



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Bacharelado em Engenharia
de Computação

AllBot: Plataforma de Robôs para Investimento no Mercado de Criptomoedas

Daniel Santos Silva Júnior
Guilherme Pereira da Silva
Ranzeus Naarson Oliveira Muniz

Orientador: Prof. Romeryto Vieira Lira, MSc.

Campina Grande, Setembro de 2021

©Daniel Santos Silva Júnior

©Guilherme Pereira da Silva

©Ranzeus Naarson Oliveira Muniz



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Bacharelado em Engenharia
de Computação

AllBot: Plataforma de Robôs para Investimento no Mercado de Criptomoedas

Daniel Santos Silva Júnior
Guilherme Pereira da Silva
Ranzeus Naarson Oliveira Muniz

Monografia apresentada à Coordenação do
Curso Superior de Bacharelado em Enge-
nharia de Computação do IFPB - Campus
Campina Grande, como requisito parcial
para conclusão do curso de Bacharelado em
Engenharia de Computação.

Orientador: Prof. Romeryto Vieira Lira, MSc.

Campina Grande, Setembro de 2021

S586a Silva Júnior, Daniel Santos

AllBot: plataforma de robôs para investimento no mercado de criptomoedas / Daniel Santos Silva Júnior, Guilherme Pereira da Silva, Ranzeus Naarson Oliveira Muniz. - Campina Grande, 2022.

80 f. : il.

Trabalho de Conclusão de Curso (Curso de Bacharelado em Engenharia de Computação) - Instituto Federal da Paraíba, 2022.

Orientador: Prof. Me. Romeryto Vieira Lira.

1. Robôs de investimentos 2. Criptomoedas 3. Aplicação web. I. Silva, Guilherme Pereira da. II. Muniz, Ranzeus Naarson Oliveira. III. Título.

CDU 004

AllBot: Plataforma de Robô para Investimento no Mercado de Criptomoedas

Daniel Santos Silva Júnior
Guilherme Pereira da Silva
Ranzeus Naarson Oliveira Muniz

Romeryto Vieira Lira
Orientador

Arley Willer Neves da Silva
Membro da Banca

Ewerthon Dyego de Araújo Batista
Membro da Banca

Campina Grande, Paraíba, Brasil
Setembro/2021

"A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo".
(Albert Einstein)

"Tenho a impressão de ter sido uma criança brincando à beira-mar, divertindo-me em descobrir uma pedrinha mais lisa ou uma concha mais bonita que as outras, enquanto o imenso oceano da verdade continua misterioso diante de meus olhos".
(Isaac Newton)

"Ciência é uma equação diferencial. Religião é a condição de contorno".
(Alan Turing)

Agradecimentos

Agradecimentos Daniel Júnior

Primeiramente gostaria de agradecer ao meu Senhor, Jesus Cristo, autor da minha fé, o qual com sua graça me sustenta todos os dias.

A minha mãe Rossiane, um ser encantador que não mediu esforços para me proporcionar uma educação de qualidade e que de uma forma brilhante me motiva a ser uma pessoa melhor. A meu pai Daniel, cujo nome carrego com bastante orgulho, que não está mais entre nós, mas que nunca saiu de minha memória. A minha irmã Dayane por está comigo em todos os momentos e por me encher de orgulho, a minha avó Lindalva Rocha que sempre foi a base da família, a meu avô Antônio que para mim é exemplo de sabedoria e honestidade, a minha avó Lindalva Santos sinônimo de serenidade, a minha bisavó Maria José que tem um carinho enorme por mim, a Eddward (Xetta) por toda a parceria e por ser como um irmão para mim, como também a meu amor Jamila Monteiro que é minha companheira de vida e sempre me ajudou a não desistir de meus sonhos.

Aos meus professores do IFPB, em especial ao professor Jerônimo Rocha, que sempre me ajudou nos projetos de pesquisa, me orientando e dando ensinamentos valiosos. Ao professor Romeryto que esteve nos orientando nesse trabalho e que foi de fundamental importância desde a concepção até o desenvolvimento, muito obrigado por acreditar em nós.

A meu amigo e parceiro de projeto, Ranzeus Muniz, que me acompanha desde o ensino médio e que acreditou sempre no meu potencial.

A Guilherme Pereira, meu parceiro de projeto que de uma maneira bastante motivada, foi fundamental para a realização deste trabalho.

A meu grande amigo, Leonardo Silva, que desempenhou um papel de extrema relevância ao longo desses anos de estudo. Você é luz meu irmão.

Quero agradecer também a Moabe Barbosa, um amigo que a vida me deu e que também foi essencial para a realização deste trabalho.

Ao IFPB por toda a infraestrutura e todos os esforços para proporcionar um ensino de qualidade para mim e para meus colegas de curso.

Agradecimentos Guilherme Pereira da Silva

Desenvolver este projeto acabou sendo uma das vivências mais importantes da minha vida. Não foi fácil, e com certeza eu não teria conseguido chegar até aqui sem ajuda. Agradeço primeiramente a Deus, que fez com que meus objetivos fossem alcançados durante todos os meus anos de estudos e sempre me guiou pelo caminho certo para que eu pudesse chegar a conclusão deste curso.

A meus pais, Eptácio e Eliete, por nunca terem medido esforços para me proporcionar um ensino de qualidade durante todo o meu período escolar, sempre confiando e me proporcionando o carinho e o afeto que nunca me deixaram desistir. Jamais serei capaz de retribuir todo o amor e incentivo que recebi de vocês.

A meu irmão, Elton, com quem tenho ótimas recordações a longo desses anos, que me incentivou, apoiou e impactou diretamente na minha formação acadêmica. (E que também está nesse momento de sua vida. Saiba que o apoio totalmente e acredito no seu sucesso).

A minha namorada, Emily, que sempre esteve ao meu lado, pela compreensão incondicional e pelo apoio demonstrado ao longo de todo o período em que me dediquei a este trabalho. Que nunca me negou apoio, carinho e incentivo. E além, me ajudou no desenvolvimento, que foi fundamental na elaboração deste trabalho.

