

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE MINAS GERAIS (IFMG)
CAMPUS BAMBUÍ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Higor Pereira Silva

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA *BACKTESTING* DE
ESTRATÉGIAS DE INVESTIMENTO BASEADAS EM ORDENAÇÃO DE ATIVOS**

BambuÍ - MG
2025

HIGOR PEREIRA SILVA

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA *BACKTESTING* DE
ESTRATÉGIAS DE INVESTIMENTO BASEADAS EM ORDENAÇÃO DE ATIVOS**

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus* Bambuí para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Marcos Roberto Ribeiro

Catálogo na Fonte Biblioteca IFMG - *Campus Bambuí*

S586d Silva, Higor Pereira

Desenvolvimento de uma ferramenta para backtesting de estratégias de investimento baseadas em ordenação de ativos [manuscrito] / Higor Pereira Silva. – 2025.

55 f. : il.

Orientador: Marcos Roberto Ribeiro.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais. *Campus Bambuí*, 2025.

1. Backtesting. 2. Mercado financeiro. 3. Análise técnica. 4. Ordenação de ativos. I. Ribeiro, Marcos Roberto. II. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus Bambuí*. III. Título.

CDD 005.636

Catálogo: João Batista Rodrigues - CRB-6/2022

Higor Pereira Silva

DESENVOLVIMENTO DE UMA FERRAMENTA PARA *BACKTESTING* DE ESTRATÉGIAS DE INVESTIMENTO BASEADAS EM ORDENAÇÃO DE ATIVOS

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus* Bambuí para obtenção do grau de Bacharel em Engenharia de Computação.

Aprovado em 10 de Fevereiro de 2025 pela banca examinadora:

Prof. Dr. Marcos Roberto Ribeiro – IFMG – *Campus* Bambuí – (Orientador)

Prof. Me. Claudio Ribeiro de Sousa – IFMG - *Campus* Bambuí

Prof. Natalia Camillo do Carmo – IFMG - *Campus* Bambuí



Documento assinado eletronicamente por **Marcos Roberto Ribeiro, Professor**, em 10/02/2025, às 21:09, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Claudio Ribeiro de Sousa, Professor EBTT**, em 10/02/2025, às 21:09, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Natalia Camillo do Carmo, Professora Substituta**, em 10/02/2025, às 21:09, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2195435** e o código CRC **4307A87B**.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Cíntia e Patrik, e à minha avó, Cleli, que sempre me apoiaram em minhas decisões. Agradeço também à minha companheira, Gabriela, que suportou meu mau humor.

Agradeço a todos os meus professores, que contribuíram imensuravelmente para o meu desenvolvimento. Em especial, agradeço ao Marcos Roberto, que teve muita paciência comigo nesta etapa de finalização do curso.

“O importante é não parar de questionar. A curiosidade tem sua própria razão de existir. ”

Albert Einstein

RESUMO

O mercado financeiro atua como um canal para a alocação de recursos entre investidores, empresas e governos, oferecendo liquidez e auxiliando na determinação dos preços dos ativos. Estratégias de investimento, por sua vez, são métodos estruturados que orientam decisões de compra e venda de ativos, buscando otimizar retornos e minimizar riscos. Com os avanços tecnológicos, investidores individuais passaram a acessar ferramentas capazes de criar e executar estratégias complexas, antes restritas a grandes corporações. A eficácia dessas estratégias pode ser avaliada por meio de *backtesting*. Diante desse contexto, o presente trabalho apresenta o desenvolvimento de uma ferramenta para *backtesting* de estratégias de investimento, baseadas em ordenação de ativos, permitindo avaliar o desempenho em cenários variados e explorar diferentes combinações de parâmetros. A ferramenta manipula parâmetros, identifica configurações eficazes e avalia estratégias em listas de ativos ordenadas. Os experimentos realizados demonstraram a flexibilidade do sistema e sua capacidade de avaliação de estratégias de investimento. Os resultados evidenciaram o potencial da abordagem para capturar dinâmicas de mercado e validar hipóteses de forma clara e replicável, contribuindo para o uso de estratégias de investimento mais eficientes no mercado financeiro.

Palavras-chave: *Backtesting*. Mercado Financeiro. Análise técnica. Ordenação de ativos. Portfólio.

ABSTRACT

The financial market serves as a channel for resource allocation among investors, corporations, and governments, providing liquidity and aiding in asset price determination. Investment strategies, in turn, are structured methods that guide buy and sell decisions, aiming to optimize returns and minimize risks. With technological advancements, individual investors have gained access to tools capable of creating and executing complex strategies, previously exclusive to large corporations. Autonomous agents employ both fundamental and technical analysis to operate continuously while avoiding emotional biases. The effectiveness of these strategies is evaluated through backtesting. This study presents a functional architecture for backtesting strategies, already validated, enabling performance evaluation across varied scenarios and exploring different parameter combinations. The tool manipulates parameters, identifies effective configurations, and ranks strategies in specific asset lists. The experiments conducted demonstrated the system's flexibility and its ability to support informed decision-making. The results highlight the approach's potential to capture market dynamics and validate hypotheses clearly and reproducibly, contributing to the adoption of more efficient investment strategies in financial market.

Keywords: Backtesting. Financial Market. Fundamental Analysis. Technical Analysis. Investment Strategies.

LISTA DE FIGURAS

Figura 1 - Cruzamento de média móvel	19
Figura 2 - Solução	26
Figura 3 - Etapas de desenvolvimento	28
Figura 4 - Diagrama de classe	32
Figura 5 - Parâmetros de estratégia e ranqueador	34
Figura 6 - Instâncias de Runner	35
Figura 7 - Método rank() da MARanker	41
Figura 8 - Retorno get_history_interval()	42
Figura 9 - Simulação de portfólios S&P500	46
Figura 10 - Simulação de portfólios IBOV	47
Figura 11 - Simulação de portfólios IBRA	48

LISTA DE QUADROS

Quadro 1 - Trabalhos e ferramentas com <i>backtesting</i>	24
Quadro 2 - Ferramentas utilizadas	27
Quadro 3 - Especificações do equipamento	28
Quadro 4 - Informações dos ativos salvos em CSV	43

LISTA DE TABELAS

Tabela 1 - <i>Backlog</i> de produto	31
Tabela 2 - Parâmetros para combinação	34
Tabela 3 - Combinações	35
Tabela 4 - Portfólio inicial venda	37
Tabela 5 - Portfólio atualizado após vendas	37
Tabela 6 - Portfólio atualizado após compra	39
Tabela 7 - Lista de ordenação de ativos.	40
Tabela 8 - <i>DataFrame</i> de dados históricos	41
Tabela 9 - Parâmetros e variações testados na simulação	44
Tabela 10 - Resultados da validação no S&P 500	45
Tabela 11 - Resultados da validação no IBOV	46
Tabela 12 - Resultados da validação no IBrA	48

LISTA DE SIGLAS

- API – *Application Programming Interface*
- B3 – Brasil, Bolsa e Balcão
- CVM – Comissão de Valores Mobiliários
- CSV – *Comma Separated Values*
- EMA – *Exponential Moving Average*
- IBOV – Índice Bovespa
- IBrA – Índice Brasil Amplo
- IFR – Índice de força relativa
- JSON – *JavaScript Object Notation*
- MACD – *Moving Average Convergence and Divergence*
- RSI – *Relative Strength Index*
- SMA – *Simple Moving Average*
- S&P500 – Standard & Poor's 500

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	<i>Objetivos específicos</i>	14
1.2	Justificativa	14
2	FUNDAMENTOS TEÓRICOS	16
2.1	Mercado financeiro	16
2.2	Análise técnica	17
2.3	Portfólio e estratégias de investimento	19
2.4	<i>Backtesting</i>	20
2.5	Estado da arte	22
2.5.1	<i>Trabalhos acadêmicos</i>	22
2.5.2	<i>Ferramentas</i>	22
2.5.3	<i>Comparação dos trabalhos</i>	23
3	METODOLOGIA	25
3.1	Classificação da pesquisa	25
3.2	Solução	25
3.3	Materiais e tecnologias	26
3.3.1	<i>Tecnologias</i>	26
3.3.2	<i>Ambiente de desenvolvimento</i>	27
3.4	Métodos e procedimentos	28
4	DESENVOLVIMENTO	30
4.1	Levantamento de requisitos	30

4.2	Modelagem do sistema	32
4.3	Avaliação	33
4.4	Execução	36
4.5	Ordenação de ativos	39
4.6	Obtenção e manipulação dos dados	40
4.7	Validação	43
4.7.1	S&P 500	44
4.7.2	IBOV	45
4.7.3	IBrA	47
5	CONCLUSÃO	50
5.1	Trabalhos futuros	51
	REFERÊNCIAS	52

1 INTRODUÇÃO

O mercado financeiro é um sistema no qual se realizam negociações de ativos financeiros, como ações, títulos, moedas e derivativos. Ele facilita a transferência de recursos entre investidores, empresas e governos, proporcionando liquidez e ajudando a determinar os preços dos ativos, de forma a desempenhar um papel crucial na organização econômica e social dos países. Nas últimas décadas, houve uma reestruturação devido ao avanço tecnológico, possibilitando pessoas físicas e investidores individuais a participarem das opções de investimento, o que, antes, era disponível apenas para grandes empresas, gerando, assim, mais liquidez (ARAÚJO, 2021).

Dentre esses avanços tecnológicos, estão os agentes autônomos voltados ao mercado financeiro. Eles trabalham na análise de mercado, empregando técnicas e algoritmos de análise de ativos financeiros, de modo que não haja influência externa, como a emocional, além de permitir operações ininterruptas.

Estes agentes usam como base indicadores de mercado que se originam da análise técnica, a qual utiliza padrões e tendências de preços passados para prever movimentos futuros do mercado (NTI; ADEKOYA; WEYORI, 2020). Essa abordagem baseia-se no fato de que todas as informações relevantes já estão refletidas nos preços e que padrões tendem a se repetir. Algumas ferramentas, como gráficos e médias, são comumente utilizadas para identificar oportunidades no mercado.

Com base nessas análises, os ativos financeiros podem ser filtrados de acordo com seu potencial de desempenho. Este processo considera fatores como o nível de risco, expectativa de retorno, a força das tendências e a volatilidade histórica. Este processo torna-se essencial para a construção de um portfólio e a tomada de decisões (PEREIRA; OLIVEIRA, 2021).

Com base nos indicadores das análises, diversas estratégias podem ser implementadas para se negociar no mercado financeiro. Para avaliar uma estratégia, é interessante fazer simulações com dados históricos para verificar o desempenho ao longo do tempo. Esse processo é conhecido como *backtesting*, o qual, quando indica um bom desempenho, sugere uma estratégia interessante de investimento (ARNOTT; HARVEY; MARKOWITZ, 2018).

Além disso, ao se analisar um conjunto de estratégias, juntamente com dados históricos dos ativos do mercado financeiro e aplicar o *backtesting* para testá-las, torna-se possível implementar uma arquitetura flexível. Essa arquitetura permitirá, automaticamente, a variação de múltiplos parâmetros das estratégias, além de possibilitar a combinação desses parâmetros de diversas maneiras, proporcionando uma avaliação mais abrangente.

Na arquitetura proposta neste trabalho, o usuário desenvolveu seu próprio

algoritmo de ordenação (*ranking*) de ativos, enquanto a ferramenta realizou a avaliação das estratégias com base nos parâmetros configurados. O principal diferencial dessa arquitetura é sua capacidade de aceitar e manipular vários parâmetros, realizar variações e combinações entre eles e executar o *backtesting* nestas combinações, facilitando, assim, a identificação dos parâmetros mais eficazes.

1.1 Objetivos

O principal objetivo deste trabalho foi desenvolver uma ferramenta para *backtesting* de estratégias de investimento baseadas na ordenação de ativos, permitindo a avaliação e a comparação de diferentes abordagens no mercado financeiro.

1.1.1 Objetivos específicos

Para alcançar o objetivo principal, foram estabelecidos os seguintes objetivos específicos:

- estabelecer as funcionalidades da arquitetura de *backtesting* para avaliação de estratégias considerando-se o lucro;
- modelar e implementar as classes principais que compõem o sistema;
- validar a ferramenta desenvolvida por meio de experimentos em dados reais.

1.2 Justificativa

Segundo Martins (2023), o número de pessoas físicas no mercado financeiro aumentou consideravelmente entre os anos de 2018 e 2022. Isso ocorreu devido a diversos fatores, sendo o principal deles o avanço tecnológico. Em janeiro de 2018, havia cerca de 700 mil pessoas físicas investindo na bolsa de valores. Já em janeiro de 2022, esse número passou para mais de 5 milhões, representando um aumento percentual de 616%.

Considerando a complexidade do mercado financeiro e a diversidade de estratégias de investimento disponíveis, surge uma demanda crescente por ferramentas que facilitem a avaliação e comparação delas. O desenvolvimento de uma arquitetura de análise de estratégias e *backtesting* é fundamental para fornecer aos desenvolvedores uma plataforma para testar suas técnicas de investimento.

Estas ferramentas podem democratizar o acesso a análises sofisticadas de investimento que, anteriormente, estavam disponíveis apenas para grandes instituições. Ao permitir uma avaliação mais eficaz das estratégias, essas ferramentas também podem contribuir para uma melhor gestão do risco e otimização das carteiras

de investimento. Além disso, a evolução contínua das tecnologias financeiras e o volume crescente de dados de mercado exigem métodos avançados para processar e analisar informações, comprovando que a arquitetura de *backtesting* proposta fornece uma contribuição oportuna e valiosa para o campo.

O presente trabalho se destaca ao automatizar esse processo de comparação de múltiplas estratégias, o que permite aos investidores tomarem decisões mais informadas e rápidas, identificando as estratégias mais promissoras para implementação no mercado financeiro. Isso permite aos investidores uma análise mais abrangente e eficaz de suas estratégias, facilitando a tomada de decisões. A capacidade de avaliar diversas estratégias lado a lado e de identificar rapidamente quais são as mais promissoras, com base em métricas predefinidas, representa um avanço significativo em relação às ferramentas atualmente disponíveis.

