



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Instituto de Matemática e Estatística

Juliana Costa da Silva


**Um robô em *Python* para operações de trading nas exchanges
de criptomoedas**

Rio de Janeiro

2020

Juliana Costa da Silva

**Um robô em *Python* para operações de trading nas exchanges de
criptomoedas**



Monografia apresentada, como requisito parcial para obtenção do título de Bacharel, ao Instituto de Matemática e Estatística, da Universidade do Estado do Rio de Janeiro.

Orientador: Prof. Dra. Rosa Maria da Costa
Coorientador: Prof. Dr. Igor Machado Coelho

Rio de Janeiro
2020

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/D

D979 Costa da Silva, Juliana
Um robô em *Python* para operações de trading nas exchanges de criptomoedas / Juliana Costa da Silva. – Rio de Janeiro, 2020-
44 f.

Orientador: Prof. Dra. Rosa Maria da Costa
Monografia (Bacharelado) – Universidade do Estado do Rio de Janeiro, Instituto de Matemática e Estatística, Instituto de Matemática e Estatística, 2020.

1. AA.. 2. BB.. 3. Python.. I. Prof. Dra. Rosa Maria da Costa. II. Universidade do Estado do Rio de Janeiro. III. Instituto de Matemática e Estatística. IV. Título

CDU 02:141:005.7

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta monografia, desde que citada a fonte.

Assinatura

Data

Juliana Costa da Silva

**Um robô em *Python* para operações de trading nas exchanges de
criptomoedas**

Monografia apresentada, como requisito parcial para obtenção do título de Bacharel, ao Instituto de Matemática e Estatística, da Universidade do Estado do Rio de Janeiro.

Aprovada em de março de 2020.

Banca Examinadora:

Prof. Dra. Rosa Maria da Costa (Orientador)
Instituto de Matemática e Estatística – UERJ

Prof. Dr. Igor Machado Coelho (Coorientador)
Unidade – Instituição

primeiro membro titular da banca
Instituto de Matemática e Estatística – UERJ

primeiro membro suplente da banca
Instituto de Matemática e Estatística – UERJ

Rio de Janeiro

2020

DEDICATÓRIA

Aos professores, amigos e familiares que de alguma forma contribuíram no desenvolvimento desse trabalho de conclusão de curso.

AGRADECIMENTOS

A minha orientadora Rosa Maria por compartilhar todo o seu conhecimento sempre trazendo boas ideias.

Ao meu coorientador Igor Machado pela orientação nestes últimos meses.

A minha família: meu pai Jorge Luiz e minha mãe Solange Costa pelo apoio e suporte durante esses anos de graduação.

Ao professor Igor Machado pela orientação nesses últimos meses.

Aos amigos Jadson Santos, Mário Gomes e Rafael Teixeira, que foram fundamentais nesse processo.

A todos os amigos que conheci no curso e que estiveram comigo nesta jornada.

Vamos inventar o amanhã em vez de ficar nos
preocupando com o que aconteceu ontem.

Steve Jobs

RESUMO

COSTA DA SILVA, J.S. *Um robô em Python para operações de trading nas exchanges de criptomoedas*. 2020. 44 f. Monografia (Bacharelado em Ciência da Computação) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

As criptomoedas são conhecidas por serem extremamente voláteis com os preços sofrendo grandes variações até mesmo em questão de minutos. Investidores podem realizar trading a qualquer hora do dia. Combinados, esses fatores limitam a capacidade do ser humano realizar operações de trading. Primeiro, em muitas situações, investidores estão impedidos de reagirem rapidamente às variações de preço das moedas para conseguirem o melhor trade. O tempo entre as transações e alguns problemas na redução da velocidade das operações nas corretoras aumentam este problema. Segundo, investidores não podem dedicar o tempo necessário para sempre garantirem o melhor trade. Isso exigiria passar o tempo todo monitorando as corretoras de criptomoedas em todo o mundo. Existem algumas soluções para este problema como o uso de robôs, ferramentas automáticas que conduzem trades e executam operações para os investidores humanos.

Palavras-chave: AA. BB. Python. Robô.

ABSTRACT

COSTA DA SILVA, J.S. *Title of dissertation*. 2020. 44 f. Monografia (Bacharelado em Ciência da Computação) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2020.

Cryptocurrencies are known to be extremely volatile with prices subject to wide variations even in a matter of minutes. Investors can trade at any time of the day. Combined, these factors limit the human being's ability to carry out trading operations. First, in many situations, investors are prevented from reacting quickly to currency price changes to achieve the best trade. The time between transactions and some problems in reducing the speed of operations at brokerage firms increase this problem. Second, investors cannot devote the necessary time to always guarantee the best trade. This would require spending all of your time monitoring cryptocurrency brokers around the world. There are some solutions to this problem, such as the use of robots, automatic tools that conduct trades and perform operations for human investors.

Keywords: Lattes. Python. Robô.

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Problema de Pesquisa	10
1.2	Objetivos	11
1.3	Justificativa	11
1.4	Metodologia	11
1.5	Estrutura do Trabalho	12
2	TECNOLOGIA LEDGER DISTRIBUÍDA	13
2.1	Entendendo a tecnologia Distributed Ledger	13
2.2	Exemplos reais da tecnologia Distributed Ledger	13
2.3	Blockchain	14
2.4	Como isso funciona	14
2.5	O que acontece quando alguém cria um novo bloco	16
2.6	IOTA Tangle	16
2.7	Transações Seguras	16
2.8	IOTA Mining - Confirmação das transações	17
2.9	Taxas nas transações da moeda IOTA	18
2.10	Velocidade da transação da moeda IOTA	18
2.11	Como o IOTA pode ser usado	18
2.12	Considerações finais do IOTA	19
3	SISTEMAS DE TRADING	20
3.1	Sistemas automatizados de trading	20
3.1.1	<u>Vantagens dos sistemas automatizados</u>	21
3.1.2	<u>Desvantagens dos sistemas automatizados</u>	22
3.2	Como evitar golpes	23
3.3	Antes de decidir automatizar	23
4	SISTEMA PROPOSTO	25
4.1	Download de um código fonte aberto	25
4.2	Começando do zero: Montar um time de desenvolvimento e começar a programar	25
4.3	Eliminando o trabalho pesado: Pagar um time de desenvolvimento externo para criar o robô	27
5	ESTUDO DE CASO	29
5.1	Primeiros passos no desenvolvimento do robô	29
5.1.1	<u>Comprando bitcoins e obtendo as APIs</u>	29
5.1.2	<u>APIS</u>	34
5.1.3	<u>Padrão REST x RPC</u>	34

5.1.4	<u>Utilizando as APIs para autenticação no código e execução das ordens</u>	. . 35
5.2	MyRobô em execução 35
5.2.1	<u>Início da operação do MyRobô</u> 38
6	CONSIDERAÇÕES FINAIS 42
	REFERÊNCIAS 43

1 INTRODUÇÃO

É genial festejar o sucesso, mas é mais importante aprender com as lições do fracasso.

Bill Gates

Existem muitos robôs de criptomoedas. Um dos mais populares é o robô de arbitragem. Robôs de arbitragem são ferramentas que examinam os preços das criptomoedas nas corretoras visando obter uma vantagem em alguma discrepância de preço. Por causa do preço de algumas criptomoedas como o bitcoin variar de corretora para corretora, robôs que podem se mover muito rápido acabam ganhando uma vantagem frente às corretoras que demoraram para atualizarem seus preços. Outros tipos de robôs podem usar os dados disponíveis no histórico de preços para testarem estratégias de arbitragem(trading). Enquanto outros robôs são programados para executarem trading a partir de determinados sinais como preço ou volume do trading.

1.1 Problema de Pesquisa

Investidores podem se inscrever de forma gratuita em programas de robôs para realizarem trading de criptomoedas. Por outro lado, alguns robôs cobram taxas que podem se tornar bem expressivas. Investidores procuram por robô ou robôs que possam atender seus objetivos e então fazem o download do código a partir de um desenvolvedor. Cada robô requer requerimentos diferentes em termos de software e hardware. Alguns robôs podem ser bem úteis. Para maximizar o impacto de um robô, investidores precisam saber a melhor maneira de utilizar a ferramenta. Por exemplo, investidores precisam ter as contas corretas nas diferentes corretoras de criptomoedas. Em muitos casos, eles ainda precisam tomar decisões como quando comprar e vender. Enquanto os robôs podem executar as ordens, não existe um substituto para uma sólida estratégia de investimento. O que um robô de criptomoedas não pode ser é uma solução rápida para enriquecer para um investidor que não está interessado em colocar tempo e esforço necessários para o sucesso. Primeiro, muitos robôs oferecem retornos baixos mesmo quando operam corretamente. Segundo, muitos robôs simplesmente não foram bem desenvolvidos. Investidores precisam lembrar que o mundo dos robôs de criptomoedas não possui regulamentação. Terceiro, e o mais importante, a utilização correta de um robô requer um conhecimento profundo do mercado de moedas digitais e um excelente suporte de um plano de investimento. Para alguns investidores, um robô pode ser uma excelente ferramenta para o trading de criptomoedas.