Aos colegas que fizeram este projeto comigo, Ranzeus e Daniel, os quais sem eles eu não o teria realizado. Que sempre mantiveram bom humor e me influenciaram na construção do mesmo. Me ensinaram a prosseguir além das dificuldades, entenderam a dificuldade do momento em que passo e mantiveram sempre a amizade acima de tudo.

Ao meu amigo Alfredo Rodrigo, que vem me acompanhando em formação desde nossos primeiros anos de vida, e que foi meu parceiro de curso durante toda a formação.

Ao professor Romeryto, por ter sido meu orientador e ter desempenhado tal função com dedicação e amizade.

Aos professores Katyusco, Fagner e Francisco Dantas pelos ensinamentos em vários projetos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso.

A Lúcio, por todos os conselhos, ajuda e paciência com a qual guiaram o meu aprendizado.

A outros professores que ajudaram em meu desenvolvimento e ensinamentos que me permitiram apresentar um melhor desempenho no processo de formação profissional ao longo do curso.

A instituição de ensino IFPB, essencial no meu processo de formação profissional, pela dedicação, e por tudo o que aprendi ao longo dos anos do curso.

Agradecimentos Ranzeus Naarson Oliveira Muniz

Inicialmente gostaria de agradecer a Deus, que me manteve firme e perseverante para alcançar essa vitória. Agradecer por ter essa oportunidade de estar vivo e saudável após todas as tribulações enfrentadas neste período complicado que estamos vivendo.

A minha mãe Rosélia, um exemplo de vida para mim, que me deu os maiores ensinamentos sobre perdão, compaixão, sabedoria e gratidão. A meu pai Rinaldo, responsável por me ensinar sobre as duras verdades da vida, através de suas "doces" e "amargas" palavras, também me proporcionou os maiores ensinamentos sobre simplicidade e integridade. Juntos responsáveis por me tornarem a pessoa que sou hoje.

A todos os meus familiares que me apoiaram nessa caminhada e deram um grande suporte principalmente nos momentos difíceis envolvendo saúde. Em especial, minha tia Rosângela, pessoa guerreira, sincera e muito importante para mim, a qual sou muito grato por todo o esforço para me proporcionar a oportunidade de estudar em um ambiente escolar melhor durante meu ensino médio.

Agradeço minha linda companheira de vida e meu amor, Heloísa Diniz, por todo o apoio durante essa caminhada em busca dos meus objetivos, por toda cumplicidade, carinho e principalmente por compreender a minha ausência enquanto eu me dedicava à realização deste trabalho.

Aos meus professores do IFPB, em especial ao professor Fagner de Araujo, o qual foi meu orientador em um projeto de pesquisa que me proporcionou um grande aprendizado e crescimento pessoal. Ao Professor Romeryto, que nos orientou neste trabalho, com grande ajuda desde a ideia inicial até a finalização.

Ao meu grande amigo e parceiro de projeto, Daniel Santos, desde o ensino médio me ajudou muito com palavras e ações. Teve um grande papel desde a concepção deste trabalho até a concretização do mesmo.

A Guilherme Pereira, grande parceiro de projeto que no seu jeito de ser e atuar, foi fundamental para a realização deste trabalho.

Aos demais amigos e companheiros de curso que nos acompanharam e ajudaram durante a jornada. Com destaque para Leonardo Silva, Moabe Barbosa muito importantes para a realização deste trabalho.

Resumo

O grande avanço da tecnologia, principalmente quanto a acessibilidade das pessoas ao mercado de investimento, possibilitou um maior contato das mesmas com ambientes como o mercado de criptoativos, esse que, por sua vez, comporta outro mercado: o de criptomoedas, o qual vem crescendo de maneira expressiva desde meados de 2017 até os dias de hoje. Este mercado é dinâmico, uma vez que dispõe de diversas criptomoedas, com várias tecnologias e propostas. Nesse cenário, o uso de robôs de investimento se mostra cada vez mais necessário, já que estes proporcionam diversos benefícios, como: a redução do fator emocional nas negociações, maior velocidade de execução de ordens de compra e venda, independência de atuação humana constante, além da possibilidade de realização de simulações em dados históricos. Nesse contexto, foi desenvolvido uma plataforma de robôs para investimento em criptomoedas, AllBot, a qual oferece a possibilidade de gerenciar robôs em um ambiente web, configurá-los dentro das características de estratégias de investimento baseadas na análise técnica, testá-los para verificar seu desempenho, dispor de uma classificação dos melhores robôs disponíveis na plataforma para ajudar na avaliação, e que conta ainda com um módulo de notificações integrado ao Telegram.

Palavras-chave: Robôs de Investimento, Aplicação Web, Criptomoedas.

Abstract

The great advance in technology, especially in terms of people's accessibility to the investment market, has made it possible for them to have greater contact with environments such as the cryptoactive market, which, in turn, includes another market: that of cryptocurrencies, which has been growing significantly from mid-2017 to the present day. This market is dynamic, as it has several cryptocurrencies, with various technologies and proposals. In this scenario, the use of investment robots is increasingly necessary, as they provide several benefits, such as: a reduction in the emotional factor in negotiations, greater speed of execution of buy and sell orders, independence from constant human action, in addition to the possibility of carrying out simulations on historical data. In this context, a robot platform for investment in cryptocurrencies, AllBot, was developed, which offers the possibility of managing robots in a web environment, configuring them within the characteristics of investment strategies based on technical analysis, testing them to verify their performance, have a ranking of the robots available on the platform to help with the evaluation, and which also has a notification module integrated to the Telegram.