2 FUNDAMENTOS TEÓRICOS

Este capítulo destaca os principais conceitos abordados no desenvolvimento deste trabalho. A Seção 2.1 engloba o mercado financeiro de renda variável, fornecendo uma introdução às suas características e funcionamento. A Seção 2.2 discute a análise técnica e seus indicadores. Em seguida, a Seção 2.3 apresenta portfólios e estratégias de investimento. Posteriormente, a Seção 2.4 explora a utilização do *backtesting* em conjunto com esses algoritmos, visando à maximização do lucro. Por fim, o capítulo encerra com a Seção 2.5, contendo a revisão do estado da arte, que contextualiza o trabalho em relação às mais recentes pesquisas e tecnologias no campo de estudo.

2.1 Mercado financeiro

O mercado financeiro é um ambiente dinâmico no qual ocorrem transações de diversos instrumentos financeiros, como ações, títulos, fundos imobiliários, moedas e *commodities*. Ele fornece o ambiente em que os participantes realizam operações de compra e venda, visando à formação de preços, alocação de recursos e gerenciamento de riscos.

Em mercados bem estruturados, as negociações são organizadas e regulamentadas, garantindo transparência, liquidez e eficiência no processo de precificação. Bolsas de valores e outras plataformas conectam investidores a oportunidades, assegurando transações seguras (SANTOS; SANTOS, 2005).

Os investimentos no mercado financeiro podem ser classificados em renda fixa e renda variável, conforme o grau de previsibilidade dos retornos. Instrumentos de renda fixa, como títulos públicos e privados, apresentam remuneração previamente definida. Por outro lado, ativos de renda variável, incluindo ações, fundos imobiliários e *commodities*, possuem retornos incertos, sujeitos às oscilações do mercado.

O investimento em renda variável está associado a um nível mais elevado de risco, pois os preços dos ativos são influenciados por fatores econômicos, políticos e comportamentais (FREITAS, 2020). A volatilidade inerente a esses ativos atrai investidores interessados em maiores retornos, embora represente, também, um risco significativo. Conforme discutido por Ferreira (2019) e Wainberg (2017), a oferta e a demanda exercem papel fundamental na determinação dos preços e dos resultados dos investimentos.

As bolsas de valores desempenham um papel essencial na negociação de ativos financeiros. Um exemplo é a Brasil, Bolsa, Balcão (B3), que organiza as transações financeiras no Brasil, proporcionando liquidez e garantindo transparência no processo de precificação (B3, 2021). No Brasil, a Comissão de Valores Mobiliá-

rios (CVM) é a entidade responsável pela supervisão e regulamentação do mercado financeiro. Vinculada ao Ministério da Fazenda, a CVM dispõe de autonomia financeira e administrativa, assegurando que as operações ocorram em conformidade com as normas e regulamentos estabelecidos (CVM, 2020). Esses elementos estruturais são fundamentais para o funcionamento eficiente do mercado financeiro, que deve se adaptar continuamente às transformações econômicas e tecnológicas.

2.2 Análise técnica

Um tipo de análise largamente utilizada no mercado financeiro é a análise técnica, que, por sua vez, é baseada na teoria de *Dow*. Pring (2014) define tal teoria como o método mais antigo para identificar tendências no mercado de ações, tendo como objetivo determinar mudanças na tendência principal. Ela surgiu a partir do trabalho de Charles Dow, que publicou diversos editoriais no *Wall Street Journal*. Os fundamentos da teoria de Dow, caracterizados por Lemos e Cardoso (2010), são os seguintes:

- a) o mercado possui três tendências (alta, baixa e lateral);
- b) o volume de negociação varia conforme a direção da tendência de mercado;
- c) tendências primárias de alta passam por três fases distintas de valorização;
- d) tendências primárias de baixa apresentam três fases distintas de desvalorização;
- e) as médias móveis incorporam todas as informações relevantes sobre os preços;
- f) a confirmação das médias móveis é essencial para validar as análises técnicas realizadas.

Com base nesses fundamentos, Lemos (2022) delinea três princípios fundamentais da análise técnica. O primeiro deles ressalta que o mercado é uma soma de todos os fatores relevantes, incluindo os psicológicos e políticos, refletidos no preço do ativo. Assim, a variação de preço é o foco principal da análise. O segundo princípio destaca que os preços seguem uma tendência ao longo do tempo, o que pode ser compreendido melhor por meio da representação gráfica dos preços.

Por último, o terceiro princípio sugere que a história se repete, pois os padrões comportamentais humanos tendem a permanecer consistentes ao longo do tempo, o que implica que estratégias baseadas em padrões passados podem ser eficazes no futuro. Esses princípios fornecem um arcabouço para interpretar e prever movimentos de mercado com base na análise técnica.

Conforme Goldberg e Nitzsch (2001), o cerne da análise técnica é reco-

nhecer as tendências, visto que os modelos mais eficazes são aqueles capazes de determinar o momento em que uma tendência se estabelece de forma sólida. Dito isso, há algumas formas de visualizar os dados de preço ao longo do tempo, permitindo aos investidores discernir padrões e tendências cruciais para tomadas de decisão. Edwards, Magee e Bassetti (2018) classificam algumas ferramentas que auxiliam na análise técnica, como gráficos, médias móveis e suas variações, linhas de suporte e resistência e alguns osciladores.

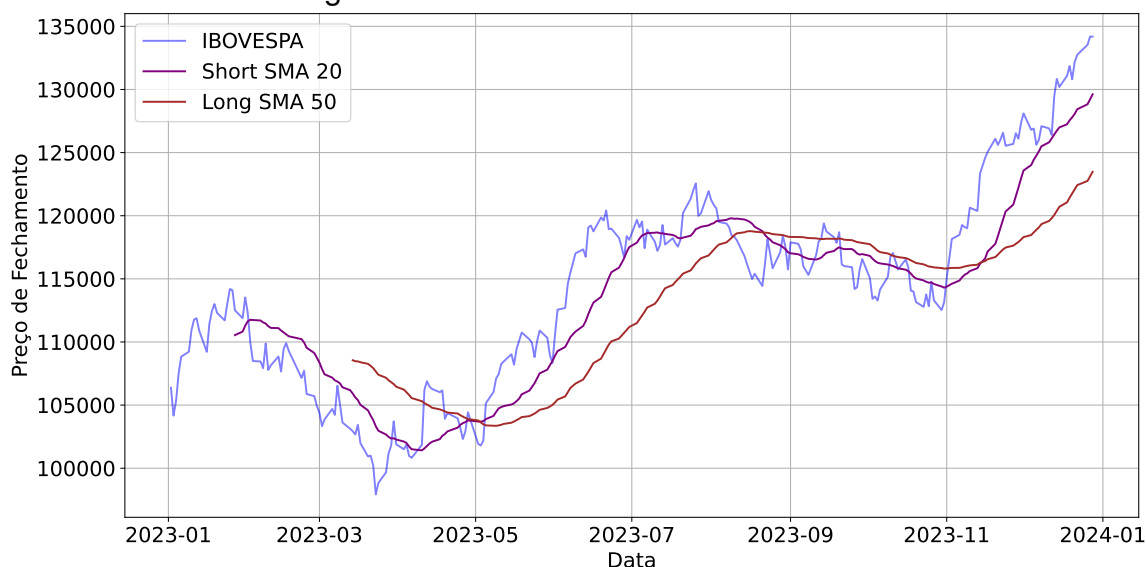
Além disso, a análise técnica pode ser utilizada como uma estratégia prática tanto para a seleção de ativos a serem adquiridos quanto para a identificação de momentos oportunos para venda. Por meio da identificação de tendências de alta ou de baixa, investidores podem estruturar decisões com base em movimentos projetados, maximizando ganhos ou minimizando perdas.

Para complementar essas ferramentas, indicadores técnicos desempenham um papel fundamental na análise técnica. Um indicador técnico é um conjunto de informações obtidas ao se realizar cálculos matemáticos nos valores de ativos, utilizado na análise técnica de mercados financeiros para prever movimentos futuros de preços com base em dados históricos de preços e volume de negociação. Esses indicadores derivam de fórmulas aplicadas a dados, como valores de abertura, fechamento, máxima, mínima e volume de negociação. Um único ponto de dado isolado não proporciona muita informação; é necessária uma série temporal desses pontos para uma análise eficaz (LEMOS, 2022). Os indicadores técnicos ajudam os *traders* a identificar tendências, pontos de reversão e possíveis sinais de compra ou venda.

As médias móveis são amplamente usadas na análise técnica para identificar tendências em dados financeiros, calculando a média dos preços de um ativo em períodos específicos, como dias ou semanas. Elas suavizam flutuações diárias, como observado na Figura 1, fornecendo um sinal claro sobre a direção do mercado. Existem médias móveis simples e ponderadas. Segundo Murphy (1999), os períodos de cálculo variam conforme o objetivo do investidor, como curto prazo (10, 20, 50 dias) e longo prazo (200 dias).

O método do cruzamento duplo, também conhecido como cruzamento de médias móveis, utiliza duas médias móveis para identificar tendências e sinais de compra ou venda. Segundo Pring (2014), quando uma média móvel mais curta cruza acima de uma média mais longa, gera-se um sinal de compra; quando cruza abaixo, gera-se um sinal de venda. Na Figura 1, é possível observar que houve três cruzamentos e, a partir disso, analisar o indicador. Este método é eficaz para capturar mudanças de tendência e fornecer indicadores para negociação.

Figura 1 – Cruzamento de média móvel



Fonte: Elaborado pelo autor, 2025.

2.3 Portfólio e estratégias de investimento

Um portfólio, no contexto do mercado financeiro, é a combinação de diversos ativos financeiros que um investidor possui. O risco é a medida da variação dos retornos de um ativo em relação ao seu valor esperado, enquanto o retorno representa o ganho ou perda de um investimento em um período específico. Os investidores buscam maximizar o retorno do portfólio e minimizar o risco, e a diversificação é uma estratégia-chave para reduzir o risco global para um nível inferior ao risco individual de cada ativo (ASSAF NETO, 2014). Ela diminui o impacto de variações adversas em retornos de ativos isolados, devido à combinação de ativos não correlacionados, que reduz a exposição a flutuações específicas de cada ativo.

A teoria moderna de portfólios, desenvolvida por Harry Markowitz, estabelece que a diversificação é mais eficiente quando os ativos têm correlações reduzidas ou negativas. A inclusão de ativos de setores econômicos distintos reduz a variância agregada do portfólio, pois ativos de um mesmo setor, geralmente, apresentam movimentos semelhantes, aumentando a exposição a riscos específicos (MARKOWITZ, 1952).

A composição de portfólios eficientes, com o objetivo de maximizar retornos e minimizar riscos, exige o uso de algoritmos especializados. Esses métodos calculam a alocação ideal de recursos, considerando o equilíbrio entre risco e retorno.

Estratégias de investimento, por sua vez, são abordagens planejadas para alocar recursos financeiros com o objetivo de atingir metas específicas, como crescimento de capital ou geração de renda. Segundo Pereira e Oliveira (2021), essas estratégias incluem o investimento em valor, que busca adquirir ativos subvalorizados.

Outra abordagem é o investimento em crescimento, focado em empresas com potencial de aumento significativo em lucros, e também a diversificação, que distribui investimentos em diferentes classes de ativos, para minimizar riscos. Complementando essas estratégias, a alocação de ativos é uma prática que envolve a divisão estratégica dos recursos entre diferentes categorias de investimento, ajustando a proporção de cada uma com base no perfil de risco e nos objetivos financeiros do investidor.

Além dessas abordagens, existem estratégias voltadas para o curto prazo, como o processo de compra e venda de ativos, popularmente conhecido como *trading*, que envolve a compra de ativos com a finalidade de vendê-los em um curto período para se obter lucro com as flutuações de preço. Esse tipo de estratégia pode ser subdividido em *day trading*, em que os ativos são comprados e vendidos no mesmo dia, e *swing trading*, no qual as posições são mantidas por alguns dias ou semanas. *Traders*, como são chamados os investidores que seguem essa estratégia, geralmente, utilizam análises técnicas e gráficos para identificar oportunidades de curto prazo e tomar decisões rápidas de compra e venda.

2.4 Backtesting

O *Backtesting* é um método utilizado em modelagem para testar a performance de uma estratégia ou modelo em dados históricos (CAMPBELL, 2007). O objetivo é avaliar a viabilidade de uma estratégia antes de aplicá-la em situações reais.

De acordo com Christoffersen (2008), o termo *backtesting* é usado de várias maneiras no campo financeiro. Ele pode se referir tanto à avaliação do desempenho histórico hipotético de uma estratégia de negociação quanto à avaliação de modelos de risco financeiro, utilizando dados históricos de previsões de risco e de lucros e perdas.

Schumann (2018) descreve *backtesting* como o processo de simular uma estratégia de investimento em dados históricos para avaliar seu desempenho e robustez antes de aplicá-la em condições reais de mercado. De forma inversa, Ni e Zhang (2005) revelam que, se uma estratégia não teve bom desempenho no passado, é provável que ela também não tenha sucesso no futuro. Já Lemos e Cardoso (2010) argumenta que, embora o passado não garanta o futuro, o *backtesting* pode oferecer uma boa indicação de como uma estratégia funcionará.

O *backtesting* é extensivamente utilizado na análise técnica para testar a viabilidade de estratégias de negociação sem envolver capital real, ao recriar condições de mercado passadas. Por meio deste processo analítico, pode-se obter uma vantagem histórica, permitindo a avaliação do desempenho da estratégia sem o risco financeiro. A essência do método está na criação de um ambiente de negociação simulado, em que os investidores podem avaliar a eficácia de suas estratégias com

base em condições de mercado passadas.

Uma distinção relevante no processo de *backtesting* está na aplicação em um único ativo ou para uma lista de ativos. O *backtesting* em um único ativo, por sua simplicidade, permite a análise isolada do desempenho da estratégia sob condições específicas, facilitando a identificação de padrões ou inconsistências. Em contrapartida, o *backtesting* em um portfólio de ativos é realizado considerando-se o conjunto de ativos simultaneamente. Esse método avalia a estratégia aplicada ao portfólio como um todo, simulando decisões de compra e venda de ativos com base em critérios definidos, como alocação de capital por setor.

O *backtesting* em portfólios não se limita à análise individual de ativos. Ele integra os resultados das opções de compra e venda para construir o portfólio em cada etapa, considerando fatores como diversificação, lucro e perda. Assim, a estratégia é testada tanto no nível individual dos ativos quanto na composição final do portfólio. Isso permite verificar a estratégia ao se balancear risco e retorno em diferentes cenários de mercado, além do desempenho dos ativos selecionados.