1.2 Objetivos

Diante disso foi desenvolvido o MyRobô. Robô que realiza compra/venda de bitcoins em duas corretoras (bitfinex/binance). Permitindo mostrar como é possível obter lucro em operações de trading ao tirar vantagem da falta de sincronismo na mudança de preço de bitcoins entre as corretoras. Além de eliminar os fatores emocionais envolvidos no processo, já que o mesmo é automatizado.

A pesquisa terá como objetivos específicos:

1. Apresentar como o MyRobô foi desenvolvido;
2. Apresentar como funciona a execução do MyRobô;
3. Explicar como funcionam as operações de trading com criptomoedas utilizando robôs;
4. Explicar o conhecimento necessário para desenvolver qualquer robô que opere no mercado de criptomoedas;
5. Elucidar os desafios que são encontrados ao realizar trading com robôs que operam no mercado de criptomoedas;
6. Explicar o funcionamento da rede onde as criptomoedas são comercializadas

1.3 Justificativa

O que motivou a realização deste trabalho foi a percepção de que muitos não entendem como funciona o mercado de trading com criptomoedas. E, por isso, acabam sendo enganados por malfeitores ao depositarem seu dinheiro em supostas empresas que dizem possuir robôs de trading de criptomoedas e que oferecem lucros exorbitantes em um curto espaço de tempo.

1.4 Metodologia

A abordagem metodológica do estudo é de caráter exploratório e descritivo. Foram consultadas fontes secundárias como artigos, sites de exchanges, livros, sistemas automatizados de trading para a apresentação do robô que foi desenvolvido com o objetivo de realizar transações nas corretoras de criptomoedas.

Para este trabalho de conclusão foi desenvolvido um robô de trading em Python3 que fez uso das APIs Binance e Bitfinex.

Foi proposto o desenvolvimento do rôbo para explicar de maneira mais clara como funcionam as operações em corretoras de criptomoedas e o necessário para investir de forma consciente nesse mercado.

1.5 Estrutura do Trabalho

O trabalho de conclusão de curso estrutura-se em 4 capítulos da seguinte forma:

O segundo capítulo traz uma explicação dos sistemas automatizados apresentando suas vantagens e desvantagens, dicas de como evitar os golpes e tudo o que é necessário saber antes de decidir otimizar o trading com um robô.

O terceiro capítulo explica como construir um trading bot de criptomoedas fazendo o download de um código fonte aberto ou começando do zero com um time de desenvolvimento.

O quarto capítulo explica como funciona a rede onde as criptomoedas são negociadas, em seu nível macro, no que diz respeito à Distributed Ledger Technology e em níveis mais específicos no que diz respeito à rede Blockchain e IOTA Tangle.

O quinto capítulo apresenta o estudo de caso, como foi desenvolvido o MyRobô e como ele opera.

2 TECNOLOGIA LEDGER DISTRIBUÍDA

Distributed Ledger é um sistema descentralizado que mantém de forma verificável e segura um registro das transações. Diferente de sistemas tradicionais de registros como bancos onde a informação é mantida por um servidor central, qualquer registro de transação pode ser vista por qualquer um.(BELIN, 2019)

Informação no sistema distributed ledger é acessível, transparente e livre de adulterações.

Bitcoin Blockchain é a primeira e mais usada tecnologia distributed ledger e com um pouco mais de dez anos de uso tem mostrado limitações no que diz respeito à escalabilidade.

Um livro de registro distribuído é como um banco de dados compartilhado e distribuído em múltiplos sites, instituições e locais. Ele possibilita que as transações estejam sob domínio público, dificultando um ataque. Os participantes em cada nó da rede podem acessar os registros distribuídos da rede e possuir uma cópia idêntica.

Além disso, qualquer modificação feita no livro de registros é refletida e copiada para todos os participantes em questão de segundos ou minutos.

2.1 Entendendo a tecnologia Distributed Ledger

Este livro de registros pode ser descrito como um registro de todas as transações ou contratos mantido de forma descentralizada em diferentes regiões e por diferentes pessoas, eliminando a necessidade de uma autoridade central. Toda a informação é armazenada de forma segura através da criptografia e pode ser acessada com o uso de chaves e assinaturas criptográficas.

Uma vez que a informação é guardada, se torna um banco de dados imutável. Enquanto os livros de registros centralizados são alvos fáceis de ataques, livros de registros descentralizados são difíceis de serem atacados já que eles precisam sofrer um ataque simultâneo. Além disso estes registros são imunes à modificações maliciosas por um único indivíduo ou grupos de indivíduos.

2.2 Exemplos reais da tecnologia Distributed Ledger

A tecnologia do livro de registros distribuído tem um grande potencial para revolucionar a maneira como governos, instituições e corporações trabalham. Isso pode ajudar o governo em arrecadações, fornecimento de passaportes. E atender outras indústrias como

música e entretenimento, arte, fornecimento de commodities.

Enquanto a tecnologia do livro de registros distribuído tem múltiplas vantagens, ainda está em sua fase inicial e precisa ser explorada para ser adotada na sua melhor forma.

2.3 Blockchain

Blockchain é uma cadeia de blocos que contém informação. Esta técnica foi originalmente descrita em 1991 por um grupo de pesquisadores e pretendia datar documentos digitais de forma que não fosse possível pré datá-los ou alterá-los. Embora não tenha quase sido usada até ser adaptada por Satoshi Nakamoto em 2009 para criar a criptomoeda Bitcoin.

Blockchain é como um livro de registros distribuído que é completamente aberto para qualquer um. Eles tem uma propriedade interessante: uma vez que os dados tenha sido gravados dentro da blockchain é muito difícil conseguir alterar. (PRADO, 2017)

2.4 Como isso funciona

Cada bloco contém dados, o hash do próprio bloco e o hash do bloco anterior. O dado inserido no bloco depende do tipo da blockchain. Por exemplo, Bitcoin blockchain grava detalhes sobre a transação como o remetente, destinatário e quantidade de moedas.

Cada bloco contém o hash. Hash pode ser comparado com uma impressão digital. Ele identifica um bloco e todo o seu conteúdo e é sempre único, como uma impressão digital.

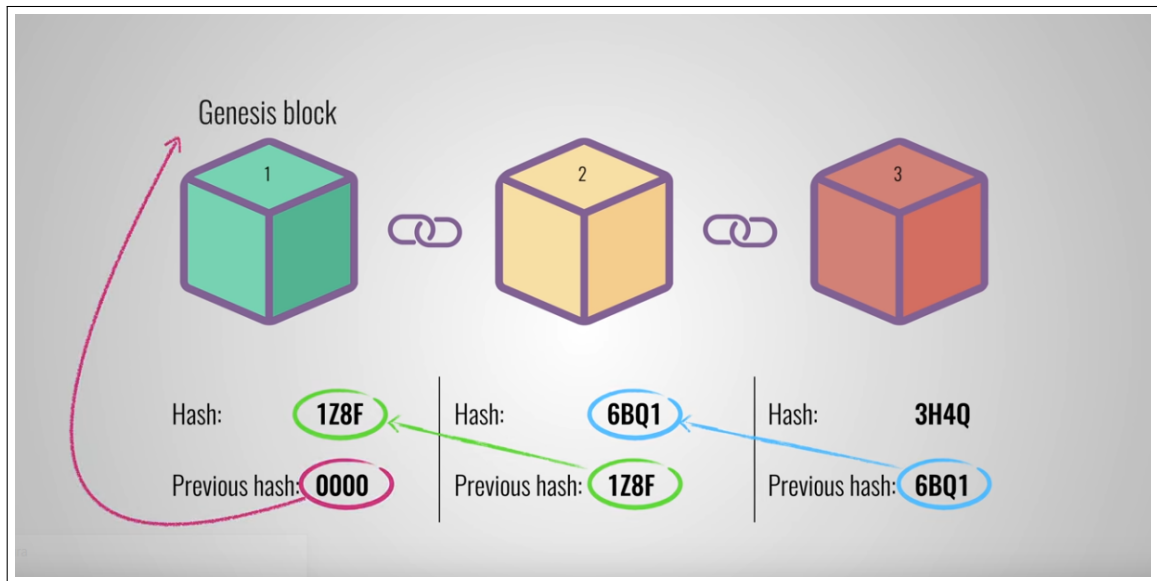
Quando o bloco é criado, o hash dele é calculado. Mudar algo dentro do bloco causa uma mudança no hash. Em outras palavras, hashes são muito úteis quando desejamos detectar mudanças nos blocos. Se a impressão digital de um bloco mudou, ele não é o mesmo bloco. (PRADO, 2017)

O terceiro elemento dentro de um bloco é o hash do bloco anterior. Isso cria uma cadeia de blocos e é essa técnica que cria uma blockchain segura.

Exemplo:

Na figura temos uma cadeia com 3 blocos. Cada bloco tem o seu hash e o hash do bloco anterior. O bloco número 3 aponta para o bloco número 2 e o bloco número 2 aponta para o bloco número 1. O primeiro bloco é um bloco especial, ele não pode apontar para blocos anteriores já que ele é o primeiro bloco. É chamado de bloco gênese. (CHANNEL, 2020) Vamos considerar que o segundo bloco foi alterado. Isso faz com que o hash do bloco mude também. Como consequência isso faz com que o bloco 3 e os blocos

Figura 1 - Exemplo da blockchain com três blocos



Fonte: (CHANNEL, 2020)

seguintes sejam invalidados. Eles não guardam mais um hash válido do bloco anterior. Então, mudar um único bloco torna todos os blocos seguintes inválidos. Porém usar hash não é o suficiente para prevenir adulterações.