Keywords: *Trading Bots, WEB Application, Cryptocurrency.*

Sumário

Lista de Abreviaturas	xii
Lista de Figuras	xiii
1 Introdução	1
1.1 Justificativa e Relevância do Trabalho	3
1.2 Objetivos	4
1.2.1 Objetivo Geral	4
1.2.2 Objetivos Específicos	5
1.3 Metodologia	5
1.4 Organização do Documento	6
2 Fundamentação Teórica	7
2.1 Mercado de Criptomoedas	7
2.1.1 Carteira de Criptomoedas	8
2.1.2 Exchanges de Criptomoedas	8
2.1.3 Livro de Ofertas	9
2.2 Robôs de Investimento	10
2.2.1 Backtest	11
2.2.2 Estrutura de um Robô de Investimento	12
2.3 Análise Técnica	12
2.3.1 Tipos de Gráficos	13
2.3.2 Indicadores	16
2.3.3 \mathcal{R} Bandas de Bollinger	17
2.3.4 Índice de Força Relativa	18
3 Desenvolvimento	19
3.1 Sistema Proposto	19
3.2 Serviço Robô	20
3.2.1 Módulo de Gerenciamento de Requisições, Respostas e Dados	20
3.2.2 Módulo de Execução dos Robôs	22
3.2.3 Módulo de Execução de Backtests	26

3.3	Estratégias	28
3.3.1	Mean Reversion	28
3.3.2	Cross Average	29
3.3.3	IFR2	30
3.4	Otimização de Estratégias	31
3.5	Aplicação Web	32
3.5.1	Módulo de Lógica de Negócios da Aplicação Web	32
3.5.2	Módulo de Interface com Usuário	38
4	Resultados	41
4.1	Resultados do Serviço Robô	41
4.2	Resultados do Serviço Web	44
5	Considerações Finais e Sugestões para Trabalhos Futuros	62
5.1	Considerações Finais	62
5.2	Sugestões para Trabalhos Futuros	63
	Referências Bibliográficas	64

Lista de Abreviaturas

API	<i>Application Programming Interface</i> (Interface de Programação de Aplicação)
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
HTTPS	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto Seguro)
DTO	<i>Data Transfer Object</i> (Objeto de Transferência de Dados)
CRUD	<i>Create Read Update Delete</i> (Criar Ler Atualizar Deletar)
MM	Média Móvel
MMA	Média Móvel Aritmética
MMS	Média Móvel Simples
MME	Média Móvel Exponencial
IFR	Índice de Força relativa
BB	ⓇBandas de Bollinger

Lista de Figuras

1.1	Capitalização de mercado das criptomoedas. (Gerado com o tradingview).	1
2.1	Funcionamento modelo Ponto a Ponto	8
2.2	Funcionamento modelo <i>exchange</i>	9
2.3	Livro de Ofertas da plataforma Vector	10
2.4	Estrutura de um Robô	12
2.5	Gráfico de Linhas Bitcoin (Gerado com o tradingview).	14
2.6	Gráfico de Barras Bitcoin (Gerado com o tradingview).	14
2.7	Estrutura de um Candlestick	15
2.8	Gráfico de Candlestick do Bitcoin (Gerado com o tradingview).	16
3.1	Estrutura do Sistema Proposto	19
3.2	Visão geral do Serviço Robô	20
3.3	Estrutura do Worker de Execução de Robôs	22
3.4	Diagrama de classe Robot	23
3.5	Estrutura de um stream Kline/Candlestick	25
3.6	Resultados de um backtest	28
3.7	Gráfico de bitcoin, com o indicador bandas de bollinger (Gerado com o tradingview).	29
3.8	Gráfico de Bitcoin, com o indicador médias móveis (Gerado com o tradingview).	30
3.9	Gráfico de bitcoin, com o indicador médias móveis e IFR (Gerado com o tradingview).	31
3.10	Diagrama de atividades de um robô otimizado.	31
3.11	Diagrama de classe genérico para subdivisões do Módulo de Lógica de Negócios da Aplicação Web	33
3.12	Diagrama de classe do Módulo Login	34
3.13	Diagrama de classes do Módulo Usuário	35
3.14	Diagrama de classes do Módulo Robô	36
3.15	Diagrama de classes do Módulo Backtest	37
3.16	Diagrama de classes do Módulo Envio de Email	37
3.17	Informações de ranking robô	40

4.1	Resultado acumulado do Robô Mean Reversion com e sem otimização. . . .	42
4.2	Resultado acumulado do Robô Mean Reversion com e sem otimização. (Resultado acumulado por tempo de operação)	43
4.3	Tela de Login.	44
4.4	Tela de Cadastro de Usuário.	44
4.5	Email Enviado ao Usuário.	45
4.6	Tela Confirmação de Email.	45
4.7	Tela de Recuperação de Senha.	46
4.8	Email de Redefinição de senha.	46
4.9	Tela Redefinição de Senha.	47
4.10	Tela do Painel de Controle (parte 1).	48
4.11	Tela do Painel de Controle (parte 2).	48
4.12	Tela do Painel de Controle Versão Mobile.	49
4.13	Tela do Perfil do Usuário.	50
4.14	Tela de Configuração do Usuário.	51
4.15	Tela de Configuração do Usuário Versão Mobile.	52
4.16	Tela de Relatórios (parte 1).	53
4.17	Tela de Relatórios (parte 2).	53
4.18	Tela de Relatórios Versão Mobile.	54
4.19	Tela dos Robôs do Usuário.	55
4.20	Tela dos Robôs do Usuário Versão Mobile.	56
4.21	Tela de Criação de Robôs.	57
4.22	Tela de Edição de Robô.	57
4.23	Ranking de Robôs.	58
4.24	Ranking de Robôs expandido.	58
4.25	Tela de Relatórios Versão Mobile.	59
4.26	Tela de Backtest de um Único Robô.	60
4.27	Captura de tela do aplicativo telegram.	61

Lista de Quadros

1	Conceitos do Mercado Financeiro	13
2	Comparativo do Robô Mean Reversion na Ripple com e sem otimização. . .	42
3	Comparativo do Robô Mean Reversion na Ethereum com e sem otimização. .	43

Capítulo 1

Introdução

Criptomoedas são moedas digitais funcionam como um meio de troca e que pode ser ou não centralizado, isto é, possuir uma unidade responsável por sua emissão ou controle. Para garantir a validade das transações se é utilizado a tecnologia de *blockchain* (cadeia de blocos) e criptografias. O mercado de criptomoedas vem ganhando cada vez mais força em todas as partes do mundo. A partir da publicação de maior relevância na área, o "*Bitcoin: A Peer-to-Peer Electronic Cash System*" NAKAMOTO (2008), foram estabelecidos os princípios de funcionamento do *Bitcoin* através do *blockchain*, onde de uma forma bastante inovadora, uma série de conceitos relacionados à criptografia e à economia monetária são relacionados. Desde então, diversas criptomoedas foram criadas, com diversas tecnologias e propostas. Na figura 1.1 é possível observar o gráfico de capitalização de mercado das criptomoedas (o valor total em dólares de todas as criptomoedas mineradas), desde 2019.