Para que o processo de *backtesting* seja bem-sucedido, é necessário desenvolver uma estratégia de negociação com objetivos bem definidos, sendo crucial, também, usar indicadores técnicos de forma eficiente. Além disso, é importante gerenciar riscos e analisar os resultados para refinar a estratégia, visando à sua aplicação no mundo real.

Técnicas avançadas de *backtesting* e ferramentas permitem que os investidores enfrentem armadilhas comuns. Elas ajudam a otimizar sistemas de negociação automatizados e também permitem comparar metodologias, como análise de cenários e testes de desempenho futuro. Isso aumenta a confiabilidade e a performance da estratégia.

De acordo com os autores Masteika, Rutkauskas e Janes (2012), os dados históricos são fundamentais no *backtesting*. Eles são usados para simular entradas e saídas de negociações, proporcionando uma base para avaliar o desempenho de uma estratégia em diversas condições de mercado. É essencial considerar o período de tempo e as condições de mercado abrangidas pelos dados, para assegurar uma avaliação extensiva. Com os dados históricos apropriados, os investidores podem testar a viabilidade de uma estratégia de negociação e aplicá-la a períodos alternativos ou a dados fora da amostra.

Em essência, o *backtesting* oferece a oportunidade de aprender com o passado, possibilitando decisões mais informadas no presente e aumentando a confiança na aplicação de estratégias em situações reais de mercado.

2.5 Estado da arte

O Estado da Arte é uma parte fundamental deste trabalho, pois oferece uma visão atualizada do cenário em que se insere. Esta seção destaca os principais trabalhos acadêmicos relevantes e ferramentas utilizadas na área de estudo.

2.5.1 Trabalhos acadêmicos

Em um estudo relacionado, Sarasa-Cabezuelo (2023) desenvolveu um conjunto de funcionalidades de *backtesting* para avaliar estratégias de negociação, gerenciar carteiras, representar resultados de simulações e otimizar uma carteira de ações. Usuários podem monitorar operações, testar estratégias em diferentes condições de mercado e acessar várias operações financeiras. A aplicação permite configurar portfólios, escolher ativos, parametrizar estratégias, carregar dados históricos e compartilhar informações entre usuários.

No entanto, a abordagem descrita por Sarasa-Cabezuelo (2023) se concentra na implementação de uma única estratégia de negociação baseada na compra ou venda de ativos financeiros em seus máximos ou mínimos em um período de tempo.

O trabalho dos autores Silveira e Rodriguez (2017) aborda a avaliação de estratégias de investimento no mercado financeiro brasileiro, com foco em estratégias baseadas em indicadores técnicos. A pesquisa se concentra no índice Ibovespa, abrangendo o período de 2013 a 2015 e o primeiro semestre de 2016.

O estudo examina um conjunto específico de indicadores técnicos e avalia como eles influenciam as estratégias de investimento no contexto do mercado brasileiro. No entanto, a baixa correlação entre os retornos dos períodos analisados evidencia a dificuldade de prever resultados futuros.

Embora o trabalho realize *backtesting* em múltiplos ativos, há limitações importantes. Os autores não utilizam filtros para otimizar os indicadores e não testam variações mais amplas de parâmetros. Além disso, há um processo manual de variação.

2.5.2 Ferramentas

O Tradingview permite aos usuários realizar *backtesting* de estratégias de investimento em dados históricos, customizando parâmetros e testando diferentes abordagens de negociação (TRADINGVIEW, 2024). Contudo, uma limitação significativa da plataforma é a incapacidade de comparar várias estratégias simultaneamente e ordená-las com base em critérios específicos de desempenho. Assim, os investidores precisam consolidar resultados manualmente, um processo que pode ser demorado

e propenso a erros. Embora o TradingView se destaque pela interface intuitiva e por sua ampla comunidade de usuários que compartilham *scripts* e ideias, a ausência de suporte nativo para múltiplas estratégias limita sua aplicabilidade para análises comparativas detalhadas.

O MetaTrader 5 (MT5) é reconhecido pela sua capacidade de realizar *backtesting* detalhado de estratégias usando dados históricos. Ele permite testar estratégias com múltiplos ativos e prazos simultaneamente, oferecendo mais indicadores técnicos e ferramentas gráficas (METAQUOTES, 2024). No entanto, não oferece suporte nativo para comparar várias estratégias simultaneamente ou para ordenação de ativos.

O NextTrade é um sistema de negociação algorítmica que permite a criação, o teste, a otimização e a implantação de estratégias de negociação (STARKS, 2024). A plataforma suporta *backtesting*, permitindo avaliar o desempenho das estratégias em dados históricos, além de possibilitar a otimização de parâmetros por meio de algoritmos genéticos. O sistema também oferece suporte a múltiplas estratégias, propiciando sua combinação para formar abordagens mais complexas. Adicionalmente, o NextTrade suporta negociação multiativa, com a possibilidade de criação de portfólios diversificados.

O QuantConnect oferece um sistema de *backtesting* para estratégias de negociação (QUANTCONNECT, 2024). Utiliza um motor de *backtesting* que simula a execução de estratégias em dados históricos, incluindo ações, futuros e criptomoedas. O ambiente de simulação replica condições reais de mercado, considerando latência e custos de transação, e também gera métricas de desempenho, como retorno, volatilidade e *drawdown*. A plataforma permite personalização de parâmetros e fornece ferramentas de visualização e relatórios detalhados para análise dos resultados.

2.5.3 Comparação dos trabalhos

O Quadro 1 compara as funcionalidades de diferentes estudos e ferramentas de *backtesting*, destacando diferenças em termos de suporte a múltiplas estratégias, ordenação dos ativos, variação de parâmetros e multiativo. Ferramentas populares como TradingView e MetaTrader 5, apesar de suas interfaces intuitivas, são limitadas na comparação simultânea de múltiplas estratégias. Essas ferramentas, embora permitam a personalização de parâmetros, não oferecem uma variação automatizada e flexibilidade para estratégias genéricas.

O estudo de Silveira e Rodriguez (2017), embora forneça suporte para múltiplas estratégias, não permite a variação dinâmica de parâmetros nem a ordenação de ativos. Isso limita a possibilidade de otimização e a personalização das estratégias. Além disso, a análise do multiativo, no contexto de Silveira e Rodriguez (2017),

Quadro 1 – Trabalhos e ferramentas com *backtesting*

<i>Backtesting</i>	Multiestratégia	Ordenação de ativos	Variação de parâmetros	Multiativo
(SARASA-CABEZUELO, 2023)	N	N	N	S
(SILVEIRA; RODRIGUEZ, 2017)	S	N	N	S
(TRADINGVIEW, 2024)	N	N	S	N
(METAQUOTES, 2024)	N	N	N	S
(STARKS, 2024)	S	N	N	S
(QUANTCONNECT, 2024)	N	N	N	N
Presente trabalho	S	S	S	S

Fonte: Elaborado pelo autor, 2025.

é restrita, o que impede uma avaliação de portfólios diversificados.

O NextTrade se diferencia pela possibilidade de montar múltiplos portfólios mesclando estratégias. Sua capacidade de personalização é limitada pela falta de uma variação automatizada de parâmetros durante o *backtesting*, o que pode retardar o processo de otimização.

O presente trabalho, por sua vez, se destaca ao combinar a capacidade de ordenação dinâmica de ativos, variação automatizada de parâmetros e a implementação de múltiplas estratégias de forma simultânea. Essas funcionalidades permitem uma análise mais robusta e adaptativa, abordando as limitações observadas em outras ferramentas.

Além disso, oferece suporte a multiativo, permitindo a construção de portfólios diversificados ao longo do *backtesting*. A diversificação é um dos parâmetros-chave que o sistema pode otimizar, oportunizando um processo de avaliação contínuo. A cada iteração de *backtesting*, o sistema prioriza os ativos com base na estratégia definida pelo usuário, aplicando variações automáticas aos parâmetros durante todo o processo.

Ao realizar uma avaliação comparativa entre diferentes estratégias e parâmetros ao longo do tempo, ele gera novas estratégias derivadas e permite aos investidores escolherem as mais eficientes. Isso torna o trabalho uma contribuição relevante para investidores que precisam de uma ferramenta eficiente e adaptável, superando algumas das barreiras comuns em plataformas amplamente utilizadas, que, muitas vezes, limitam a flexibilidade e a profundidade da análise.

3 METODOLOGIA

Este capítulo descreve a metodologia adotada no desenvolvimento deste trabalho. Inicia-se com a classificação da pesquisa, na Seção 3.1, seguida pela apresentação da solução proposta, na Seção 3.2. Em seguida, na Seção 3.3, são abordados os materiais e tecnologias utilizados, e, por fim, a Seção 3.4 aborda os métodos e os procedimentos adotados.

3.1 Classificação da pesquisa

De acordo com Gil (2002), o presente estudo caracteriza-se, quanto à natureza, como uma pesquisa aplicada, pois seu objetivo é resolver um problema prático relacionado ao desenvolvimento de uma ferramenta de *backtesting* para avaliar estratégias de investimento baseadas na ordenação de ativos no mercado financeiro. Os objetivos e procedimentos são descritivos e exploratórios, o que se justifica pela análise de estratégias existentes e pela investigação de novas funcionalidades para a arquitetura proposta.

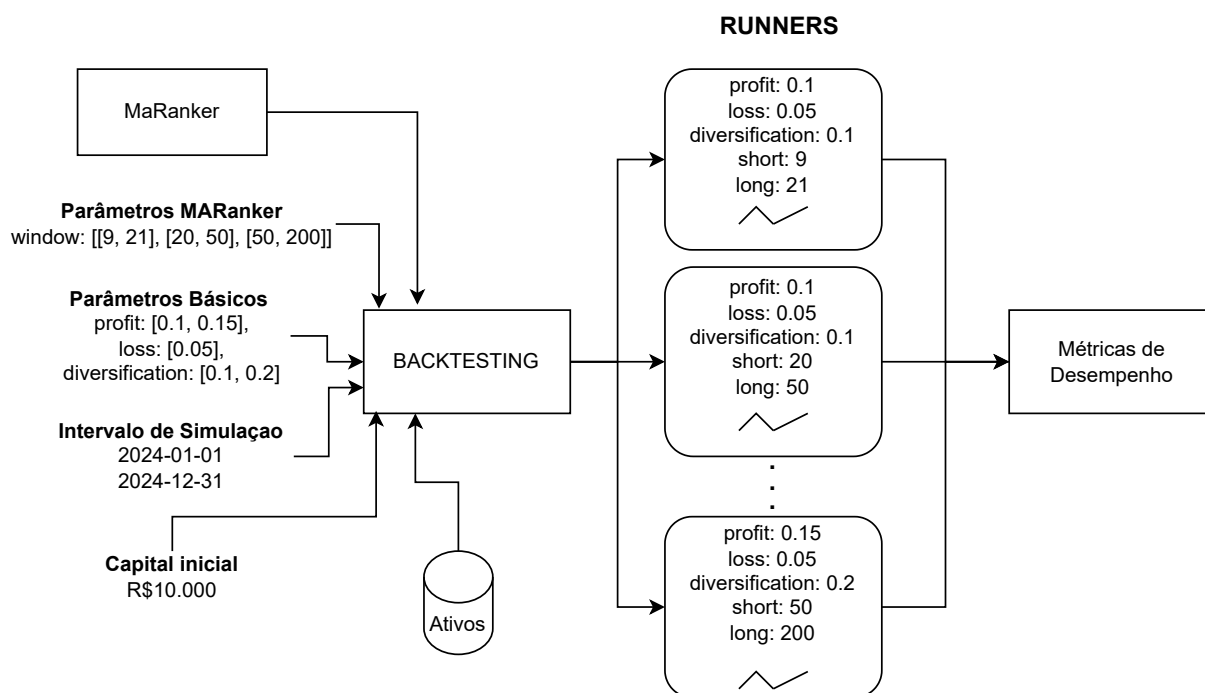
Conforme Wazlawick (2009), o trabalho pode ser classificado como uma pesquisa que tem o objetivo de apresentar uma solução considerada superior em relação às existentes. A ferramenta proposta oferece uma abordagem inovadora para o *backtesting* de estratégias de investimento. Ademais, segundo Gerhardt e Denise Tolfo Silveira (2009), os procedimentos adotados configuram uma pesquisa experimental. Essa classificação se deve à manipulação de variáveis durante o desenvolvimento e teste da ferramenta em diferentes cenários de mercado. A abordagem é quantitativa, centrada na análise de dados históricos e na aplicação de métodos estatísticos para avaliar a eficácia das estratégias de investimento.

3.2 Solução

No presente trabalho, foi desenvolvida uma ferramenta para *backtesting* de estratégias de investimento baseadas na ordenação de ativos. A arquitetura permite a simulação e análise de desempenho de estratégias em diferentes cenários de mercado. A Figura 2 apresenta uma visão geral da solução.

O usuário, desenvolvedor e investidor, implementa algoritmos para ordenação de ativos financeiros. Essa ordenação é essencial para definir estratégias de investimento e otimizar parâmetros por meio de simulações. O processo inicia-se quando o usuário herda a classe abstrata *Ranker* e implementa o método *rank()* em um arquivo Python. Esse método é responsável por ordenar os ativos com base em critérios específicos.

Figura 2 – Solução



Fonte: Elaborado pelo autor, 2025.

Além disso, o usuário fornece a lista de ativos, define os parâmetros do algoritmo, especifica as regras de compra e venda, determina o período da simulação e informa o capital inicial. Com essas informações, a arquitetura do sistema realiza o *backtesting*, ajustando e combinando automaticamente os parâmetros do algoritmo para testar diferentes configurações e avaliar seu desempenho.

Os resultados das simulações são analisados com base em métricas específicas, como retorno e valor final do portfólio. A arquitetura apresenta a análise de performance das estratégias testadas, permitindo que o usuário compare o desempenho entre diferentes estratégias e configurações de parâmetros.

3.3 Materiais e tecnologias

A presente seção descreve quais tecnologias e ferramentas foram usadas para se desenvolver o trabalho. A Seção 3.3.1 apresenta as principais tecnologias utilizadas, e a Seção 3.3.2 descreve o ambiente dos experimentos e os equipamentos usados.

3.3.1 Tecnologias

Para a preparação do ambiente de desenvolvimento, foram utilizadas as tecnologias citadas no Quadro 2. A Python foi escolhida como a linguagem de progra-

mação principal para o desenvolvimento da arquitetura de *backtesting* devido à sua simplicidade, versatilidade e vasta comunidade de suporte (RIBEIRO, 2022b).

