

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ECONOMIA**

O USO DE ROBÔS NO MERCADO FINANCEIRO

Lucas Martins Stalter

JUIZ DE FORA
2023

Lucas Martins Stalter

O USO DE ROBÔS NO MERCADO FINANCEIRO

Trabalho de conclusão de curso apresentado ao curso de Ciências Econômicas da Universidade Federal de Juiz de Fora, como requisito parcial à obtenção do título de Bacharel em Ciências Econômicas

Orientador: Prof. Paulo César Coimbra Lisboa

JUIZ DE FORA

2023

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Stalter, Lucas Martins.

O uso de robôs no mercado financeiro. / Lucas Martins Stalter. -- 2023.

51 p. : il.

Orientador: Paulo César Coimbra Lisboa

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Juiz de Fora, Faculdade de Economia, 2023.

1. Robôs. 2. Algoritmo . 3. Day trade. 4. investidor . I. Lisboa, Paulo César Coimbra , orient. II. Título.



UNIVERSIDADE FEDERAL DE JUIZ DE FORA
REITORIA - FACECON - Depto. de Economia

FACULDADE DE ECONOMIA / UFJF

ATA DE APROVAÇÃO DE MONOGRAFIA II

NA DATA DE 13/07/2023, A BANCA EXAMINADORA, COMPOSTA PELOS
PROFESSORES

1 – PAULO CÉSAR COIMBRA LISBÔA - ORIENTADOR; E
2 – ROGÉRIO SILVA DE MATTOS,

REUNIU-SE PARA AVALIAR A MONOGRAFIA DO ACADÊMICO LUCAS MARTINS
STALTER,
INTITULADA: O USO DE ROBÔS NO MERCADO FINANCEIRO.

APÓS PRIMEIRA AVALIAÇÃO, RESOLVEU A BANCA SUGERIR ALTERAÇÕES AO
TEXTO APRESENTADO, CONFORME RELATÓRIO SINTETIZADO PELO ORIENTADOR.
A BANCA, DELEGANDO AO ORIENTADOR A OBSERVÂNCIA DAS ALTERAÇÕES
PROPOSTAS, RESOLVEU APROVAR A REFERIDA MONOGRAFIA.



Documento assinado eletronicamente por **Paulo César Coimbra Lisbôa, Professor(a)**, em 14/07/2023, às 15:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Rogério Silva de Mattos, Professor(a)**, em 15/07/2023, às 01:00, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1367691** e o código CRC **ABF1C1E1**.

RESUMO

Este trabalho de conclusão de curso apresenta um panorama geral da criação dos algoritmos de trading na bolsa de valores, seguido de uma linha do tempo apresentando alguns momentos históricos marcantes, como também mostra fatores psicológicos que afetam o investidor na hora de fazer suas operações, por fim, demonstramos um código de IA que realiza operações no mercado futuro bem como a sua análise expositiva sobre os resultados alcançados no *backtest*. O objetivo principal foi mostrar que o uso de robôs de trading no mercado financeiro é funcional e que pode evoluir ainda mais com a popularização e a evolução da plataforma, que neste trabalho foi utilizada a MetaTrader junto de sua linguagem, a MQL5, para a construção e demonstração de resultados. Usando uma estratégia de médias móveis chegamos a um resultado positivo dentro do *backtest* no período analisado, concluindo que os algoritmos de investimento podem ser sim satisfatórios desde que bem programado e calibrado, mas como todo investimento possui prós e contras e é critério de cada investidor decidir se essa modalidade combina com seu estilo de investir.

Palavras chave: Algoritmo, day trading, *backtest*, robô, *MetaTrader*, IA.

ABSTRACT

This undergraduate thesis provides an overview of the development of trading algorithms in the stock market, followed by a timeline highlighting significant historical moments. It also explores the psychological factors that influence investors when making trading decisions. Furthermore, it demonstrates an AI code that executes operations in the futures market, along with an expositional analysis of the results achieved in the backtest.

The main objective was to demonstrate the functionality of using trading robots in the financial market and how it can further evolve with the popularization and advancement of the platform. In this study, the MetaTrader platform and its programming language, MQL5, were utilized for building and showcasing the results. By employing a moving average strategy, a positive result was achieved during the analyzed backtest period, concluding that investment algorithms can indeed be satisfactory when well-programmed and calibrated. However, like any investment, there are pros and cons, and it is up to each investor to decide if this approach aligns with their investment style.

Keywords: Algorithm, day trading, backtest, robot, MetaTrader, AI.

Lista de ilustrações

Ilustração 1: Planilha de resultados do backtest.....	9
Ilustração 2: Fórmula MMA.....	24
Ilustração 3: Fórmula MME.....	24
Ilustração 4: Código Fonte.....	30
Ilustração 5: Estratégia de entrada.....	36
Ilustração 6: Variáveis globais.....	37
Ilustração 7: Função OnInIt/OnDnit.....	38
Ilustração 8: Função de compra e venda.....	39
Ilustração 9: Fechamento de ordens.....	40
Ilustração 10: Função OnTick.....	42
Ilustração 11: Variável bool.....	43
Ilustração 12: Horário limite.....	43
Ilustração 13: Configurações.....	44
Ilustração 14: Parâmetros de entrada.....	44
Ilustração 15: Resultados.....	45
Ilustração 16: Dados.....	45

Lista de gráficos

Gráfico 1: Black Monday	12
Gráfico 2: Mini-contratos da S&P ESMO, 06 maio de 2010	14
Gráfico 3: Volatilidade de mini-contratos de futuros S&P.....	15
Gráfico 4: Linhas Fibonacci.....	22
Gráfico 5: Projeção Fibonacci.....	22
Gráfico 6: Bandas de Bollinger.....	23
Gráfico 7: Média Móvel aritmética.....	25
Gráfico 8: Cruzamento de médias móveis.....	26
Gráfico 9: HiLo Activator.....	27
Gráfico 10: Índices.....	46
Gráfico 11: Dados.....	46
Gráfico 12: Evolução do patrimônio no período 01/01/2022 até 05/12/2022...	47

SUMÁRIO

1. INTRODUÇÃO	4
2. REVISÃO DE LITERATURA	6
3. METODOLOGIA	7
3.1 ALGORITMO	7
3.2 BACKTEST	7
4. HISTÓRICO	9
4.1. NASCIMENTO	9
4.2. CRISES	10
4.2.1. BLACK MONDAY	10
4.2.1.2 PORTIFÓLIO, FUTUROS, ROBÔS E QUEDA.....	11
4.2.2 FLASH CRASHS	12
5. A PSICOLOGIA POR TRÁS.....	16
5.1 VIESES	17
5.1.1 AVERSÃO A PERDA.....	18
5.1.2 EXCESSO DE CONFIANÇA	18
5.1.3 VIÉS DE ANCORAGEM.....	18
5.1.4 EFEITO MANADA	18
5.2 SOLUÇÃO	18
6. INDICADORES	20
6.1 FIBONACCI.....	20
6.2 BANDAS DE BOLLINGER.....	22
6.3 MÉDIA MÓVEL.....	23
6.3.1 MÉDIA MÓVEL ARITMÉTICA (MMA).....	23
6.3.2 MÉDIA MÓVEL EXPONENCIAL (MME).....	24
6.3.3 CRUZAMENTO DE MÉDIAS MÓVEIS.....	24
6.4 HILO ACTIVATOR.....	26
6.5 ÍNDICE DE FORÇA RELATIVA (IFR)	27
7. O ALGORITMO.....	29
8.2.1 PRIMEIRO BLOCO	34
7.2.2 SEGUNDO BLOCO	35
7.2.3 TERCEIRO BLOCO	36
7.2.4 QUARTO BLOCO	37
7.2.4.1 COMPRA E VENDA	37
7.2.4.2 FECHAMENTO DE ORDENS	39
7.2.4.3 VELAS E FUNÇÃO ONTICK	40
7.2.3 BACKTESTS.....	43
9. CONSIDERAÇÕES FINAIS	47
REFERÊNCIAS.....	48

1. INTRODUÇÃO

O robô de investimento é uma ferramenta que consiste em um conjunto de algoritmos e códigos que, por meio de uma plataforma computacional, realizam operações no mercado financeiro, analisando o mercado, e realizando as operações programadas para fazer. Ao contrário do que alguns podem pensar, já existem no mercado financeiro americano desde o ano de 1983, quando foi criado por Michael Bloomberg (GUPTA, 2015), e que, por mais que o uso exacerbado dessa tecnologia possa ter levado a algumas consequências, como o crash de 1987, já que na época existiram vários investidores inseguros que se desfizeram de suas posições, fazendo com que os robôs seguissem esse efeito manada e potencializando ainda mais a queda, que atingiu patamares de -20% ao dia, e ficou conhecido como “Black Monday” (SEGAL, 2021), além desse dia, muitos são convencidos que o “Flash Crash”, quebra que durou aproximadamente 36 minutos, que aconteceu em 2010 também ocorreu devido aos algoritmos de trading .

Observando esse acontecido, é normal pensar que o uso do mesmo foi descontinuado, mas conforme os anos passaram a porcentagem de robôs operando no mercado foi superando suas marcas e chegou a 70% nas operações de ações do mercado americano em 2013 (GUPTA, 2015). Logo, não foi só nesse mercado em que a porcentagem do volume de operações com automação foi crescendo, outros mercados também tiveram um crescimento no volume de operações realizada por robôs, como consequência, não só a complexidade por trás dessa tecnologia foi aumentando, como também foram ficando acessíveis aos mais diversos públicos, tendo plataformas próprias e especializadas, como a MetaTrader 5 (MT5), onde é possível, a partir de uma linguagem de programação chamada MQL5, o usuário criar seu próprio software de trading.

Assim, dentro dessa linguagem o investidor pode criar seu próprio algoritmo, que é "uma sequência de comandos executados por um software sob determinadas condições" (DUBART, 2015) e assim ter seu próprio robô realizando suas operações, ou seja, ele pode determinar a estratégia, o ativo, os

indicadores, gatilhos, além de fazer as modificações que julgar necessárias ao longo do tempo, não tendo limitações em sua criação e em seu uso. Visto isso, criaremos um algoritmo de trading na linguagem MQL5, passando desde a “base” do algoritmo até o seu desenvolvimento final, que trabalhará com base em operações de *day trade* e para isso é necessário um ativo que tenha uma alta liquidez e uma boa volatilidade, um bom exemplo seria os mini-contratos de futuros de dólar comercial americano, e como também existe uma alta quantidade de dados sobre suas cotações, é possível fazer *backtestes* longos e assim testar o robô de melhores formas.

Além desta introdução, este trabalho contém uma seção falando sobre a história e a evolução do algoritmo de trading, uma seção sobre as ferramentas que compõe e como será feita a construção do algoritmo, uma seção falando sobre como o psicológico do indivíduo pode afetar a sua tomada de decisão na hora de investir, uma que consta com estratégias que podem ser usadas no conjunto de um algoritmo junto de seus respectivos códigos e finalizando com a seção que aborda a construção do robô.

2. REVISÃO DE LITERATURA

O algoritmo de trading é algo um pouco mais antigo do que as pessoas costumam pensar, seja por não ser tão popular no país ou seja por receio dos investidores que pensam que é uma tecnologia recente, mas artigos escritos por McCracken (2015) nos mostram que os investidores e pesquisadores tentam desenvolver meio de automatizar as operações desde os anos 60, no entanto, foi apenas nos anos 80 que, segundo McCracken (2015) conseguiu desenvolver o primeiro terminal para tal, ainda, um pouco antes, um grupo chamado LOR conseguiu desenvolver um algoritmo que utilizava de futuros para assim criar um portfólio de segurança, que segundo Morris, que discutiu a relação entre esses portfólios com a Black Monday utilizando como fonte o livro *A First-Class Catastrophe: the road to Black Monday, the worst day in Wall Street History* de Diana B. Henriques (2017), que liga não só a utilização de robôs e algoritmos, mas também a intensa utilização de futuros nas transações e as turbulências que a economia americana estava passando algumas semanas antes.

