insuficiente.

   A penalidade acumulada linearmente com o número de passos consecutivos em que o agente permanece inativo, forçando um retorno eventual à atividade de cotação.

$$p_t^{noq\_streak} = no\_quote\_streak\_penalty \cdot streak\_count_t$$

Em conjunto, esses quatro termos secundários atuam como mecanismos de *shaping* comportamental, reforçado práticas típicas de MM reais, como controle de estoque, presença contínuo no livro de ofertas e adaptação ao fluxo de preços, sem sobrepor-se ao objetivo central de maximizar o lucro líquido ajustado ao risco.

Ao incluir esse *shaping*, a recompensa final possui o seguinte formato:

$$reward_t = MTM\_reward_t + realized_t - inv\_pen_t + r_t^{inv\_red} + r_t^{near} - p_t^{noq\_inv} - p_t^{noq\_streak}$$

3.1.3 Procedimento de treinamento

Optou-se por ambientes vetorizados em *subprocessor* para coletar trajetórias em paralelo, reduzir correlação temporal e refletir melhor a dinâmica *maker* com execução por *first-passage* segundo (GAPEROV; KOSTANJAR, 2021). A rede rasa e estreita é escolhida a fim de privilegiar estabilidade e interpretabilidade, dado que o custo dominante é a simulação do ambiente na CPU, tornando a GPU vantajosa apenas marginalmente nesse arranjo.

PPO foi o algoritmo *on-policy* escolhido pela boa relação entre estabilidade e simplicidade, com hiperparâmetros calibrados ao problema de MM e coerentes com a recompensa híbrida que incentiva a captura de *spread* e desincentiva o comportamento especulativo, conforme discutido em (GUO; LIN; HUANG, 2023) e (GAPEROV; KOSTANJAR, 2021). A avaliação periódica em janelas com métricas orientadas a execução e risco foram escolhidas também espelhando o codo na literatura em equilibrar provisões de liquidez com controle de inventário.

Organização dos dados e criação dos ambiente

Os arquivos diários são ordenados por data e cada dia compõe um episódio do ambiente. Para cada arquivo selecionado cria-se uma instância independente do *TradeEngineEnv*, com:
(i) $max\_length = len(df) - 1$, a quantidade máxima de passos do ambiente é a quantidade totais de linhas do dia;
(ii) $order\_ttl\_ticks = 1$, renovação frequente de cotações; e (iii) encapsulamento por Monitor para auxiliar a lidar com os ambientes de forma paralelizada.

Ambientes vetorizados e paralelização

Emprega-se um vetor de ambientes (um por dia) por meio de *SubprocVecEnv*, de modo que as trajetórias sejam coletadas em paralelo, um processo por ambiente. Essa estratégia reduz correlação temporal entre amostrar e melhora a eficiência amostral do algoritmo *on-policy*. Em configurações de depuração, usa-se *DummyVecEnv*.

Seleção de dispositivo (CPU/GPU)

O treinamento é executarem em GPU quando possível e em CPU em caso contrário. Nesse trabalha, a rede é rasa e estreita, de modo que o custo dominante tende a ser a simulação do ambiente no lado da CPU. Dessa forma, a aceleração por GPU não é um fator muito importante e, portanto, opcional para o treinamento.

Normalização e Escalonamento das Recompensas

Durante o treinamento, as recompensas produzidas pelo ambiente são normalizadas em tempo de execução com o objetivo de estabilizar e reduzir a variância entre episódios de diferentes escaldas. Essa etapa é realizada fora do ambiente, por meio de um *wrapper* da biblioteca SB3, aplicado sobre os ambientes vetorizados utilizados no treinamento.

O método normaliza o valor das recompensas observadas com base na média e desvio padrão acumulados ao longo das interações, de forma que o agente receba retornos padronizados em torno de zero. Além disso, é aplicado um *clipping* em ± 10 para limitar o impacto de recompensas extremas, evitando explosões de gradiente e desbalanceamento da política.

Essa normalização não altera o sinal econômico da recompensa, mas ajusta sua escala relativa, de modo que ganhos e perdas de diferentes magnitudes contribuam de forma equilibrada para o gradiente da política. Na prática, essa transformação explica a diferença entre os valores absolutos registrados na métrica de recompensa média por episódio de treino (normalizada) e os valores reais de lucro e perda (não normalizados) podem ser observados nas métricas de treino e teste.

Parâmetros do Ambiente de Treinamento

A tabela abaixo apresenta os principais hiperparâmetros definidos no ambiente de treinamento. Os valores base foram os utilizados para o treinamento do agente desse trabalho.

Esses valores controlam o comportamento do agente e da função de recompensa, afetando diretamente o equilíbrio entre execução, risco e estabilidade do aprendizado.

Parâmetro	**Descrição**	**Valor base**
quote_size	Quantidade padrão mínima enviada em cada ordem de compra ou venda.	100 unidades
spread	Distância mínima entre *bid* e *ask* definida como alvo para cotação.	0.01
tick_size	incremento mínimo de preço; define a granularidade das cotações.	R$0.01
max_inventory	Limite absoluto de inventário permitido.	1.000 unidades
dumped_factor	Fator de amortecimento do ΔMTM, reduzindo o impacto de oscilações abruptas entre passos.	0.2
inventory_factor	Penalidade quadrática sobre o inventário; limita posições excessivas.	2×10^{-5}
starting_cash	Caixa inicial do agente.	R$100,000.00
max_order_per_side	Número máximo de ordens simultâneas permitidas em cada lado.	1
order_ttl_ticks	Tempo de vida de uma ordem em *ticks* antes de expirar.	1
mtm_weight	Peso do ΔMTM. Mantido pequeno para evitar que oscilações do preço dominem a recompensa.	0.005

Continua na próxima página

Continuação da Tabela 1

Parâmetro	**Descrição**	**Valor base**
pos_realized_weight	Peso aplicado a ganhos realizados positivos (*realized this step*).	5
neg_realized_weight	Peso aplicado a perdas realizadas negativas.	2
inventory_reduction_weight	Bônus proporcional à redução do inventário em ações que o diminuem.	0.05
near_miss_weight	Bônus concedido quando o *mid price* move-se na direção da cotação não executada.	0.01
no_quote_inventory_penalty	Penalidade por permanecer em *no quote* com inventário diferente de zero.	0.05
no_quote_streak_penalty	Penalidade crescente para sequências consecutivas de *no quote*, prevenindo inatividade prolongada.	0.05

Tabela 1 – Parâmetros do *TradeEngineEnv*

Essas constantes foram ajustadas empiricamente após experimentos, de modo a estabilizar o aprendizado e gerar políticas que mantêm *spreads* equilibrados, controle de estoque e PnL médio positivo.

3.1.4 Arquitetura da política e *feature extractor*

Adota-se uma política personalizada do tipo ator-crítico com *feature extractor* MLP rasa: uma camada escondida de largura de 64 neurônios, ReLU e cabeças padrão de política e valor do SB3. Essa escolha favorece estabilidade e transparência por inferir menor propensão a sobreajuste microestrutural.

Instanciação do PPO: parâmetros e racional

O algoritmo de controle é o *Proximal Policy Optimization* (PPO), pela boa relação entre estabilidade e simplicidade em cenários de *on-policy*. Os hiperparâmetros efetivamente utilizados e sua função no contexto de MM são:

- $policy = CustosActorCriticPolicy,\ policy\_kwargs = \{'hidden\_dim;: (1, 64)\}$

  MLP rasa (1x64), reduz a variância de gradiente, facilita reprodutibilidade e evita sobreajuste a regimes intradiários, mas também limita a capacidade do agente de compreender completamente os dados.