Computadores nos dias atuais são muito rápidos e podem calcular milhões de hashes por segundo. É possível adulterar um bloco e recalculá-los todos os hashes de todos os outros blocos para tornar a blockchain válida novamente. Para amenizar isso a blockchain usa uma coisa chamada prova de trabalho (proof-of-work).

É um mecanismo que desacelera a criação de novos blocos. No caso das Bitcoins, é necessário em torno de 10 minutos para calcular a prova de trabalho necessária e colocar um novo bloco na cadeia. Esse mecanismo dificulta muito a alteração dos blocos. Se ocorrer a alteração de um bloco, será necessário recalcular a prova de trabalho para todos os blocos seguintes. A segurança da blockchain vem do uso do hashing e da força de trabalho (proof of work).

Porém, existe mais uma forma que as blockchains usam para garantirem segurança, a distribuição. Ao invés de usarem uma central única para controlar a cadeia, blockchains usam a rede peer-to-peer e qualquer um pode se juntar a cadeia. Quando alguém se junta a cadeia, adquire uma cópia completa da blockchain. (ROSIC, 2016)

2.5 O que acontece quando alguém cria um novo bloco

O bloco é enviado para todo mundo na rede. Cada nó verifica o bloco para ter certeza que ele não foi adulterado. Todos os nós da rede criam um consenso. Eles concordam sobre os blocos que são válidos e os que não são. Blocos que foram adulterados são rejeitados por outros nós na rede.

Para ter sucesso na adulteração de uma blockchain é necessário alterar todos os blocos na cadeia, refazer a prova de trabalho para cada bloco e conseguir o controle de mais de cinquenta por cento da rede peer-to-peer. Só assim o bloco adulterado será aceito por todos na rede. Isso é quase impossível de se fazer.

2.6 IOTA Tangle

O tangle apresenta soluções para as limitações do blockchain. O projeto IOTA foi criado inicialmente em 2015 por Sergey Ivancheglo, David Sontesbo, Serguei Popov e Dominik. Desde que o IOTA foi criado ele é administrado pela fundação IOTA, localizada na Alemanha.

Como outros projetos de criptomoedas, IOTA tinha como objetivo criar um sistema de pagamento que pudesse processar rápido, em segurança, custasse muito pouco e fosse escalável.

IOTA não tem intenção de se limitar à transações financeiras, mas permitir a transferência de qualquer tipo de dado de uma máquina para outra. O projeto IOTA tem sua própria criptomoeda chamada MIOTA.

Antes que IOTA fosse oficialmente lançada, conseguiu fundos através de uma ICO. No total, o projeto recebeu mais de 3000 Bitcoins.

O mais interessante do projeto moeda IOTA é que não usa a tecnologia blockchain como Bitcoin e Ethereum. Ao invés disso, usa uma tecnologia chamada Directed Acyclic Graph que o projeto IOTA chama de Tangle. Todos os pedaços de dados estão linkados.

Um grafo é formado conectando diferentes pontos, no tangle cada um desses pontos é direto, significando que os pontos(as transações) estão conectados numa direção significativa, sempre para frente. Os pontos não podem se conectar de maneira que formem um ciclo. E um ponto não pode fazer um ciclo voltando para si mesmo.

2.7 Transações Seguras

Para uma transação ser completada e adicionada ao tangle, precisa verificar transações antigas. Isso possibilita que o tangle calcule quando uma transação é autêntica pois cada

IOTA que já existiu e cada transação que já aconteceu pode ter seu caminho traçado para a origem da primeira transação.

Se cada transação tiver de provar que ela pode estar lá, fará isso provando de onde ela veio. Pode parecer que este processo é lento mas na realidade ele é bem eficiente. Graças a um outro princípio matemático chamado cumulative weight.

Quanto mais confiável e verificável uma transação se torna maior será seu peso. Quando novas transações fizerem referência a essas transações com peso alto, a rede pode assumir rapidamente que elas são legítimas sem precisar recalcular toda a trajetória histórica do tangle.

Esse processo de confirmar transações pode ser usada em qualquer nova transação que entra no tangle e como isso é feito automaticamente pelo dispositivo, cada transação precisa confirmar duas outras transações anteriores. O resultado é uma rápida e segura criptomoeda com uma escalabilidade ilimitada e uma segura distributed ledger.

2.8 IOTA Mining - Confirmação das transações

Tangle não usa miners. Quando as transações são confirmadas na Bitcoin, as pessoas que voluntariamente usam o poder de seus computadores são beneficiadas com Bitcoins extras. Estes miners não estão envolvidos na transação. Eles só confirmam transações de outras pessoas.

É o mecanismo chamado de prova de trabalho (Proof of work). O problema é que usando somente PoW para confirmar uma transação fica cada vez mais caro e difícil para o miner, mais poder computacional e eletricidade são necessários.

O resultado disso são transações lentas e custosas. Isso significa que Bitcoin Blockchain só consegue confirmar 7 transações por segundo. Na rede IOTA, as coisas são diferentes, todo mundo que precisa usar a rede deve contribuir para a confirmação de transações de outras pessoas.

Exemplo:

Juliana enviou fundos para Miguel.

Durante essa transferência, Juliana também precisa confirmar as transações de duas outras pessoas, Carlos e Raquel.

Juliana faz isso contribuindo com o poder computacional do seu computador.

Na próxima vez que Carlos e Raquel quiserem enviar o dinheiro de alguém, eles precisam fazer a mesma coisa, confirmar transações de outras duas pessoas.

A ideia por trás disso é fazer com que cada vez mais pessoas usando o sistema IOTA torne ele mais escalável. Com isso não existiria um limite no número de transações que a rede poderia processar por segundo.

2.9 Taxas nas transações da moeda IOTA

A grande vantagem do Tangle é já que não existe nenhum IOTA mining, as transações são livres de taxas. Todos contribuem para a rede confirmando transações de terceiros. O único custo é a eletricidade. Isso torna a moeda IOTA o ideal no que diz respeito ao sistemas de pagamento.

2.10 Velocidade da transação da moeda IOTA

IOTA ainda é muito recente, não existe nada oficial sobre o tempo das transações. Existem muitos fatores que determinam a rapidez com que os fundos chegam na carteira.

Primeiro, como todo mundo que deseja enviar fundos precisa confirmar duas transações anteriores, eles precisam realizar uma rápida prova de trabalho (Proof of work). O tempo que isso leva depende da performance do computador do usuário. Quanto mais rápido for a realização de uma prova de trabalho, mais rápido vai ser verificada a transação de alguém.

Quanto mais pessoas usarem a rede da moeda IOTA, mais e mais transações serão processadas. Infelizmente, IOTA ainda é um projeto novo, não existem muitas pessoas usando.

Para ter certeza que a rede é operável, o time IOTA criou um coordenador. É como um protocolo centralizado que performa trabalhos como um usuário individual. Quando mais pessoas usarem o IOTA não será necessário um coordenador.

Por isso, nesta fase de desenvolvimento, as transações podem levar minutos, se a rede não estiver com uma boa performance, pode levar mais tempo ainda. Para a rede da moeda IOTA funcionar com todo seu potencial, é necessário mais pessoas usando.

2.11 Como o IOTA pode ser usado

Se a plataforma for habilitada para processar instantaneamente, sem cobrar taxas e com um número ilimitado de transações, se torna perfeito para um sistema global de pagamento.

As pessoas e organizações poderiam enviar fundos para qualquer lugar do mundo, sem taxas, ao pressionar um botão. É esperado que o projeto IOTA processe mais do que transações financeiras. O que tem sido chamado de Machine-to-Machine Economy.

Um exemplo disso é o sistema de cobrança da indústria de veículos elétricos. IOTA chegou a divulgar um facilitador de cobranças na Holanda (Países Baixos) que possibilita as pessoas pagarem pela eletricidade que eles usam de forma automática.

Isso pode ser aplicado para praticamente qualquer coisa que precise transmitir dados de uma máquina para outra.

O projeto IOTA tem potencial para ser usado em qualquer escala global. Para isso, seus desenvolvedores precisam concluir o projeto, possibilitando que o IOTA opere em escala global e ilimitada, com transações gratuitas, podendo processar qualquer tipo de dado de forma instantânea.

2.12 Considerações finais do IOTA

O projeto moeda IOTA é bem ambicioso, porém, é importante observar que está em sua fase inicial. A rede já apresentou alguns problemas técnicos.

O protocolo coordenador é centralizado, isso significa que caso ele pare de funcionar apropriadamente, toda a rede corre riscos. Isso já aconteceu muitas vezes.

O coordenador central também foi instalado para proteger a rede de 34 por cento dos ataques. Então, se alguém adquirir um grande poder computacional na rede será possível realizar alterações.

Bitcoin também corre risco de sofrer um ataque. Porém, neste caso, os hackers precisam controlar 51 por cento do poder total do hashing. Isso é virtualmente impossível agora já que a blockchain do Bitcoin possui muitos miners.