No entanto, mesmo com esses problemas causados pela utilização de inteligência artificial, os robôs continuaram evoluindo, mas continuam com seus movimentos mecânicos, então qual seria o sentido de os usar se um humano consegue fazer o mesmo trabalho? A resposta, para Mark Douglas está no psicológico de cada um, o que atrapalha o processo de tomada de decisão, e por ser algo muito difícil de superar, muitos acabam desistindo do mercado, no entanto, com o uso de robôs é possível apagar o efeito de viés produzido pelo cérebro e assim conseguir ter resultados melhores do que os humanos. O interessante é que Mark Douglas (2000), em seu livro *Trading in the zone* liga vários desses vieses a experiências passadas na infância e que justificam ações da vida adulta de uma pessoa, fazendo ligação dessas com possíveis pensamentos e ações na hora de realizar operações.

Por fim, existem muitos artigos e profissionais que indicam várias estratégias para se usar em um algoritmo, no entanto, nenhum deles é realmente certo ou errado, mas sim de quanto risco e de quanto retorno ao investidor está disposto a se expor.

3. METODOLOGIA

Ao final deste trabalho, será mostrado a construção e os elementos que compõe a construção e testes que um algoritmo passa para provar o seu pleno funcionamento técnico e mecânico, para isso é necessário um conhecimento desde sobre o que é um algoritmo, que seria uma base inicial, e passando para os *backtests*, que vão ser os “testes” do robô.

3.1 ALGORITMO

Um algoritmo pode ser descrito como uma forma de organizar a sua lógica, ou seja, podemos dizer que é uma sequência de passos que devem ser seguidos de forma ordenada e sem ambiguidade, para que determinado problema seja resolvido, por exemplo, pegar um balde, seguir até um rio, enchê-lo e depois voltar para casa seria um algoritmo, que diferente do que muitos pensam não está somente ligado a programação, mas sim ao conjunto de ações que formam um padrão.

Aqui iremos utilizar um algoritmo de trading, ou seja, da mesma forma que o investidor faz a análise, toma uma decisão e, conseqüentemente, faz uma operação de compra ou de venda, esses que seriam os passos de um *trader*, o algoritmo de trading irá realizar essas ações a partir da linguagem MQL5.

Visto isso, é importante saber como se dará a construção do algoritmo em si, onde será utilizado o *MetaEditor 5*, uma ferramenta vinculada ao MQL5 onde é possível programarmos os códigos da linguagem MQL5 e conseguimos realizar os *backtests*, além de já conseguirmos importar para a própria plataforma de negociação da *MetaTrader*, que é o terminal mais acessível e amplo entre as diversas opções de linguagens e plataformas.

3.2 BACKTEST

Com o devido algoritmo programado, vamos para o que chamados de *backtests*, onde analisamos o comportamento da estratégia ao longo do tempo, consideramos mais confiável quanto maior o período de tempo testado, por exemplo, o período de um ano tem muito mais confiabilidade que um período de

6 meses por exemplo, e assim esses testes nos dão uma série de informações para assim julgarmos se determinada estratégia ou conjunto delas são confiáveis e consistentes.

Temos diversas informações importantes que recebemos dos *backtests*, como, por exemplo, os ganhos e as perdas, a curva de capital, que nos mostra a capacidade da estratégia tem de multiplicar o capital aplicado, o *drawdown* relativo e absoluto, onde um é referente quanto aos ganhos e depois as perdas e o outro é referente as perdas do total de capital inicial, e por fim temos a planilha de resultados, que nos traz a maior parte das informações.

Ilustração 1: Planilha de resultados do *backtest*



FONTE: INFOMONEY,2017

Visto todas essas informações, decidimos se a estratégia está pronta para ser usada ou se ainda é necessário fazer ajustes e repetir o processo até o robô estar bem calibrado para uso.

4. HISTÓRICO

Traders e pesquisadores, desde muito tempo, tentam automatizar o mercado de ações, um bom exemplo disso são as tentativas que ocorreram ao final de década de 60 com um aparelho chamado *Telequote III*. Tal foi lançado em 1964 e continha algumas funcionalidades que instigavam os investidores, como a recuperação de informações de alguma ação específica, como também fornece informações de vendas contínuas de várias ações simultaneamente (LACY, 1968), no entanto, mesmo com esse avanço tecnológico considerável, foi apenas em 82 que Michael Bloomberg teve sucesso em criar o primeiro mecanismo de automação no mercado de ações.

4.1. NASCIMENTO

Michael Bloomberg, nascido em 1942, era sócio geral do banco de investimento Salomon Brothers, o qual foi comprado por outra firma no ano de 1981, o que deu a Bloomberg um valor de 10 milhões de dólares que o ajudou a financiar a sua própria empresa de serviços de dados financeiros em 1982, a *Innovative Market Systems (IMS)* ou em tradução literal para o português o “sistema inovador de mercado”, tal empresa veio a se chamar Bloomberg LP em 1986. Além do valor recebido que o ajudou a financiar a IMS, Bloomberg também recebeu o financiamento de 30 milhões de dólares do bom norte-americano de investimentos Merrill Lynch para ajudar no desenvolvimento do terminal de Bloomberg (Gupta, 2015), e no final de 1982 as primeiras unidades originais de seu produto, que veio a ter o nome de *Market Master at First*, já estavam a serviço da Merrill Lynch (MCCRACKEN, 2015) a qual tinha 30% da empresa nos primeiros anos.

Essa relação fez com os dois tivessem um acordo de exclusividade, o que impedia Bloomberg de comercializar seu terminal para outros bancos e empresas, no entanto, depois de uma certa pressão imposta sobre a Merrill Lynch, esse acordo acabou e isso possibilitou um crescimento astronômico que variavam de 25 a 30% ao ano, além de continuar atualizando seu produto, como cores e telas planas (MCCRACKEN, 2015).

Um fato interessante, é que na época que Bloomberg começou seu projeto, as bolsas de Nova Iorque e de Tokyo já estavam se modernizando, ou seja, estavam ficando eletrônicas, que é necessário para outros serviços feitos de forma online e isso permitiu que o terminal de Bloomberg funcionasse sem problemas, tanto que o terminal funciona até atualmente, muito mais sofisticado e com mais funcionalidades, todavia, por mais que os robôs de investimento tenham dominado o mercado, chegando a fazer parte de 70% do mercado, eles podem ter causado alguns crashes no mercado desde o início do seu funcionamento.

4.2. CRISES

Muitos discutem o peso que os robôs tiveram em certas crises e são os principais culpados em “*flash crashes*”, como o de 2010 por exemplo. Podemos ressaltar que a primeira crise que tivemos envolvimento de robôs foi a *Black Monday*, que aconteceu em 1987.

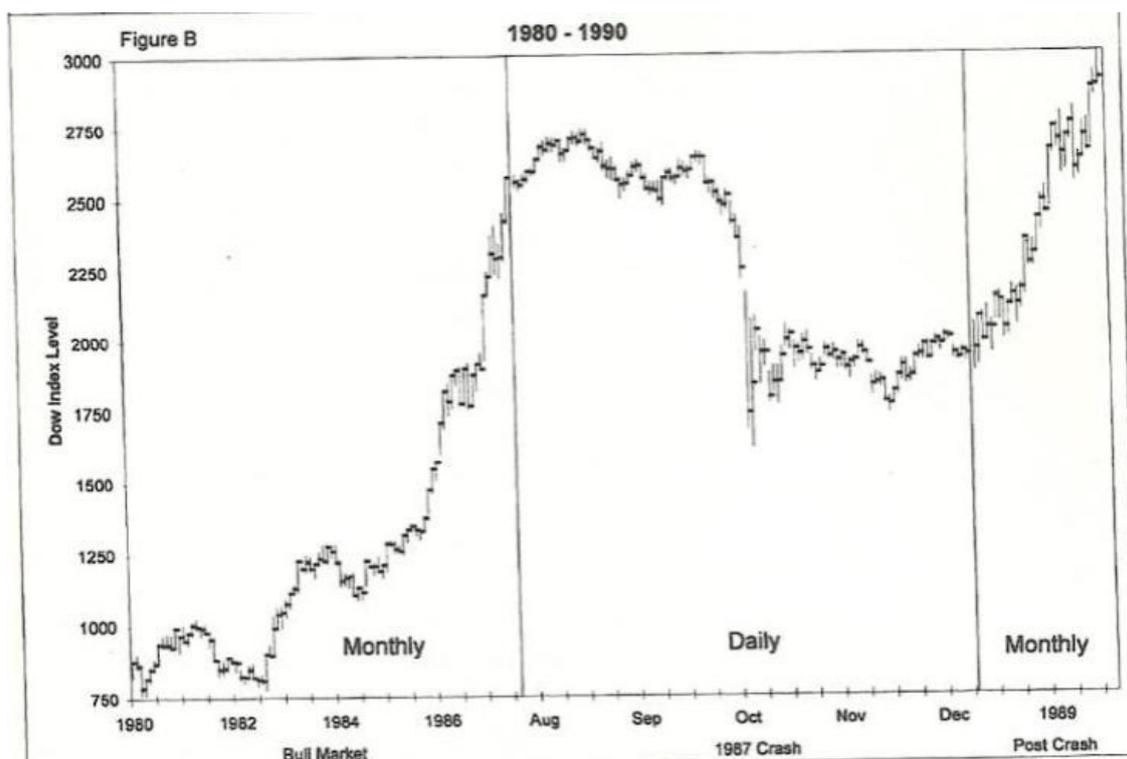
4.2.1. BLACK MONDAY

Poucos anos após a implementação dos primeiros robôs de trading no mercado financeiro por Bloomberg, mais especificamente em 19 de outubro de 1987 aconteceu o que os investidores chamam de “*Black Monday*”. Neste dia, as bolsas de várias cidades do mundo, como Nova Iorque, Tokyo, Hong Kong e Berlim vieram um rompimento de suporte que fez com que os mercados mundiais caíssem (SEGAL, 2021) e assim acontecendo o que consideram a maior queda da bolsa de Nova Iorque, com uma queda registrada de 22,6%, outras bolsas registraram uma queda considerável também, um exemplo seria a de Tokyo que teve uma queda de 15,6% (REIS, 2018). Em relação à causa, não se sabe ao certo o que realmente impactou diretamente na “*Black Monday*”, no entanto, por conta das exportações americanas apresentarem números cada vez menores, ocasionou em um aumento no déficit da conta corrente americana, chegando a um valor de 16 bilhões de dólares uma semana antes do crash, assim a bolsa americana começou a apresentar quedas

durante sucessivos dias até o dia 19 (REIS, 2018) onde aconteceu o rompimento de suporte e o mercado caiu de maneira desenfreada.

O gráfico abaixo (gráfico 1) mostra o período entre os anos 80 a 90, com uma abertura que representa só o ano de 1987, assim conseguimos ver claramente a queda e o impacto que a mesma teve.

Gráfico 1 – Black Monday



Fonte: Siegel (1988, p. 226)

4.2.1.2. PORTIFÓLIO, FUTUROS, ROBÔS E QUEDA

Tanto Henriques (2017) como Reis (2018) comentam sobre o peso que os contratos futuros, que em definição são “um acordo entre duas partes de comprar ou vender um ativo em uma determinada data no futuro, por um preço específico” (HULL, Pag 8), tiveram na “*Black Monday*”. Nessa época, no final dos anos 70 e início dos 80, algumas bolsas como a *Chicago Board Trade* e a *Chicago mercantile Exchange (Merc)* começaram a negociar futuros de vários outros ativos, como taxa de juros, títulos hipotecários e índices de ações. Tais dominaram o mercado e as ideias de mercados autocorretivos, a chegada dos

computadores como também a entrada de sistemas tecnológicos nas bolsas de valores afirmava e sinalizava a modernização nos mercados.

Vendo essas mudanças, a companhia *Leland O'Brien Rubinstein Associates* (LOR) criou um “portifólio de insegurança”, que era um conjunto de algoritmos que Morris (2017) explicou como:

“was a set of algorithms that clicked in during a market downturn to limit losses. When the insurance algorithms were triggered, computers would sell futures to lock in a pricing floor, and then reverse the process as markets recovered”.