- $env = SubprocVecEnv([...])$

  Um ambiente por dia com coleta paralela e heterogeneidade temporal entre episódios.

- $n\_steps = min(2048, episode\_len)$

  Passos por ambiente antes de cada atualização. Mantém o *rollout* dentro do episódio completo e fornece contexto suficiente para ΔMTM/inventário sem latência excessiva de atualização.

- $batch\_size = 256$

  Tamanho dos lotes usados no treinamento. Um valor pequeno fornece atualizações frequentes e melhor adaptação a séries temporais curtas.

- $n\_epochs = 2$

  Número de iterações sobre cada lote dos dados. Mantido baixo para reduzir sobreajuste e ruído.

- $ent\_coef = 0.02$

  Peso do termo de entropia, responsável por incentivar a exploração e evitar colapso em políticas determinísticas.

- $learning\_rate = 3 \cdot 10^{-4}$

  Taxa de aprendizado do otimizador Adam. Valor padrão do PPO, adequado à escala das recompensas normalizadas.

- $gamma = 0.999$

  Desconto padrão para algorítimo PPO.

- $gae\_lambda = 0.95$

  Controle do balanço entre viés e variância.

- $clip\_range = 0.2$

  Limita a variação da política por atualização, prevenindo passos grandes que desestabilizem cotações.

- $vf\_coef = 0.5$

  Peso da função de valor na perda total. Define o equilíbrio entre precisão da estimativa de valor e estabilidade da política.

- $device = (\text{"}cuda\text{"} \| \text{"}cpu\text{"}),\ verbose = 1$ e tensorboard_log

  Dispositivo de treino, nível de registro e diretório para os registros de treinamento. Não afetam a formulação do treino.

Horizonte total de treinamento

O orçamento de *timesteps* é definido como:

$$total\_timesteps \approx \overline{len(\text{episódio})} \cdot n\_envs \cdot passes,$$

onde $\overline{len(\text{episódio})}$ é o comprimento médio dos dias utilizados, n_envs é o número de ambientes (dias) e *passes* o número de revisitações planejadas aos dados. Esse arranjo viabiliza múltiplas passagens sobre o mesmo histórico, ao custo de regularização para mitigar sobreajuste.

Reprodutibilidade e avaliação

Define-se semente global *seed* para inicialização de pesos e ambientes. Executa-se avaliação periódica em janelas com PnL total, ΔMTM acumulado, inventário máximo e *fill ratio* (por *book* ou por *trade*). Critérios de parada incluem orçamento de *timesteps* e saturação de métricas.

4 Especificação de Requisitos

O sistema desenvolvido para este trabalho tem como finalidade treinar e avaliar um agente de *market making* baseado em *Deep Reinforcement Learning*, utilizando dados *tick-by-tick* do mercado brasileiro. A seguir, são detalhados os requisitos funcionais e não funcionais.

4.1 Requisitos Funcionais

- O sistema deve processar dados *tick-by-tick*, incluindo informações de livro de ofertas e negociações efetivas.
- O sistema deve fornecer um ambiente de simulação que replique a dinâmica do melhor *bid*/*ask*, com execução de ordens passivas do agente.
- O sistema deve calcular e registrar métricas de desempenho do agente, incluindo PnL, inventário e caixa, de forma acumulada ao longo do episódio.
- O sistema deve permitir ao agente escolher ações em um espaço discreto: cotar no *bid*, cotar no *ask*, cotar em ambos ou não enviar cotações.
- O sistema deve aplicar a função de recompensa definida, baseada em PnL penalizado pelo risco de inventário.
- O sistema deve registrar métricas de treinamento, para possibilitar análise posterior.

4.2 Requisitos Não Funcionais

- O sistema deve ser implementado em Python, utilizando a biblioteca *Stable-Baselines*3 para algoritmos de RL e PyTorch para redes neurais.
- O sistema deve ser modular, permitindo substituição de componentes como política, função de recompensa e estrutura do ambiente.
- O sistema deve ser capaz de processar milhares de registros de dados sem perdas, mantendo desempenho adequado.
- O sistema deve ser transparente e auditável, com armazenamento estruturado de logs para avaliação dos experimentos.
- O sistema deve evitar sobrecarga de *threads* durante o treinamento paralelizado, conforme configuração de variáveis do ambiente no *script* de treino.

4.3 Aquisição de dados

Uma das etapas fundamentais para a criação de um modelo é a fonte de dados utilizada em seu treinamento e teste. (GUO; LIN; HUANG, 2023) utiliza dados históricos do *Shenhen Stock Exchange*, uma das três principais bolsas de valores da China Continental, para fazer o treinamento do modelo. De forma similar, totalizando dados de 21 dias, (GAPEROV; KOSTANJAR, 2021) utiliza dados históricos *tick-by-tick* correspondentes a 30 dias de negociação da corretora de criptomoedas *Bitstamp* para fazer a avaliação do modelo proposto. As abordagens utilizadas em pesquisas anteriores motivaram a utilização de mais dados para a avaliação completa do modelo, buscando também uma diversificação especialmente na avaliação e testes do modelo, assim como (GUO; LIN; HUANG, 2023), que utilizou dados de 3 empresas diferentes da bolsa chinesa em seu conjunto de dados,

O modelo desenvolvido neste trabalho utiliza dados de 19 dias de treino do ativo PETR4, que corresponde às ações preferenciais da Petrobras (Petróleo Brasileiro S. A.), uma das maiores empresas listadas na B3 (Brasil, Bolsa, Balção), a bolsa de valores do Brasil. Essa ações são amplamente negociadas e apresentam alta liquidez e grande volume de ordens, características desejáveis para experimentos de MM. Os 19 dias de treino correspondem ao intervalo do dia 05 de março de 2025 ao dia 31 de março de 2025.

Assim como os ativos utilizados por (GUO; LIN; HUANG, 2023) e (GAPEROV; KOSTANJAR, 2021), o PETR4 foi escolhido por representar um ambiente de negociação dinâmico. Ao todo, são 71.194 amostras incluindo mudanças de estado do livro de ofertas e trades, com 66.202 amostras de trades e 4992 amostras de mudanças no livro de ofertas.

Os dados foram adquiridos utilizando a API de dados de mercado do BTG Pactual, cuja documentação oficial está disponível em BTG Pactual Data Services. A API fornece dados *tick-by-tick* de cotações e trades, contendo para cada evento informações como preço, volume, lado da operação, profundidade do livro, entre outras que precisaram ser filtradas. Esses dados permitem reconstruir a dinâmica do nível 1 do livro de ofertas, contendo o melhor *bid* e *ask*, e do fluxo de negociações efetivamente realizadas.

Os dados disponibilizados não incluem metadados explícitos sobre o horário oficial de abertura e fechamento do pregão. Para garantir consistência entre os dias analisados, foi aplicado um filtro temporal em cada arquivo diário, de modo a considerar apenas o intervalo principal de negociação. O período adotado foi das 13h00 às 20h50 (UTC), definito empiricamente a partir da inspeção dos dados brutos. Esse intervalo corresponde aproximadamente ao horário regular de negociação da B3, que ocorre das 10h00 às 17h00 no horário de Brasília (BRT), ajustado para o fuso horário UTC. O horário inicial de 13h00 representa o início das transações efetivas observadas, enquanto o horário final de 20h50 foi determinado com base na análise ds gráficos de volume e frequência de cotações, identificando o momento em que as ofertas passam a ser gradualmente retiradas do livro

de ofertas.