Porém no caso do IOTA, se o coordenador central parar de funcionar, a rede fica sob ameaça. Mesmo que a ideia central do IOTA seja ser descentralizado, isso não será possível enquanto existir um coordenador central.

Existe um receio na comunidade/rede de que a moeda IOTA sofra um replay attack. Um replay attack acontece quando os hackers são capazes de repetir uma transação sem o remetente saber disso. Se forem bem sucedidos, a carteira de alguém pode ser roubada.

Em 2017, pesquisadores do Massachusetts Institute of Technology(MIT) publicaram um artigo acadêmico afirmando que a moeda IOTA tinha muitas falhas de segurança.

Isso tudo mostra que o projeto moeda IOTA está longe de se tornar um produto finalizado.

3 SISTEMAS DE TRADING

Não são as crises que mudam o mundo, e sim nossa reação a elas.

Zygmunt Bauman

Este capítulo aborda os conceitos essenciais para compreender a proposta deste trabalho. É apresentado o sistema automatizado de trading com suas vantagens e desvantagens (Seção 1.1), o que é necessário saber antes de decidir otimizar e como evitar golpes.

3.1 Sistemas automatizados de trading

O que é um sistema automatizado de trading?

É um sistema que permite aos usuários determinar regras específicas para a entrada e saída de uma operação de trading, uma vez programado, pode ser executado automaticamente por um computador.(LIEW, 2020)

Uma das grandes vantagens da estratégia automatizada é que ela remove a emoção humana das decisões de trading já que os trades são colocados assim que certas condições são atingidas.

A entrada e saída de uma operação de trade pode ser baseada em condições simples ou estratégias complicadas que requer um conhecimento profundo da linguagem de programação usada na plataforma de trading.

Sistemas automatizados de trading requerem o uso de um software ligado a um direct access broker, e qualquer regra específica precisa ser escrita nessa linguagem proprietária da plataforma. A plataforma TradeStation, por exemplo, usa a linguagem de programação EasyLanguage. Por outro lado, a plataforma NinjaTrader usa NinjaScript.

Algumas plataformas de trading possuem estratégias pré-definidas que permitem aos usuários selecionar indicadores técnicos para a construção de uma série de regras que posteriormente serão automatizadas no trade. Usuários também podem definir o tipo de ordem(market, limit) , quando o trade será acionado ou mesmo usar as opções default da plataforma.

A maioria dos traders escolhem desenvolver seus próprios indicadores e estratégias. Eles normalmente trabalham perto dos programadores para construir o sistema. Enquanto isso requer mais esforço do que usar as opções default da plataforma, isso possibilita uma maior flexibilidade, e os resultados podem ser melhores. Como tudo no mundo dos tradings, infelizmente, não existe uma estratégia perfeita de investimento que irá garantir o sucesso.

Uma vez que as regras estejam estabelecidas, o computador pode monitorar o mercado pesquisando oportunidades de compra e venda baseadas nas especificações das estratégias do trading. Dependendo das regras especificadas, assim que um trade inicia, qualquer ordem vai ser gerada automaticamente (stop losses, trailing stops).

No mercado que se movimenta bem rápido, essas ordens instantâneas podem significar a diferença entre uma perda pequena ou uma perda catastrófica.

3.1.1 Vantagens dos sistemas automatizados

Existe uma longa lista de vantagens ao usar um computador para monitorar o mercado procurando por oportunidades de trading e execução de trades, incluindo:

Minimizar as emoções

Sistemas automatizados de trading minimizam as emoções durante o processo de trading. Como as ordens são executadas automaticamente assim que as regras definidas no trade são atingidas, traders não vão poder hesitar ou questionar a operação de trade. (BERTOLUCCI, 2018)

Backtesting

As regras de trading são testadas em dados históricos do mercado para determinar a viabilidade da ideia. Quando é construído um sistema para automatizar o trading, as regras precisam ser absolutas, sem nenhuma margem para interpretação. O computador não pode fazer chutes e precisa ser definido exatamente o que ele precisa fazer. Traders podem usar essas regras específicas e realizar testes em dados históricos antes de arriscarem o dinheiro numa operação real de trading. Um backtesting cuidadoso permite refinar e validar uma ideia de trading, além de determinar a expectativa do sistema, quanto um trader espera ganhar ou perder.

Manter a disciplina

Como as regras de trading são estabelecidas e executadas de forma automática, a disciplina é mantida mesmo em um mercado volátil. A disciplina normalmente é perdida por conta de fatores emocionais como o medo da perda ou o desejo de ganhar mais de um trade. O trading automático ajuda a manter a disciplina pois o plano traçado é seguido precisamente. Por exemplo, uma ordem para comprar 10 bitcoins não será confundida com uma ordem para comprar 100 bitcoins. Uma das grandes vantagens do trading é pla-

nejar e executar. Traders que ignoram as regras estão alterando toda a expectativa que o sistema poderia ter. Não existe nada como um plano de trading que funcione 100 por cento do tempo. Perdas fazem parte do jogo. Porém perdas podem ser traumatizantes psicologicamente, e por isso um trader que teve duas ou três perdas seguidas pode decidir pular o próximo trade. Se o próximo trade era de sucesso, o trader destruiu qualquer expectativa que o sistema tinha de ganhar. Concluindo, sistemas automáticos de trading permitem que os traders tenham resultados consistentes, seguindo o plano.

Diversificar o trading

Sistemas automáticos de trading permitem ao usuário realizar múltiplas operações de trading e várias estratégias ao mesmo tempo. Isso tem o potencial de espalhar o risco enquanto cria uma proteção contra posições de perda. O que pode ser muito difícil para um humano executar, pode ser realizado por um computador em questão de milissegundos. O computador é capaz de pesquisar oportunidades de trading por todo mercado, gerar ordens e monitorar os trades.

3.1.2 Desvantagens dos sistemas automatizados

Sistemas automatizados de trading possuem muitas vantagens mas existe uma outra realidade que os traders precisam estar cientes.

Falhas mecânicas

A teoria por trás do trade torna ele simples: preparar um software, programar as regras e deixar a execução fluir. Na realidade, esse sistema automatizado é um método sofisticado de trading mas não infalível. Dependendo da plataforma de trading, uma ordem pode residir em um computador, não em um servidor. Isso significa que se a conexão com a internet for perdida, uma ordem pode não ser executada no mercado. Ainda podem existir discrepâncias entre o trade teórico gerado pela estratégia e o que é executado pela plataforma tornando eles trades reais. Muitos traders devem esperar uma curva de aprendizado quando estão usando o sistema automatizado, por isso é uma boa ideia começar com uma quantia pequena enquanto o processo está sendo refinado.

Monitoramento

Embora fosse ótimo ligar o computador e deixar o sistema funcionar sozinho, sistemas automáticos de trading precisam de monitoramento. Isso por conta das falhas tecnológicas,

como perda de conectividade, perda de energia, falha no computador, e outras peculiaridades do sistema. É possível que nestes sistemas automatizados ocorram anomalias que podem resultar em ordens erradas, perdas de ordens ou ordens duplicadas. Se o sistema é monitorado, estes eventos podem ser identificados e resolvidos rapidamente.

Otimização Excessiva

Embora não específico para os sistemas automatizados de trading, traders que empregam técnicas de backtesting podem criar sistemas que são ótimos no papel mas performam mal no mercado. Otimização excessiva refere-se ao excesso do ajuste de curva que produz um plano de trading não confiável. É possível, por exemplo, refinar uma estratégia para alcançar resultados excepcionais nos dados históricos em que foi testada. Traders assumem, às vezes, incorretamente que um plano de trading precisa ser 100 por cento rentável ou nunca deve experimentar um drawdown para ser um plano viável. Com isso, parâmetros podem ser ajustados para criarem um plano quase perfeito que irá falhar completamente assim que for colocado em situações reais no mercado.

3.2 Como evitar golpes

Alguns sistemas oferecem grandes lucros por um preço pequeno. Para determinar quando um sistema é legítimo ou não:

1. Examinar tudo antes de pagar por uma conta de trading e sempre fazer perguntas. Caso contrário, dinheiro pode ser perdido no final;
2. Fazer uma pesquisa sobre o sistema em questão. Ler os termos e condições antes;
3. Procurar depoimentos de pessoas que já utilizaram o serviço. Verificar sites de terceiros;
4. Verificar se o sistema possui um período de teste. Muitos sites enganosos não possuem esse período para avaliação.

3.3 Antes de decidir automatizar

A palavra otimização faz parecer que é um processo simples mas é necessário estar ciente de algumas coisas antes de decidir automatizar:

1. Ter certeza que é realmente necessário um sistema automatizado. Existem promessas de se ganhar dinheiro mas pode-se levar muito tempo. Sistemas automatizados

são complexos e sem conhecimento e experiência, dinheiro pode ser perdido;

2. Ter certeza que os prós e contras são entendidos;
3. Manter as estratégias simples no início e só depois ir para estratégias mais complicadas;
4. Lembrar que não existe uma solução única;
5. Selecionar a estratégia preferida e o quanto deseja customizar.