O autor completa que existia uma “enorme escala” de carteiras protegidas por seguros e que isso poderia causar um grande volume de vendas acionadas por robôs que poderia causar uma sobrecarga nas operações, o que de fato, aconteceu. Quanto mais os preços dos ativos caíam, mais os algoritmos de seguros aceleravam suas vendas, com isso, faziam os preços despencarem ainda mais, fazendo os preços de alguns ativos chegar a zero.

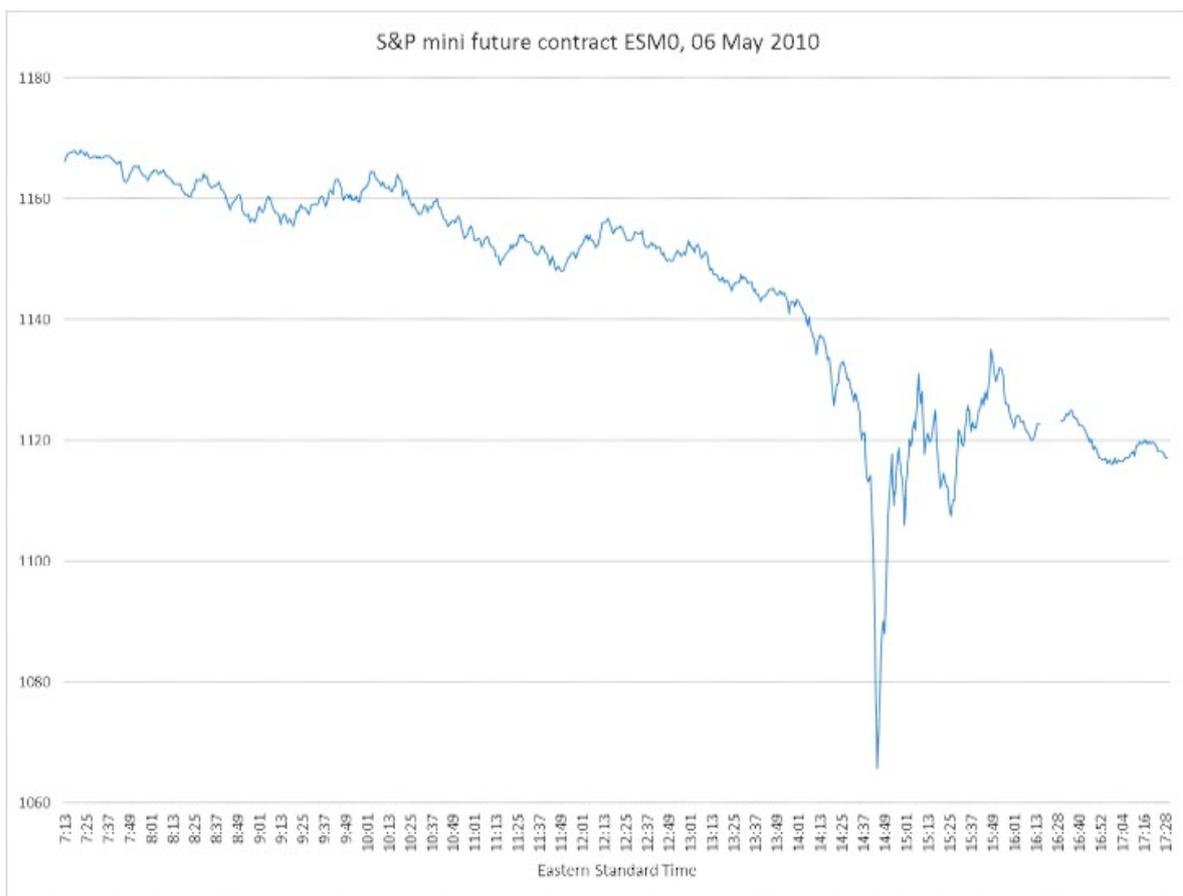
Visto isso, podemos resumir a *Black Monday* como uma ingenuidade tecnológica, já que os investidores e as empresas de “seguros de capitais” usaram para proteger os seus investimentos, o que parecia tudo normal até certo ponto. No entanto, com a intensa utilização de futuros para isso, por exemplo, o capital da LOR que protegia 80% de seus ativos utilizando o mercado de futuros (LELAND, RUBISTEIN, 2016), essa massa acarretou nessa quebra histórica da bolsa de Nova Iorque.

4.2.2. FLASH CRASHS

Conceitualmente, os *flashs crashes*, ou queda relâmpago em português, são quedas bruscas de preços, mas que ao mesmo tempo que caem de maneira extraordinariamente rápida, conseguem se recuperar na mesma proporção e fecham o dia no mesmo nível de preço que estava antes da queda, parecendo que o mesmo nunca aconteceu.

Um exemplo famoso sobre isso, é a queda relâmpago que aconteceu em maio de 2010 na S&P500, onde conseguimos ver que o dia abriu com uma leve queda, que se intensificou depois que rompeu a linha dos 1140 pontos as 14:01 e derretendo após abaixar dos 1120 pontos, até que as 14:37 temos uma reta para baixo, no entanto, os preços começam a se recuperar logo em seguida, chegando e ultrapassando o patamar dos 1120 pontos as 14:49, continuando a oscilar até o fechamento. No entanto, o nível de preço se manteve no mesmo, ou seja, tivemos uma recuperação em mais ou menos 23 minutos. É interessante analisar junto disso que a volatilidade chegou a 300% durante a queda, seguido de outro pico menor até chegar a um patamar normal de oscilação (LARANJEIRA, 2016).

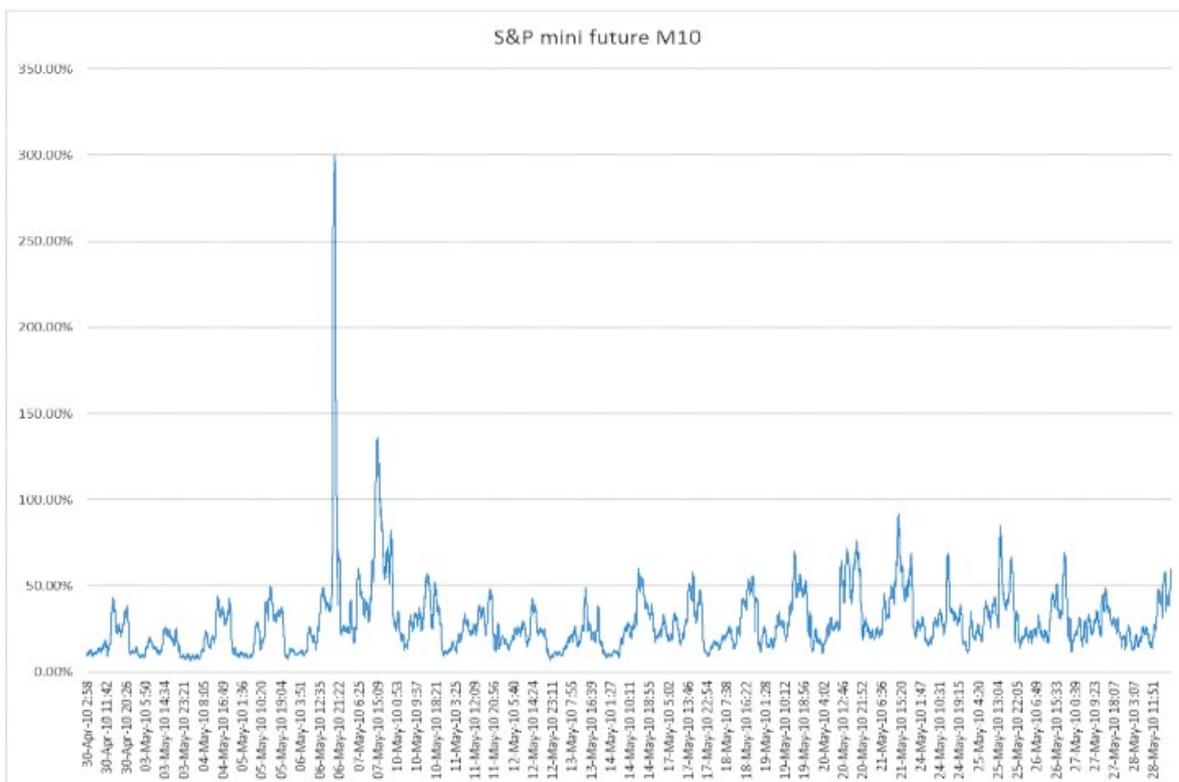
gráfico 2 – Mini-contratos¹ da S&P ESMO, 06 maio de 2010



Fonte: GIANT STEPS CAPITAL, 2016

¹ Instrumento financeiro derivativos negociados no mercado de futuro que permite que os investidores especulem sobre a variação do preço do dólar em uma escala menor do que no contrato cheio.

gráfico 3: Volatilidade de mini-contratos de futuros S&P



Fonte: GIANT STEPS CAPITAL (2016)

Visto esse exemplo, conseguimos ter uma ideia da agressividade que uma queda relâmpago tem quando vemos o volumoso número de ordens, como também, a velocidade que essas ordens têm que ter pra conseguir atingir esse número de volatilidade. Segundo Will Kenton (2022), especialista em economia, leis de investimento e regulações, os *flashes crashes* são “exacerbados” por aberrações de mercado que normalmente acontecem por vendas pesadas de robôs de alta frequência, que são gatilhos de venda de grandes quantidades de ativos para evitar que ocorram perdas. Kelton (2022) ainda fala que quanto mais digitalizada as negociações ficam, maior a probabilidade de outras quedas relâmpagos acontecerem, já que essas falhas são resultados de algoritmos ao invés de quedas convencionais que normalmente tem uma notícia, empresa ou política envolvida, e isso pode desencadear um efeito dominó, resultando que o algoritmo não é o único efeito que pode causar isso, mas que também podem existir outros fatores que contribuem para tal, no entanto, pode ser difícil identificá-lo.

Por fim, as bolsas mundiais tentam utilizar mecanismos para evitar essas quedas, que vem aumentando de maneira considerável nos ultimas anos, um deles seria a utilização dos *circuit breakers*, que é uma “pausa” ou parada no sistema quando as quedas chegam a certos níveis, no entanto, estimativas apontam que mesmo assim cerca de 12 *mini-flash crashes* acontecem no mercado por dia (ALBUQUERQUE, 2021).

5. A PSICOLOGIA POR TRÁS

Para fazer operações não é necessário apenas técnica e conhecimento das ferramentas e das estratégias usadas, é preciso apenas uma plataforma que tenha acesso a algum ativo de interesse do investidor e ele apertar um botão que irá realizar a compra ou venda, por exemplo, o *trader* pode comprar um ativo fictício que esteja a R\$ 2.00 e o vender a R\$ 3.00, e ele pode usar apenas da sua vontade para isso, no entanto, os indivíduos de modo geral, não conseguem reproduzir fielmente as técnicas de trading, o que acaba fazendo com que o *trader* tenha prejuízos constantes no mercado o que leva a muitos desistirem da atividade de *day trading*. É isso que a pesquisa de Fernando Chague (2020), Ph. D em economia pela Universidade da Carolina do Norte e de Bruno Giovannet (2020), Ph. D em economia pela Columbia University, nos mostra as estatísticas das pessoas que começaram as atividades em contratos futuros de mini índice.

Esse estudo nos mostra três pontos relevantes, sendo o primeiro que mostra que 99,43% dos *traders* não continuam operando, assim apresentando operações em menos de 300 pregões, a segunda mostra que os *traders* que continuaram apresentando operações tem um saldo final negativo e a última nos apresenta que, apenas ,127 indivíduos apresentaram um lucro bruto médio diária acima de 100 reais nos mais de 300 pregões analisados. Os pesquisadores ainda chegam a algumas conclusões em cima desses resultados, sendo que mesmo que levassem em conta um possível nível de aprendizado nos primeiros 200 pregões, os resultados ainda seriam semelhantes. Já em relação aos 127 ganhadores, seus lucros variavam entre 103,00 reais para o indivíduo com o menor lucro médio bruto diário a até 4.032,00 reais para o que teve o maior lucro médio bruto diário.