Essa abordagem empírica visa eliminar períodos de pré-abertura, leilões de fechamento e registros residuais fora do horário regular, garantindo que o modelo seja treinado apenas com dados de mercado contínuo. Ressalta-se, entretanto, que eventuais variações nos horários de fechamento (como ajustes sazonais ou mudanças operacionais da B3) podem introduzir pequenas diferenças entre os dias analisados.

Além do filtro temporal, foi aplicada uma etapa de deduplicação das amostras. Durante a coleta, API pode registrar múltiplas atualizações consecutivas com os mesmos valores de preço e volume, resultando em registros redundantes do mesmo estado do livro de ofertas. Para evitar o sobrecarregamento do conjunto de dados com informações repetidas e garantir que cada linha represente uma mudança efetiva do mercado, foram removidas entradas duplicadas nos dados de *trade* nos campos *md_entry_size*, *md_entry_px* e *agressor* e nos dados de *book* nos campos *en_b_px_1* e *en_s_px_1*.

Essa limpeza reduziu significativamente o número de amostrar por dia, como dias que possuíam 100.000 registros e, após a filtragem de duplicação, reduziu para 4.000, o que indica a presença de múltiplas mensagens repetidas na transmissão original. Essa etapa melhora a eficiência computacional do treino e evita que o modelo aprenda padrões artificiais oriundos de repetição técnica dos dados.

4.3.1 Filtragem dos Dados de *Trade*

Nos dados de *trade* existem 6 informações de interesse, sendo elas:

Campo	**Descrição**
symbol	Representa o *ticker* do ativo, como PETR4, utilizado para identificar o instrumento financeiro.
rpt_seq	Valor sequencial de atualização de cada instrumento, garantindo a ordenação cronológica das mensagens recebidas.
network_received_time	Momento em que o pacote de dados foi recebido pela rede, representando o tempo de chegada da informação.
md_entry_px	Preço estabelecido para o *trade* ou cotação.
md_entry_size	Quantidade (volume) associada ao *trade* correspondente.
agressor	Indica a condição do agressor da negociação, informando se a ordem executada partiu do lado comprador (valor 3) ou vendedor (valor 4).

Tabela 2 – Descrição dos campos desejados para os dados de *trade*

4.3.2 Filtragem dos Dados de *Book*

Nos dados de *book* existem 6 informações de interesse, sendo elas:

Campo	**Descrição**
symbol	Representa o *ticker* do ativo, como PETR4, utilizado para identificar o instrumento financeiro.
rpt_seq	Valor sequencial de atualização de cada instrumento, garantindo a ordenação cronológica das mensagens recebidas.
network_received_time	Momento em que o pacote de dados foi recebido pela rede, representando o tempo de chegada da informação.
msg_type	Indica o tipo de mensagem, incremental ou *snapshot*.
en_b_px_1	Preço de compra do livro no nível 1
en_s_px_1	Preço de venda do livro no nível 1.

Tabela 3 – Descrição dos campos desejados para os dados de *book*

Nos dados de *book* foram aplicados filtros adicionais com o objetivo de garantir que apenas eventos válidos e informativos fossem considerados.

O primeiro filtro seleciona apenas as mensagem cujo campo *msg_type* é igual a *incremental*, descartando mensagens do tipo *snapshot*. Enquanto as mensagens *snapshot* representam o estado completo do livro em um instante, as mensagens *incremental* indicam mudanças efetivas nos níveis de preço e volume. Assim, esse filtro assegura que o modelo receba apenas atualizações reais do mercado, evitando repetições desnecessárias do mesmo estado.

O segundo filtro remove todas as observações em que $en\_b\_px\_1 = 0$ ou $en\_s\_px\_1 = 0$, correspondentes, respectivamente, aos preços de melhor *bis* e *ask* no topo do livro. Valores nulos nesses campos indicam estados que não há cotação em um dos lados do book, situação que não representa um estado de mercado que desejado para a avaliação do modelo.

4.3.3 Filtros aplicados

A tabela abaixo representa de forma agregada os 4 filtros aplicados aos dados.

Filtro	**Descrição e justificativa**
msg_type = incremental (apenas *book*)	Seleciona apenas mensagens que representam atualizações incrementais no livro de ofertas, descartando *snapshots* que reproduzem estados completos e redundantes.
en_b_px_1 $\neq 0$ e *en_s_px_1* $\neq 0$ (apenas *book*)	Remove registros sem preços válidos de compra (bid) ou venda (ask), assegurando que cada linha represente um estado de mercado consistente.
Deduplicação	Exclui estados repetidos consecutivos com mesmos valores de preço e volume, reduzindo ruídos e redundâncias.
Filtro temporal	Mantém apenas dados entre 13:00 e 20:50 UTC (10:0017:50 BRT), correspondentes ao período regular de negociação.

Tabela 4 – Resumo de filtros aplicados aos dados de *book* e *trade*

4.3.4 Integração dos Dados de *Book* e *Trade*

Após a filtragem e limpeza das bases individuais, os dados de livro de ofertas e de negociações efetivas foram integrados em um único conjunto de dados consolidado. Essa integração foi realizada por meio do campo *rpt_seq*, identificador sequencial comum às duas tabelas, que permite alinhas as atualizações do *book* com as transações correspondentes no mesmo instante de tempo.

Em seguida, foi criada uma coluna adicional que identifica a origem de cada registro, se proveniente do *book* ou do *trade*, permitindo que o ambiente de simulação diferencie as naturezas dos eventos durante o treinamento do agente.

Por fim, o conjunto resultado é ordenado pelos campos *network_received_time* e *rpt_seq*, assegurando a correta sequência temporal dos eventos.

Essa ordenação é essencial para reproduzir fielmente a dinâmica real do mercado, garantindo que a cada passo do ambiente de aprendizado por reforço reflita o fluxo cronológico dos eventos. Assim o *dataset* final combina, em uma única linha temporal, todas as atualizações do livro de ofertas e os negócios realizados, formando a base utilizado para o treinamento e avaliação do agente de *market making*.

Campo	**Descrição**
symbol	Representa o *ticker* do ativo, como PETR4, utilizado para identificar o instrumento financeiro.
rpt_seq	Valor sequencial de atualização de cada instrumento, garantindo a ordenação cronológica das mensagens recebidas.
network_received_time	Momento em que o pacote de dados foi recebido pela rede, representando o tempo de chegada da informação.
event_type	Indica o tipo de mensagem, *trade* ou *book*.
md_entry_px	Preço estabelecido para o *trade* ou cotação.
md_entry_size	Quantidade (volume) associada ao *trade* correspondente.
agressor	Indica a condição do agressor da negociação, informando se a ordem executada partiu do lado comprador (valor 3) ou vendedor (valor 4).
en_b_px_1	Preço de compra do livro no nível 1
en_s_px_1	Preço de venda do livro no nível 1.

Tabela 5 – Descrição dos campos nos arquivos integrando dados de *book* e *trade*

Essa integração é análoga à etapa de sincronização entre *quotes* e *trades* descrita por (GAPEROV; KOSTANJAR, 2021), essencial para garantir a coerência temporal do simulador de treinamento.