4 SISTEMA PROPOSTO

A concordância faz com que permaneçamos estacionados. A discordância faz com que crescamos.

Mário Sergio Cortella

Este capítulo aborda o que é necessário fazer para construir um robô de criptomoedas. Além de apresentar opções, como fazer download de um código fonte aberto e adaptá-lo ou começar a desenvolver do zero o robô.

4.1 Download de um código fonte aberto

Existem muitas maneiras de se construir um trading bot de criptomoedas. A mais barata e mais simples é simplesmente usar um código aberto. Isso requer um conhecimento técnico menor e ajuda a manter custos e tempo de desenvolvimento reduzidos.

Claro que para desenvolver um robô com suas próprias características, continuar o desenvolvimento, e resolver bugs e problemas com segurança, é necessário pelo menos um desenvolvedor de software.

Uma vez que o robô já tenha sido desenvolvido, o lado negativo é que fica mais difícil desenvolver suas características e adaptar o algoritmo.

Melhores robôs de criptomoedas com código aberto(BITCONGRESS, 2020):

1. Gekko;
2. Zenbot.

4.2 Começando do zero: Montar um time de desenvolvimento e começar a programar

Desenvolver um trading software de criptomoedas pode ser um projeto pessoal e financeiramente rentável. A parte mais importante é ter certeza que existe um time dedicado e apaixonado com as habilidades e experiências necessárias.

O mais básico trading bot pode ser feito em poucas semanas. Exemplo: um robô que realiza trading entre duas exchanges.

Para criar um robô mais sofisticado que opere em múltiplas exchanges, será necessário mais tempo. Esse tempo adicional é necessário para criar o algoritmo e garantir que não existe nenhuma falha de segurança a ser explorada por hackers.(HEDGETRADE, 2019)

Passos importantes ao desenvolver um trading bot do zero:

1. Decidir a linguagem de programação a ser usada

É uma boa ideia usar uma linguagem de programação com a qual haja familiaridade. Python, Javascript, Perl e C são as linguagens mais comumente usadas. A vantagem de usar uma linguagem de programação bem conhecida é a facilidade de conseguir outros desenvolvedores para ajudar na escrita e correção do código futuramente.

2. Achar as APIs

Antes de começar a desenvolver o robô é necessário pesquisar e selecionar todas as APIs que vão permitir que o robô acesse e realize trade. Todas as corretoras de criptomoedas oferecem APIs para acesso aos dados.

3. Lista com as principais corretoras:

Binance

Bitfinex

Coinbase

Kraken

Bittrex

CEX.IO

Poloniex

4. Criar contas em todas as corretoras que serão usadas

Criar contas é relativamente simples. Mas é importante ter em mente que exchanges diferentes podem ter procedimentos distintos para a criação de contas. Algumas requerem que informações pessoais sejam autenticadas enquanto outras permitem a realização de trading anônimo. Autenticações requerem mais tempo. É importante ter isso em mente na hora de planejar o projeto.

5. Escolha um modelo de trading bot

Trend following, Arbitrage ou Market Making , etc. Tenha em mente que quanto mais complexo o modelo do robô maior tempo será necessário no seu desenvolvimento.

6. Arquitetura

A arquitetura do robô vai ter uma grande implicação em como ele vai funcionar. A chave para um robô operar é decidir sobre os algoritmos que ele irá usar para interpretar os dados. Algoritmos de trading são uma grande indústria que fazem bilhões

de dólares a cada ano. Para qualquer algoritmo, o modelo matemático em que ele se baseia precisa ser sólido. Caso não seja, o robô mostrará ser não confiável e dinheiro será perdido. Parte do processo é determinar o tipo de dados que o algoritmo precisa interpretar. Para modelos complexos de trading é necessário identificar coisas como ineficiências do mercado. Isso significa que será necessário analisar dados históricos. Tudo isso precisa ser considerado antes de começar a criação do robô.

7. Criação

Essa é a parte do processo que mais consome tempo. Se existir um time de desenvolvedores trabalhando em diferentes partes do projeto, é preciso ter certeza que existe uma boa comunicação. Uma opção é usar o Slack ou um programa similar em que os membros do time possam se comunicar. Agendar pequenas reuniões semanais para ter certeza que todos os membros do time estão alinhados com o desenvolvimento.

8. Testando

É necessário ter certeza que o robô funciona como o esperado e é capaz de lidar com as variações e flutuações de dados que irá receber. Fatores como risco vs. retorno e erros de modelo como overfitting precisam ser analisados neste estágio. Também é nessa hora que ocorrem os ajustes de performance. A otimização de como é esperado que o robô se comporte.

9. Deploy do robô

Colocar o robô em produção para receber dados reais do mercado. Nesse momento muitos acham que irão fazer fortuna mas nenhuma plataforma na história foi colocada em produção sem experimentar problemas. Um bom robô de trading vai evoluindo aos poucos. Monitoramento constante da performance do robô é estritamente recomendável, pelo menos nos primeiros meses. Depois disso, você pode se sentir confiante o suficiente para deixar o robô em execução sem muito supervisão.

4.3 Eliminando o trabalho pesado: Pagar um time de desenvolvimento externo para criar o robô

Para facilitar o desenvolvimento, uma opção é contratar um time de desenvolvedores externos para fazer o trabalho pesado. Um bom time de desenvolvimento pode entregar o projeto em um curto espaço de tempo e garantir que o robô vai funcionar da melhor forma. Já que o robô vai estar movimentando grandes quantias de dinheiro de terceiros, confiabilidade é muito importante. O mercado de criptomoedas é construído

sobre confiança, logo, o robô precisa ser 100 por cento confiável para garantir sucesso.
(DAVIES, 2020)

5 ESTUDO DE CASO

A insatisfação é o primeiro passo para o progresso de um homem ou de uma nação.

Oscar Wilde

Este capítulo tem como objetivo apresentar o MyRobô, bot que compra e vende bitcoins nas corretoras Binance e Bitfinex, e como ele foi desenvolvido.

Linguagem de programação: Python3.

APIs usadas: Binance, Bitfinex.

5.1 Primeiros passos no desenvolvimento do robô

5.1.1 Comprando bitcoins e obtendo as APIs

Antes de começar a desenvolver o código do robô é necessário comprar bitcoins e ter acesso aos sites das corretoras onde é possível obter as APIs.

1. Cadastro na BitPreço e depósito em reais. BitPreço é um marketplace de bitcoins onde é possível depositar valores em reais e comprar bitcoin pelo menor preço, já que o site reuni uma lista de corretoras e os preços das bitcoins de cada uma. Figura1.
2. É necessário realizar um cadastro nas corretoras Bitfinex e Binance. Figura2.Figura3.
3. Após o cadastro nas corretoras, duas compras são realizadas no site da BitPreço. A primeira é a compra de bitcoin na corretora Bitfinex e a segunda é a compra de bitcoin na corretora Binance. Bitfinex e Binance são corretoras internacionais de compra/venda de criptomoedas. Figura4.
4. Após as operações serem realizadas, é possível entrar nos sites destas duas corretoras e verificar o saldo disponível em bitcoins. Figura 5. Figura 6.
5. No site das corretoras também é necessário obter as chaves públicas e privadas. O robô usa estas chaves para fazer requisições às APIs Binance e Bitfinex. Figura 7. Figura 8.

Figura 2 - Exemplo de cadastro e depósito em reais na BitPreço

bitPreço

VOLUME 24H: BRL 81,63612718 ÚLTIMO PREÇO: R\$ 43,354,77 MENOR: R\$ 42,888,70 MAIOR: R\$ 43,560,83 VARIAÇÃO: 2,50%

Juliana Costa da Silva

CONTA LIMITE OPERACIONAIS SEGURANÇA

Informações Gerais

Razão Social ou Nome completo: Juliana Costa da Silva Data de nascimento: Não consta CPF/CNPJ: Requerido

Telefone para contato: +55 (00) 00000-0000 E-mail: jujurj.juliana@gmail.com

Endereço + número (Domicílio Fiscal): XXXXXXXXXXXXXXXXXXXXXXXX Cidade: XXXXXXXXXXXXXXXX Estado: RJ Cep: Requerido

Verificação de documentos: Pendente

Contas Bancárias

*TODOS OS CAMPOS ACIMA SÃO OBRIGATORIOS.