Visto esses dados, é comum se perguntar o que impede que os *traders* tenham um resultado verdadeiramente consistente? A resposta para isso, segundo, Mark Douglas (2000), está na mente do investidor, ou seja, existe uma estrutura de funcionamento em nossas mentes e essas não se adequam as características do mercado

Para Mark Douglas (2000), existe algo que separa os *traders* de sucesso dos *traders* que fracassam, e isso seria o seu modo de pensar, ou seja, além de

todas as técnicas aprendidas, ainda é necessário “pensar como um *trader*”, em outras palavras, o investidor tem que “alcançar um estado mental” que permita que mesmo em condições adversas o investidor consiga fazer o que ele tem que fazer. Já essas condições seriam, em conceito, os fatores que podem prejudicar o indivíduo no ato, em um cenário de mercado, fazer a sua operação da maneira que planejou fazer. Vamos ignorar as situações adversas externas ao *trader* e focar em algumas que estão ligadas diretamente as experiências vividas e pela mentalidade do indivíduo, logo, esses são chamados de vieses.

5.1 VIESES

Podemos falar que os vieses são “falhas no sistema” do nosso cérebro, “pré-julgamentos” que influenciam no processo de tomada de decisão dos indivíduos. Isso ocorre por que o nosso cérebro usa de atalhos para tomar decisões e assim fazer com que o processamento das ações aconteça de forma mais rápida com menos gasto de energia, no entanto, por mais em um primeiro momento essa descrição pareça algo bom, ela acaba atrapalhando no processo de tomada de decisão.

Esses vieses são algo que todos os investidores tem, sendo em menor ou maior escala, dependendo das experiências de vida de cada indivíduo, Segundo Mark Douglas (2000), esses vêm de experiências passadas, traumas e medos que foram desenvolvidos ao longo da vida, como por exemplo, uma pessoa que tem muito medo de cachorros pode ter passado por uma má experiência da infância e assim seu cérebro, para evitar sofrer da mesma dor, faz com que a pessoa evite cachorros o máximo possível, trazendo isso para a realidade dos investimentos, sejam quais tipo sejam, Mark Douglas(2000) separa os investidores em 3 tipos, os que já apresentam ganhos frequentes, os que passam pela “montanha Russa”, ou seja, apresentam ganhos e perdas constantes, e por último, os que apresentam apenas perdas constantes, sendo os que superaram os vieses, os que estão no processo, e os que ainda tem os vieses muito presentes nas operações, esses que se criam logo nas primeiras operações, nas primeiras perdas e nos primeiros ganhos, fazendo o cérebro criar os vieses para assim proteger a dor que o investidor poderá vir a sofrer em decorrência dos resultados das próprias operações.

Mark Douglas (2000) fala que é possível superar esses vieses, e apresenta uma série de exercícios em seu livro “*Trading in the Zone*”, onde aprofunda ainda mais nas naturezas, origens e soluções para eles. A seguir estão alguns dos vieses que mais atrapalham os investidores em suas operações.

5.1.1 AVERSÃO A PERDA

Na aversão a perda o investidor, por medo de tomar prejuízos, não busca investimento com maior rentabilidade, assim, fazendo escolhas irracionais para evitar a dor da perda, como consequência, o *trader* pode vender ações que estão com boa rentabilidade com medo que elas caiam, do mesmo jeito que pode manter operações perdedoras para tentar conseguir algum ganho no futur.

5.1.2 EXCESSO DE CONFIANÇA

O excesso de confiança acontece com mais frequência em investidores mais experientes, que pensam que já dominam o mercado por fazerem uma boa quantidade de operações positivas, o que pode resultar em análises erradas por não acreditar em possíveis projeções ou profissionais e seguir somente o seu “*feeling*”.

5.1.3 VIÉS DE ANCORAGEM

Esse viés se dá quando o investidor utiliza de alguma experiência ou dados passados para tentar explicar suas ações no presente ou no futuro, por exemplo, se o preço de um ativo era \$50 e aparece uma nova empresa no mercado do mesmo ramo, o agente acredita que o preço desse novo ativo pode chegar no mesmo nível de preço.

5.1.4 EFEITO MANADA

Como o nome já diz, o “efeito manada” ocorre quando um grupo muito grande de investidores seguem um grande player ou simplesmente seguem um fluxo muito grande de compras ou vendas, fazendo com o preço de um ativo suba ou desça rapidamente.

5.2 SOLUÇÃO

Existem algumas formas de solucionar os vieses, mas para isso é necessário um período de tempo e nem sempre os investidores conseguem “se livrar” totalmente deles, para isso, uma forma simples, seria o uso de robôs para fazer o trabalho do investidor, já que por meio de algoritmos é possível replicar uma operação que um humano faria, no entanto, sem as emoções e sem os vieses para atrapalhar e assim tendo operações quase perfeitas.

6. INDICADORES

Os investidores, de maneira geral, possuem suas próprias estratégias e maneiras de lerem o mercado, sejam elas fundamentalistas, técnicas ou até mesmo o que alguns chamam de “*tape reading*”. De maneira resumida, a análise fundamentalista leva em consideração o estudo da situação financeira, econômica e das perspectivas da empresa, assim tentando identificar o potencial de crescimento do lucro da empresa no futuro, podemos dividir a análise em 3 grandes partes, a macroeconômica, a setorial e a própria empresa. Já a análise técnica, também chamada de grafista, leva em conta padrões passados pelos ativos, além de utilizar de ferramentas, também chamadas de indicadores, que ajudam nessa leitura, e assim fazer análises e operações com base nesses padrões, por fim, o “*tape reading*” usa de certas ferramentas para ler o mercado e as operações, de uma maneira mais rústica, o “*tape reading*” faz a leitura por dentro do gráfico da análise técnica.

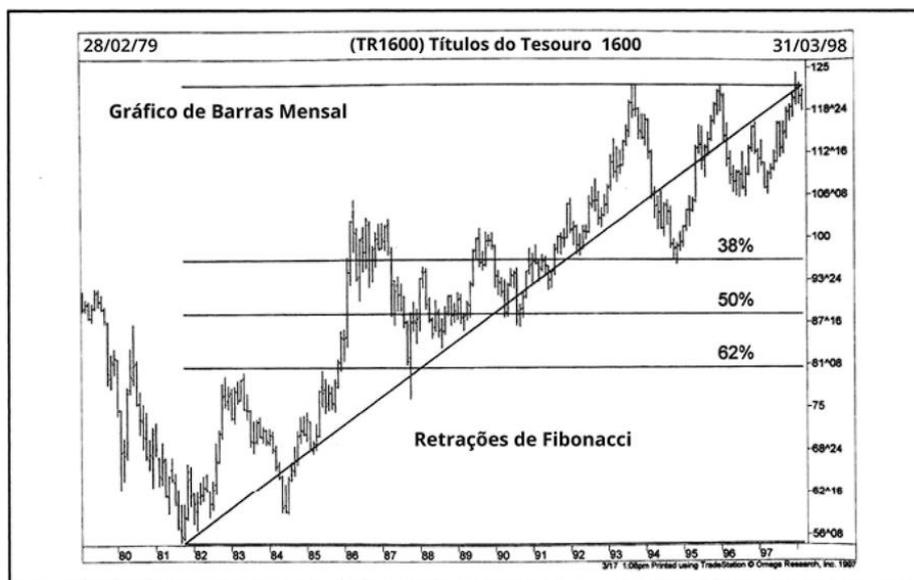
No algoritmo de *trading*, o robô utiliza de códigos de programação para reconhecer os padrões da análise gráfica e assim exercer as suas operações, podendo contar uma ou mais estratégias que se utilizam de indicadores para tornar possível cada uma delas e dentro das centenas que existem.

6.1 FIBONACCI

Esse indicador vem da teoria de sequência de Fibonacci, que é uma teoria matemática desenvolvida pelo matemático Leonardo Fibonacci (1170-1250), e é constituída por uma série de números que, começando entre 1 e 1, os próximos números sempre serão resultado das somas dos anteriores, ou seja, 1,1,2,3,5,8 e continuando até o infinito (TORO INVESTIMENTOS, 2022).

Essa teoria envolve mais algumas propriedades, no entanto, vamos focar no que tange as retrações que envolve o indicador Fibonacci, onde Murphy (1986, pag. 454) afirma que números usados com mais frequência para análise na retração são o 62%, 38%, e 50% e que levando em conta a repetição de movimento de mercado, temos os números 33%, 50% e 67% e ainda usa como exemplo que em retrações mínimas fica em torno de 38% e nas máximas em torno de 62%.

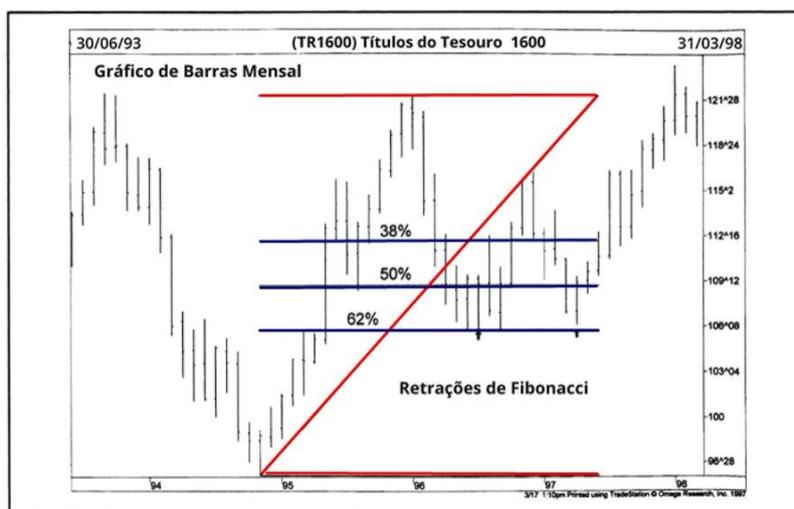
Gráfico 4: Linhas Fibonacci



Fonte: Murphy (1986, p. 457)

Quando falamos em alvos para as operações usando o Fibonacci procuramos utilizar das linhas traçadas a partir de topos e fundos significados para assim termos onde marcar as linhas de retração.

Gráfico 5: Projeção Fibonacci



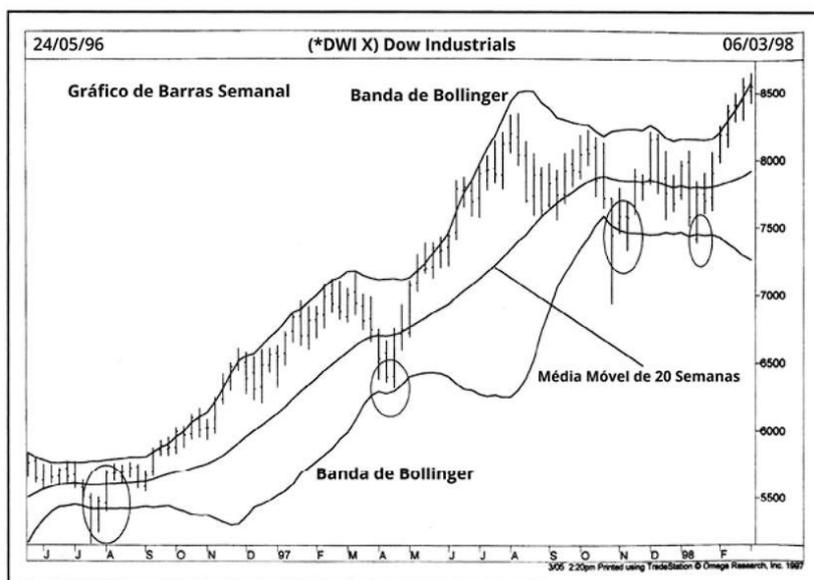
Fonte: Murphy (1986, p. 458)

6.2 BANDAS DE BOLLINGER

Criado por Bollinger na década de 80, as bandas de Bollinger funcionam com dois desvios-padrão nas extremidades e uma média móvel ao meio, assim conseguimos ver a dispersão em volta do preço médio, ou seja, quando temos uma volatilidade baixa, acontece uma contração nas bandas, e quando temos uma volatilidade mais alta, as bandas se abrem, assim conseguimos localizar os pontos onde está acontecendo a sobrevenda e a sobrecompra do ativo em questão (Murphy, 2021).