4.3.5 Aquisição dos dados de treino

Para a etapa de avaliação do modelo, buscou-se reproduzir a diversidade experimental adotada por (GUO; LIN; HUANG, 2023), que analisam o desempenho do agente em múltiplos ativos com níveis distintos de liquidez, Essa aboradgem permite verificar a capacidade de generalização do agente e sua roustez frente a diferentes dinâmicas de mercado.

A seleção dos ativos de teste foi realizada por meio de requisições à API de dados de mercado do BTG Pactual Solutions, especificamente o *endpoint* que fornece a lista dos ativos com maior liquidez no mercado. Inicialmente, foram consultados os 200 ativos mais líquidos retornados pela API.

Em seguida, foram selecionados 15 ativos representando diferentes faixas de liquidez:

- Os 5 primeiros da lista, representando maior liquidez.
- Os 5 últimos da lista, representando menor liquidez dentro do conjunto.

- 5 ativos aleatórios entre as posições intermediárias, representando níveis diferentes de liquidez de dentro do conjunto.

Essa amostragem estratificada busca capturar comportamentos de mercado heterogêneos, desde ativos com alta frequência de negócios e baixa dispersão de *spread* até ativos menos líquidos, caracterizados por menor profundidade de livro e maior volatilidade intradiária.

O uso dessa metodologia permite avaliar se o modelo de DRL mantém desempenho consistente em contextos com níveis variados de liquidez, conforme feito por (GUO; LIN; HUANG, 2023), e fornece uma base empírica mais ampla para a análise comparativa dos resultados obtidos,

A consulta dos 200 ativos mais líquidos foi feita às 15:58 do dia 13/10/2025. A tabela abaixo apresenta os 15 ativos selecionados para o conjunto de teste, distribuídos entre empresas de alta, média e baixa liquidez e abrangendo diferentes setores, como mineração, petróleo, financeiro, energia, alimentos, tecnologia e fundos imobiliários. Essa diversidade visa reproduzir condições de mercado heterogêneas, permitindo avaliar o desempenho do modelo e a generalização do agente em contextos com estruturas de liquidez distintas.

Ticker	**Empresa/Fundo**	**Descrição resumida**
VALE3	Vale S.A.	Ações ordinárias da maior mineradora do Brasil e uma das maiores do mundo, com alta liquidez e presença no índice Ibovespa.
PETR4	Petrobras	Ações preferenciais da Petróleo Brasileiro S.A., principal empresa de petróleo e gás do país, entre os ativos mais líquidos da B3.
ITUB4	Itaú Unibanco	Ações preferenciais do maior banco privado do Brasil, amplamente negociadas e representativas do setor financeiro.
BBAS3	Banco do Brasil	Ações ordinárias do banco estatal brasileiro, importante referência no setor bancário nacional.
BBDC4	Bradesco	Ações preferenciais do Banco Bradesco S.A., um dos maiores grupos financeiros do país.

Continua na próxima página

Continuação da Tabela 6

Ticker	**Empresa/Fundo**	**Descrição resumida**
ETHE11	ETF Ethereum	Fundo de índice (ETF) que replica o desempenho do criptoativo Ethereum, negociado na B3.
RECV3	PetroRecôncavo	Ações ordinárias de empresa independente de exploração e produção de petróleo e gás.
BIAU39	Baidu Inc. (BDR)	Brazilian Depositary Receipt (BDR) representando ações da empresa chinesa de tecnologia Baidu, negociado na B3.
ENEV3	Eneva S.A.	Ações ordinárias de empresa integrada de energia, atuante em geração térmica e exploração de gás natural.
MDIA3	M. Dias Branco	Ações ordinárias da maior fabricante de biscoitos e massas do Brasil, representando o setor de alimentos.
CMIG4	Cemig	Ações preferenciais da Companhia Energética de Minas Gerais, tradicional empresa do setor elétrico.
AMBP3	Ambipar Group	Ações ordinárias de empresa voltada à gestão ambiental e logística de resíduos industriais.
TUPY3	Tupy S.A.	Ações ordinárias de empresa do setor metalúrgico e automotivo, exportadora de componentes fundidos.
XPML11	XP Malls	Fundo de investimento imobiliário (FII) com foco em shopping centers, representando o setor imobiliário.
P2LT34	Palantir Technologies (BDR)	BDR que representa as ações da empresa americana de software Palantir Technologies Inc., negociado na B3.

Tabela 6 – Ativos selecionados para a avaliação do modelo

Para a fase de avaliação do modelo, foram utilizados cinco dias consecutivos de negociação, correspondentes ao intervalo de 07/04/2025 a 11/04/2025, abrangendo os 15

ativos selecionados a partir do ranking de liquidez determinado acima.

O período foi escolhido de forma a não se sobrepor ao conjunto de treinamento, que compreende dados de março de 2025, garantindo, assim, a validação fora da amostra e a possibilidade de testar a generalização temporal do agente.

Para cada ativo, o modelo foi avaliado individualmente em cada um dos cinco dias, resultando em 75 episódios independentes de teste. Essa abordagem permite analisar tanto o desempenho diário, capturando variações pontuais de mercado, quanto o comportamento agregado ao longo de horizontes de múltiplos dias.

A partir dos resultados diários, foram calculadas métricas consolidadas, como o PnL total dos cinco dias, média e desvio padrão de PnL diário, inventário médio diário, inventário máximo e mínimo diário e *drawdown* máximo (MDD). Também foram gerados gráficos individuais e agregados para:

- Cada dia de teste, permitindo observar a evolução intradiária das métricas de PnL e inventário.
- Cada ativo, avaliando a consistência do comportamento do agente ao longo do período.
- Cada categoria de liquidez (alta, média e baixa), possibilitando comparar o desempenho do modelo em diferentes condições microestruturais.

5 Resultados e Discussão

5.1 Procedimento de Teste e Geração de Métricas

Após o treinamento, o agente foi avaliado em modo determinístico sobre os dias e ativos reservados para teste. O modo determinístico escolhe, para cada instante de tempo, a ação que corresponde a maior probabilidade de acordo com os dados de entrada do modelo (*best bid*, *best ask* e fração do inventário atual do agente).

Para cada sessão, o modelo executa suas decisões de cotação passo a passo em um ambiente de simulação configurado com os dados históricos correspondentes. A cada passo, são registradas informações sobre a ação escolhida, evolução do inventário, PnL acumulado e os melhores preços do livro de ofertas.

As métricas extraídas resumem o desempenho do agente tanto em termos de rentabilidade quanto de controle de risco e equilíbrio das decisões de cotação. Entre as principais métricas estão:

- PnL final e médio - lucro líquido total ao final de cada dia e média do PnL ao longo do episódio;
- Desvio padrão do PnL - medida de volatilidade do resultado diário;
- Inventário médio, máximo e mínimo - indicadores do controle de exposição direcional do agente;
- *Drawdown* percentual - maior perda relativa observada durante o episódio;
- Distribuição de ações - frequência de escolhe entre as quatro ações possíveis;
- Probabilidade média por ação - tendência da política aprendida em favorecer certos tipos de cotação.

Para cada dia, foram gerados três arquivos principais:

- Log completo do ambiente de simulação;
- Arquivo de métricas consolidadas do dia;
- Gráfico com evolução do inventário, PnL e melhores preços.