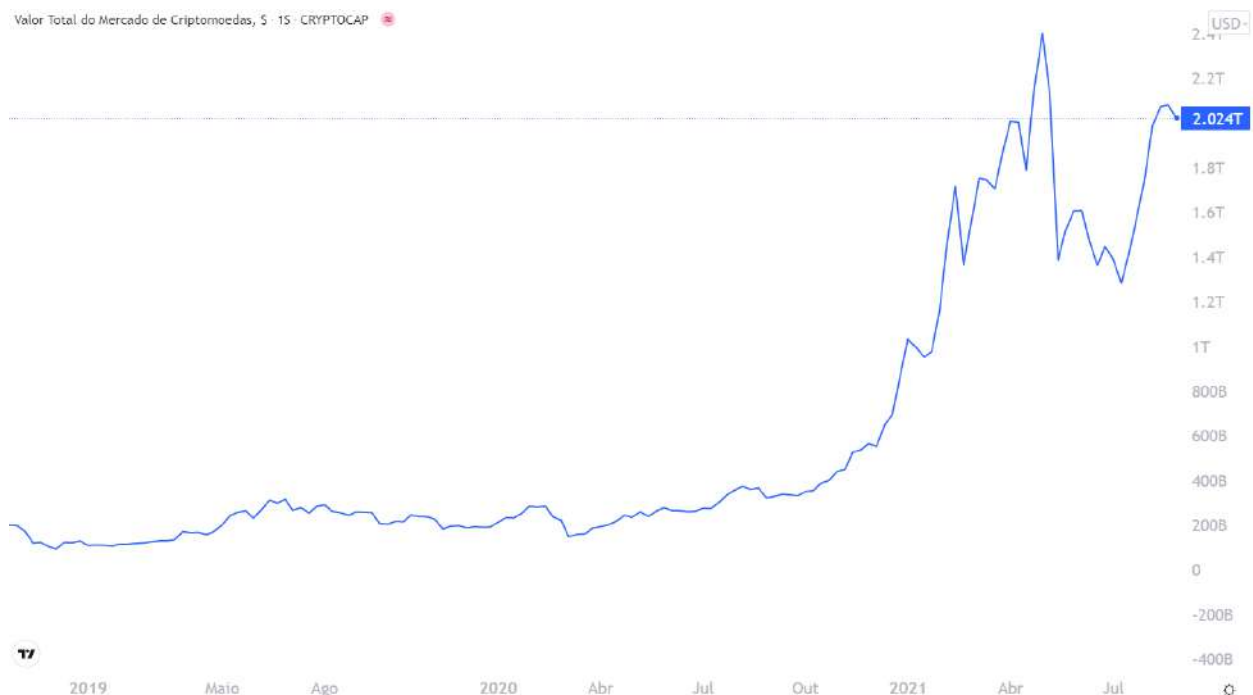


Figura 1.1: Capitalização de mercado das criptomoedas. (Gerado com o tradingview).

Liu and Tsyvinski (2021) dizem que o mercado de criptomoedas caracteriza um investimento especulativo, pois, nesse mercado, o investidor depende exclusivamente do equilíbrio da oferta entre compradores e vendedores. Dessa forma, não existe então nenhum tipo de retorno previsto, dividendo, lucro, ou qualquer outro recurso que assegure determinado patamar de cotação.

De modo geral, o investidor de criptomoedas busca um retorno bem acima do que é oferecido no mercado de ações, renda fixa, imóveis e ouro. Isso porque o potencial das criptomoedas é sub-explorado, tanto em adoção, ou seja, poucos as adotam, quanto de funcionalidades, em outras palavras, nem todas as possíveis funcionalidades das mesmas são exploradas. Visto que elas possuem diversas utilidades, desde meio de pagamentos à reserva de valor em momentos de alta volatilidade do mercado, já que em geral, elas não estão correlacionadas ao mercado tradicional.

Existem duas principais formas de obtenção de criptomoedas: a mineração e o câmbio de moedas. A mineração é o processo de adicionar registros de transações de uma moeda virtual, completamente digital, à *blockchain* (livro onde são registradas as transações) e obter o valor em criptomoeda resultante, chamado criptoativo. Muitas pessoas se aventuram gastando fortunas em tecnologia para minerar moedas digitais, em especial o Bitcoin. Conforme Silva et al. (2021) nesse contexto, pode-se identificar várias dificuldades para a realização dessa atividade. Uma delas, é a constante busca por formas de melhorar o rendimento de maneira mais inteligente, possibilitando otimizar o lucro a partir do monitoramento de variáveis importantes, como por exemplo: A taxa de processamento por consumo de energia, valores de tarifação de energia (dependendo de horários dos dias e semana) e turnos mais vantajosos para mineração, além de maneiras de economizar energia sem diminuir a taxa de processamento. Tendo em vista tais dificuldades, o câmbio se tornou a forma mais bem vista para se obter lucro com tais moedas, baseando-se na valorização das mesmas.

De acordo com NUNES et al. (2021) ao se tratar de um mercado especulativo e bastante competitivo, a todo momento surgem novas estratégias e técnicas de negociação. Nesse cenário, o mercado financeiro, em geral, têm sido alvo de grandes mudanças. Devido ao rápido desenvolvimento computacional, as transações têm sido cada vez mais automatizadas através de plataformas eletrônicas de negociação. Diante disso, os robôs de investimento surgem como uma excelente forma de facilitar o trabalho humano na tomada de decisões mais lógicas e na procura dos melhores investimentos com base na análise do mercado, calculando a probabilidade de rentabilidade em uma negociação.