Visto isso, normalmente as bandas são usadas como alvos, explicando de maneira mais clara, assim que o preço do ativo passa a média para cima, a banda superior se torna o alvo do “preço máximo” e quando passa para baixo da média temos o “preço mínimo”, o que pode ser até usado para observar uma possível reversão de tendência, Murphy (2021, p. 300) utiliza o seguinte exemplo: “Em uma tendência forte de alta, os preços geralmente flutuam entre a banda superior e a média de 20 dias, nesse caso, um cruzamento abaixo da média de 20 dias alerta para uma reversão de tendência para baixo. “

Gráfico 6: Bandas de Bollinger



Fonte: Murphy (1986, p. 300)

6.3 MÉDIA MÓVEL

Em seu livro, “Análise Técnica do Mercado Financeiro”, Jonh J. Murphy (1986, p. 282) define a média móvel como sendo “a média em um certo conjuntos de dados”, logo, quando esse conjunto se movimenta ao longo do tempo, o resultado dessa média acompanha o movimento da série temporal e assim conseguimos seguir e identificar as novas tendências, enquanto elas se formam.

Visto isso, podemos escolher uma série de tempo que varie 10 ou até 200 dias, podendo o investidor escolher o período que melhor se encaixe no seu operacional, no entanto, vale lembrar que medias longas tem uma sensibilidade menor em relação as mais curtas, o que pode ser menos ou mais vantajoso dependendo do cenário escolhido.

6.3.1 MÉDIA MÓVEL ARITMÉTICA (MMA)

Também chamada de média Móvel simples, a média móvel aritmética (MMA) representa o valor médio de um determinado ativo em determinado tempo, utilizando, normalmente, os preços de fechamento. (NELOGICA)

Murphy (1986) aponta duas críticas sobre essa Média, uma é que “apenas o período coberto pela média é levado em consideração” (1986, pág. 285) e a segunda pode ser melhor entendida ao ser vista junto de sua formula:

Ilustração 2: Fórmula MMA

$$MMA = \frac{V_1 + V_2 + \dots + V_N}{N}$$

Fonte: Nelogica (s.d)

Observando a fórmula acima, vemos que todos os valores tem o mesmo peso, ou seja, compartilham de igual importância dentro do *price action*.

Gráfico 7: Média Móvel aritmética



Fonte: Murphy (1986, p. 285)

6.3.2 MÉDIA MÓVEL EXPONENCIAL (MME)

Diferente da MMA, a Média Móvel Exponencial tenta resolver o problema do *price action* utilizando de uma ponderação da série temporal, ou seja, atribui uma importância maior aos fechamentos mais recentes e menores para os antigos, podendo ser visível com a fórmula abaixo:

Ilustração 3: fórmula MME

$$MME = Preço \cdot K + MME_{ontem} \cdot (1 - K)$$

$$K = \frac{2}{N+1}$$

Fonte: Nelogica (s.d)

6.3.3 CRUZAMENTO DE MÉDIAS MÓVEIS

O cruzamento de médias móveis é caracterizado pelo uso de duas linhas simultâneas com períodos distintos, sendo uma mais curta e uma mais longa,

assim temos o comportamento no curto prazo (média curta) e no longo prazo (média longa). Passando para parte mais prática, fazemos operações quando essas linhas se cruzam, por exemplo, um sinal de entrada seria quando a curta cruzar acima da longa, indicando uma possível tendência de alta, e caso esse cruzamento ocorrer com a longa acima, indica que o mercado pode estar inclinando para uma tendência de baixa o que indica um ponto de venda (ISMAR, 2021). Ainda existem outras formas de usar esse cruzamento, no entanto, vamos nos concentrar nesse caso por enquanto.

Gráfico 8: Cruzamento de médias móveis



Fonte: Nelogica (s.d)

6.4 HILO ACTIVATOR

O HiLo Activator é outro dos indicadores de tendencia que é muito usado pelos *traders*, e segundo Pinheiro (2019) esse indicador é:

” Basicamente as linhas do Hilo Activator são duas médias móveis deslocadas um período para o futuro. A linha inferior utiliza os valores das mínimas dos *candles* anteriores em sua base de cálculo. E da mesma forma a linha superior utiliza os valores das máximas dos *candles* para ser composta.”

Assim é necessário que o *trader* observe o sinal de virada do HiLo para realizar as suas operações, por exemplo, ao ver a linha superior “surgindo” o indicador está indicando uma possível tendência de queda, e quando a linha inferior surge, pode estar se formando uma possível tendência de alta.

Gráfico 9: HiLo Activator



Fonte: Neologica (2019)

6.5 ÍNDICE DE FORÇA RELATIVA (IFR)

Desenvolvido pelo engenheiro mecânico J.welles Wilder Jr.(1978) que em seu livro “*New Concepts in Technical Trading System*” descreve o IFR como uma ferramenta que pode dar uma nova dimensão a interpretação grafista quando é plotada junto de um gráfico diário e ainda adiciona alguns fatores de interpretação, os quais são:

- 1- Os topos e fundos são indicados quando o indicador está acima de 70 ou abaixo de 30;
- 2- Formas gráficas que podem aparecer no indicador podem não aparecer no gráfico de *candle*;
- 3- Oscilações de falha acima do 70 e abaixo do 30 podem ser falsas indicações de reversão de mercado;
- 4- Suportes e resistências aparecem antes e mais claramente no IFR do que no gráfico de *candle*;
- 5- Uma divergência entre o indicador e o *price action* é um ponto forte de indicação de ponto de inflexão.

Wilder(1978) ainda, antes de começar com a real explicação de seu indicador, comenta sobre o oscilador de momentum, no qual o IFR é baseado, e aponta para 3 problemas principais desse oscilador, o primeiro é que o oscilador faz um movimento errado dentro da sua configuração, o segundos é que não sabemos o quão alto é o alto e o quão baixo é o baixo, para assim poder fazer as operações e o terceiro é que torna necessário ter uma quantidade da dados enormes, e pra resolver isso Winder propôs a seguinte formula:

Ilustração 3: Formula IFR

The equation for the Relative Strength Index, RSI, is:

$$RSI = 100 - \left[\frac{100}{1 + RS} \right]$$

$$RS = \frac{\text{Average of 14 day's closes UP}}{\text{Average of 14 day's closes DOWN}}$$

Fonte: Wilder Jr. (1978, p. 65)

Para chegar a esta fórmula, Wilder seguiu alguns passos, como pegar os fechamentos positivos e negativos e dividir cada um por 14, fazendo uma média dos fechamentos e dividindo os positivos pelos negativos, adicionando 1,00 para dividir por 100 e depois dividindo por 100 novamente.

No entanto, de forma mais prática, o investidor usa o indicador para achar os pontos de sobrevenda e sobrecompra e assim ter uma ferramenta a mais para usar em suas análises e operações.

7. O ALGORITMO

Um algoritmo, segundo Ziviani (2011, pag 1) é uma “sequência de ações executáveis para a obtenção de uma solução para um determinado tipo de problema” e que pode ser representado de diversas maneiras, como em uma lista de instruções, um diagrama de fluxo ou, como neste caso, em um código de programação, é importante ressaltar que o mesmo não está ligado só a elementos de computação, mas sim no dia-a-dia das pessoas.

No caso deste trabalho, foi criado um algoritmo, dentro da linguagem MQL5, que busca fazer operações automatizadas dentro do mercado financeiro, mais especificamente, busca fazer operações de *day trade* no ativo do mini dólar futuro, que costuma ser mais liquido e tem uma boa volatilidade diária. O código está apresentado a seguir:

Ilustração 4: Código Fonte

```

14 enum ESTRATEGIA_ENTRADA
15     {
16         MEDIA_MOVEL
17     };
18 //--
19 // input
20
21 // input
22 sinput string Entrada;
23 input ESTRATEGIA_ENTRADA      estrategia = MEDIA_MOVEL;
24
25 sinput string s1;
26 input int mm_rapida_periodo    = 14;
27 input int mm_lenta_periodo    = 50;
28 input ENUM_TIMEFRAMES mm_tempo_grafico = PERIOD_CURRENT;
29 input ENUM_MA_METHOD mm_metodo = MODE_EMA;
30 input ENUM_APPLIED_PRICE mm_preco = PRICE_CLOSE;
31
32 sinput string lotes;
33 input int num_lots            = 5;
34 input double TK              = 6;
35 input double SL              = 4;
36
37 sinput string hora;
38 input string hora_limite_fecha_op = "17:00";

```

```

39
40
41
42 int mm_rapida_Handle;
43 double mm_rapida_Buffer[];
44
45
46 int mm_lenta_Handle;
47 double mm_lenta_Buffer[];
48
49
50 int expert_magic = 1;

52 MqlRates velas[];
53 MqlTick tick;
54
55 CSymbolInfo m_symbol;
56
57
58 bool emOperacao=false;
59 bool acionouEntrada=false;
60
61 int OnInit()
62 {
63
64     mm_rapida_Handle = iMA(_Symbol,mm_tempo_grafico,mm_rapida_periodo,0,mm_metodo,mm_preco);
65     mm_lenta_Handle = iMA(_Symbol,mm_tempo_grafico,mm_lenta_periodo,0,mm_metodo,mm_preco);
66
67
68     CopyRates(_Symbol,_Period,0,4,velas);
69     ArraySetAsSeries(velas,true);
70
71
72     ChartIndicatorAdd(0,0,mm_rapida_Handle);
73     ChartIndicatorAdd(0,0,mm_lenta_Handle);
74
75     m_symbol.Name(_Symbol);
76     m_symbol.Refresh();
77
78     ///---
79     return(INIT_SUCCEEDED);
80 }

82 void OnDeinit(const int reason)
83 {
84     ///---
85     IndicatorRelease(mm_rapida_Handle);
86     IndicatorRelease(mm_lenta_Handle);
87 }
88
89 void OnTick()
90 {
91     ///---
92
93     if(acionouEntrada && !TemPosicoesAbertas())
94     {
95         acionouEntrada = false;
96         emOperacao = false;
97     }
98
99     CopyBuffer(mm_rapida_Handle,0,0,4,mm_rapida_Buffer);
100    CopyBuffer(mm_lenta_Handle,0,0,4,mm_lenta_Buffer);
101
102
103
104    CopyRates(_Symbol,_Period,0,4,velas);
105    ArraySetAsSeries(velas,true);
106
107
108    ArraySetAsSeries(mm_rapida_Buffer,true);
109    ArraySetAsSeries(mm_lenta_Buffer,true);

```

```

112 SymbolInfoTick(_Symbol,tick);
113
114
115 bool compra_mm_cros = mm_rapida_Buffer[0] > mm_lenta_Buffer[0] &&
116                      mm_rapida_Buffer[2] < mm_lenta_Buffer[2] ;
117
118
119 bool venda_mm_cros = mm_lenta_Buffer[0] > mm_rapida_Buffer[0] &&
120                      mm_lenta_Buffer[2] < mm_rapida_Buffer[2];
121
122                      //---
123 bool Comprar = false;
124 bool Vender = false;
125
126 if(estrategia == MEDIA_MOVEL)
127 {
128     Comprar = compra_mm_cros;
129     Vender = venda_mm_cros;
130 }
131
132 if(TemPosicoesAbertas()==false && !emOperacao)
133 {
134     if(Comprar)
135     {
136         CompraAMercado();
137         emOperacao = true;
138     }else if(Vender)
139     {
140         VendaAMercado();
141         emOperacao = true;
142     }
143 }
144
145
146
147 bool temosNovaVela = TemosNovaVela();
148
149
150
151
152
153
154 if(TimeToString(TimeCurrent(),TIME_MINUTES) == hora_limite_fecha_op && PositionSelect(_Symbol)==true)
155 {
156     Print("-----> Fim do Tempo Operacional: encerrar posições abertas!");
157
158     if(PositionGetInteger(POSITION_TYPE) == POSITION_TYPE_BUY)
159     {
160         FechaCompra();
161     }
162     else if(PositionGetInteger(POSITION_TYPE) == POSITION_TYPE_SELL)
163     {
164         FechaVenda();
165     }
166 }
167 }
168
169
170
171
172
173
174 if(TimeToString(TimeCurrent(),TIME_MINUTES) == hora_limite_fecha_op && PositionSelect(_Symbol)==true)
175 {
176     Print("-----> Fim do Tempo Operacional: encerrar posições abertas!");
177
178     if(PositionGetInteger(POSITION_TYPE) == POSITION_TYPE_BUY)
179     {
180         FechaCompra();
181     }
182     else if(PositionGetInteger(POSITION_TYPE) == POSITION_TYPE_SELL)
183     {
184         FechaVenda();
185     }
186 }
187 }