Quadro 2 – Ferramentas utilizadas

Ferramenta	Versão	URL
Visual Studio Code	1.92	https://code.visualstudio.com
Python	3.12.4	https://www.python.org
Git	2.45.2	https://git-scm.com
Yfinance	0.2.50	https://github.com/ranaroussi/yfinance
Pandas	2.2.2	https://pandas.pydata.org

Fonte: Elaborado pelo autor, 2025.

A linguagem Python possui bibliotecas consolidadas que facilitam a manipulação de dados, visualização de gráficos e integração com APIs financeiras, essenciais para o processamento e análise de dados históricos no contexto do mercado financeiro (RIBEIRO, 2022a).

A biblioteca *Yfinance* foi utilizada para facilitar a obtenção de dados financeiros históricos de diferentes ativos do mercado. Ela permite uma integração direta com a API do Yahoo Finance, fornecendo acesso fácil e eficiente a informações como preços históricos, volume de negociação e outros indicadores financeiros. A utilização da *Yfinance* simplifica o processo de coleta de dados, essencial para a realização do *backtesting* das estratégias de investimento (AROSSI, 2024).

A biblioteca Pandas foi essencial para a manipulação e análise dos grandes volumes de dados financeiros necessários para o *backtesting*, com suas estruturas de dados, como *DataFrames*, que são tabelas bidimensionais semelhantes a planilhas, permitindo o armazenamento e a manipulação de dados em linhas e colunas.

A biblioteca Pandas permite o processamento eficiente de dados temporais, além de oferecer uma ampla gama de funcionalidades para limpeza, transformação e agregação de dados. Isso facilita a preparação dos dados históricos, ajustando-os conforme necessário para a simulação precisa das estratégias de investimento (MCKINNEY, 2024).

3.3.2 Ambiente de desenvolvimento

O desenvolvimento dos componentes do trabalho proposto foi realizado em um computador pessoal com as configurações do Quadro 3.

Quadro 3 – Especificações do equipamento

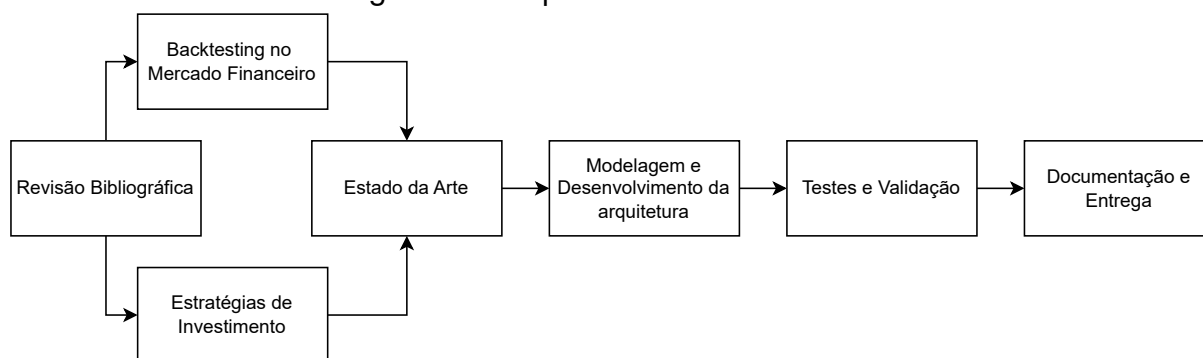
Item	Detalhes
Modelo	Lenovo Gaming 3i i5-11300H
Processador	Intel® Core™ i5-11300H @ 3.10GHz
Processador gráfico	NVIDIA® GeForce® GTX 1650, 4GB, GDDR6
Memória RAM	16 GB, 3200MHz, DDR4
Unidade de estado sólido	500 GB
Arquitetura do sistema operacional	64-bit
Sistema operacional	Windows 11 Home

Fonte: Elaborado pelo autor, 2025.

3.4 Métodos e procedimentos

Para a realização deste trabalho, foram definidas etapas, como ilustrado na Figura 3. O processo foi iniciado com a fase de levantamento e análise, em que foram selecionadas e estudadas as principais estratégias de investimento no mercado financeiro. Nesta fase, também foram realizadas análises aprofundadas das metodologias existentes para *backtesting* no contexto do mercado financeiro. Isso incluiu uma revisão das ferramentas e técnicas mais relevantes empregadas na área, permitindo uma compreensão do estado da arte.

Figura 3 – Etapas de desenvolvimento



Fonte: Elaborado pelo autor, 2025.

Na fase de modelagem e desenvolvimento, foi projetada a arquitetura do sistema, com a definição de módulos e classes, além da preparação dos dados históricos de mercado dos ativos de interesse. Essa etapa incluiu a filtragem, integração e configuração dos parâmetros necessários para simular diferentes períodos econômicos. Nesta fase, também foi implementado um algoritmo de ordenação de ativos que faz uso de médias móveis e foi aplicado em dados reais.

As simulações são avaliadas por meio de métricas específicas, como taxa de retorno, volatilidade e outras medidas de performance financeira, permitindo uma análise comparativa entre as diferentes estratégias testadas. Finalmente, a etapa de

documentação e entrega consistiu na elaboração da monografia e na preparação de toda a documentação necessária para o uso e manutenção futura da arquitetura desenvolvida.

Para gerenciar o desenvolvimento do projeto, foi adotada a metodologia *Scrum*, que se mostrou ideal para lidar com a complexidade do processo e a necessidade de adaptação contínua. O *Product Backlog* foi utilizado para organizar as tarefas e processos em uma lista ordenada, garantindo que todas as etapas até o momento fossem abordadas de forma estruturada. Cada *Sprint*, com duração de 7 a 15 dias, foi planejada para focar em conjuntos específicos de tarefas, permitindo entregas incrementais e revisões constantes (SCHWABER; SUTHERLAND, 2020).

4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento da arquitetura proposta, abrangendo desde a concepção inicial até a avaliação do sistema. A organização do processo de implementação reflete as principais etapas realizadas, iniciando pela modelagem do sistema.

Na Seção 4.1, são apresentados os requisitos do sistema, estabelecendo a base para as decisões de modelagem. Na Seção 4.2, são descritos os conceitos fundamentais e as decisões de *design* que orientaram essa fase. Para facilitar a compreensão do sistema como um todo, adotou-se uma abordagem *top-down* na explicação, permitindo uma visão clara das relações entre os componentes no contexto geral.

O processo inicia-se na Seção 4.3, que aborda a avaliação do sistema, com ênfase na variação dos parâmetros da estratégia e na análise dos resultados obtidos. A Seção 4.4 detalha os componentes responsáveis pela orquestração do fluxo de informações entre as etapas do processo, bem como o gerenciamento de portfólio.

Na Seção 4.5, são descritas as estratégias de ordenação de ativos e algoritmos. Em seguida, a Seção 4.6 aborda os métodos empregados para a obtenção e manipulação de dados, destacando as práticas adotadas para assegurar sua consistência nas análises realizadas. Após a conclusão da implementação, foi conduzida a validação do sistema. Na Seção 4.7, são apresentados os critérios utilizados e os experimentos realizados para verificar a confiabilidade e a aplicabilidade da solução desenvolvida.

Os códigos-fonte empregados no desenvolvimento da arquitetura e na implementação dos algoritmos estão disponíveis publicamente em um repositório do GitHub¹. Além do código, o repositório contém instruções detalhadas para a reprodução dos resultados apresentados.

4.1 Levantamento de requisitos

A Tabela 1 apresenta o *backlog* de produto do projeto. Consiste em uma lista ordenada de tudo o que é necessário para o sucesso do projeto. Contém itens de trabalho que podem incluir funcionalidades, correções de *bugs*, melhorias e quaisquer outras tarefas necessárias.

No *backlog* de produto, os itens são organizados em diferentes níveis de detalhamento com estimativa de término em horas. Os principais componentes incluem épicos, histórias de usuário, tarefas e subtarefas. Os épicos representam grandes blocos de funcionalidade, englobando objetivos amplos do projeto, necessitando ser

¹ <https://portbackrank.github.io>

decompostos em partes menores para viabilizar sua implementação.

As histórias de usuário descrevem funcionalidades sob a perspectiva do usuário. Elas especificam o que o usuário necessita ou deseja realizar no sistema. Essas descrições servem como base para o desenvolvimento de funcionalidades, garantindo que as entregas estejam alinhadas com as expectativas do usuário.

As tarefas e subtarefas correspondem às atividades necessárias para implementar uma história de usuário ou um épico. Elas representam unidades menores de trabalho, permitindo maior organização e controle do progresso dentro do *backlog*. Cada item no *backlog* de produto é priorizado com base em seu valor para o cliente e na necessidade do negócio. A priorização ajuda a equipe a focar nas tarefas mais importantes primeiro.

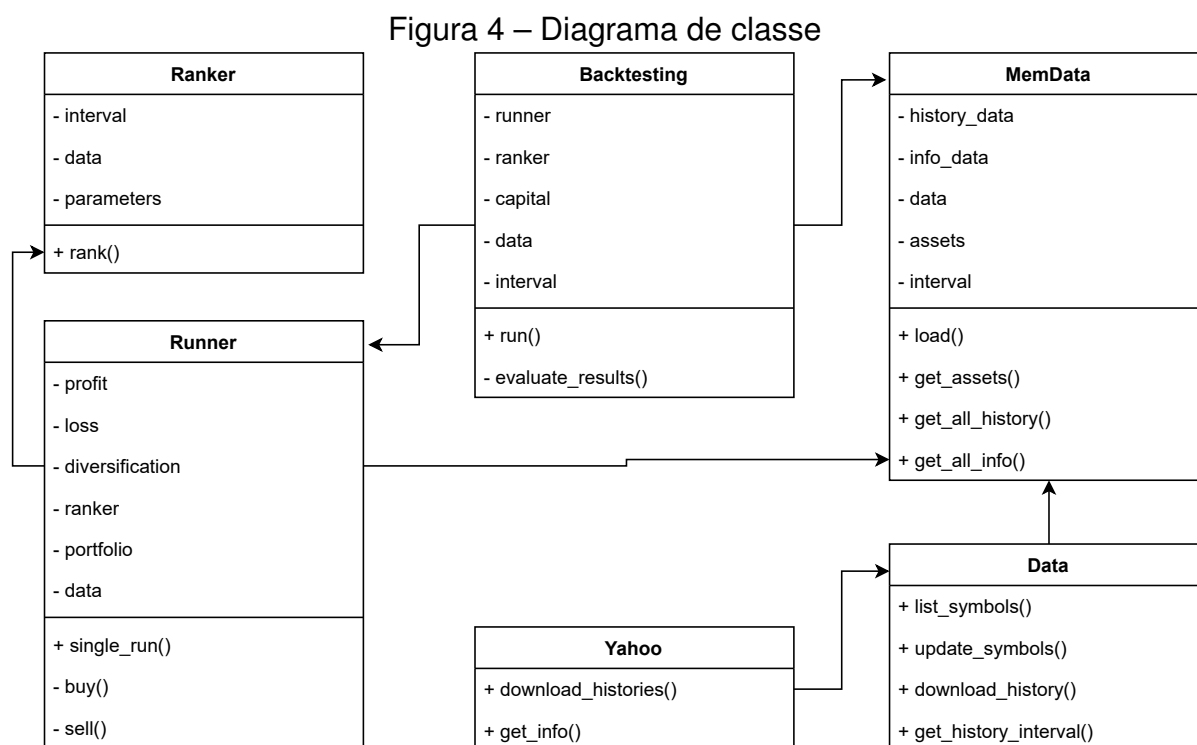
Tabela 1 – *Backlog* de produto

Atividade	Importância	Estimativa	Tipo	Descrição
Redação da Monografia	10	120	Épico	Redação da monografia final do projeto
Revisão Bibliográfica	10	60	Épico	Revisão das principais estratégias de investimento e técnicas de <i>backtesting</i>
Estratégias de Investimento	3	20	Tarefa	Estudo detalhado das estratégias de investimento
<i>Backtesting</i> no Mercado Financeiro	3	20	Tarefa	Análise das metodologias de <i>backtesting</i> no mercado financeiro
Estado da Arte	4	30	História	Pesquisa e documentação do estado da arte
Modelagem e desenvolvimento da arquitetura	10	60	Épico	Desenvolvimento da arquitetura e integração com técnicas simples de investimento
Testes e validação	9	20	Épico	Testes da arquitetura com diferentes algoritmos de ordenação de ativos
Documentação e Entrega	10	30	Épico	Documentação final e entrega do projeto

Fonte: Elaborado pelo autor, 2025.

4.2 Modelagem do sistema

O diagrama de classes apresentado na Figura 4 modela a interação entre diferentes componentes da arquitetura. As classes principais são `Ranker`, `Data`, `YahooData`, `MemData`, `Runner` e `Backtesting`, cada uma com responsabilidades específicas que colaboram para a execução e avaliação dos algoritmos.



Fonte: Elaborado pelo autor, 2025.

A classe `Ranker` é uma classe abstrata que torna a arquitetura genérica e extensível. Ela é projetada para ser implementada pelo usuário, permitindo a definição de algoritmos personalizados de ordenação. Sua função principal é gerenciar a execução e a adição de algoritmos, fornecendo a lógica de ordenação para os demais componentes. Essa classe é utilizada diretamente pelo `Runner`.

A classe `Data` é responsável pela obtenção, preparação, armazenamento em disco e atualização dos dados necessários para a execução dos algoritmos. Esses dados incluem informações de ativos e históricos financeiros que são fundamentais para as operações de ordenação. Os dados históricos utilizados pela `Data` são obtidos a partir da classe `YahooData`, que utiliza a biblioteca `yfinance` para realizar o *download* das informações financeiras. A classe `YahooData` foi projetada para permitir o desacoplamento da fonte de dados, oferecendo flexibilidade caso seja necessário alterar o provedor de dados históricos no futuro.

A `MemData` complementa a funcionalidade da `Data`. Para otimizar o desempenho, ao informar o período do *backtesting*, ela carrega os dados de `Data` em memó-

ria e os mantém disponíveis durante a execução do sistema. Isso reduz o tempo de acesso e agiliza operações que dependem de grandes volumes de dados. Sem esse mecanismo, os dados de um mesmo ativo podem ser lidos diversas vezes do disco, o que torna o acesso lento. A `MemData` evita essa redundância, garantindo maior eficiência. Essa otimização é essencial, pois cada ciclo de compra executado pelo `Runner` requer a ordenação atualizada dos ativos.