O uso de estratégias automatizadas de investimento são cada vez mais necessárias, pois proporcionam diversos benefícios, como: minimizar as emoções durante o processo de negociação em um mercado agitado, maior velocidade de execução, independência de atuação humana constante, além de possibilitar a realização de testes em dados históricos.

Existem muitos robôs de investimento em criptomoedas simples e disponíveis para

compra, assinatura ou obtenção de forma gratuita. Possuem uma simples proposta de valor, oferecendo aos usuários a capacidade de automatizar suas negociações e esperar que o robô obtenha lucros. No entanto, nem sempre é esse o caso. Enquanto robôs básicos podem ser conectados à conta de negociação de um usuário podendo executar ordens de compra e venda automaticamente, os usuários têm controle limitado sobre a estratégia do mesmo, tornando-os menos flexíveis às variações de mercado (TRADINGSTRATEGYGUIDES, 2019).

Levando em conta este contexto, neste trabalho, apresenta-se uma plataforma de robôs para investimento em criptomoedas, que dispõe de estratégias de negociação amplamente consolidadas do mercado, oferecendo aos usuários um conjunto de funcionalidades relacionadas ao gerenciamento e análise, proporcionando, uma total transparência do real desempenho das estratégias de negociação.

1.1 Justificativa e Relevância do Trabalho

Os robôs de investimento proporcionam diversos benefícios para o usuário. Uma grande vantagem que podemos destacar é a economia de tempo. O investidor não precisa ficar acompanhando o mercado o dia inteiro, podendo focar em outras atividades, sem arriscar perder oportunidades no mercado. Outro benefício seria a facilidade de atuação, se despreendendo da necessidade de análise gráfica e aplicar técnicas a cada operação no mercado, diminuindo as hipóteses de erros de cálculos. E como conclui Lichtman (2019)

"indivíduos com pouco conhecimento ou experiência em mercados financeiros tendem a se beneficiar da automação de investimentos, que proporcionam a confiança necessária para que o investidor assuma riscos dentro do seu perfil e alinhados aos seus objetivos financeiros".

Para Raimundo (2018) o uso de algoritmos de negociação proporciona diversos benefícios tais como: maior velocidade na entrada de ordens, controle de falhas mecânicas e capacidade de *backtest*. Sendo essa uma característica fundamental a se destacar. Como o robô é um *software*, as chances de perder dinheiro devido a atrasos na execução ou esquecimentos são praticamente nulas, principalmente se ele estiver hospedado em um servidor de qualidade.

Outro benefício proporcionado pelos robôs é o fato deles pouparem o investidor de tomar decisões, cujo emocional pode influenciar. Uma vez configurado, o robô não considera fatores emocionais nas decisões, possuem regras claras de quando e como agir, proporcionando em alguns casos melhores resultados quando comparados às operações realizadas por humanos que seguem a mesma estratégia.

Diante disso, muitos trabalhos acadêmicos relacionados à temática do mercado de criptomoedas buscam validar estratégias de negociação, explorando a eficácia e discutindo a viabilidade de uma aplicação financeira seguindo essas estratégias.

Em Garcia (2021) é implementado um mecanismo de inteligência computacional baseados em séries temporais para predição do comportamento das criptomoedas. Nesse trabalho, foram identificadas prováveis tendências e comportamentos que a série de preços do Bitcoin tomou em períodos específicos de tempo, porém, não foi implementado um protótipo automatizado da estratégia.

Já Silva and Vieira (2021) apresentam um sistema de automatização de operações com criptomoedas. Nesse trabalho, é desenvolvido uma estrutura para envio de ordens às *exchanges* de moedas digitais, fornecendo ao usuário uma visualização dos estados de envio das ordens, além de levantar algumas informações e melhorias que podem auxiliar no desenvolvimento de sistemas automatizados. Embora tenha sido realizada a integração com as corretoras de criptomoedas para a realização das negociações, o usuário não consegue modificar as estratégias, assim como solicitar a execução de testes em dados históricos.

O trabalho de Lichtman (2019) mostra um algoritmo de automação baseado na metodologia de média-variância tratado no mesmo, obtendo resultados positivos, no entanto, não é disponibilizada uma plataforma para que o usuário possa realizar testes e implementação, para isso, é necessário criar um robô baseado no algoritmo estudado no trabalho.

No artigo de Duarte et al. (2019) foi desenvolvido um algoritmo que usa estatística, aprendizado de máquina e redes neurais para a otimização dos lucros obtidos no mercado financeiro, através de um algoritmo que retorna os resultados via web. Entretanto, também não possui plataforma para simulação em dados históricos, ou até implementação do mesmo.

Sendo assim, justifica-se a criação de uma plataforma de robôs para investimento em criptomoedas, que conta com uma robusta arquitetura de execução de ordens em tempo real, fornecendo ao usuário uma interação com diversas estratégias de negociação, tendo diversos parâmetros personalizáveis, que também dispõe de recursos avançados de análise, através de um módulo de *backtests* (um teste de estratégia, onde é analisado o histórico de cotações de um ativo, para ver como ela teria se comportado no passado) em dados históricos, garantindo uma total transparência do desempenho das estratégias de negociação.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo deste trabalho consiste no desenvolvimento de um protótipo de uma plataforma de robôs para investimento que atue no mercado de criptomoedas, proporcionando uma fácil usabilidade para pessoas com menos conhecimento na área. Tal plataforma deve gerenciar a execução de robôs em tempo real, gerir um módulo de notificação de transações e informações aos usuários através de plataformas de comunicação como o Telegram, assim como permitir a realização de *backtests*.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos são considerados:

- Criar um sistema web onde o usuário possa se cadastrar e adquirir uma licença de teste, para ter acesso à plataforma de robôs para investimento;
- Desenvolver uma aplicação *frontend* com interface intuitiva para onde o usuário possa configurar seus robôs e utilizar seus recursos;
- Implementar a integração com a *exchange* Binance para ter acesso aos dados de mercado e envio das ordens de negociação;
- Desenvolver robôs que utilizem de estratégias conhecidas da análise técnica;
- Implementar o serviço que gerencia a execução dos robôs em tempo real;
- Integrar a plataforma a API do Telegram para notificar os usuários com as ações tomadas pelos robôs;
- Criar uma ampla base histórica de dados de mercado, e implementar o módulo de *backtest*, onde será possível testar robôs em dados históricos;
- Gerar relatórios com resultados de *backtests* de todo o portfólio de robôs criando um ranking;
- Implementar um modelo de otimização de estratégia buscando obter melhores resultados.