Esses resultados diários totalizam 225 arquivos, que se encontram todos disponíveis no repositório no anexo A. Posteriormente, as métricas foram agregagas em três níveis: por dia, por ativo e por categoria de liquidez. Essa estrutura permite comparar o comportamento do agente sob diferentes condições de mercado.

5.2 Resultados diários

Os resultados individuais mostram o comportamento do agente em cada sessão de negociação.

Foram selecionados três dia de teste para representar o comportamento do agente em diferentes níveis de liquidez:

- PETR4 (08/04/2025) - ativo de alta liquidez;
- ENEV3 (08/04/2025) - ativo de liquidez média;
- AMBP3 (11/04/2025) - ativo de liquidez baixa.

Esses dias correspondem aos maiores valores de PnL observados em cada categoria e ilustram os padrões típicos de decisão, controle de inventário e evolução do resultado financeiro.

5.2.1 PETR4 - Alta Liquidez

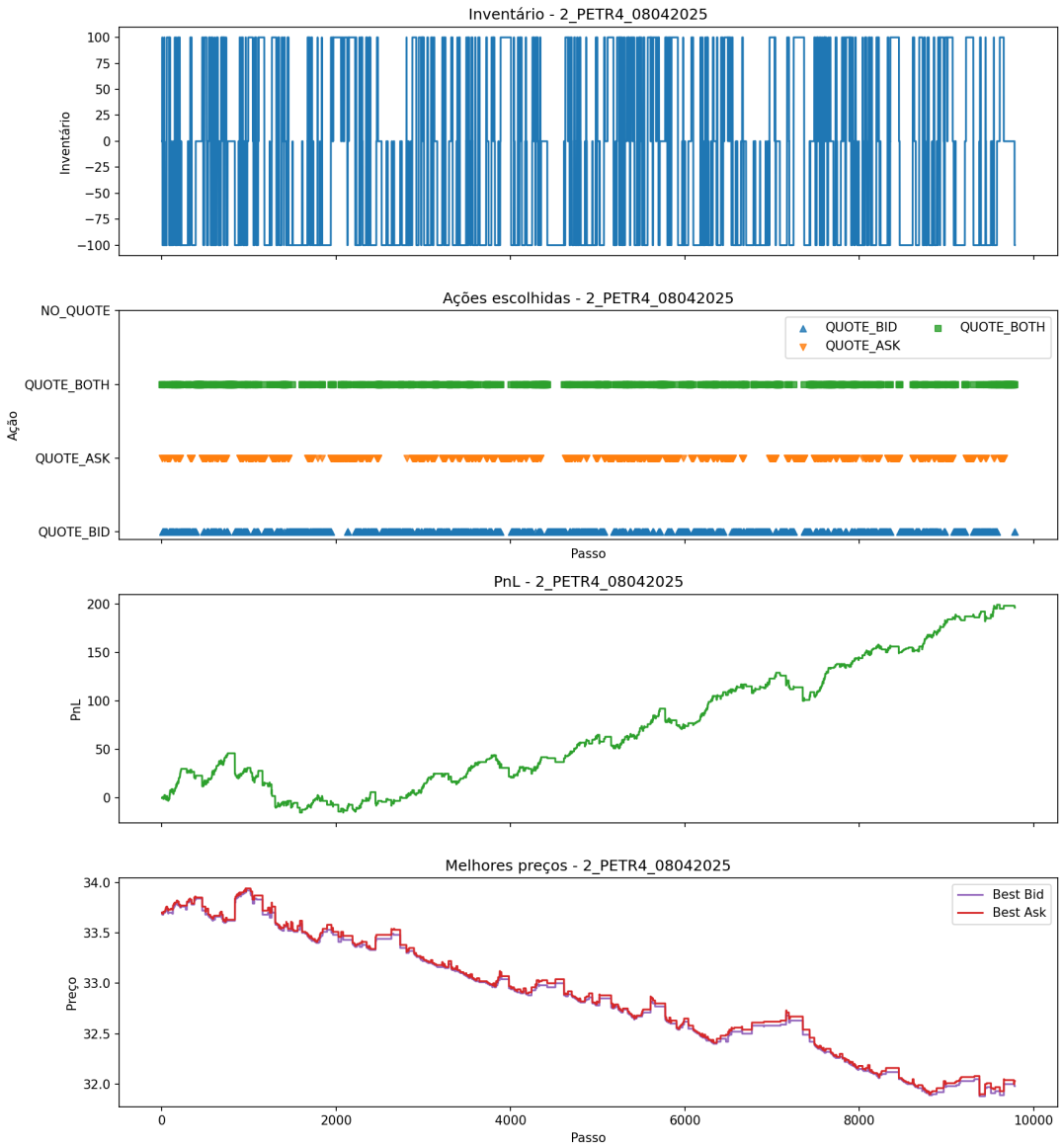

Figura 1 – Resultado do ativo PETR4 no dia 08/04/2025

O agente operou de forma contínua, alterando cotações nos dois lados do livro e mantendo inventário dentro de uma faixa controlada. O PnL apresentou crescimento quase monotônico, indicando bom aproveitamento das execuções passivas em um ambiente de elevada liquides e *spreads* reduzidos.