A classe `Runner` atua como o executor principal do sistema. Sua função é percorrer os dados diários, utilizando o `Ranker` para ordenar os ativos e tomar decisões de compra e venda com base nesses *rankings*. O `Runner` é responsável por instanciar o `Ranker` com os parâmetros apropriados, que podem variar conforme as estratégias definidas pela classe `Backtesting`. Esses parâmetros incluem configurações específicas para compra, bem como parâmetros globais aplicados tanto para compra quanto para venda, como lucro esperado, limites de perda e níveis de diversificação. Além disso, o `Runner` mantém um portfólio dos ativos, permitindo controle sobre as operações realizadas.

A classe `Backtesting` desempenha um papel central na análise do desempenho dos algoritmos. Ela permite avaliar os resultados das estratégias testadas, fornecendo métricas de retorno e eficiência. Esse sistema é organizado de forma modular para separar responsabilidades, facilitar a expansão ou modificação, aumentar a manutenibilidade e permitir a adição de novas estratégias.

4.3 Avaliação

A classe `Backtesting` gerencia o fluxo principal das simulações, coordenando a criação de instâncias da classe `Runner` e a execução das estratégias de investimento. Ela centraliza o controle das simulações, configurando os parâmetros necessários para cada instância de `Runner` e avaliando os resultados.

A inicialização da classe recebe o intervalo de tempo da simulação, o capital inicial, um objeto da classe `Ranker` e dicionários com configurações. O intervalo de tempo é utilizado para carregar os dados históricos na instância de `MemData`, responsável pelo gerenciamento desses dados, e o objeto do `Ranker` define as estratégias de ordenação dos ativos.

No método `run()`, da classe `Backtesting`, as combinações de parâmetros para o `Runner` e o `Ranker` são definidas a partir de dicionários. Exemplos desses dicionários `parameter_grid` e `ranker_grid` são representados nas Figuras 5(a) e 5(b), que especificam as variações que devem ser analisadas.

A chave `window` retrata uma lista de janelas de médias móveis, onde cada elemento é um par de valores, sendo que o primeiro valor do par corresponde à média móvel de curto prazo, enquanto o segundo valor representa a média móvel de longo

Figura 5 – Parâmetros de estratégia e ranqueador

<pre>{ "profit": [0.1, 0.15], "loss": [0.05], "diversification": [0.1, 0.2] }</pre>	<pre>{ "window": [[9, 21], [20, 50], [50, 200]] }</pre>
(a) – <i>parameter_grid</i>	(b) – <i>ranker_grid</i>

Fonte: Elaborado pelo autor, 2025.

prazo. A estratégia de ordenação utilizada neste contexto baseia-se em cruzamentos de médias móveis. No entanto, diferentes implementações de Ranker podem utilizar outros parâmetros, como indicadores técnicos ou métricas alternativas, de acordo com a abordagem de ordenação adotada.

Os termos utilizados no dicionário de *parameter_grid* possuem significados específicos para a estratégia de investimento. Profit refere-se ao percentual de lucro desejado para a estratégia, ou seja, o retorno esperado sobre o investimento. Loss, por sua vez, define o limite de perda aceitável, expresso como um percentual do valor inicial investido. O parâmetro Diversification está relacionado à proporção do portfólio que será alocada em um mesmo setor e indústria. A escolha dos fatores da estratégia depende do ordenador adotado, podendo incluir diferentes variáveis.

Tabela 2 – Parâmetros para combinação

Profit	Loss	Diversification	Short	Long
0,1	0,05	0,1	9	21
0,1	0,05	0,2	20	50
0,15	0,05	0,1	50	200
0,15	0,05	0,2		

(a) – Parâmetros de execução

(b) – Parâmetros de estratégia

Fonte: Elaborado pelo autor, 2025.

A função `product()` gera todas as combinações possíveis entre os parâmetros das Tabelas 2(a) e 2(b), criando o conjunto de entradas para as instâncias de Runner (12 combinações). Cada combinação dos parâmetros Profit, Loss e Diversification é associada aos parâmetros Long e Short, gerando as configurações mostradas na Tabela 3.

A função interna `run_simulation()` é responsável por criar cada instância de Runner, como ilustrado na Figura 6. Para cada combinação de parâmetros, o método inicializa o Runner com os valores de lucro, perda e diversificação, além da instância configurada do Ranker. O método `single_run()`, do Runner, executa a

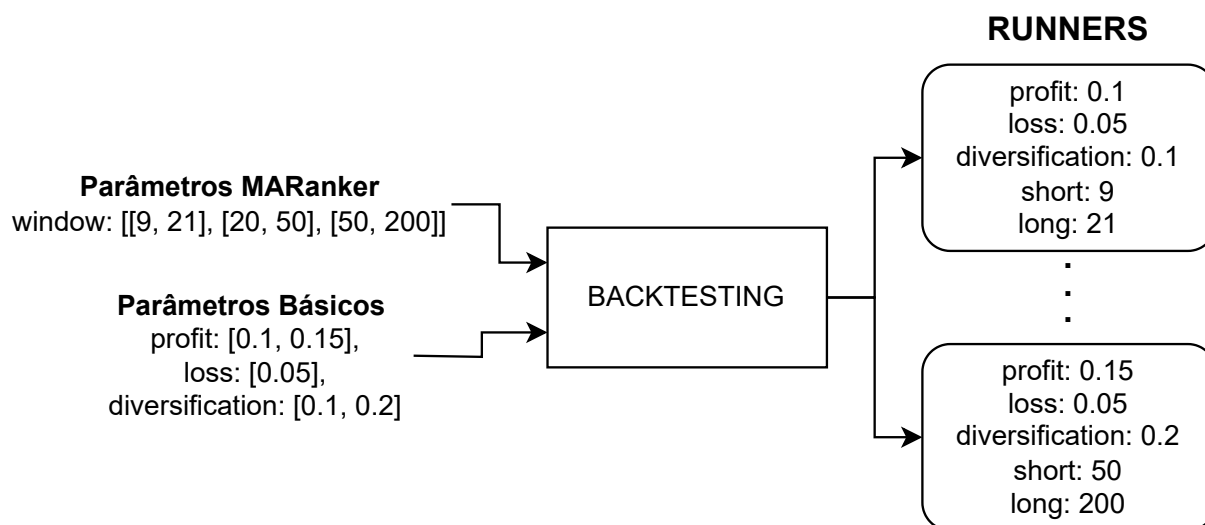
Tabela 3 – Combinações

Profit	Loss	Diversification	Window
0,1	0,05	0,1	9, 21
0,1	0,05	0,1	20, 50
0,1	0,05	0,2	50, 200
0,1	0,05	0,2	9, 21
0,1	0,05	0,1	20, 50
0,1	0,05	0,1	50, 200
0,15	0,05	0,2	9, 21
0,15	0,05	0,2	20, 50
0,15	0,05	0,1	50, 200
0,15	0,05	0,1	9, 21
0,15	0,05	0,2	20, 50
0,15	0,05	0,2	50, 200

Fonte: Elaborado pelo autor, 2025.

simulação com base nesses parâmetros, retornando os resultados individuais.

Figura 6 – Instâncias de Runner



Fonte: Elaborado pelo autor, 2025.

Os resultados de cada simulação incluem métricas como saldo final, valor do portfólio e retorno total. Essas informações são avaliadas pelo método `_evaluate_results()` e consolidadas em um `DataFrame`².

Este `DataFrame` reúne os dados de todas as simulações, facilitando a análise comparativa entre diferentes configurações. Além disso, são armazenados, para cada configuração, os dados de compra e venda da simulação, em arquivos JSON, nomeados com suas respectivas variações.

² Um `DataFrame` é uma estrutura de dados tabular bidimensional.

A escolha de variar todos os parâmetros do `Runner` e do `Ranker` simultaneamente permite explorar as interações entre diferentes configurações. Essa abordagem garante que as combinações sejam avaliadas de forma abrangente, reduzindo o risco de ignorar interdependências entre os parâmetros.

4.4 Execução

A classe `Runner` simula o processo de compra e venda de ativos em um intervalo de tempo. Sua implementação explora como estratégias de ordenação de ativos e parâmetros ajustáveis afetam o desempenho do portfólio.

O portfólio armazena os ativos adquiridos, incluindo símbolo, quantidade, preço de aquisição, setor e data de compra. Já os métodos `_buy()` e `_sell()` gerenciam operações de compra e venda com base em critérios configuráveis.

A lógica da venda verifica, diariamente, se algum ativo no portfólio atende aos critérios de venda, os quais incluem os parâmetros `profit` (lucro desejado) e `loss` (limite de perda aceitável). O método adota uma abordagem consistente ao incluir restrições operacionais, como o volume diário disponível no mercado, e ao respeitar a ordem de entrada e saída *First In, First Out* (FIFO) na realização das vendas. Essas decisões foram tomadas para aproximar o comportamento da simulação ao contexto de operações financeiras reais, em que o volume negociado pode limitar as transações.

A venda de um ativo no portfólio ocorrerá quando a variação percentual entre o preço de compra e o preço atual atingir os limites predefinidos de lucro ou perda. A variação percentual de um ativo é calculada como a diferença entre o preço atual e o preço de compra, dividida pelo preço de compra. Além disso, para garantir a viabilidade da venda, verifica-se se a quantidade a ser liquidada não ultrapassa o volume diário negociado no mercado.

Um exemplo é a simulação realizada na data de 2024-02-14, utilizando-se um limite de lucro (*profit*) de 10% e um limite de perda (*loss*) de -5%. Além disso, o caixa disponível era de R\$ 100,00. A Tabela 4 apresenta a composição do portfólio na data da simulação, contendo as quantidades adquiridas, os preços de compra e os preços dos ativos naquela data.

Com base nos critérios estabelecidos para a simulação, dois ativos atingiram os limites definidos e, portanto, foram vendidos. O ativo `P1GR34.SA` apresentou um preço atual de R\$ 460,00, em comparação com seu preço de compra de R\$ 394,68, resultando em uma variação percentual de 16,5%. Como essa variação supera o limite de 10%, o ativo foi vendido com lucro. A quantidade total de 4 unidades foi liquidada por um valor de R\$ 1840,00, somando-se ao caixa, resultando em R\$ 1940,00.

Já o ativo `F1TN34.SA` apresentou um preço atual de R\$ 130,00, inferior ao

Tabela 4 – Portfólio inicial venda

Símbolo	Quant.	Compra	Setor	Atual	Variação (%)
HFOF11.SA	18	70,51	<i>Financial</i>	68,00	-3,58%
P1GR34.SA	4	394,68	<i>Financial Services</i>	460,00	+16,53%
F1TN34.SA	7	142,24	<i>Technology</i>	130,00	-8,62%
S2GM34.SA	67	22,50	<i>Basic Materials</i>	23,30	+3,56%
NEOE3.SA	19	19,13	<i>Utilities</i>	18,13	-5,22%
C1MG34.SA	23	15,60	<i>Consumer Cyclical</i>	14,87	-4,67%

Fonte: Elaborado pelo autor, 2025.

de compra, de R\$ 142,24, resultando em uma variação de -8,6%. Como essa variação ultrapassa a perda máxima permitida de -5%, o ativo foi vendido para evitar maiores prejuízos. A venda das 7 unidades resultou em um valor de R\$ 910,00, que foi somado ao saldo de caixa, totalizando R\$ 2850,00 após ambas as transações. Depois da verificação e venda dos ativos P1GR34.SA e F1TN34.SA, o portfólio atualizado é mostrado na Tabela 5.

Tabela 5 – Portfólio atualizado após vendas

Símbolo	Quant.	Compra	Data de compra	Setor	Atual
HFOF11.SA	18	70,51	2024-01-04	<i>Financial</i>	68,00
S2GM34.SA	67	22,50	2024-02-02	<i>Basic Materials</i>	23,30
NEOE3.SA	19	19,13	2024-02-02	<i>Utilities</i>	18,13
C1MG34.SA	23	15,60	2024-02-05	<i>Consumer Cyclical</i>	14,87

Fonte: Elaborado pelo autor, 2025.

O processo da compra baseia-se no *ranking* gerado pela classe Ranker. Implementações como a MARanker, abordada na Seção 4.5, definem a prioridade dos ativos a serem adquiridos. A decisão de compra leva em consideração o caixa disponível e o limite de diversificação por setor, configurado pelo parâmetro *diversification*. Isso garante que o portfólio mantenha uma distribuição equilibrada entre setores, evitando concentração excessiva em um único segmento.

O método `rank()`, da classe Ranker, é chamado diariamente para ordenar os ativos e indicar aqueles que possuem maior prioridade de aquisição. Para cada ativo no *ranking*, é realizada uma verificação dos dados históricos simulados, garantindo que a compra respeite o volume diário disponível e o preço de fechamento.

Após isso, é calculada a quantidade máxima de unidades a serem adquiridas. Esse cálculo leva em conta o limite de investimento por setor, o caixa disponível e o volume diário negociado. Primeiramente, define-se o limite de investimento para cada setor; caso este ainda não esteja presente no portfólio, o limite é definido com base no saldo disponível. Se já houver ativos do mesmo setor, o valor é ajustado para

preservar a diversificação.

Em seguida, calcula-se a quantidade de unidades a serem compradas, considerando-se o saldo disponível, o limite por setor e o volume diário negociado. A compra ocorre somente se todas essas condições forem atendidas, garantindo que haja saldo suficiente, que o setor não ultrapasse o limite e que o ativo tenha liquidez. Somente após essa avaliação, a compra é efetivada, registrando o ativo no portfólio e atualizando o saldo de caixa.

No contexto da simulação, no mesmo dia 2024-02-14, ocorreu uma operação de compra de três ativos, com o saldo de caixa disponível para novas aquisições de R\$ 2.850,00, e o limite de diversificação setorial em 20% configurado. O portfólio é representado pela Tabela 5.

Obteve-se o *ranking* para o dia em questão, resultando em uma lista extensa de ativos, sendo os três primeiros GEPA4.SA, A1MT34.SA e AZEV4.SA. O primeiro ativo a ser comprado foi GEPA4.SA, do setor *Utilities*, com preço de R\$ 21,05. O portfólio já possuía R\$ 344,47 alocados no setor *Utilities*, e o limite de diversificação era de R\$ 829,88. O saldo disponível para o setor era de R\$ 485,41, permitindo a compra de 23 unidades, totalizando R\$ 483,15. O saldo de caixa, após a compra, foi de R\$ 2.369,85.