Seu banco: 184 - Banco Itaú BBA S.A. Agência: 0000 Conta: Conta Corrente Conta Poupança

SALVAR

Fonte: (BITPRECO, 2020a)

Figura 3 - Exemplo de cadastro na corretora Bitfinex

BITFINEX

FEATURES SECURITY MORE PROFESSIONAL TOKEN SALES AFFILIATE PROGRAM

LOG IN SIGN UP English

Sign Up

Username Email address

Password Confirm password

Timezone: (GMT+00:00) UTC

Referrer code

Captcha Text: AMV736

By signing up on Bitfinex, you agree to our Terms & Conditions and Anti-Spam Policy

Open Account

USD EUR

SYMBOL	LAST PRICE	24H CHANGE	24H HIGH	24H LOW	24H VOLUME
BTC/USD	10,054	-0.5%	10,187	9,962.4	87,802,461 USD

Fonte: (BITFINEX, 2020b)

Figura 4 - Exemplo de cadastro na corretora Binance

Fonte: (BINANCE, 2020b)

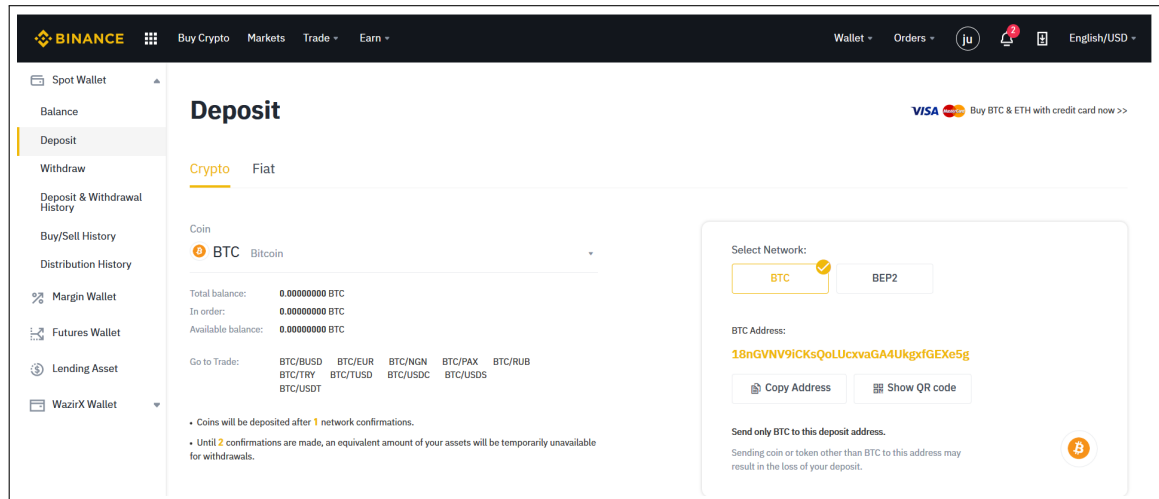
Figura 5 - Exemplo de compra de bitcoins pelo site da bitPreço

Preços	Quantidade	Total R\$
43.416,20 a 43.416,20	0,00201694	87,57
43.416,20	0,00201694	87,57
43.416,00 a 43.371,98	0,50122167	21.741,22
43.359,26 a	0,44833843	19.436,77

Preços	Quantidade	Total R\$
43.608,85 a 43.642,47	0,65846614	28.718,26
43.608,85	0,13309048	5.803,92
43.608,85	0,00150000	65,41
43.608,86	0,00201896	88,04

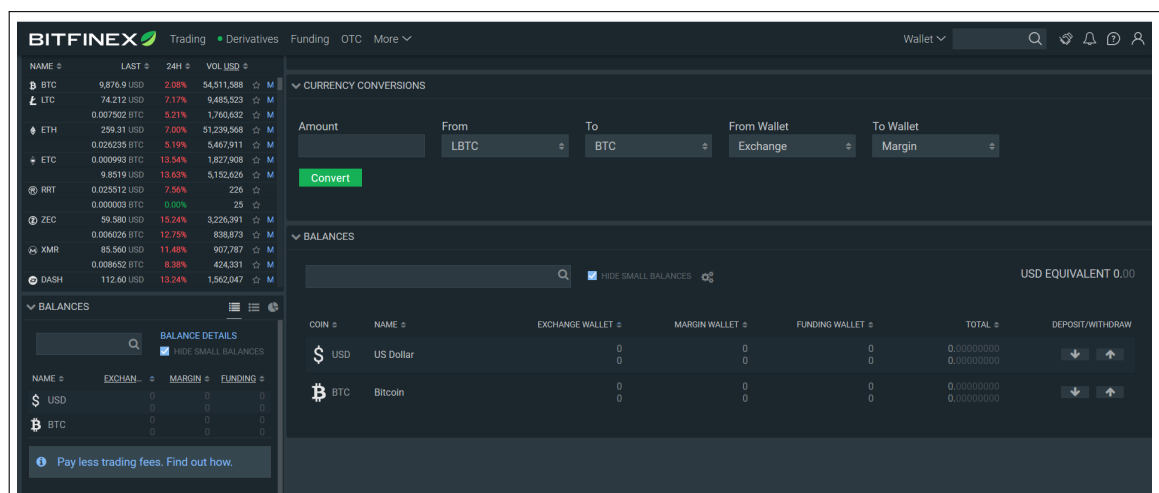
Fonte: (BITPRECO, 2020b)

Figura 6 - Saldo na exchange Binance



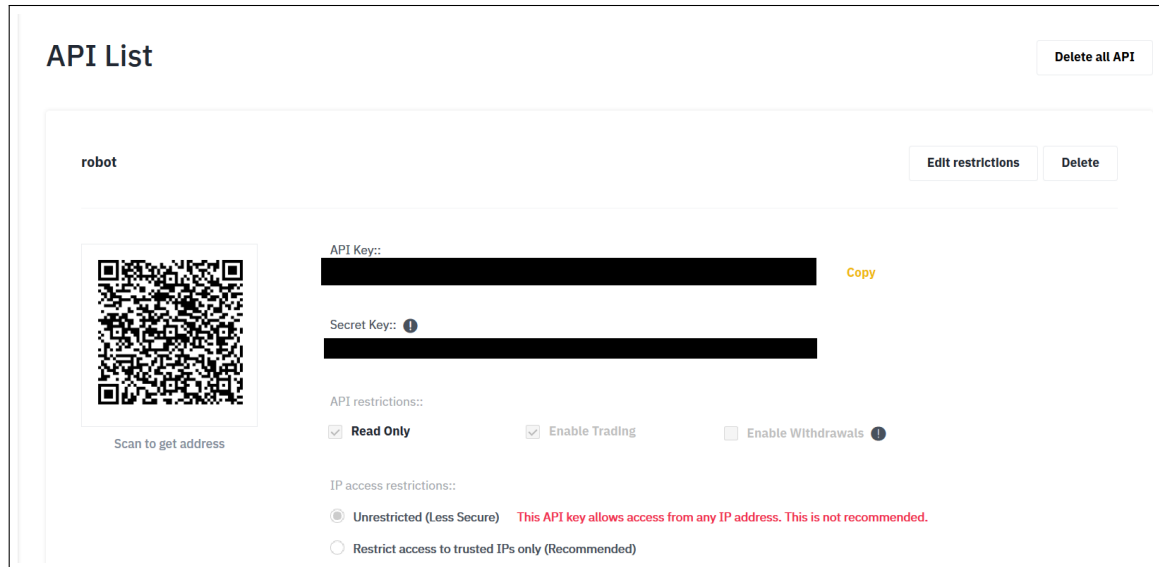
Fonte: (BINANCE, 2020c)

Figura 7 - Saldo na exchange Bitfinex



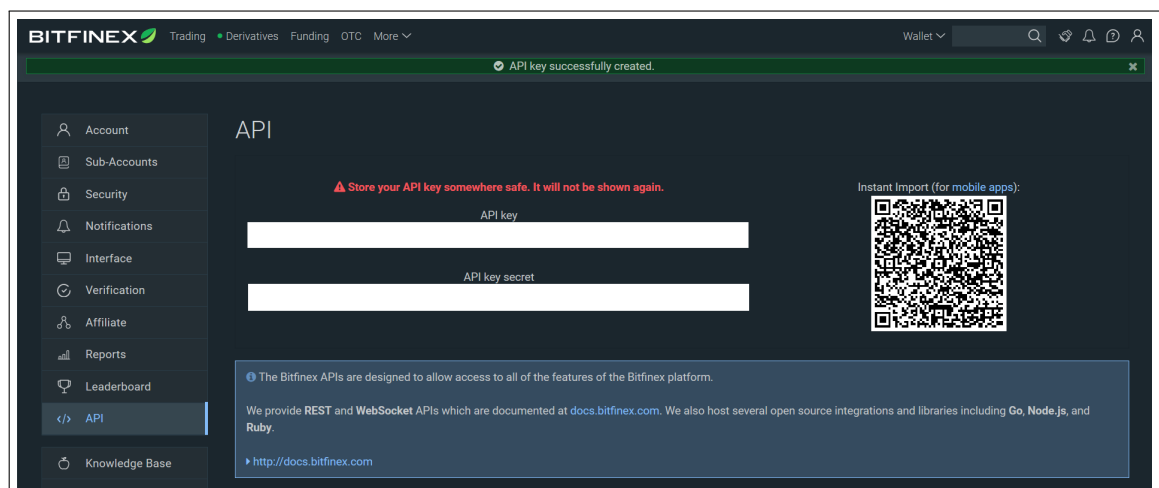
Fonte: (BITFINEX, 2020c)

Figura 8 - ApiKey Binance



Fonte: (BINANCE, 2020a)

Figura 9 - ApiKey Bitfinex



Fonte: (BITFINEX, 2020a)

5.1.2 APIS

API é uma interface de programação de aplicações. Pode ser definida como um conjunto de protocolos, procedimentos e ferramentas que possibilitam a interação entre duas aplicações. É o software intermediário que envia a mensagem para o servidor e retorna a resposta para o cliente. APIs ajudam os programadores a agilizarem a escrita de código, por tornarem muitas partes do código acessíveis para serem usadas e reutilizadas nas aplicações. Deixando assim os desenvolvedores livres para focarem no que é realmente útil na aplicação.