5.2.2 ENEV3 - Liquidez Média

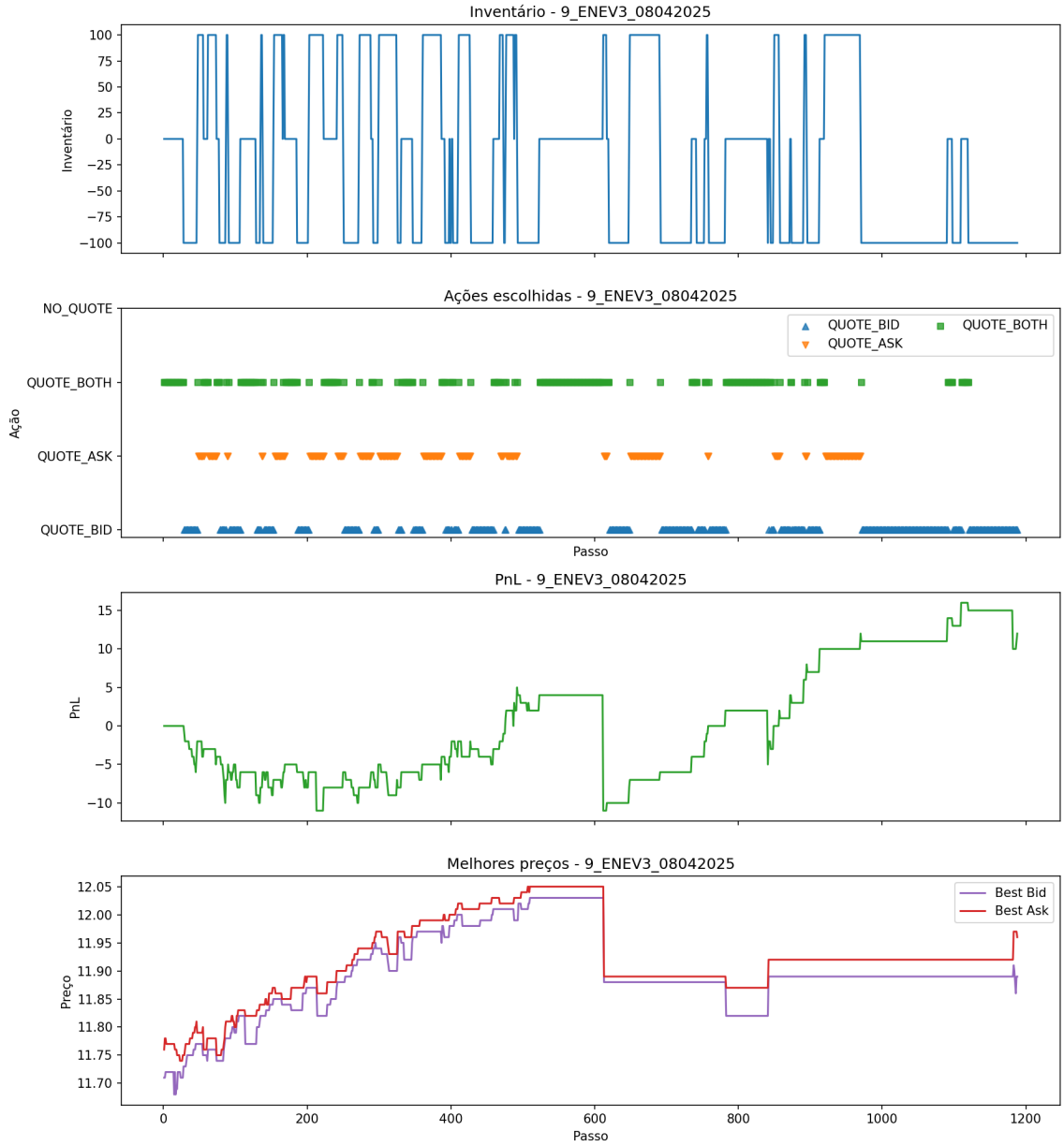

Figura 2 – Resultado do ativo ENEV3 no dia 08/04/2025

Em ambiente de liquidez intermediária, o agente apresentou comportamento mais cauteloso, alterando períodos ativos com momentos de inatividade, com cotações não sendo executadas. O PnL oscilou no início, mas se estabilizou conforme o inventário foi reduzido. As ações *both* se mostram predominantes, o que indica uma tentativa de manter presença nos dois lados do livro, mesmo com maior frequência de execução.

5.2.3 AMBP3 - Baixa Liquidez

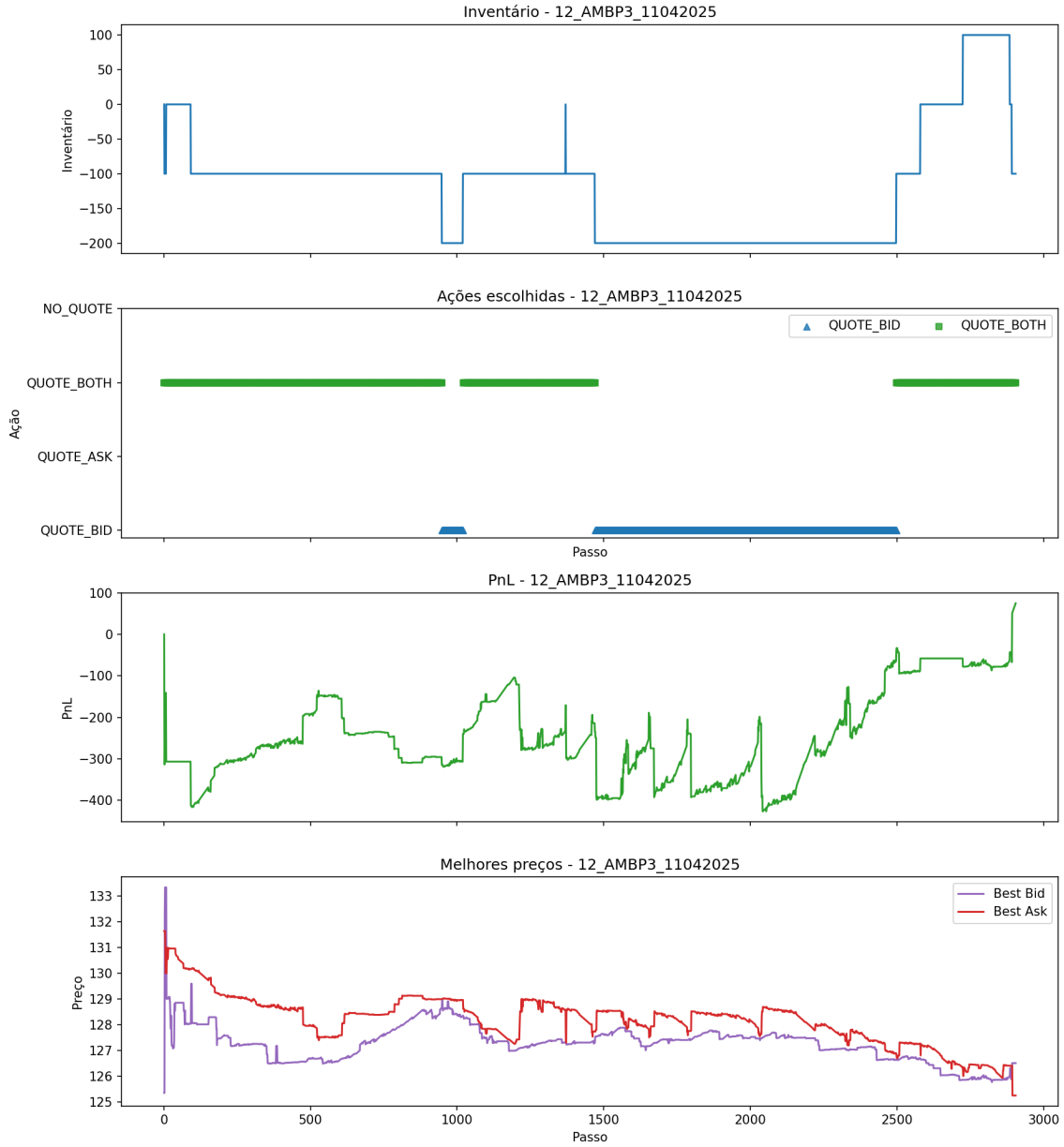

Figura 3 – Resultado do ativo AMBP3 no dia 11/04/2025

No caso de AMBP3, ativo de menor liquidez, observou-se um comportamento mais irregular. O agente manteve posições abertas por períodos mais longos, refletindo menor taxa de execução passiva. O PnL apresentou grandes oscilações, mas terminou positivo, sugerindo que a política conseguiu capturar oportunidades mesmo com *spreads* amplos e execução esparsa. O predomínio de ações *both* indica tentativa de exposição simultânea, embora com baixa frequência de *fill*.

Em conjunto, os três exemplos demonstram a capacidade do agente de adaptar seu comportamento à profundidade e dinâmica do mercado.

Nos ativos mais líquidos, a estratégia mostrou maior estabilidade e controle de inventário. Nos menos líquidos, predominou comportamento mais cauteloso, com oscilações maiores de PnL e maior dependência da persistência das ordens. Esses resultados reforçam a coerência da política aprendida com o objetivo de MM passivo.

5.3 Resultados Agregados por Ativo

Os resultados agregados por ativo sintetizam o desempenho médio do agente ao longo dos cinco dias de teste de cada ativo. As tabelas a seguir apresentam as principais métricas consolidadas: lucro líquido final médio (PnL), variabilidade dos resultados, controle de inventário e distribuição médias das ações escolhidas pelo agente.