1.3 Metodologia

A metodologia de desenvolvimento foi construída com finalidade aplicada, onde o objetivo é o desenvolvimento de uma plataforma que possibilite a execução e testes de estratégias para operações em mercado de criptomoedas, com o intuito de obter lucro na variação de preço das criptomoedas. As atividades deste trabalho são descritas a seguir.

- Construção de um referencial teórico com trabalhos que empregam sistemas de robôs para investimento; As fontes de referências abrangem artigos, teses, monografias, livros, etc;
- Desenvolvimento de uma aplicação web para cadastro de usuários e gerenciamento de robôs para investimentos em criptomoedas;
- Pesquisa e implementação das estratégias de negociação já consolidadas da análise técnica, escolhendo estratégias populares e de fácil compreensão;

- Realização de integração com a *exchange* para obtenção de dados e envio de ordens, a partir dos robôs, em tempo real;
- Implementação do módulo de notificação integrado ao Telegram, de modo a proporcionar o acompanhamento das ações dos robôs;
- Implementação de um módulo de *backtest* em dados históricos para análise e otimização de estratégias.

1.4 Organização do Documento

Neste Capítulo 1 apresentamos uma introdução do que se trata o sistema junto com sua justificativa para a necessidade do desenvolvimento do mesmo, foi definido os objetivos na qual esse sistema foi alinhado e a metodologia na qual se seguirá em seu desenvolvimento. Já no Capítulo 2 demonstramos a fundamentação teórica, onde abordaremos conceitos relacionados ao mercado de criptomoedas, assim como análise técnica e os robôs de investimento. No Capítulo 3 serão exibidos as ferramentas utilizadas, a arquitetura proposta e como se deu o desenvolvimento. A seguir, no Capítulo 4 serão expostas evidências do protótipo implementado, assim como resultados de *backtests* realizados. No Capítulo 5, serão demonstrados as considerações finais e sugestões para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, são introduzidos alguns conceitos básicos que são necessários para o entendimento e desenvolvimento do projeto. Nas próximas seções, são feitas contextualizações sobre o mercado de criptomoedas, análise técnica, e como os robôs de investimento se inserem nesse contexto.

2.1 Mercado de Criptomoedas

"As criptomoedas, assim denominadas, são resultantes de uma viabilização baseada em métodos criptográficos — um conjunto de técnicas que permitem proteger dados transmitidos e armazenados —, a partir da transformação de informações legíveis em códigos ininteligíveis, conseqüentemente, pertencentes ao mundo digital, gerando uma moeda desregulamentada[...]"(CARVALHO et al., 2017).

A maioria das moedas digitais tem como sua principal característica não possuir uma autoridade governamental para reger, fazendo com que seu valor seja totalmente definido por sua especulação e seguindo esse propósito tornando transações comerciais mais fáceis, preservação de valor para valorização da mesma e até como finalidade de unidade de conta.

Uma moeda que possui finalidade de unidade de conta é quando os produtos são precificados ou até mesmo o cálculo econômico de dada região é definido baseado na moeda em questão, essa característica as criptomoedas ainda não possui aplicação prática, pois segundo Ulrich (ULRICH, 2017) a volatilidade de seus preços estão bastante elevados, ocasionando a impossibilidade de haver esta implementação.

"O mercado de criptomoedas tem grande impacto na economia, além de fomentar vários ramos da ciência como a matemática, direito, economia, computação e em especial “big data”, criptografia e desenvolvimento de hardware."(SICHEL; CALIXTO, 2018). Portanto, não pode ser visto como "uma tecnologia passageira", tendo em vista que o crescimento gigantesco do Bitcoin (primeira moeda digital) quando foi criada em 2008 possuía um valor por volta de US\$0,08, hoje, possui preços por volta de US\$48.780, com um volume de

movimentação diária por volta de US\$32.059.380. (COINMARKETCAP, 2021). O grande crescimento do valor da moeda trouxe visualização não somente para o Bitcoin como para várias outras criptomoedas isso ocorreu também devido à mídia social que ajudou a popularizar a mesma, fazendo com que fosse criada várias *exchanges* para a negociação da mesma.

2.1.1 Carteira de Criptomoedas

Assim como é necessária uma carteira para guardar o dinheiro físico, com as criptomoedas não é diferente. As carteiras virtuais são *softwares* que além guardar as criptomoedas, permitem enviar, receber e verificar um saldo ali dentro.

Existem diferentes tipos de carteiras para atender diferentes interesses pessoais dos usuários. Em geral, são divididas em duas categorias: as carteiras quentes que são as mais comuns no universo das criptomoedas, são conectadas à internet, acessadas geralmente através de aplicativos *mobile* ou navegadores (*web*). São muito usadas por geralmente serem mais baratas. Já as frias são as que não podem ser acessadas pela internet. Nesse caso, as criptomoedas ficam guardadas de forma segura em dispositivos físicos, sejam no formato de *pen-drive* ou até mesmo papel. (Wuzu, 2021)

2.1.2 Exchanges de Criptomoedas

Assim como temos as corretoras de bolsa de valores no mercado tradicional de investimento, também temos as *exchanges* para o mercado de criptomoedas. As *exchanges* tem como objetivo permitir o câmbio entre as moedas, sejam elas moedas digitais como o Bitcoin ou moedas fiduciárias a exemplo do dólar. Alguns exemplos são a Binance ou a Poloniex. Ao criar uma conta na *exchange*, é gerado uma conta na mesma onde o usuário pode fazer depósitos para realizar as operações.