O segundo ativo foi A1MT34.SA, do setor *Technology*, com preço de R\$ 75,12. Como o setor não possuía investimentos, o limite de diversificação para ele era de R\$ 829,88. O volume diário disponível para compra de A1MT34.SA era de 4 unidades; portanto, o número máximo de unidades a ser comprado era 4, totalizando R\$ 300,48. O saldo de caixa, após a compra, foi de R\$ 2.069,37.

O terceiro ativo foi AZEV4.SA, do setor *Industrials*, com preço de R\$ 15,60. O limite de diversificação para o setor era de R\$ 829,88, e o saldo disponível para compra de AZEV4.SA, R\$ 829,88. Como o preço do ativo era R\$ 15,60, o valor máximo para a compra era de 53 unidades, totalizando R\$ 826,80. O saldo de caixa, após a compra, resultou em R\$ 1.242,57.

Após as compras, o portfólio foi ajustado com os novos ativos presentes na Tabela 6, e o saldo de caixa final foi de R\$ 1.242,57. O processo continua enquanto houver saldo disponível e ativos que permitam novas compras, respeitando-se os critérios de diversificação estabelecidos.

O método principal da classe, `single_run()`, encapsula todo o fluxo de execução. Ele inicializa o `Ranker` com os parâmetros configurados, percorre os dados financeiros diários no intervalo especificado, aplicando a lógica de compra e venda para atualizar o portfólio e o saldo disponível. O *ranking* gerado pelo `Ranker` orienta as compras, enquanto os critérios de venda são verificados no portfólio atual. O resultado final, incluindo o saldo de capital restante e os ativos em carteira, avalia a eficácia da estratégia utilizada.

Tabela 6 – Portfólio atualizado após compra

Símbolo	Quant.	Compra	Data de compra	Setor	Atual
HFOF11.SA	18	70,51	2024-01-04	<i>Financial</i>	68,00
S2GM34.SA	67	22,50	2024-02-02	<i>Basic Materials</i>	23,30
NEOE3.SA	19	19,13	2024-02-02	<i>Utilities</i>	18,13
C1MG34.SA	23	15,60	2024-02-05	<i>Consumer Cyclical</i>	14,87
GEPA4.SA	23	21,05	2024-02-14	<i>Utilities</i>	21,05
A1MT34.SA	4	75,12	2024-02-14	<i>Technology</i>	75,12
AZEV4.SA	53	15,60	2024-02-14	<i>Industrials</i>	15,60

Fonte: Elaborado pelo autor, 2025.

A relação entre o `Runner` e o `Ranker` é detalhada na Seção 4.5. Essa análise destaca como diferentes implementações da classe `Ranker` influenciam a lógica de compra e venda. Além disso, a separação das responsabilidades em métodos como `_buy()` e `_sell()` facilita a manutenção e a experimentação com diferentes configurações.

4.5 Ordenação de ativos

A classe `Ranker` foi projetada como uma classe abstrata para ser herdada e implementada de acordo com as necessidades específicas de estratégias de ordenação. Sua arquitetura exige a implementação do método abstrato `rank()`, que deve, obrigatoriamente, retornar uma lista de símbolos de ativos ordenados.

O método `rank()` recebe um dicionário de parâmetros, permitindo flexibilidade na configuração de estratégias quantitativas. Por exemplo, em uma implementação baseada em médias móveis, um dicionário contém a chave `window` associada a uma lista com dois valores `[9, 21]`. Esses valores representam as janelas de médias móveis utilizadas na estratégia. Durante a inicialização, os parâmetros são extraídos e atribuídos às variáveis internas. Dessa forma, a parametrização permite que o usuário modifique dinamicamente os períodos das médias móveis sem alterar o código-fonte da classe.

Essa escolha por dicionários permite encapsular os parâmetros de forma organizada e generalista. Em dicionários mais elaborados, o fornecimento de parâmetros adicionais oferece maior controle, possibilitando a implementação de estratégias baseadas em indicadores técnicos ou combinações de critérios de forma mais eficiente.

A classe `Ranker` recebe uma instância da classe `MemData`, que fornece acesso a dados financeiros históricos ou informações complementares, como setor ou indicadores fundamentais das empresas. Isso possibilita o uso de estratégias sim-

ples, baseadas apenas em listas de ativos, ou técnicas mais avançadas, que utilizam múltiplas fontes de dados e cálculos complexos.

Neste trabalho, foi implementada a classe `MARanker`, que utiliza o conceito de cruzamento de médias móveis, detalhado no Capítulo 2. Essa técnica analisa tendências de mercado e identifica pontos de inflexão nos preços dos ativos. Ela utiliza médias móveis de curto e longo prazo para ordenar os ativos com base na força do cruzamento, refletida pela proporção entre essas médias.

A `MARanker` detecta momentos em que a média móvel de curto prazo cruza acima da média de longo prazo, um sinal comum em análises técnicas para indicar alta nos preços. O método `rank()` calcula as médias móveis para cada ativo e verifica se, em uma data específica, ocorreu o cruzamento desejado. Quando essa condição é atendida, um índice de força é calculado com base na razão entre as médias, indicando a intensidade do movimento e auxiliando na ordenação dos ativos.

O comportamento da `MARanker` é parametrizado por `short` e `long`, que definem os períodos das médias móveis. Janelas curtas favorecem movimentos rápidos, enquanto janelas longas capturam tendências mais estáveis. A Tabela 7 exemplifica os ativos ordenados, e o código da Figura 7 ilustra o funcionamento do algoritmo.

Tabela 7 – Lista de ordenação de ativos.

Ativo	Força (%)
PETR4.SA	5,31
VALE3.SA	3,57
ITUB4.SA	1,67
...	...

Fonte: Elaborado pelo autor, 2025.

A `MARanker` foi projetada para operar no fluxo de execução do `Runner`. A interação entre o `Runner` e as estratégias de ordenação conecta diretamente as decisões de compra ao comportamento dos *rankings* diários, integrando as estratégias ao processo de simulação.

4.6 Obtenção e manipulação dos dados

De posse dos símbolos dos ativos, a classe `YahooData` é responsável por baixar os dados históricos dos ativos. Para isso, foi utilizado o método `history()`, da biblioteca `yfinance`, que retorna um `DataFrame` com as informações detalhadas, como mostrado na Tabela 8. Os dados obtidos têm como índice a data (`Date`), o que facilita operações de manipulação temporal, como filtragens e agregações.

Além disso, os dados são armazenados em cache, para que, caso seja um ativo e período repetido, não seja necessário fazer a requisição novamente na

Figura 7 – Método rank() da MARanker

```

1 def rank(self, date: str = None) -> List[str]:
2     dict_data = self.data.get_all_history()
3     ranked_symbols = []
4     for symbol, df_data in dict_data.items():
5         short = 'w' + str(self._short)
6         long = 'w' + str(self._long)
7         df_data[short] = df_data['Close'].rolling(self._short).mean()
8         df_data[long] = df_data['Close'].rolling(self._long).mean()
9         strength = float('-inf')
10        if date in df_data.index:
11            idx = df_data.index.get_loc(date)
12            if isinstance(idx, slice):
13                idx = idx.start
14            if idx >= 1:
15                latest = df_data.iloc[idx]
16                prev = df_data.iloc[idx-1]
17                if prev[short] <= prev[long] and latest[short] >
18                    ↪ latest[long]:
19                    strength = (latest[short] / latest[long] - 1) * 100
20        else:
21            continue
22        ranked_symbols.append((symbol, strength))
23    ranked_symbols.sort(key=lambda x: x[1], reverse=True)
24    return [x[0] for x in ranked_symbols]

```

Fonte: Elaborado pelo autor, 2025.

biblioteca *yfinance*. É importante destacar que o sistema foi projetado para suportar dados históricos de qualquer ativo disponível na plataforma Yahoo Finance.

Tabela 8 – *DataFrame* de dados históricos

Date	Open	High	Low	Close	Volume
2020-12-18	29,0000	30,0	27,0000	30,0000	1095400
2020-12-19	27,5100	30,5	25,7199	26,2999	169200
2020-12-20	26,0400	26,1	24,9000	26,1000	80500
...
2025-01-06	27,1000	27,1	25,5499	26,4000	49000

Fonte: Elaborado pelo autor, 2025.

Com os dados obtidos e devidamente salvos, a classe *Data* implementa alguns métodos para manipulação deles. Um dos métodos mais relevantes nesse processo é o `get_history_interval()`, responsável por retornar os dados históricos filtrados dinamicamente com base em um intervalo de tempo especificado.

Esse método aceita como entrada a lista de ativos, a data inicial (`start_date`), a data final (`end_date`) e a coluna desejada (`column_filter`, por pa-

drão `Close`). O resultado é uma lista de dicionários em que cada elemento contém o símbolo do ativo e um `DataFrame` correspondente aos dados filtrados, como evidenciado na Figura 8. Eles possuem a `Date` como *index* e também `Close` e `Volume`. Esse formato de retorno permite uma consulta otimizada, pois o `DataFrame` é facilmente acessado via chave, evitando a necessidade de percorrer a lista.

Figura 8 – Retorno `get_history_interval()`

```
[
  {
    "symbol": "PETR4.SA",
    "data": [
      {
        "Date": "2012-12-18",
        "Open": 29.0000,
        "High": 30.0,
        "Low": 27.0000,
        "Close": 30.0000,
        "Volume": 1095400,
        "Dividends": 1.6000
      },
      ...
    ]
  },
  {
    "symbol": "EQPA3.SA",
    "data": [
      ...
    ]
  },
  ...
]
```

Fonte: Elaborado pelo autor, 2025.

Além dos métodos mencionados, destaca-se, também, o `get_info()`. O método `get_info()` permite carregar informações adicionais sobre os ativos, como o setor ao qual pertencem, informações essenciais para cálculos de diversificação ou para alguns algoritmos de ordenação, os quais também são salvos em cache, desta vez, em CSV. O Quadro 4 apresenta essas informações.

Durante o desenvolvimento, foi identificado um gargalo no acesso ao cache, devido às múltiplas consultas aos mesmos dados. Esse comportamento, embora esperado por causa do grande volume de dados financeiros e da natureza intensiva das análises, ressaltou a necessidade de uma solução que mantivesse os dados relevantes diretamente em memória, visando à aceleração do processamento.

Para mitigar esse problema, foi projetada a classe `MemData`, que funciona como uma interface para os dados financeiros, sendo responsável pelo carregamento

Quadro 4 – Informações dos ativos salvos em CSV

Campo	Valor
<i>address1</i>	Avenida Henrique Valadares, 28
<i>address2</i>	19th floor
<i>city</i>	Rio De Janeiro
<i>state</i>	RJ
<i>zip</i>	20241-030
<i>country</i>	<i>Brazil</i>
<i>website</i>	https://petrobras.com.br
<i>industry</i>	<i>Oil & Gas Integrated</i>
<i>industryKey</i>	<i>oil-gas-integrated</i>
<i>industryDisp</i>	<i>Oil & Gas Integrated</i>
<i>sector</i>	<i>Energy</i>
<i>currency</i>	BRL

Fonte: Elaborado pelo autor, 2025.

e gerenciamento das informações em memória. Essa abordagem reduziu significativamente o tempo de acesso a históricos de preços e dados complementares dos ativos, sendo particularmente útil em cenários que envolvem operações iterativas, como o processo de *backtesting*.

O método `load()`, que representa a implementação central da classe, é responsável por carregar os dados históricos e informações adicionais dos ativos dentro de um intervalo de tempo predefinido. Os dados são armazenados em dois dicionários: o `history_data()`, que contém os históricos de preços, e `info_data()`, que armazena informações complementares dos ativos.

Além disso, a classe disponibiliza métodos auxiliares para a manipulação dos dados em memória, como `get_all_history()`, que retorna todo o histórico carregado, e `get_all_info()`, que fornece acesso às informações complementares. A lista completa de ativos gerenciados pela instância é obtida por meio do método `get_assets()`.

Essa implementação foi projetada para otimizar operações com *DataFrames* em larga escala, evitando acessos frequentes ao cache ou a fontes externas. Em cenários como o *backtesting*, em que múltiplas simulações são realizadas sobre os mesmos dados, o uso da `MemData` reduziu consideravelmente o tempo de execução.

4.7 Validação

Para validar a abordagem proposta, foi utilizada a classe `MARanker`. A validação foi realizada utilizando-se ativos pertencentes aos índices Standard & Poor's

500 (S&P 500), Ibovespa (IBOV) e Índice Brasil Amplo (IBrA). O S&P 500 é um dos principais indicadores do desempenho das 500 maiores empresas listadas nas bolsas dos Estados Unidos. O IBOV representa as ações mais negociadas na B3, enquanto o IBrA abrange um conjunto mais amplo de ativos do mercado brasileiro. Essa abordagem permitiu testar a estratégia em diferentes mercados.

Após a obtenção dos símbolos, o sistema consulta informações adicionais dos ativos, como o setor e a indústria. Nem todos os ativos possuíam essa informação disponível no Yahoo Finance, e, portanto, foram descartados. Os parâmetros foram definidos conforme a Tabela 9.

A validação foi conduzida no intervalo anual (2024-01-01 a 2024-12-31), com capital inicial de R\$10.000.

Foram testadas combinações de médias móveis de curto prazo (*short*) e longo prazo (*long*), cobrindo um espectro de sensibilidades às tendências de mercado. O impacto das configurações de *diversification*, *profit* e *loss* também foi avaliado.

Tabela 9 – Parâmetros e variações testados na simulação

Parâmetro	Valores
Intervalo	2024-01-01 a 2024-12-31
Capital Inicial	R\$10.000
Diversificação	0,1 e 0,2
Lucro	0,10 e 0,15
Perda	0,05
Média Móvel Curto Prazo	9, 20, 50
Média Móvel Longo Prazo	21, 50, 200

Fonte: Elaborado pelo autor, 2025.

Os resultados obtidos com o *MARanker* destacam o impacto significativo das configurações de médias móveis e do nível de diversificação nos retornos totais. Nas seções a seguir, são apresentados os resultados obtidos para cada um desses índices, destacando as particularidades de cada ambiente e os impactos de cada um desses parâmetros.

4.7.1 S&P 500

A Tabela 10 apresenta os resultados obtidos na validação da estratégia utilizando ativos do índice S&P 500. Além disso, a Figura 9 ilustra a variação do desempenho da estratégia para cada configuração testada.