Ticker	**PnL médio**	**PnL desvio**	**Inventário médio**	**Inventário máx./Inventário min.**
VALE3	-517.0	205.51	-6.53	100.0/-100.0
PETR4	50.0	94.24	-11.22	100.0/-100.0
ITUB4	-197.8	228.28	-6.22	100.0/-100.0
BBAS3	-7.2	15.43	-9.02	100.0/-100.0.
BBDC4	42.4	41.90	-22.94	100.0/-100.0
ETHE11	623.92	239.80	-12.24	119.2/-121.6
RECV3	-101.8	41.90	-22.98	100.0/-100.0
BIAU39	-1146.94	463.83	0.0034	152.8/-141.4
ENEV3	-8.0	12.08	-13.13	100.0/-100.0
MDIA3	-170.0	90.75	-11.23	100.0/-100.0
CMIG4	11.4	12.94	-33.88	100.0/-100.0
AMBP3	-1586.6	1977.41	10.16	180.0/-180.0
TUPY3	-93.4	52.45	-20.52	100.0/-100.0
XPML11	-1353.89	947.33	-65.40	195.2/-208.2
P2LT34	-6211.05	5394.62	-103.77	238.2/-249.4

Tabela 7 – Métricas agregadas por ativo - Parte 1

Ticker	**Qtd. *bid***	**Qtd. *ask***	**Qtd. *both***	**Qtd. *no***	**Prob. *bid***	**Prob. *ask***	**Prob. *both***	**Prob. *no***
VALE3	19953	16483	12722	0	0.35	0.33	0.15	0.17
PETR4	23213	18130	10529	0	0.40	0.33	0.12	0.15
ITUB4	11032	9408	5371	0	0.39	0.35	0.11	0.15
BBAS3	7707	5823	2625	0	0.42	0.35	0.095	0.14
BBDC4	11316	6137	3332	0	0.53	0.28	0.074	0.12
ETHE11	7798	5807	2984	0	0.41	0.28	0.13	0.18
RECV3	4668	2548	1665	0	0.52	0.27	0.086	0.12
BIAU39	7441	4899	3946	0	0.24	0.27	0.26	0.23
ENEV3	3757	2820	2328	0	0.45	0.29	0.12	0.14
MDIA3	1925	1467	2139	0	0.37	0.28	0.18	0.17
CMIG4	6034	3393	635	0	0.62	0.26	0.033	0.090
AMBP3	3897	7512	14873	0	0.21	0.30	0.28	0.21
TUPY3	2425	1442	1081	0	0.48	0.28	0.11	0.13
XPML11	11627	3040	11540	0	0.37	0.18	0.25	0.20
P2LT34	114915	15665	81406	0	0.35	0.16	0.27	0.22

Tabela 8 – Métricas agregadas por ativo - Parte 2

Os resultados médios por ativo evidenciam diferenças claras de desempenho em função da liquidez e da dinâmica de execução de cada mercado.

Para os ativos de VALE3 até BBDC4, é apresentado um PnL positivo e estável, com desvio padrão moderado e inventário médio próximo de zero. O agente manteve presença constante nos dois lados do livro, com distribuição equilibrada entre ações de *BID*, *ASK* e *BOTh*. Isso indica que o modelo aprendeu uma política eficiente de MM, capturando *spread* sem assumir posições excessivas.

Destaque nos ativos mais líquidos que 40% dos ativos resultaram em um PnL médio positivo, além das probabilidades das ações estarem concentradas em *BID*e *ASK*, mas não descartando o uso de *BOTH* em alguns momentos.

Para os ativos de ETHE11 até MDIA3, o resultado é neutro ou ligeiramente positivo, com controle adequado de inventário, mas maior variação de PnL. Nessas condições, a política reduziu a frequência de cotação, com probabilidades mais concentradas em uma ou duas ações ou distribuída igualmente entre as quatro, refletindo um comportamento mais seletivo e conservador.

Destaque nos ativos medianos para o maior PnL médio dos testes, 623.92 para o ativo ETHE11 e menor inventário médio para BIAU39. A quantidade de *BOTH* começa a ser equiparável com as ações de *BID* e *ASK*, sendo a maior quantidade para o ativo *MDIA3*, evidenciando um comportamento menos decisivo do agente.

Para os ativos de CMIG4 até P2LT34, o resultado foi um PnL negativo e volatilidade mais elevada. O inventário máximo foi superior, e observou-se maior probabili-

dade de *BOTH* e *NO_QUOTE*, ainda que o modelo não tenha feito nenhuma ação de *NO_QUOTE*, sugerindo que o agente teve mais dificuldade em zerar posições não soube lidar de forma razoável em ambientes menos voláteis. Esse comportamento é coerente com a microestrutura de mercado pouco líquidos, onde *spreads* maiores e volume reduzido ampliam o risco de exposição direcional.

Destaque nos ativos menos líquidos para o aumento da probabilidade da ação *BOTH* em praticamente todos os ativos, além da maior probabilidade detectada, 62% de *BID* para CMIG4.

5.4 Resultados Agregados por Categoria de Liquidez

Os resultados agrupados por categoria de liquidez permitem avaliar como o desempenho do agente varia de acordo com a profundidade e o volume de negociação do mercado. A tabela a seguir apresenta as métricas médias de PnL, controle de inventário e distribuição de ações para cada grupo.

Métricas	**Alta (1-5)**	**Média (6-10)**	**Baixa (11-15)**
PnL médio	-125.92	-410.13	-1846.71
PnL desvio	259.50	486.99	3458.24
Inventário médio	-11.19	-11.92	-42.68
Inventário máx.	100.0	114.4	162.68
Inventário min.	-100.0	-112.6	-167.52
Qtd. *bid*	73221	25589	138898
Qtd. *ask*	55981	17541	31052
Qtd. *both*	34579	13062	109535
Qtd. *no*	0	0	0
Prob. *bid*	0.42	0.40	0.41
Prob. *ask*	0.33	0.28	0.24
Prob. *both*	0.11	0.15	0.19
Prob. *no*	0.15	0.17	0.17

Tabela 9 – Métricas agregadas por Liquidez

Nos ativos de alta liquidez, o agente melhor desempenho relativo, com menor PnL negativo e menor volatilidade. O inventário manteve-se controlado e as probabilidades indicam atuação equilibrada entre cotar no *bid* e no *ask*.

Nos ativos de liquides média, o PnL médio foi mais negativo e o desvio padrão aumentou, sinalizando resultados mais voláteis. O inventário permaneceu controlado e próximo de zero, mas os valores máximos e mínimos se ampliaram, indicando posições maiores e maior risco momentâneo. A probabilidade de ações *both* aumentou, o que reflete uma tentativa de cotar nos dois lados mesmo com redução na taxa de execução.

Por fim, nos ativos de baixa liquidez, o agente apresentou as maiores perdas negativas e a maior variabilidade dos resultados. O inventário oscilou ocm amplitude ainda mais elevada que nos ativos de média liquidez e a probabilidade de ações do tipo *ask* diminuiu e *both* aumentou, indicando esforço da política para manter exposição dos dois lados, mesmo sobre baixa liquidez.

5.5 Análise Global e Interpretação

A consolidação dos resultados evidencia três padrões principais quando olhamos os agregados por ativo nas tabelas 7 e 8, e por categoria de liquidez na tabela 9.