Como é possível ver na figura 2.1 e na figura 2.2 existem duas formas de se obter moedas virtuais: no modelo Ponto a Ponto é feita uma transferência para o comprador por uma carteira virtual e o comprador paga o preço cobrado pelo vendedor, sem a necessidade de terceiros.

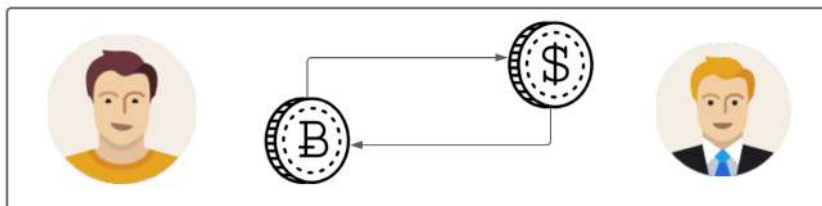


Figura 2.1: Funcionamento modelo Ponto a Ponto

No modelo *exchange* uma plataforma conecta usuários interessados em comprar, vender ou trocar moedas, assim, o cliente tem a possibilidade de criar uma ordem de compra ou venda, então, a *exchange* faz o casamento dessas ordens encontrando o comprador com o vendedor e realizando a operação, semelhante ao que ocorre numa feira, além de realizar operações com mais segurança. (BitcoinTrade, 2021)

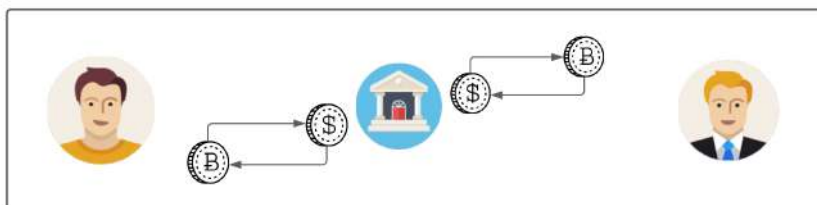


Figura 2.2: *Funcionamento modelo exchange*

2.1.3 Livro de Ofertas

O livro de ofertas é onde se encontram as intenções de compra e venda das moedas negociados na *exchange*, como se fosse um balcão ou uma vitrine.

Todas as ofertas de compra de alguma moeda são listadas em ordem decrescente conforme o valor. Desse modo, no topo do livro estão as ordens de compra de maior valor.

Já as ofertas de venda, são listadas em ordem crescente conforme o valor, assim as ofertas de menor valor estão especificadas no topo da lista.

Quando existem ofertas de compra de mesmo valor, a ordenação considera a ordem cronológica mostrando da mais antiga para a mais atual. O mesmo ocorre quando existem ofertas de venda de mesmo valor. Na figura 2.3 temos o livro de ofertas do Bitcoin.

Qtde	Compra	Venda	Qtde
0,6974	47.118	47.119	1,3673
0,0014	47.117	47.120	0,0497
0,0300	47.114	47.121	0,0345
0,2123	47.112	47.125	0,5306
0,0024	47.110	47.126	0,1438
1,0614	47.108	47.129	1,0613
1,6953	47.106	47.130	0,0050
0,1947	47.105	47.131	0,0575
1,3716	47.104	47.132	1,6964
0,0032	47.102	47.139	0,0024
0,0020	47.101	47.142	0,1999
0,0260	47.100	47.144	0,0195
0,0027	47.099	47.145	0,0505
0,0005	47.098	47.146	0,3312
0,2486	47.096	47.147	0,0016
0,0005	47.095	47.150	1,2000
0,0001	47.093	47.153	0,9043
2,3404	47.092	47.155	0,2000
0,5679	47.091	47.157	0,7758
0,1082	47.090	47.158	0,4288
0,1847	47.087	47.159	0,0024
0,0030	47.084	47.160	2,9966
0,7499	47.083	47.161	0,2012
1,8000	47.082	47.162	0,0010
0,0455	47.081	47.163	0,0005
2,0900	47.080	47.165	0,0004
1,1457	47.078	47.166	1,2737
0,1084	47.077	47.167	0,0003

Figura 2.3: Livro de Ofertas da plataforma Vector

Os dois principais tipos de ordens são as de mercado e as limitadas. A limitada é a que vai para o livro de ofertas, onde o usuário define o preço que quer comprar ou vender a criptomoeda, e assim que esse preço chegar ao topo do livro e tiver uma contraparte com quantidade suficiente, será executada. Já a de mercado é executada sem ser definido um preço, o usuário apenas determina uma quantidade e aceita o preço atual do mercado.

2.2 Robôs de Investimento

Antes de falarmos sobre robôs de investimentos, podemos falar um pouco sobre o que são os *softwares* robôs, ou "bots" como comumente são chamados na literatura. Os *softwares* robôs um novo paradigma de interface; que conectam usuários a serviços de *software*. Embora os usuários de *bots* geralmente sejam humanos, eles não precisam ser: podem ser programas, sistemas ou mesmo outros *bots*. Um *bot* é a interface que fornece os serviços ao usuário, ou seja, é tudo o que é necessário para apresentar o serviço ao usuário (LEBEUF et al., 2019).

Bots são frequentemente usados em grupo ou configurações colaborativas para automatizar tarefas que normalmente requerem interação humana (LEBEUF; STOREY; ZAGALSKY, 2017).

"Os Robôs de investimento são *softwares*, programas ou sistemas, voltados para o mercado financeiro, consistem em algoritmos que realizam as análises no mercado e realizam negociações"(Rodrigues, 2019). Esse tipo de negociação usa plataformas eletrônicas que registram ordens de compra e venda através de um algoritmo que executa instruções de negociação pré-programadas pelos investidores, os quais, possuem critérios muito bem definidos, de quando deve ser realizada a entrada e a saída de uma negociação.