Os resultados indicam variações significativas no desempenho da estratégia conforme a configuração utilizada, sendo que a melhor delas obteve um retorno de 20,33%, enquanto a pior apresentou 3,41%.

Tabela 10 – Resultados da validação no S&P 500

Profit	Loss	Diversificação	Médias	Caixa	Portfólio	Retorno
0,10	0,05	0,1	[9, 21]	7,91	10605,52	6,13%
0,10	0,05	0,1	[20, 50]	3,89	10643,40	6,47%
0,10	0,05	0,1	[50, 200]	4,12	10337,35	3,41%
0,10	0,05	0,2	[9, 21]	0,64	12032,38	20,33%
0,10	0,05	0,2	[20, 50]	1,18	10957,90	9,59%
0,10	0,05	0,2	[50, 200]	0,08	10992,57	9,93%
0,15	0,05	0,1	[9, 21]	2,66	10433,14	4,36%
0,15	0,05	0,1	[20, 50]	7,17	10545,25	5,52%
0,15	0,05	0,1	[50, 200]	7,23	10382,04	3,89%
0,15	0,05	0,2	[9, 21]	2,44	11401,79	14,04%
0,15	0,05	0,2	[20, 50]	5,95	10694,78	7,01%
0,15	0,05	0,2	[50, 200]	3,90	11191,93	11,96%

Fonte: Elaborado pelo autor, 2025.

O melhor desempenho ocorreu com médias móveis de curto prazo (9, 21) e menor diversificação (0,2). Esse resultado sugere que a estratégia se beneficiou da rápida adaptação às oscilações do mercado, aproveitando reversões de curto prazo e mantendo posições mais concentradas no período analisado.

O pior desempenho foi registrado com médias móveis longas (50, 200) e maior diversificação (0,1). Essa configuração teve menor eficiência na captura de reversões rápidas e diluiu o impacto de movimentos positivos ao distribuir o capital entre mais ativos.

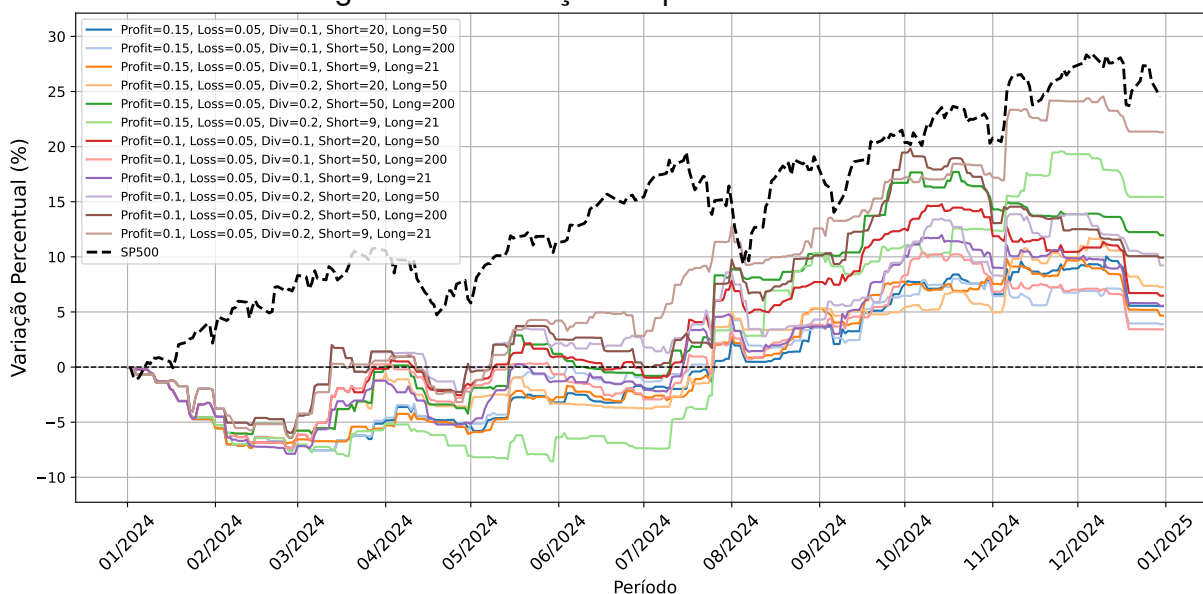
No geral, os retornos foram positivos na maioria das configurações. O sucesso das médias curtas e da menor diversificação indica que, no período analisado, o S&P 500 favoreceu estratégias mais concentradas e reativas às oscilações do mercado.

4.7.2 IBOV

Os testes demonstraram que todas as configurações resultaram em retornos negativos, refletindo a desvalorização do IBOV ao longo do período avaliado. A configuração mais favorável apresentou um retorno de -18,73%, enquanto a menos eficiente obteve -37,10%. A Tabela 11 apresenta os resultados obtidos na validação da estratégia utilizando ativos do índice IBOV.

O melhor desempenho foi alcançado com menor diversificação (0,2) e médias móveis curtas (9, 21). Apesar do cenário desfavorável, essa configuração conseguiu reduzir parte das perdas, possivelmente, devido à concentração em ativos menos

Figura 9 – Simulação de portfólios S&P500



Fonte: Elaborado pelo autor, 2025.

Tabela 11 – Resultados da validação no IBOV

Profit	Loss	Diversificação	Médias	Caixa	Portfólio	Retorno
0,10	0,05	0,1	[9, 21]	1,26	6812,18	-31,87%
0,10	0,05	0,1	[20, 50]	293,72	6660,26	-30,46%
0,10	0,05	0,1	[50, 200]	10,34	6852,81	-31,37%
0,10	0,05	0,2	[9, 21]	0,36	7649,21	-23,50%
0,10	0,05	0,2	[20, 50]	0,53	6678,73	-33,21%
0,10	0,05	0,2	[50, 200]	1,21	6289,04	-37,10%
0,15	0,05	0,1	[9, 21]	0,10	6770,71	-32,29%
0,15	0,05	0,1	[20, 50]	2,15	6876,75	-31,21%
0,15	0,05	0,1	[50, 200]	0,46	6930,68	-30,69%
0,15	0,05	0,2	[9, 21]	0,20	8127,15	-18,73%
0,15	0,05	0,2	[20, 50]	0,26	6737,44	-32,62%
0,15	0,05	0,2	[50, 200]	1,31	6598,22	-34,00%

Fonte: Elaborado pelo autor, 2025.

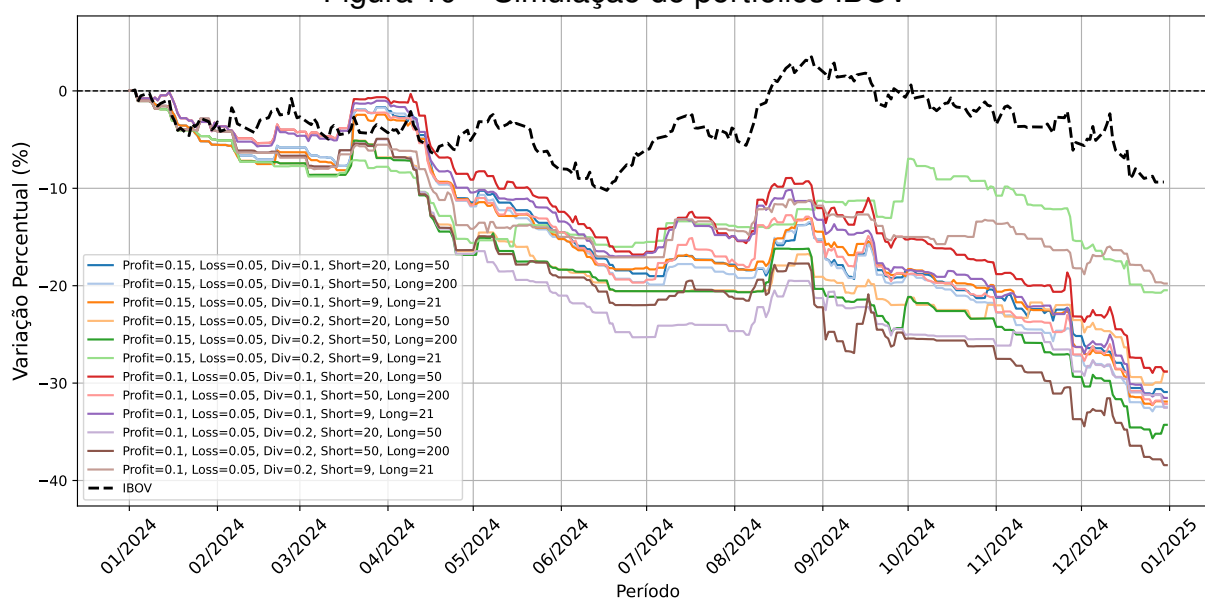
voláteis, diminuindo a exposição a quedas expressivas.

A pior configuração utilizou menor diversificação (0,2) associada a médias móveis longas (50, 200). Essa abordagem não apresentou bons resultados, pois a estabilidade dos preços no IBOV comprometeu a eficácia da estratégia, dificultando a identificação de movimentos relevantes de reversão.

De modo geral, os resultados indicaram que a estratégia baseada em médias móveis não foi a mais adequada para o IBOV, nesse contexto. Considerando-se a estabilidade nos preços e a tendência de queda, é possível observar que a con-

figuração atual não gerou os retornos esperados. No entanto, a ferramenta oferece flexibilidade para testar diferentes estratégias, permitindo que sejam exploradas abordagens alternativas que possam se adequar melhor ao comportamento do mercado. Esse é um dos principais diferenciais da ferramenta, que possibilita a análise de múltiplos cenários e a adaptação das estratégias conforme as condições de mercado. O gráfico da Figura 10 apresenta a variação de cada configuração no período em questão.

Figura 10 – Simulação de portfólios IBOV



Fonte: Elaborado pelo autor, 2025.

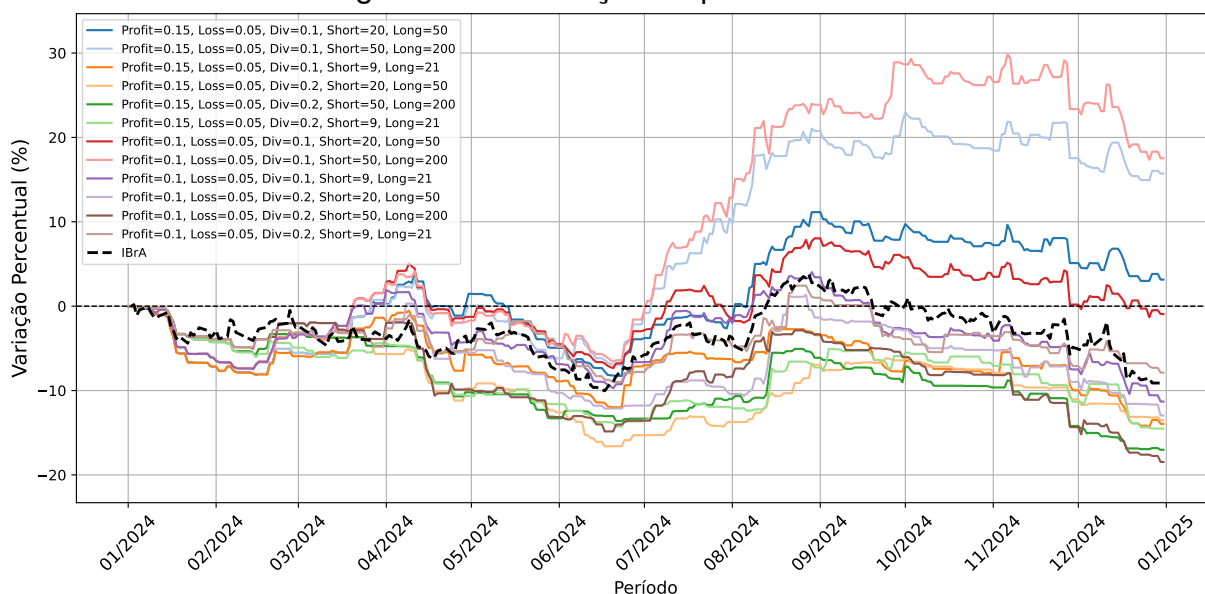
4.7.3 IBrA

Os resultados revelam um comportamento distinto do IBrA em relação ao IBOV. O índice apresentou menor volatilidade e uma distribuição mais equilibrada de retornos. A melhor configuração obteve um retorno de 17,53%, enquanto a pior registrou -18,46%. O gráfico da Figura 11 ilustra a variação do desempenho de cada configuração ao longo do tempo, destacando as diferenças no impacto das variadas estratégias. A Tabela 12 expõe os resultados obtidos na validação da estratégia utilizando ativos do índice IBrA.

O melhor desempenho foi observado com maior diversificação (0,1) e médias móveis longas (50, 200). Esse resultado sugere que tendências de longo prazo foram mais eficazes na captura de oportunidades dentro do IBrA, que possui uma composição diversificada e menos dependente de oscilações bruscas.

Em contrapartida, a pior performance ocorreu com menor diversificação (0,2) e médias móveis longas (50, 200). Essa abordagem mostrou-se desfavorável, possivelmente devido à concentração do capital nos mesmos ativos, reduzindo o im-

Figura 11 – Simulação de portfólios IBrA



Fonte: Elaborado pelo autor, 2025.

Tabela 12 – Resultados da validação no IBrA

Profit	Loss	Diversificação	Médias	Caixa	Portfólio	Retorno
0,10	0,05	0,1	[9, 21]	3,470792	8865,479241	-11,31%
0,10	0,05	0,1	[20, 50]	0,028930	9906,491872	-0,93%
0,10	0,05	0,1	[50, 200]	0,624381	11751,990289	17,53%
0,10	0,05	0,2	[9, 21]	0,789081	9211,412172	-7,88%
0,10	0,05	0,2	[20, 50]	0,143161	8703,956417	-12,96%
0,10	0,05	0,2	[50, 200]	0,027621	8153,726339	-18,46%
0,15	0,05	0,1	[9, 21]	0,496236	8604,468524	-13,95%
0,15	0,05	0,1	[20, 50]	40,294888	10274,674850	3,15%
0,15	0,05	0,1	[50, 200]	0,627241	11570,473863	15,71%
0,15	0,05	0,2	[9, 21]	0,915197	8548,180490	-14,51%
0,15	0,05	0,2	[20, 50]	1,240052	8645,189261	-13,54%
0,15	0,05	0,2	[50, 200]	0,532182	8297,229388	-17,02%

Fonte: Elaborado pelo autor, 2025.

pacto positivo dos movimentos favoráveis.