```

```

176 void CompraAMercado()
177 {
178     MqlTradeRequest  requisicao;
179     MqlTradeResult   resposta;
180
181     ZeroMemory(requisicao);
182     ZeroMemory(resposta);
183
184
185     requisicao.action      = TRADE_ACTION_DEAL;
186     requisicao.magic       = expert_magic;
187     requisicao.symbol      = _Symbol;
188     requisicao.volume      = num_lots;
189     requisicao.price       = NormalizeDouble(tick.ask,_Digits);
190     requisicao.sl          = m_symbol.NormalizePrice(NormalizeDouble(tick.ask - SL,_Digits));
191     requisicao.tp          = m_symbol.NormalizePrice(NormalizeDouble(tick.ask + TK,_Digits));
192     requisicao.deviation   = 0;
193     requisicao.type        = ORDER_TYPE_BUY;
194     requisicao.type_filling = ORDER_FILLING_FOK;
195
196     //--
197     OrderSend(requisicao,resposta);
198     //--
199     if(resposta.retcode == 10008 || resposta.retcode == 10009)
200     {
201         Print("Ordem de Compra executada com sucesso!");
202     }
203     else
204     {
205         Print("Erro ao enviar Ordem Compra. Erro = ", GetLastError());
206         ResetLastError();
207     }
208 }
209
210
211 void VendaAMercado()
212 {
213     MqlTradeRequest  requisicao;
214     MqlTradeResult   resposta;
215
216     ZeroMemory(requisicao);
217     ZeroMemory(resposta);
218
219
220
221     requisicao.action      = TRADE_ACTION_DEAL;
222     requisicao.magic       = expert_magic;
223     requisicao.symbol      = _Symbol;
224     requisicao.volume      = num_lots;
225     requisicao.price       = NormalizeDouble(tick.bid,_Digits);
226     requisicao.sl          = m_symbol.NormalizePrice(NormalizeDouble(tick.bid + SL,_Digits));
227     requisicao.tp          = m_symbol.NormalizePrice(NormalizeDouble(tick.bid - TK,_Digits));
228     requisicao.deviation   = 0;
229     requisicao.type        = ORDER_TYPE_SELL;
230     requisicao.type_filling = ORDER_FILLING_FOK;
231     //--
232     OrderSend(requisicao,resposta);
233     //--
234     if(resposta.retcode == 10008 || resposta.retcode == 10009)
235     {
236         Print("Executada!");
237     }
238     else
239     {
240         Print("Erro ao enviar Ordem Venda. Erro = ", GetLastError());
241         ResetLastError();
242     }
243 }

```

```

247 void FechaCompra()
248 {
249     MqlTradeRequest  requisicao;
250     MqlTradeResult   resposta;
251
252     ZeroMemory(requisicao);
253     ZeroMemory(resposta);
254
255
256     requisicao.action      = TRADE_ACTION_DEAL;
257     requisicao.magic       = expert_magic;
258     requisicao.symbol      = _Symbol;
259     requisicao.volume      = num_lots;
260     requisicao.price       = 0;
261     requisicao.type        = ORDER_TYPE_SELL;
262     requisicao.type_filling = ORDER_FILLING_RETURN;
263
264     //---
265     OrderSend(requisicao,resposta);
266     //---
267     if(resposta.retcode == 10008 || resposta.retcode == 10009)
268     {
269         Print("Executada!");
270     }
271     else
272     {
273         Print("Erro = ", GetLastError());
274         ResetLastError();
275     }
276 }
279 void FechaVenda()
280 {
281     MqlTradeRequest  requisicao;
282     MqlTradeResult   resposta;
283
284     ZeroMemory(requisicao);
285     ZeroMemory(resposta);
286
287     requisicao.action      = TRADE_ACTION_DEAL;
288     requisicao.magic       = expert_magic;
289     requisicao.symbol      = _Symbol;
290     requisicao.volume      = num_lots;
291     requisicao.price       = 0;
292     requisicao.type        = ORDER_TYPE_BUY;
293     requisicao.type_filling = ORDER_FILLING_RETURN;
294
295     //---
296     OrderSend(requisicao,resposta);
297
298     //---
299     if(resposta.retcode == 10008 || resposta.retcode == 10009)
300     {
301         Print("Ordem de Compra executada com sucesso!");
302     }
303     else
304     {
305         Print("Erro ao enviar Ordem Compra. Erro = ", GetLastError());
306         ResetLastError();
307     }
308 }
309

```

```

305 bool TemosNovaVela()
306 {
307
308     static datetime last_time=0;
309
310     datetime lastbar_time= (datetime) SeriesInfoInteger(Symbol(),Period(),SERIES_LASTBAR_DATE);
311
312
313     if(last_time==0)
314     {
315
316         last_time=lastbar_time;
317         return(false);
318     }
319
320     if(last_time!=lastbar_time)
321     {
322
323         last_time=lastbar_time;
324         return(true);
325     }
326
327     return(false);
328 }

```

```

331 bool TemPosicoesAbertas()
332 {
333     bool pos = PositionSelect(_Symbol);
334
335     if ( pos == true ){
336         return true;
337     }
338     return false;
339 }
341 void OnTradeTransaction(const MqlTradeTransaction & trans,
342                         const MqlTradeRequest & request,
343                         const MqlTradeResult & result)
344 {
345     ResetLastError();
346     if(trans.type == TRADE_TRANSACTION_DEAL_ADD && trans.symbol == _Symbol) //Nesse Ativo
347     {
348         if(HistoryDealSelect(trans.deal))
349         {
350             ENUM_DEAL_ENTRY deal_entry = (ENUM_DEAL_ENTRY) HistoryDealGetInteger(trans.deal, DEAL_ENTRY);
351             if((ENUM_DEAL_ENTRY)deal_entry == DEAL_ENTRY_IN && emOperacao)
352             {
353                 acionouEntrada = true;
354             }
355         }
356     }
357 }

```

Fonte: Elaborado pelo autor, 2023

7.2.1 PRIMEIRO BLOCO

Neste primeiro bloco, colocamos a estratégia de entrada por meio de uma variável enum (enumerador) que chamamos de “ESTRATEGIA_ENTRADA” e assim conseguimos controlar os parâmetros passados, já que a enumeração introduz novas constantes nomeadas (Manual MQL5). Após declararmos a enum, declaramos as variáveis e os parâmetros de entrada que o algoritmo irá seguir.

Ilustração 5: Estratégia de entrada

```

14 enum ESTRATEGIA_ENTRADA
15     {
16         MEDIA_MOVEL
17     };
18 //--
19 // input
20 // input
21 // input
22 sinput string Entrada;
23 input ESTRATEGIA_ENTRADA      estrategia = MEDIA_MOVEL;
24
25 sinput string s1;
26 input int mm_rapida_periodo    = 14;
27 input int mm_lenta_periodo    = 50;
28 input ENUM_TIMEFRAMES mm_tempo_grafico = PERIOD_CURRENT;
29 input ENUM_MA_METHOD mm_metodo = MODE_EMA;
30 input ENUM_APPLIED_PRICE mm_preco = PRICE_CLOSE;
31
32 sinput string lotes;
33 input int num_lots            = 5;
34 input double TK               = 6;
35 input double SL               = 4;
36
37 sinput string hora;
38 input string hora_limite_fecha_op = "17:00";

```

Fonte: Elaborado pelo autor (2023)

7.2.2 SEGUNDO BLOCO

No segundo bloco declaramos as variáveis globais da função, que são os elementos que estão “fora” do código, ou seja, não são locais de algum bloco. Nesse caso, as nossas variáveis são as médias lentas e rápidas declaradas a partir de *handles* para controlá-las e buffers para o armazenamento dos dados. Também declaramos as velas para armazenamento de dados e uma variável do tipo *bool* para saber se a vela está ou não dentro de uma operação.

Ilustração 6: Variáveis globais

```

44 int mm_rapida_Handle;
45 double mm_rapida_Buffer[];
46
47 /
48 int mm_lenta_Handle;
49 double mm_lenta_Buffer[];
50
51
52 int expert magic = 1;
53
54 MqlRates velas[];
55 MqlTick tick;
56
57 CSymbolInfo m_symbol;
58
59 // Controle de operações
60 bool emOperacao=false;
61 bool acionouEntrada=false;

```

Fonte: Elaborado pelo autor (2023)

7.2.3 TERCEIRO BLOCO

Aqui começamos a fazer o robô iniciar de fato, declarando as funções *OnInit* e *OnDeInit*, que são as funções responsáveis pelo ligamento e desligamento do robô dentro da plataforma. Na função *OnInit* necessitamos de um *handle* para carregar o indicador, e para funcionar de maneira plena declaramos o símbolo (o ativo em questão), seguido dos dados que já declaramos nos blocos anteriores, que seria o valor da variável, o período, o “recuo”, a estratégia e o preço do ativo, em seguida salvamos as *candles* por meio do *CopyRates* e ordenando com o *ArraySetSeries*, por fim adicionamos os indicadores. Já na função *OnDeInit* só removemos o indicador do gráfico.

Ilustração 7: Função OnInit/OnDnit

```

66 int OnInit()
67 {
68
69     mm_rapida_Handle = iMA(_Symbol,mm_tempo_grafico,mm_rapida_periodo,0,mm_metodo,mm_preco);
70     mm_lenta_Handle  = iMA(_Symbol,mm_tempo_grafico,mm_lenta_periodo,0,mm_metodo,mm_preco);
71
72
73     CopyRates(_Symbol,_Period,0,4,velas);
74     ArraySetAsSeries(velas,true);
75
76     // Para adicionar no gráfico o indicador:
77     ChartIndicatorAdd(0,0,mm_rapida_Handle);
78     ChartIndicatorAdd(0,0,mm_lenta_Handle);
79
80     m_symbol.Name(_Symbol);
81     m_symbol.Refresh();
82
83     //---
84     return(INIT_SUCCEEDED);
85 }
86
87 void OnDeinit(const int reason)
88 {
89     //---
90     IndicatorRelease(mm_rapida_Handle);
91     IndicatorRelease(mm_lenta_Handle);
92 }

```

Fonte: Elaborado pelo autor (2023)

7.2.4 QUARTO BLOCO

No quarto bloco entramos dentro da função *onTick*, onde colocamos as funções responsáveis pela compra e venda a mercado, as responsáveis pelo fechamento dessas operações e a lógica por trás da Inteligência artificial (IA).

7.2.4.1 COMPRA E VENDA

Como toda a operação manual, onde necessitamos abrir ordens de compra e venda para, de fato, estarmos realizando uma operação, a IA também precisa fazer o mesmo, que fazemos dentro da função *MqITraderRequest*.