No caso do MyRobô, uma API REST é necessária para solicitar informações das corretoras e enviar informações. Quando é necessário checar o saldo de uma das carteiras, como a carteira bitcoin, é só fazer uma chamada get http, usando o endpoint disponibilizado pela própria corretora, informando a carteira desejada. E, o saldo é retornado. Quando uma compra é solicitada, o mesmo procedimento é realizado, porém, agora uma chamada post http é realizada.

Todas essas solicitações são transportadas entre cliente(o MyRobô solicitando uma ação) e servidor(corretoras Binance ou Bitfinex) no formato JSON. JSON é um formato utilizado para guardar e transportar dados. O MyRobô converte objetos python em JSON e envia para as corretoras. No retorno, o MyRobô recebe a resposta em formato JSON e converte em objetos Python.

JSON é muito fácil de ser lido.

Exemplo de como os dados são encapsulados em JSON para realizar uma solicitação de compra na corretora Bitfinex.

```
body = "request": /v1/order/new,  
"nonce": nonce,  
"symbol": btc,  
"amount": 0.1,  
"price": 10,  
"exchange": bitfinex,  
"side": side,  
"type": market
```

Os dados são guardados como pares de chave/valor. Isso significa que a chave fica do lado esquerdo e o valor do lado direito.

5.1.3 Padrão REST x RPC

REST é tudo sobre uma relação cliente servidor em que os dados do servidor ficam disponíveis em representações simples como JSON e XML através de um método chamado

hypermedia.

Hypermedia é um fundamento do REST, é essencialmente a ideia de prover links para todos os recursos.

O estilo REST deve ser tratado como um recurso que provê dados de domínio. O lado bom disso é que os dados são segregados em diferentes domínios. Isso facilita muito as coisas quando existem um grande número de apps fazendo requisição de dados. Isso separa os dados da aplicação ou da lógica do negócio.

Antes do padrão REST se tornar popular, o que aconteceu quando empresas como Twitter e Facebook nomearam suas respectivas APIs de REST, muitas APIs eram construídas usando o padrão RPC/XML.

O estilo RPC é bom quando só é necessário que um trabalho seja bem feito. Isso acaba sendo útil para um ou dois clientes por ser um único serviço. RPC endpoint pode implementar a lógica do negócio dentro do serviço, garantindo que isso faça apenas uma coisa. Isso adiciona simplicidade e clareza para o serviço.

5.1.4 Utilizando as APIs para autenticação no código e execução das ordens

Algumas requisições às APIs só são possíveis utilizando autenticação. Autenticação é quando uma entidade prova uma identidade. Ou seja, autenticação prova quem é quem. Nesta autenticação, informações como chave privada são passadas no cabeçalho das requisições.

Métodos de Autenticação API Keys

Nesse método, um valor único de chave de API é usado para cada acesso do usuário, garantindo que o usuário é conhecido. Quando o mesmo usuário tentar entrar novamente no sistema, a mesma chave será usada. Como acontece com o MyRobô.

5.2 **MyRobô em execução**

Nos testes do MyRobô foi utilizado o tipo de ordem à mercado.

A ordem à mercado é utilizada quando se deseja comprar ou vender bitcoins de imediato, ou seja, ao lançar a ordem é utilizado o menor preço de compra e/ou o maior preço de venda, disponíveis no livro de ofertas.

O robô foi desenvolvido em Python3. Para rodá-lo é necessário executar o script RobotController.py. Figura 12.

Antes do robô iniciar suas operações é necessário atentar para o seguinte:

O MyRobo realiza trading à mercado. Compra e vende bitcoins em corretoras de criptomoedas visando obter lucro. Se o desejado é vender bitcoins, o robô precisa

Figura 10 - Exemplo do código do MyRobô que mostra a Autenticação Bitfinex

```
base_url = "https://api.bitfinex.com"
key = [REDACTED]
secretKey = [REDACTED]

def headers(self, nonce, path, body):
    body = json.dumps(body)
    body = body.encode('utf8')
    body = b64encode(body)

    h = hmac.new(self.secretKey.encode('utf8'), body, hashlib.sha384)

    signature = h.hexdigest()

    return {
        "x-bfx-apikey": self.key,
        "x-bfx-payload": body,
        "x-bfx-signature": signature,
        "content-type": "application/json"
    }
```

Figura 11 - Exemplo do código do MyRobô que mostra a execução de ordem Bitfinex

```
def order(self, symbol, amount, price, side, _nonce):
    nonce = self.nonce() + _nonce

    path = "/v1/order/new"

    body = {
        "request": path,
        "nonce": nonce,
        "symbol": symbol,
        "amount": amount,
        "price": price,
        "exchange": "bitfinex",
        "side": side,
        "type": "market"
    }
```

Figura 12 - Exemplo do código do MyRobô que mostra a execução de ordem Binance

```
def order(self, symbol, amount, price, side):

    nonce = self.nonce()
    path = "/api/v3/order"

    body = {
        "symbol": symbol,
        "side": side,
        "type": "market",
        "timeInForce": "GTC",
        "quantity": amount,
        "price": price,
        "recvWindow": "20000",
        "timestamp": nonce
    }
```

Figura 13 - Exemplo do MyRobô sendo colocado em execução no terminal

```
10/02/2020 23:49 <DIR> .
10/02/2020 23:49 <DIR> ..
21/01/2020 17:19 2.373 checarSaldos.py
01/12/2019 01:36 7.968 Database.py
10/02/2020 23:40 11.361 ExchangeBinance.py
10/02/2020 23:45 20.220 ExchangeBitfinex.py
16/10/2019 00:09 740 MoedaSpread.py
16/10/2019 00:09 1.744 MoedaTransporte.py
21/01/2020 17:20 122 nonce.txt
21/01/2020 17:24 20.502 RobotController.py
10/02/2020 23:48 <DIR> __pycache__
8 arquivo(s) 65.030 bytes
3 pasta(s) 542.504.861.696 bytes disponíveis

C:\Users\Juliana\Desktop\tradingbot>RobotController.py
```

olhar o livro de ofertas das corretoras procurando pelo maior valor de compra à mercado. Se o desejado é comprar bitcoins, o robô precisa olhar o livro de ofertas das corretoras procurando pelo menor valor de venda à mercado.

5.2.1 Início da operação do MyRobô

O robô inicia sua operação procurando pelo maior preço de compra e menor preço de venda nas corretoras Bitfinex e Binance. Se ele verificou que o maior preço de compra está na bitfinex(precisa vender) e o menor preço de venda está na binance(precisa comprar), o robô faz o seguinte:

Inicia sua operação olhando a carteira bitcoin na bitfinex, verificando se existe saldo disponível, e se esse saldo é maior que o valor mínimo para realizar uma operação de venda. Este valor mínimo é exigido pela própria corretora(bitfinex). É realizada uma consulta à api da bitfinex por esses valores. Figura13.

Então uma chamada à api bitfinex é realizada informando a moeda que desejo vender (seu símbolo), quanto desejo vender, o valor de compra, o tipo da operação, neste caso, o tipo é “sell”, e o nonce. Em criptografia, um nonce é um número arbitrário que só pode ser usado uma vez.

```
data = self.order(symbol, carteira, valorCompra, "sell", nonce)
```

A api retorna o resultado da operação, se foi um sucesso ou se ocorreu algum erro. Após um resultado de sucesso, o robô precisa checar a carteira USD bitfinex para verificar se o valor já foi depositado. Mesmo após a api retornar sucesso é necessário esperar alguns segundos, entre 30 e 60s, para o valor cair na carteira.

Quando o valor cair na carteira USD significa que o robô já pode comprar bitcoin na corretora binance com o menor preço de venda.

As consultas por valores de compra/venda das moedas não necessitam de autenticação. São chamadas públicas às apis. Somente as consultas que envolvem visualização ou alteração das carteiras é que precisam ser autenticadas.

No início, foi verificado que a Binance possuía o menor preço de venda.