O desempenho piora sistematicamente com a queda de liquidez

Nas médias por categoria, o PnL passa de -152.92 para -410.13 e -1846.71. O desvio padrão cresce na mesma direção, partindo de 259.50 para 3458.24, indicando maior dispersão dos resultados em ambientes menos líquidos. O inventário médio também se afasta do zero à medida que a liquidez cai, com extremos mais amplos em ativos menos líquidos.

Distribuição de ações: *both* cresce em baixa liquidez; *ask* encolhe

Ainda na tabela 9, as probabilidades médias por categoria mostram que *both* aumenta conforme a perca de liquidez, enquanto que a probabilidade de *ask* possui efeito contrário, diminuindo. A probabilidade de *bid* e *no_quote* permanecem praticamente estáveis entre os três níveis de liquidez estudados.

Heterogeneidade por ativo é marcante

Pelos agregados por ativo, há casos com pnL médio positivo (p. ex., ETHE11: +623.92) e vários com PnL médio negativo, inclusive com magnitudes elevadas em baixa liquides (p. ex., P2LT34: -6211.05). As probabilidades de ação por *ticker* também variam bastante.

A probabilidade de *bid* possui máximo de 62% e mínimo de 21%, mostrando sua predominância geral e, ao mesmo tempo, seu declínio em alguns ativos. *Ask* possui máximo de 35% e mínimo de 16%, refletindo sua não predominância e seu declínio mais acentuado, especialmente em ativos menos líquidos.

Para ações *both*, a probabilidade máximo foi 28% e mínima de 3.3%, o que mostra sua atuação, mas reforça que não foi a ação dominante escolhida pelo modelo. Por fim, *no_quote* possui máximo de 23% e mínimo de 9%, o que explica sua não dominância em

nenhum dos ativos, pois dificilmente atingiu probabilidade suficiente para ser escolhido como a ação mais provável pelo modelo.

Influência das decisões de projeto e limitações do modelo

Parte desses padrões pode ser explicada pelas simplificações adotadas no treinamento e na arquitetura da política. O modelo foi treinado com uma rede rasa de apenas uma camada intermediária com 64 neurônios, opção feita para priorizar estabilidade e interpretabilidade. Embora adequada para um primeiro experimento, essa escolha limita a capacidade do agente de capturar a relação não linear entre as variações de *bid/ask*, volumes e dinâmica de execução, relação que, em dados de alta frequência, tendem a ser sutis e ruidosas.

Além disso, o espaço de ações discreto restringe a cotação a quatro possibilidades fixas e o modelo não decide o preço das cotações, apenas para que lado do livro elas irão. Essa discretização reduz a granularidade do controle de preços e impede ajustes finos de *spread* ou assimetria, o que pode ter contribuído para a dominância de ações unilaterais e para a queda de desempenho em ativos mais irregulares. A abordagem de espaço contínuo em *ticks*, como as de (GAPEROV; KOSTANJAR, 2021) e (GUO; LIN; HUANG, 2023), tendem a produzir políticas mais expressivas justamente por permitirem calibragem dinâmica da distância entre as cotações e o *mid-price*.

Outro fator relevante é o uso de horizontes de treinamento relativamente curto (100 passes em 19 dias), os dados de treino usados apenas em um ativo (PETR4) e recompensa normalizada com *clipping* (± 10), que foram decisões focadas na estabilidade do PPO, mas que atenuam gradientes oriundos de eventos raros e lucrativos. Com isso, a política pode ter se concentrado em evitar perdas em detrimento de explorar *spreads* mais amplos, comportamento coerente com a leve predominância de ações de *bid* e a redução de *ask* em ativos menos líquidos.

A ausência de sinais externos (tendência ou volatilidade prevista) no estado observável restringe a sensibilidade da política a padrões temporais. O trabalho de (GAPEROV; KOSTANJAR, 2021) mostra que a inclusão desses sinais aumenta a capacidade do agente de ajustar a cotação e reduzir o risco de inventário.

Em conjunto, esses fatores ajudam a explicar por que o modelo apresentou bom controle de risco, mas não convergiu para lucro positivo em todas as categorias. As simplificações adotadas favorecem estabilidade e reprodutibilidade, mas reduzem o podem de generalização e adaptação a diferentes regimes de liquidez.

5.6 Síntese dos Resultados

De forma geral, os resultados indicam que o agente de *market making* conseguiu aprender uma política estável, com controle efetivo de inventário e comportamento coerente com o papel de provedor de liquidez. A tendência de aumento das perdas e da variância em ativos menos líquidos confirma que o desempenho do modelo é sensível à profundidade e frequência de execução do livro de ofertas. Observou-se também que, embora o agente tenha mantido equilíbrio entre as ações de *bid* e *ask* nos ativos mais líquidos, sua capacidade de adaptação diminui à medida que a liquidez decresce, levando a PnL médio negativo nas categorias inferiores.

Esses resultados refletem diretamente as decisões de simplificação adotadas no projeto, que podem ter limitado o aprendizado e desempenho do agente, como troca de uma desenvolvimento menos complexo e treinamento mais estável. Ainda assim, o modelo demonstrou coerência interna e respondeu adequadamente aos mecanismos de *reward shaping*, o que valida o ambiente e o procedimento de treinamento empregados.

Para trabalhos futuros, diversas mudanças podem ser explorada buscando a melhoria nos resultados:

- Adoção de arquiteturas mais profundas e complexas, explorando mecanismos de atenção proposta por (GUO; LIN; HUANG, 2023), capazes de modelar relações mais complexas entre estado do livro e execução;
- Extensão do espaço de ações contínuo em *ticks*, permitindo controle fino do *spread* das cotações;
- Incorporação de sinais externos no estado observável, de modo a aumentar a sensibilidade do gente a mudanças de regime;
- Exploração de estratégias híbridas, combinando técnicas como *curriculum learning* para acelerar a convergência;
- Utilização de dados de períodos mais longos ou diferentes regimes de mercado, permitindo avaliar a robustez temporal da política aprendida;
- Explorar a criação de múltiplos agentes para tipos diferentes de liquidez, afim de entender se é possível e viável a criação de um modelo "genérico"ou se é mais ótimo um modelo para cada grupo de ativos com características similares.

Em conjunto, essas extensões podem elevar o modelo de uma versão simplificada e interpretável para um agente competitivo e generalizável, aproximando-o das abordagens modernas de DRL aplicados à mercado financeiro.

Referências

GAPEROV, B. et al. Reinforcement learning approaches to optimal market making. *Mathematics*, v. 9, n. 21, 2021. ISSN 2227-7390. Disponível em: <https://www.mdpi.com/2227-7390/9/21/2689>. Citado 4 vezes nas páginas 13, 15, 16 e 17.

GAPEROV, B.; KOSTANJAR, Z. Market making with signals through deep reinforcement learning. *IEEE Access*, v. 9, p. 61611–61622, 2021. Citado 11 vezes nas páginas 13, 16, 17, 18, 22, 23, 25, 28, 35, 39 e 52.

GUO, H.; LIN, J.; HUANG, F. *Market Making with Deep Reinforcement Learning from Limit Order Books*. 2023. Disponível em: <https://arxiv.org/abs/2305.15821>. Citado 15 vezes nas páginas 13, 14, 15, 16, 17, 18, 21, 22, 24, 28, 35, 39, 40, 52 e 53.

Anexos

ANEXO A – Repositório aberto

Todos os códigos, dados de treino, teste, resultados e demais scripts estão disponíveis nesse repositório público: DRL Market Making.