Desta forma, a análise indica que estratégias baseadas em médias móveis curtas não foram as mais eficientes para o IBrA. Por outro lado, as médias móveis longas revelaram bons resultados, pois mantiveram a variação do portfólio superior ao índice IBrA como um todo, sugerindo que a dinâmica desse mercado favorece abordagens de médio a longo prazo.

A validação reforça o papel da MARanker como uma estratégia eficiente para identificar pontos de reversão e priorizar ativos em conformidade com os princípios da

análise técnica. Baseada em conceitos de reversão à média, a *MARanker* demonstrou capacidade de capturar movimentos significativos do mercado, maximizando retornos ajustados ao risco. Sua integração à arquitetura de *backtesting* permitiu a simulação de cenários complexos, evidenciando como diferentes parâmetros influenciam o desempenho em contextos variados.

5 CONCLUSÃO

Conclui-se que o objetivo principal deste trabalho foi alcançado, com o desenvolvimento de uma ferramenta para *backtesting* de estratégias de investimento baseadas na ordenação de ativos. A solução proposta permitiu a realização de análises parametrizadas, simulações com dados históricos e comparações de estratégias em diferentes configurações.

Para atingir esse objetivo, foi desenvolvida uma arquitetura modular, composta por componentes para a aquisição de dados, processamento de estratégias e análise dos resultados. A implementação foi conduzida por meio de etapas estruturadas, incluindo levantamento de requisitos, modelagem, implementação e validação empírica.

A arquitetura implementada permite a integração de algoritmos de ordenação de ativos personalizados, como o *MARanker*, fundamentado no conceito de cruzamento de média móvel e médias móveis. O desempenho desse algoritmo foi avaliado em cenários simulados, destacando sua capacidade de identificar pontos de reversão e de ordenação. Além disso, a estrutura desenvolvida viabiliza a variação e a combinação de parâmetros, o que possibilita avaliar um amplo espectro de estratégias. A solução proposta permitiu a realização de análises parametrizadas, simulações baseadas em dados históricos e a comparação de estratégias com diferentes configurações, fornecendo subsídios para o entendimento das dinâmicas do mercado financeiro.

Duas principais dificuldades foram enfrentadas durante o desenvolvimento: a primeira foi a modelagem do sistema para torná-lo modularizado e genérico. O desafio consistiu em equilibrar flexibilidade e simplicidade, assegurando que a arquitetura pudesse suportar a integração de diferentes algoritmos de ordenação e configurações de estratégia sem comprometer a clareza e a manutenibilidade do código. Esse esforço foi essencial para garantir que o sistema pudesse ser adaptado a diferentes contextos de aplicação.

A segunda dificuldade foi manter a velocidade do sistema, visto que o uso de dados históricos em larga escala poderia resultar em lentidão durante o processamento. Para superar esse obstáculo, foi implementada a paralelização de algumas funções críticas, reduzindo significativamente o tempo de execução. Além disso, a classe *MemData* foi projetada para armazenar dados frequentemente acessados diretamente na memória, minimizando operações de leitura e escrita em disco e otimizando o desempenho do sistema.

Por fim, a arquitetura desenvolvida cumpre o objetivo de oferecer uma base metodológica para avaliar estratégias de investimento no mercado financeiro. Sua estrutura permite manipular dados históricos, ajustar múltiplos parâmetros e realizar testes sistemáticos, possibilitando a análise de diferentes cenários e hipóteses de mer-

cado. Torna-se uma ferramenta funcional para apoiar estudos e experimentos relacionados à avaliação de estratégias de investimento.

5.1 Trabalhos futuros

A arquitetura desenvolvida apresenta oportunidades significativas para expansão e aprimoramento. Uma possível evolução seria incluir outras abordagens de ordenação e estratégias mais sofisticadas, como aquelas baseadas em aprendizado de máquina.

Adicionalmente, a arquitetura pode ser aprimorada para incluir a ordenação de ativos também no momento da venda destes, uma funcionalidade atualmente implementada apenas para a compra. Isso possibilitaria uma análise mais completa de estratégias que consideram tanto a entrada quanto a saída no mercado. Atualmente, a arquitetura realiza a saída baseada no lucro ou prejuízo.

Uma possível direção para ampliar a aplicação da ferramenta seria o desenvolvimento de diversas aplicações *web* e dispositivos móveis. A criação de interfaces responsivas para diferentes plataformas e dispositivos portáteis garantiria maior acessibilidade e flexibilidade no uso das estratégias de investimento.

Outra evolução seria a criação de um conjunto de ordenadores pré-configurados. Sem necessidade de criar algoritmos do zero, poderia acelerar a implementação e a experimentação de diferentes abordagens.

Atualmente, a arquitetura realiza as operações com dados diários, o que é adequado para muitos cenários. No entanto, seria de bom proveito explorar a utilização de dados com uma granularidade menor, como dados de minutos ou horas. Isso permitiria uma análise mais detalhada e estratégias mais específicas, ajustadas a diferentes intervalos temporais e condições de mercado.

Por fim, o desenvolvimento de um sistema intermediário que converta linguagem natural ou de marcação em estratégias de investimento configuráveis representaria uma contribuição significativa. Essa funcionalidade simplificaria o uso da ferramenta para um público mais amplo, eliminando a necessidade de conhecimento técnico aprofundado para configurar estratégias no sistema.

REFERÊNCIAS

- ARAÚJO, P. H. d. C. **Criação de um método para balanceamento de estratégias automatizadas intradiárias para transação de contratos de derivativos**. 2021. Trabalho de Conclusão de Curso (Especialização em Gestão Estratégica) – Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, 2021. Disponível em: <http://hdl.handle.net/1843/38290>. Acesso em: 01/07/2024.
- ARNOTT, R.; HARVEY, C. R.; MARKOWITZ, H. M. A Backtesting Protocol in the Era of Machine Learning. **SSRN**, Elsevier, Rochester, 2018. Disponível em: <https://dx.doi.org/10.2139/ssrn.3275654>. Acesso em: 01/06/2024.
- AROUISSI, R. **yfinance**: Yahoo Finance Market Data in Python. 2024. Disponível em: <https://github.com/ranaroussi/yfinance>. Acesso em: 17/08/2024.
- ASSAF NETO, A. **Mercado Financeiro**. 12. ed. São Paulo: Atlas, 2014.
- BRASIL, BOLSA, BALCÃO (B3). **Quem Somos**. 2021. Disponível em: https://www.b3.com.br/pt_br/b3/institucional/quem-somos/. Acesso em: 21/04/2024.
- CAMPBELL, S. D. A review of backtesting and backtesting procedures. **Journal of Risk**, Infopro Digital Risk, London, v. 9, n. 2, p. 1–17, 2007. Disponível em: <https://doi.org/10.21314/JOR.2007.146>. Acesso em: 14/02/2025.
- CHRISTOFFERSEN, P. Backtesting. **SSRN**, Elsevier, Rochester, 2008. Disponível em: <https://dx.doi.org/10.2139/ssrn.2044825>. Acesso em: 01/06/2024.
- COMISSÃO DE VALORES MOBILIÁRIOS (CVM). **Sobre a CVM**. 2020. Disponível em: <https://www.gov.br/cvm/pt-br/acao-a-informacao-cvm/institucional/sobre-a-cvm>. Acesso em: 21/04/2024.
- EDWARDS, R. D.; MAGEE, J.; BASSETTI, W. H. C. **Technical Analysis of Stock Trends**. 11. ed. Boca Raton: CRC Press, 2018. Disponível em: <https://doi.org/10.4324/9781315115719>. Acesso em: 14/07/2024.
- FERREIRA, R. G. **O que é Renda Variável: Vale a Pena? Quais são os Riscos? Tudo o que Você Precisa Saber Antes de Investir**. 2019. Disponível em: <https://clubedovalor.com.br/o-que-e-renda-variavel/>. Acesso em: 19/04/2024.
- FREITAS, J. P. R. N. d. **Análise comparativa dos principais investimentos no mercado financeiro brasileiro para pessoas físicas**. 2020. Trabalho de Conclusão de Curso (Graduação em Gestão da Informação) – Universidade Federal de Uberlândia (UFU), Uberlândia, 2020. Disponível em: <https://repositorio.ufu.br/handle/123456789/30247>. Acesso em: 04/07/2024.

GERHARDT, T. E.; SILVEIRA, D. T. (org.). **Métodos de pesquisa**. Porto Alegre: Editora da UFRGS, 2009. Disponível em: <https://lume.ufrgs.br/handle/10183/52806>. Acesso em: 17/08/2024.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GOLDBERG, J.; NITZSCH, R. **Behavioral Finance**. Nova York: John Wiley & Sons, 2001.

LEMOS, F. A. C. d. A. **Análise Técnica dos Mercados Financeiros**. São Paulo: Saraiva, 2022.

LEMOS, F. A. C. d. A.; CARDOSO, C. **Análise Técnica Clássica**: Com as mais recentes estratégias da Expo Trader Brasil. São Paulo: Saraiva, 2010.

MARKOWITZ, H. M. Portfolio Selection. **The Journal of Finance**, Wiley, v. 7, n. 1, p. 77–91, 1952. Disponível em: <https://doi.org/10.2307/2975974>. Acesso em: 17/06/2024.

MARTINS, L. G. S. **Crescimento de pessoas físicas no mercado acionário brasileiro (2018-2022)**. 2023. Trabalho de Conclusão de Curso (Bacharel em Ciências Econômicas) – Pontifícia Universidade Católica de Goiás (PUC), Goiânia, 2023. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/6986>. Acesso em: 29/05/2024.

MASTEIKA, S.; RUTKAUSKAS, A. V.; JANES, A. A. Continuous futures data series for back testing and technical analysis. In: INTERNATIONAL CONFERENCE ON ECONOMICS BUSINESS AND MARKETING MANAGEMENT (CEBMM), II., 2012, Singapore. **Proceedings [...]** Singapore: IACSIT Press, 2012. p. 265–269. Disponível em: <https://hdl.handle.net/10863/23717>. Acesso em: 11/07/2024.

MCKINNEY, W. **Pandas Documentation**. 2024. Disponível em: <https://pandas.pydata.org/>. Acesso em: 17/08/2024.

METAQUOTES LTD. **MetaTrader 5**. Sobre a empresa. 2024. Disponível em: <https://www.metatrader5.com/pt/company>. Acesso em: 17/06/2024.

MURPHY, J. J. **Technical analysis of the financial markets: A Comprehensive Guide to Trading Methods and Applications**. New York: Penguin, 1999.

NI, J.; ZHANG, C. An Efficient Implementation of the Backtesting of Trading Strategies. In: INTERNATIONAL SYMPOSIUM ON PARALLEL AND DISTRIBUTED PROCESSING AND APPLICATIONS (ISPA), III., 2005, Nanjing, China. **Proceedings [...]** Berlin: Springer, 2005. v. 3758, p. 126–131. Disponível em: http://dx.doi.org/10.1007/11576235_17. Acesso em: 11/07/2024.

NTI, I. K.; ADEKOYA, A. F.; WEYORI, B. M. A systematic review of fundamental and technical analysis of stock market predictions. **Artificial Intelligence Review**, Springer, v. 53, p. 3007–3057, 2020. Disponível em: <https://doi.org/10.1007/s10462-019-09754-z>. Acesso em: 22/05/2024.

PEREIRA, P. L. V.; OLIVEIRA, A. B. Strategies of portfolio investment with estimates of bull and bear markets. **Brazilian Review of Finance**, Fundação Getulio Vargas (FGV), v. 19, n. 4, p. 160–185, 2021. Disponível em: <https://doi.org/10.12660/rbfin.v19n4.2021.80765>. Acesso em: 17/05/2024.

PRING, M. J. **Technical Analysis Explained: The Successful Investor's Guide to Spotting Investment Trends and Turning Points**. 5. ed. New York: McGraw-Hill Education, 2014.

QUANTCONNECT. **QuantConnect**. 2024. Disponível em: <https://www.quantconnect.com>. Acesso em: 21/07/2024.

RIBEIRO, M. R. **Python Avançado**. Belo Horizonte: IFMG, 2022. Disponível em: <http://hdl.handle.net/20.500.14387/1809>. Acesso em: 17/08/2024.

RIBEIRO, M. R. **Python Básico**. Belo Horizonte: IFMG, 2022. Disponível em: <http://hdl.handle.net/20.500.14387/1808>. Acesso em: 17/08/2024.

SANTOS, J. O. d.; SANTOS, J. A. R. d. Mercado de capitais: racionalidade versus emoção. **Revista Contabilidade & Finanças**, Universidade de São Paulo (USP), v. 16, p. 103–110, 2005. Disponível em: <https://doi.org/10.1590/S1519-70772005000100008>.

SARASA-CABEZUELO, A. Development of a Backtesting Web Application for the Definition of Investment Strategies. **Knowledge**, v. 3, n. 3, p. 414–431, 2023. ISSN 2673-9585. Disponível em: <https://doi.org/10.3390/knowledge3030028>.

SCHUMANN, E. Backtesting. **SSRN**, Elsevier, Rochester, 2018. Disponível em: <https://dx.doi.org/10.2139/ssrn.3374195>.

SCHWABER, K.; SUTHERLAND, J. **The 2020 Scrum Guide**. 2020. Disponível em: <https://scrumguides.org/scrum-guide.html>. Acesso em: 18/08/2024.

SILVEIRA, F. d. S. S. A.; RODRIGUEZ, A. R. S. C. Backtesting de estratégias de investimento com indicadores de análise técnica no mercado acionário brasileiro: estudo de caso. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO (SIMEP), V., 2017, Joinville. **[Anais...]** Joinville: UDESC/UNIVILLE, 2017. Disponível em: <https://dx.doi.org/10.29327/15421.5-25>. Acesso em: 14/07/2024.

STARKS, A. **NextTrade**. 2024. Disponível em: <https://github.com/austin-starks/NextTrade>. Acesso em: 21/07/2024.

TRADINGVIEW. **Gráficos em tempo real**. 2024. Disponível em: <https://br.tradingview.com/>. Acesso em: 29/04/2024.

WAINBERG, R. **O que é renda variável?** 2017. Disponível em:
<https://www.suno.com.br/artigos/renda-variavel-7-dicas/>. Acesso em: 19/04/2024.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. Rio de Janeiro: Elsevier, 2009.