Para realizar a compra de bitcoins na binance, preciso transferir o saldo da carteira USD Bitfinex para carteira USD Binance. Para isso utilizo uma chamada à api da bitfinex. Preciso sacar o valor na bitfinex para depositar na binance. Ou seja, preciso realizar uma transferência.

```
data = self.withdraw(name, amount, address, nonce)
```


Figura 14 - Exemplo do código do MyRobô que mostra a consulta aos saldos das carteiras Bitfinex

```
def exibirSaldoCarteirasUsadas(self, moeda):
    carteira = {
        moeda: self.getCoinAvailableExchange(moeda),
    }

    return carteira

def getCoinAvailableExchange(self, symbol):
    data = self.getWalletBalance()

    if data[0] == 200:
        for index in range(len(data[1])):
            for key in data[1][index]:
                if data[1][index]['type'] == 'exchange':
                    if data[1][index]['currency'] == symbol:
                        return data[1][index]['available']

def getWalletBalance(self):
    nonce = self.nonce()
    path = "/v1/balances"

    body = {
        "request": path,
        "nonce": nonce
    }

    headers = self.headers(nonce, path, body)
    body = json.dumps(body)

    r = requests.post(self.base_url + path, headers=headers, data=body, verify=True)

    if r.status_code == 200:
        return r.status_code, r.json()
    else:
        return r.status_code, r.json()
```

Figura 15 - Exemplo do código do MyRobô que mostra a transferência de saldo carteira USD da Bitfinex para Binance

```
def withdraw(self, withdraw_type, amount, address, _nonce):
    nonce = self.nonce() + _nonce
    path = "/v1/withdraw"

    body = {
        "request": path,
        "withdraw_type": withdraw_type,
        "walletselected": "exchange",
        "amount": amount,
        "address": address,
        "nonce": nonce
    }

    headers = self.headers(nonce, path, body)

    body = json.dumps(body)

    r = requests.post(self.base_url + path, headers=headers, data=body, verify=True)

    if r.status_code == 200:
        return r.status_code, r.json()
    else:
        return r.status_code, r.json()
```

name: nome da moeda. Importante usar o nome especificado pela corretora. Cada corretora tem sua nomenclatura para símbolos e nomes das criptomoedas

amount: quantidade a ser transferida. Valor disponível na carteira USD.

Address: endereço de depósito da carteira USD Binance.

nonce: Em criptografia, um nonce é um número arbitrário que só pode ser usado uma vez. É basicamente uma palavra de uso único, daí o nome (N = Number (Número) e Once = Uma vez, em inglês), embora N seja de número, o processo também pode usar letras. Nas operações envolvendo as APIs, nonce é um número transmitido através do cabeçalho.

Uma vez que a transferência é realizada, recebo uma mensagem de sucesso ou falha.

Se uma mensagem de sucesso for recebida, preciso checar a carteira USD Binance, verificando se o valor já foi depositado. O robô entra em um loop, a cada 10 minutos checa novamente a carteira de destino. Esse processo costuma ser o mais lento, transferência entre carteiras de corretoras diferentes. Na primeira vez que for realizada uma transferência entre carteiras bitcoin, esse processo pode demorar até 24 horas. Conforme aumenta a quantidade de transferências a serem feitas entre as mesmas carteiras, reduz-se o tempo de espera para o valor chegar ao seu destino.

Após o dinheiro ser transferido para a binance, de fato cair na carteira USD da binance, o robô dá início ao processo de compra da bitcoin na corretora. É necessário verificar se o valor de venda da bitcoin se manteve o mesmo ou se ocorreu alteração, se ainda é viável realizar a compra e conseguir lucro.

Figura 16 - Exemplo do código do MyRobô que mostra a compra de bitcoins na Binance

```
def comprarMoedaBinance(self, symbol, precoMinimo):
    #consultando o saldo da carteira USD na Binance
    carteiraUsd = self.getCoinAvailableExchange("USD")

    #consultando o preço venda da criptomoeda bitcoin
    valorVenda = self.getLowestAsk(symbol)

    #precoMaximo: valorVenda verificado na corretora Binance antes do robô iniciar a operação
    while float(valorVenda) > float(precoMaximo):
        print("O preco maximo para compra da moeda ", symbol, "é ", precoMaximo)
        print("O preco atual é ", valorVenda, ". Aguardar o preco atual ficar inferior ou igual ao precoMaximo.")
        time.sleep(60)
        valorVenda = self.getHighestBid(symbol)

        if(valorVenda == "error"):
            continue

        while valorVenda == "error":
            print("Excesso de chamadas à api em um curto espaço de tempo. Esperando " + tempo + " segundos antes de consultar novamente o valor")
            tempo = tempo + 10
            time.sleep(tempo)
            valorVenda = self.getHighestBid(symbol)

    #Binance trabalhar com um limite de casas decimais, além disso, só aceita valores do tipo String. Precisa converter para string os campos float
    valorVenda = float(valorVenda)
    valorVenda = "%.8f" % valorVenda
    valorVenda = str(valorVenda)

    #quantia disponível na carteira USD, considerando um limite de casas decimais
    carteiraUsd = "%.7f" % carteiraUsd

    #determinando amount. Calculo do quanto é possível comprar em bitcoins considerando o valorVenda atual e o quanto tem disponível na carteira USD
    amount = float(carteiraUsd)/float(valorVenda)
    amount = "%.7f" % amount
    amount = str(amount)
    amount = amount[:7]

    #executando a ordem de compra
    resultadoCompra = self.order(symbol, amount, valorVenda,"buy")
```

Está comparação com o último valor de venda da bitcoin na binance, antes do robô iniciar a operação, e o valor atual, é realizada com uma nova consulta à api da Binance.

Se não existir diferença ou a mesma for irrelevante(ainda é possível obter lucro), o processo de compra da moeda é iniciado. Uma chamada à api binance é realizada informando:

```
resultadoCompra = self.order(symbol, amount, valorVenda,"buy")
```

Uma vez que a compra tenha sido realizada, uma mensagem de sucesso será retornada. E então o valor será depositado na carteira bitcoin Binance. O robô espera até que o novo valor seja inserido de fato na carteira e dá a operação como concluída.

6 CONSIDERAÇÕES FINAIS

Diante de tudo que foi apresentado, é importante ressaltar para o investidor a importância de um bom entendimento do mercado de criptomoedas antes de tomar a decisão de investir em robôs de trading. Evitando com isso, cair em armadilhas de empresas fraudulentas.

Blockchain é sem sombra de dúvidas a rede mais segura e mais utilizada até o momento mas existem alternativas que em longo prazo podem vir a ser bem mais atrativas, como a rede da moeda IOTA.

Ao construir um robô de trading de criptomoedas é importante manter registros de todas as operações na rede blockchain. Futuramente, em cima destes registros históricos podem ser aplicadas técnicas como *deep learning*. Deep Learning é um subconjunto de machine learning onde redes neurais artificiais inspiradas no cérebro humano, aprendem a partir de uma grande quantidade de dados.

Com isso, aumentam as chances de obtenção de lucro através da análise destes dados, como variações de taxas e preços de moedas. Possibilitando prever quando taxas de compra/venda para determinadas moedas envolvidas na transação são uma vantagem ou quando podem acarretar prejuízo.

REFERÊNCIAS

- BELIN, O. The difference between blockchain distributed ledger technology. 2019. Disponível em: <https://tradeix.com/distributed-ledger-technology/>. Acesso em: 02/03/2020.
- BERTOLUCCI, G. Diferença entre trading bots e humanos, confira. 2018. Disponível em: <https://livecoins.com.br/diferenca-entre-trading-bots-e-humanos-confira/>. Acesso em: 05/02/2020.
- BINANCE. *API key Binance*. 2020. Disponível em: <https://www.binance.com/>. Acesso em: 14/02/2020.
- BINANCE. *Cadastro Binance*. 2020. Disponível em: <https://www.binance.com/>. Acesso em: 14/02/2020.
- BINANCE. *Saldo na exchange Binance*. 2020. Disponível em: <https://www.binance.com/>. Acesso em: 14/02/2020.
- BITCONGRESS. Best crypto trading bots 2020 – automated crypto profits! 2020. Disponível em: <https://www.bitcongress.org/bitcoin/trading/best-bitcoin-trading-bots/>. Acesso em: 04/02/2020.
- BITFINEX. *API key Bitfinex*. 2020. Disponível em: <https://www.bitfinex.com/>. Acesso em: 14/02/2020.
- BITFINEX. *Cadastro Bitfinex*. 2020. Disponível em: <https://www.bitfinex.com/>. Acesso em: 14/02/2020.
- BITFINEX. *Saldo na exchange Bitfinex*. 2020. Disponível em: <https://www.bitfinex.com/>. Acesso em: 14/02/2020.
- BITPRECO. *Cadastro BitPreço*. 2020. Disponível em: <https://bitpreco.com/>. Acesso em: 14/02/2020.
- BITPRECO. *Compra de Bitcoins no site da BitPreco*. 2020. Disponível em: <https://bitpreco.com/>. Acesso em: 14/02/2020.
- CHANNEL, S. E. *Como um blockchain funciona - Explicação simples*. 2020. Disponível em: https://www.youtube.com/watch?v=SSo_EIwHSd4. Acesso em: 14/02/2020.
- DAVIES, A. How to build a crypto trading bot. 2020. Disponível em: <https://www.devteam.space/blog/how-to-build-a-crypto-trading-bot/#1>. Acesso em: 03/02/2020.
- HEDGETRADE. How to code a trading bot. 2019. Disponível em: <https://hedgetrade.com/how-to-code-a-trading-bot/>. Acesso em: 02/02/2020.
- LIEW, L. What is algorithmic trading and how do i learn it? 2020. Disponível em: <https://algotrading101.com/learn/what-is-algorithmic-trading/>. Acesso em: 02/02/2020.

PRADO, J. O que é blockchain? [indo além do bitcoin]. 2017. Disponível em: <https://tecnoblog.net/227293/como-funciona-blockchain-bitcoin/>. Acesso em: 02/02/2020.

ROSIC, A. What is blockchain technology? a step-by-step guide for beginners. 2016. Disponível em: <https://blockgeeks.com/guides/what-is-blockchain-technology/>. Acesso em: 01/02/2020.