Ilustração 8: Função de compra e venda

```

176 void CompraAMercado()
177 {
178     MqlTradeRequest  requisicao;
179     MqlTradeResult  resposta;
180
181     ZeroMemory(requisicao);
182     ZeroMemory(resposta);
183
184
185     requisicao.action      = TRADE_ACTION_DEAL;
186     requisicao.magic      = expert_magic;
187     requisicao.symbol     = _Symbol;
188     requisicao.volume     = num_lots;
189     requisicao.price      = NormalizeDouble(tick.ask, _Digits);
190     requisicao.sl         = m_symbol.NormalizePrice(NormalizeDouble(tick.ask - SL, _Digits));
191     requisicao.tp         = m_symbol.NormalizePrice(NormalizeDouble(tick.ask + TK, _Digits));
192     requisicao.deviation  = 0;
193     requisicao.type       = ORDER_TYPE_BUY;
194     requisicao.type_filling = ORDER_FILLING_FOK;
195
196     //---
197     OrderSend(requisicao, resposta);
198     //---
199     if(resposta.retcode == 10008 || resposta.retcode == 10009)
200     {
201         Print("Ordem de Compra executada com sucesso!");
202     }
203     else
204     {
205         Print("Erro ao enviar Ordem Compra. Erro = ", GetLastError());
206         ResetLastError();
207     }
208 }
209
212 void VendaAMercado()
213 {
214     MqlTradeRequest  requisicao;
215     MqlTradeResult  resposta;
216
217     ZeroMemory(requisicao);
218     ZeroMemory(resposta);
219
220
221     requisicao.action      = TRADE_ACTION_DEAL;
222     requisicao.magic      = expert_magic;
223     requisicao.symbol     = _Symbol;
224     requisicao.volume     = num_lots;
225     requisicao.price      = NormalizeDouble(tick.bid, _Digits);
226     requisicao.sl         = m_symbol.NormalizePrice(NormalizeDouble(tick.bid + SL, _Digits));
227     requisicao.tp         = m_symbol.NormalizePrice(NormalizeDouble(tick.bid - TK, _Digits));
228     requisicao.deviation  = 0;
229     requisicao.type       = ORDER_TYPE_SELL;
230     requisicao.type_filling = ORDER_FILLING_FOK;
231     //---
232     OrderSend(requisicao, resposta);
233     //---
234     if(resposta.retcode == 10008 || resposta.retcode == 10009)
235     {
236         Print("Executada!");
237     }
238     else
239     {
240         Print("Erro ao enviar Ordem Venda. Erro = ", GetLastError());
241         ResetLastError();
242     }
243 }

```

Fonte: Elaborado pelo autor (2023)

Podemos observar que dentro do *MqlTraderRequest* é necessário que seja colocado, na ordem do código, o tipo de execução, o número mágico do robô, o ativo, o número de lotes, o preço de venda, o stop, o alvo, o desvio de preço, o tipo da ordem e o preenchimento da ordem.

7.2.4.2 FECHAMENTO DE ORDENS

Já para a realização do fechamento das operações, tanto de compra como de venda, a função é basicamente a mesma, apenas realizando a ordem contrária da trade, no entanto, enquanto utilizamos uma “*fok*” nas entradas, utilizamos um “*return*” na saída, que é para ele retornar à operação.

Ilustração 9: Fechamento de ordens

```

247 void FechaCompra()
248 {
249     MqlTradeRequest
250     MqlTradeResult
251
252     ZeroMemory(requisicao);
253     ZeroMemory(resposta);
254
255
256     requisicao.action      = TRADE_ACTION_DEAL;
257     requisicao.magic      = expert_magic;
258     requisicao.symbol     = _Symbol;
259     requisicao.volume     = num_lots;
260     requisicao.price      = 0;
261     requisicao.type       = ORDER_TYPE_SELL;
262     requisicao.type_filling = ORDER_FILLING_RETURN;
263
264     ///---
265     OrderSend(requisicao,resposta);
266     ///---
267     if(resposta.retcode == 10008 || resposta.retcode == 10009)
268     {
269         Print("Executada!");
270     }
271     else
272     {
273         Print("Erro = ", GetLastError());
274         ResetLastError();
275     }
276 }

```

```

279 void FechaVenda()
280 {
281     MqlTradeRequest  requisicao;
282     MqlTradeResult   resposta;
283
284     ZeroMemory(requisicao);
285     ZeroMemory(resposta);
286
287     requisicao.action      = TRADE_ACTION_DEAL;
288     requisicao.magic       = expert_magic;
289     requisicao.symbol      = _Symbol;
290     requisicao.volume     = num_lots;
291     requisicao.price       = 0;
292     requisicao.type        = ORDER_TYPE_BUY;
293     requisicao.type_filling = ORDER_FILLING_RETURN;
294
295
296     OrderSend(requisicao,resposta);
297
298     //---
299     if(resposta.retcode == 10008 || resposta.retcode == 10009)
300     {
301         Print("Ordem de Compra executada com sucesso!");
302     }
303     else
304     {
305         Print("Erro ao enviar Ordem Compra. Erro = ", GetLastError());
306         ResetLastError();
307     }
308 }
309

```

Fonte: Elaborado pelo autor (2023)

7.2.4.3 VELAS E FUNÇÃO ONTICK

Para começar a função *Ontick* é necessário “carregar” os buffers por meio um *CopyBuffer* para carregar as variáveis do indicador, depois colocamos a variável das *candles*, ordenamos os vetores e colocamos essas informações dentro da vela. Partindo para a parte logica, onde colocamos a estratégia funcionar, inserimos a compra quando o buffer da média rápida for maior que o buffer da média lenta e a venda quando o contrário acontece, lendo esses movimentos a cada 2 velas.

Ilustração 10: Função OnTick

```

89 void OnTick()
90 {
91
92
93     if(acionouEntrada && !TemPosicoesAbertas())
94     {
95         acionouEntrada = false;
96         emOperacao = false;
97     }
98
99     CopyBuffer(mm_rapida_Handle,0,0,4,mm_rapida_Buffer);
100    CopyBuffer(mm_lenta_Handle,0,0,4,mm_lenta_Buffer);
101
102
103
104    CopyRates(_Symbol,_Period,0,4,velas);
105    ArraySetAsSeries(velas,true);
106
107
108    ArraySetAsSeries(mm_rapida_Buffer,true);
109    ArraySetAsSeries(mm_lenta_Buffer,true);
110
111
112    SymbolInfoTick(_Symbol,tick);
113
114
115    bool compra_mm_cros = mm_rapida_Buffer[0] > mm_lenta_Buffer[0] &&
116                        mm_rapida_Buffer[2] < mm_lenta_Buffer[2] ;
117
118
119    bool venda_mm_cros = mm_lenta_Buffer[0] > mm_rapida_Buffer[0] &&
120                        mm_lenta_Buffer[2] < mm_rapida_Buffer[2];
121

```

Fonte: Elaborado pelo autor (2023)

Com as variáveis *bool*, definimos que o robô não pode fazer aleatoriamente operações de compra e venda, logo em seguida atribuímos isso as limitações de compra e venda e que a operação vai ser feita a mercado, como também um *bool* pra reconhecer as novas *candles*.

Ilustração 11: Variável `bool`

```

123 bool Comprar = false;
124 bool Vender = false;
125
126 if(estrategia == MEDIA_MOVEL)
127 {
128     Comprar = compra_mm_cros;
129     Vender = venda_mm_cros;
130 }
131
132 if(TemPosicoesAbertas()==false && !emOperacao)
133 {
134     if(Comprar)
135     {
136         CompraAMercado();
137         emOperacao = true;
138     }else if(Vender)
139     {
140         VendaAMercado();
141         emOperacao = true;
142     }
143 }
144
145
146
147 bool temosNovaVela = TemosNovaVela();

```

Fonte: Elaborado pelo autor (2023)

Já em relação ao horário limite de fechamento das operações, usamos *PositionSelect* para o robô reconhecer se existem operações existentes no momento, caso tenha, ele retorna “true” e automaticamente faz a operação contrária a mercado para sair do trade.

Ilustração 12: Horário limite

```

154 if(TimeToString(TimeCurrent(),TIME_MINUTES) == hora_limite_fecha_op && PositionSelect(_Symbol)==true)
155 {
156     Print("-----> Fim do Tempo Operacional: encerrar posições abertas!");
157
158     if(PositionGetInteger(POSITION_TYPE) == POSITION_TYPE_BUY)
159     {
160         FechaCompra();
161     }
162     else if(PositionGetInteger(POSITION_TYPE) == POSITION_TYPE_SELL)
163     {
164         FechaVenda();
165     }
166 }
167 }

```

Fonte: Elaborado pelo autor (2023)

7.2.3 BACKTESTS

Em um primeiro, momento nos deparamos com duas telas, onde uma podemos organizar o tempo gráfico, as datas de início e de final do *backtest*, latência, banca inicial e a otimização:

Ilustração 13: Configurações

Fonte: Elaborado pelo autor (2023)

Escolhido as configurações iniciais, conseguimos organizar os parâmetros de entrada, no caso dessa EA, conseguimos escolher os valores das médias móveis e caso houvesse mais de uma estratégia dentro do robô, conseguiríamos escolher ela dentro dessa aba também.

Ilustração 14: Parâmetros de entrada

Variável	Valor	Iniciar	Passo	Parar
<input checked="" type="checkbox"/> Entrada				
<input checked="" type="checkbox"/> estratégia	MEDIA_MOVEL	MEDIA_MOVEL		MEDIA_MOVEL
<input checked="" type="checkbox"/> Media Movel				
<input checked="" type="checkbox"/> media rapida	14	14	1	140
<input checked="" type="checkbox"/> media lenta	31	50	1	500
<input checked="" type="checkbox"/> tempo grafico	current	current		1 Month
<input checked="" type="checkbox"/> metodo	Exponential	Simple		Linear weighted
<input checked="" type="checkbox"/> preço	Close price	Close price		Weighted price

Fonte: Elaborado pelo autor, 2023

Após isso, podemos iniciar o *backtest* e recebemos uma série de informações a respeito da série histórica escolhida, para esse teste foi utilizado as médias de 14 e 31 em um gráfico de *candle* no tempo de 5 minutos e utilizando 10 contratos de mini dólar, assim conseguimos esse resultado inicial:

Ilustração 15: Resultados

Depósito Inicial	12 000.00
Lucro Líquido Total	14 881.00
Lucro Bruto	555 544.90
Perda Bruta	-540 663.90

Fonte: Elaborado pelo autor (2023)

Assim conseguimos observar que tivemos um lucro total de R\$ 14,881.00 no período entre 1 de janeiro de 2022 até 6 de dezembro de 2022, com um lucro bruto de R\$ 555.544,90 e uma perda bruta de R\$ -540.663,90, logo de cara o resultado não aparenta ser tão interessante, mas nos mostra que é possível ter um retorno em cima dos algoritmos de trading. A plataforma também nos apresenta outros dados, como os da ilustração 16:

Ilustração 16:Dados

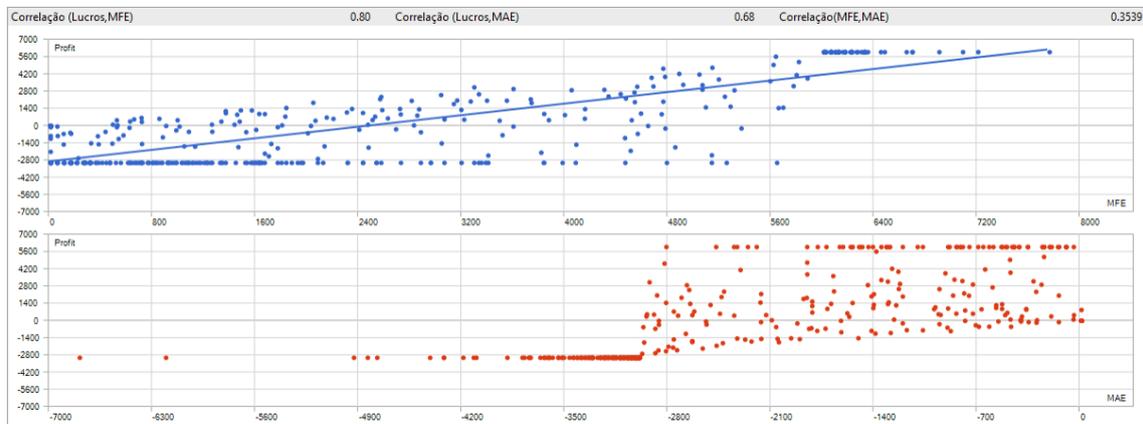
Fator de Lucro	1.03	Retorno Esperado (Payoff)	36.74	Nível de Margem	
Fator de Recuperação	0.28	Índice de Sharpe	0.18	Z-Pontuação	0.84 (59.91%)
AHPR	1.0256 (2.56%)	Correlação LR	0.14	Resultado OnTester	0
GHPR	1.0020 (0.20%)	Erro Padrão LR	11 274.91		
Total de Negociações	405	Posições Vendidas (% e ganhos)	201 (42.79%)	Posições Compradas (% de gan...	204 (37.25%)
Ofertas Total	810	Negociações com Lucro (% of t...	162 (40.00%)	Negociações com Perda (% of t...	243 (60.00%)
	Maior	lucro da negociação	6 000.00	perda na Negociação	-3 000.00
	Média	lucro da negociação	3 429.29	perda na Negociação	-2 224.95
	Máximo	ganhos consecutivos (\$)	6 (21 061.80)	perdas consecutivas (\$)	10 (-16 224.10)
	Máxima	lucro consecutivo (contagem)	24 421.40 (5)	perda consecutiva (contagem)	-16 224.10 (10)
	Média	ganhos consecutivos	2	perdas consecutivas	2

Fonte: Elaborado pelo autor (2023)

Aqui vamos analisar os fatores que mais tem peso na hora de decidir se a estratégia é viável ou não, começando pelo fator de lucro de 1.03, não muito eficiente, porém positivo, e esse resultado obtido ao dividir o lucro bruto pela perda bruta, onde números maiores que 1 nos dão resultados positivos e números menores que 1 nos dão resultados negativos. O fator recuperação mede o quanto a estratégia consegue se recuperar depois de uma perda, assim obtivemos um resultado de 0.28. O retorno esperado é o lucro líquido dividido pelo número total de operações, assim sabemos a média de lucro que cada operação realizada vende, assim obtivemos um resultado no 36,74. Partindo agora para o índice de Sharpe, talvez o indicador mais importante aqui, que nos apresenta uma correlação entre a média aritmética do lucro e seu desvio padrão,

assim apontando para a relação entre risco e retorno do robô, ajudando o usuário a tomar suas decisões de maneira mais eficiente, aqui também conseguimos um resultado positivo, atingindo 0.18.

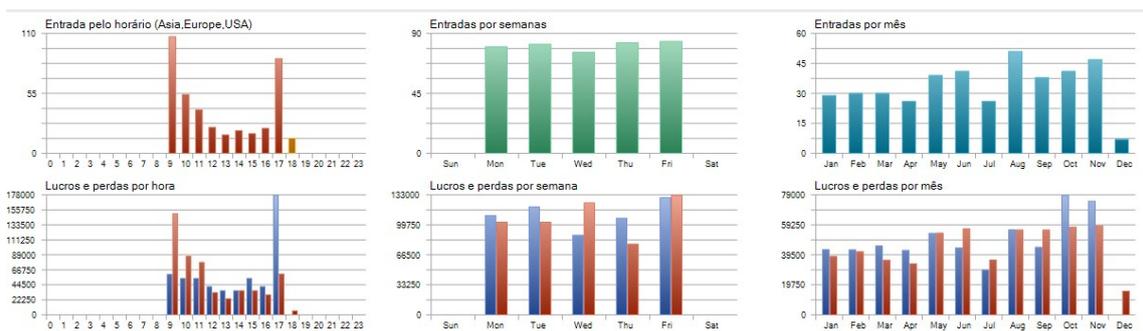
Gráfico 10: Índices



Fonte: Elaborado pelo autor (2023)

No gráfico 10, conseguimos analisar melhor os índices comentados, já as demais informações presentes também são importantes, mas não são determinantes tão grandes para a tomada de decisão do usuário. O MQL5 também conta com gráficos dentro de seu *backtest*, apresentando as entradas por horário, dia, semana e mês, como os lucros e as perdas respectivas:

Gráfico 11: Dados



Fonte: Elaborado pelo autor (2023)

Essas informações nos ajudam a ver quais os melhores horários e dias para o funcionamento o que nos permite desligar o robô nesses momentos para obtermos resultados melhores e até mesmo automatizá-lo para não ter que ser

realizado manualmente. Por fim, a plataforma nos mostra a evolução do patrimônio.

Gráfico 12: Evolução do patrimônio no período 01/01/2022 até 05/12/2022



Fonte: Elaborado pelo autor (2023)

9. Considerações Finais

Visto todo o processo de criação e desenvolvimento do algoritmo de trading, desde Bloomberg até os dias atuais, podemos concluir que essa tecnologia está em constante evolução e chegando a resultados cada vez melhores, o que pode ajudar investidores que sofrem de algum tipo de viés psicológico ou que apenas queiram utilizar de uma IA em suas operações. Já com o resultado do robô, conseguimos ter um panorama geral das capacidades e limitações desse tipo de IA, mas mesmo com todas as limitações, conseguimos ter resultados positivos e consistentes ao longo do tempo, o que permite adição de novos indicadores, como um índice de força relativa ou até mesmo as bandas de bollinger, e estratégias que aumentam o nível de complexidade do robô, aumentando o rendimento desse tipo de investimento.

REFERÊNCIAS

ALBURQUERQUE, Melissa, Flash Crash, **Pandhora**. Disponível em: <https://blog.pandhora.com/flash-crash/>. acesso em 10 jun 2022

BLOOMBERG, Michael. **Bloomberg by bloomberg**. 1° Edição. Nova Jersey: John Wiley & Sons, 27 de ago de 2001

CERBASI, Gustavo Análise Técnica x Análise Fundamentalista: Como avaliar as empresas na hora de investir? **BTG actual**, Disponível em: https://www.btgpactualdigital.com/como-investir/artigos/renda-variavel/analise-tecnica-ou-fundamentalista?cmpid=c04:m05:google:1741053764::96926179141&utm_medium=spl&utm_source=google&utm_campaign=1741053764&utm_content=96926179141&creative=593766517156&adposition=&keyword=&matchtype=&targetid=dsa1649582094409&device=c&feeditemid=&loc_interest_ms=&loc_physical_ms=1001575&placement=&s_kwid=AL!9288!3!593766517156!!!g!!&gclid=CjwKCAjwlqOXBhBqEiwA-hhitKKu-iKR-4pftQndPHYni8Hij30230QNc6_HnnUZjNcJr41kZDIkyRoCylIQAvD_BwE. Acesso em 07 jul 2022.

CHAGUE, Fernando. GIOVANNETTI, Bruno. É possível viver de day-trade em ações? **Revista Brasileira de Finanças**, Rio de Janeiro, v. 18, n. 3, p. 1-4, set de 2020.

DUBART, Caroline. Robo de investimento: Tudo o que você precisa saber dessa tecnologia. **Magnets**, 26 de jul 2016. Disponível em: <https://blog.magnetis.com.br/robo-de-investimento/#:~:text=com%20um%20rob%C3%B4%3F-,Sobre%20os%20rob%C3%B4s%20de%20investimento,conhecimento%20sobre%20o%20mercado%20financeiro>. Acesso em 19 jul 2022

GOLBERG, Steven. Could Computerized Trading Cause Another Market Crash. **Kiplinger**, 3 abr 2018. Disponível em:

<https://www.kiplinger.com/article/investing/t041-c007-s001-could-computerized-trading-cause-another-market-cr.html>. Acesso em 19 abr 2022.

GUPTA, Anupriya. History of Algorithmic Trading, HFT and News Based Trading. **QuantInsti**, 02 jun 2015. Disponível em em:<https://blog.quantinsti.com/history-algorithmic-trading-hft/>. Acesso em 18 abr 2022.

HENRIQUES, Diana. **A First-Class Catastrophe: The Road to Black Monday, the Worst Day in Wall Street History**. 1° edição. Nova York: Henry Holt and Company, 19 set 2017.

INVESTIMENTOS, Aprenda o que é e como usar a retração Fibonacci, **Toro investimentos**. Disponível em:https://blog.toroinvestimentos.com.br/trading/retracao-de-fibonacci?utm_source=google&utm_medium=cpc&utm_campaign=13987880586&utm_content=&utm_term=&gclid=CjwKCAjwlqOXBhBqEiwA-hhitCBytYTLY7vgZQyA7ENN1-GB2fzH0xE5yYs298-qIAPJhLmkdJBHghoCbGsQAvD_BwE. Acesso em 07 jul 2022.

KALOWSKI, Mark. How Robots Rule the Stock Market (SPX, DJIA). **Investopedia**, 25 jun 2019. Disponível em:<https://www.investopedia.com/news/how-robots-rule-stock-market-spx-djia/>. Acesso em 19 abr 2022

LACY, Edward. Market Quotations. **Eletronics World**, p. 38-40. Disponível em: <https://worldradiohistory.com/Archive-Electronics-World/60s/1968/Electronics-World-1968-08.pdf>. Acesso em 17 jun de 2023

LARANJEIRA, Jorge, Uma análise do flash crash, **Giant Steps**, Disponível em:<https://gscap.com.br/uma-analise-do-flash-crash/>. acesso em 10 jun de 2022.

LELAND, Hayne, RUBINSTEIN, Mark. The Evolution of Portfolio Insurance. Disponível em: https://www.researchgate.net/profile/Hayne-Leland-2/publication/265430746_The_Evolution_of_Portfolio_Insurance/links/568c1602

08ae197e42689833/The-Evolution-of-Portfolio-Insurance.pdf. Acesso em 20 set de 2022

MARK, Douglas. **Trading in the zone**. 8ª Edição. Algueirão-Mem Martins: Bookout, 1 de jan de 2019.

MCCRACKEN, Harry, How the Bloomberg Terminal Made History—And Stays Ever Relevant, **Fast Company**. Disponível em: <https://www.fastcompany.com/3051883/the-bloomberg-terminal>, acesso em 9 mai 2022

METAQUOTES. **Linguagem MQL5 referente ao terminal do cliente MetaTrader 5**.

MORRIS, Charles R., The Day Wall Street Colapse , **The New York Times**, Disponível em: <https://www.nytimes.com/2017/11/17/books/review/diana-b-henriques-a-first-class-catastrophe.html>, acesso em 15 mai 2022

MURPHY, John J. **Análise Técnica do Mercado Financeiro**: Um Guia Abrangente de Aplicações e Métodos de Negociação. São Paulo: Elsevier, 1986.

NELOGICA. Utilizando cruzamento de médias de maneira mais eficiente, **Nelogica**. Disponível em: <https://www.nelogica.com.br/conhecimento/artigos/trading/cruzamento-medidas>. Acesso em 20 set 2022

PAVINI, Angelo, Prejuízo, depressão e morte no day trade: quando tudo sai de controle, **Exame Invest**. Disponível em: <https://exame.com/invest/mercados/prejuizo-depressao-e-morte-no-day-trade-quando-tudo-sai-de-controle/>, acesso em 02/06/22

REIS, Tiago, A Segunda Férias Negra de 1987, **Suno Artigos**. Disponível em: <https://www.suno.com.br/artigos/a-segunda-feira-negra-de-1987/>, acesso em 15 mai de 2022

SEGAL, Troy. What Caused Black Monday: the stock market crash of 1987?. **Investopedia**, 28 out 2021. Disponível em: <https://www.investopedia.com/ask/answers/042115/what-caused-black-monday-stock-market-crash-1987.asp#:~:text=Key%20Takeaways,as%20well%20as%20investor%20panic>. Acesso em 18 abr 2022.

SIEGEL. Jeremy J. **Stocks for the long run: the definitive guide to financial market returns and long-term investimento strategies**. 2 ed. Nova York: McGraw-Hull, 1988.

VELLOSO, Álvaro, Como escolher um robô de investimento com base nos backtests, **INFOMONEY**, Disponível em: <https://www.infomoney.com.br/colunistas/meta-trader-experiencias/como-escolher-um-robo-de-investimento-com-base-nos-backtests/>. Acesso em 9 ago 2022

WILDER, J. Welles. **New Concepts in Technical Trading System**. 1° Edição. Winston-Salen: Hunter Publishing Company, 1978.

ZIVIADANI, Nivio. **Projeto de Algoritmos Com Implementações em Pascal E C**. 3° Edição. São Paulo: Cengage Learning, 23 de jul de 2010.