De maneira a facilitar o entendimento podemos pensar na classificação dos robôs sendo feita semelhantemente à classificação de técnicas de teste, quando divididas em 3 grupos, sendo eles: *WhiteBox*, *BlackBox* e *GrayBox*. Em que *WhiteBox* é a investigação detalhada da lógica interna e da estrutura do código. No teste *WhiteBox*, é essencial que o testador tenha total conhecimento do código-fonte. O teste *BlackBox* é uma técnica de teste sem nenhum conhecimento do funcionamento interno do aplicativo. Já o teste *GrayBox*, é uma técnica para testar o aplicativo com conhecimento limitado do funcionamento interno do aplicativo e também possui o conhecimento dos aspectos fundamentais do sistema. Ele apenas examina os aspectos fundamentais do sistema e tem pouca ou nenhuma relevância com a estrutura lógica interna do sistema. A semelhança entre as técnicas de teste e a classificação de robôs se dá pelos nomes usados e quando trocamos as questões de conhecimento do interior do que esta sendo testado pela possibilidade de alteração das características do robô (KHAN; KHAN et al., 2012).

Assim os robôs classificados como *WhiteBox*, naturalmente permitem que sejam alterados parâmetros de configurações em todos as partes possíveis da estratégia de negociação.

Já os robôs *BlackBox*, são robôs totalmente fechados, onde não é possível que o usuário realize alterações das configurações, embora seja indicado para pessoas sem experiência, não proporcionam uma transparência em seu funcionamento.

Por fim, os robôs *GrayBox*, que já possuem uma certa limitação, sendo estes capazes de alteração apenas alguns parâmetros.

Para este trabalho, com o objetivo de proporcionar uma maior facilidade para o usuário, porém, permitindo que ele contribua na personalização da estratégia de negociação, optamos por usar os robôs classificados como *GrayBox*.

2.2.1 Backtest

Para toda ferramenta com atuação no mercado se faz necessário o uso de testes para garantir seu total funcionamento, de modo a fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar. Uma das formas mais utilizadas de avaliar um robô, é através da análise de métricas como a assertividade, a porcentagem de retorno ou até mesmo tempo de operação, através dos chamados *backtests*.

Conforme Pardo (2011), o *backtest* se caracteriza pelo teste de um modelo preditivo usando dados históricos. No mercado financeiro, o *backtesting* se refere ao teste de uma estratégia de investimento utilizando dados passados do mercado, seu objetivo é criar uma

2.3.1 Tipos de Gráficos

O modo mais simples de entender a evolução dos preços de uma ação ou cotação de uma criptomoeda durante um período é através de gráficos. Eles são as principais ferramentas utilizadas pelos analistas técnicos. A seguir no quadro 1 serão apresentados alguns conceitos necessários para o entendimento dos principais tipos de gráficos.

Quadro 1: *Conceitos do Mercado Financeiro*

<i>Timeframe</i>	É o intervalo de tempo usado para agrupar as cotações em períodos ao formar os elementos do gráfico de barras ou <i>candlesticks</i> .
Preço de Abertura	É o preço de abertura de um período.
Preço de Fechamento	É o preço de fechamento de um período.
Preço de Máxima	É o maior preço atingido dentro de um período.
Preço de Mínima	É o menor preço atingido dentro de um período.

Gráfico de linhas

Nesse tipo de gráfico é possível perceber uma das mais simples representações dos preços, consiste basicamente na união de pontos que representam os preços de fechamento de cada período.

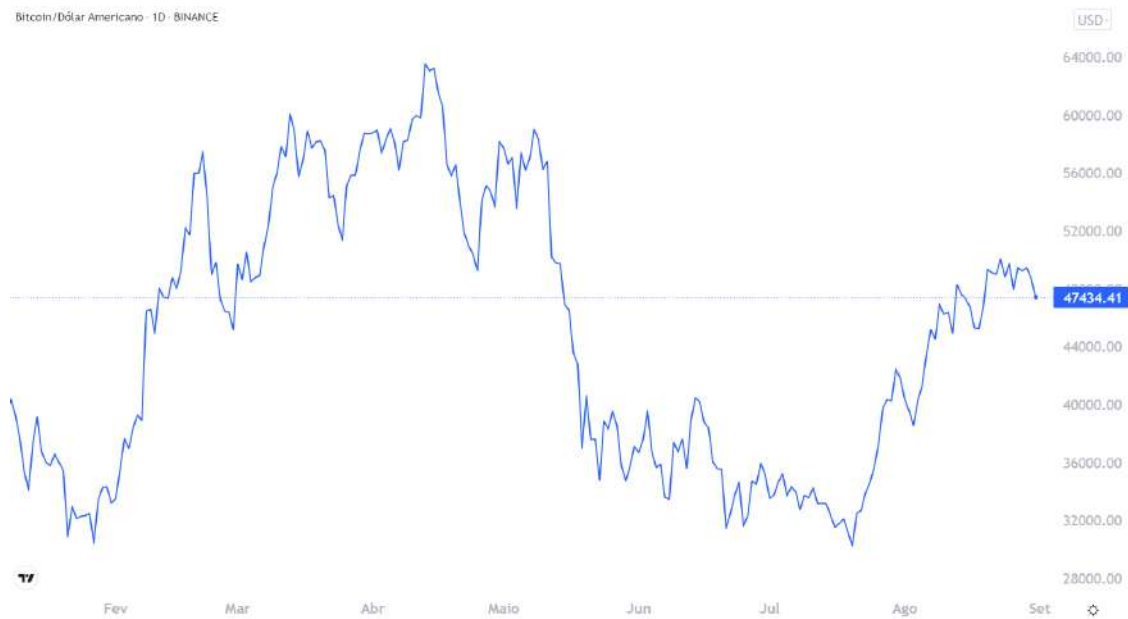


Figura 2.5: Gráfico de Linhas Bitcoin (Gerado com o tradingview).

Gráfico de Barras

Esse tipo de gráfico carrega basicamente quatro informações relacionadas a cada período, são elas: a abertura, o fechamento, a máxima e a mínima.

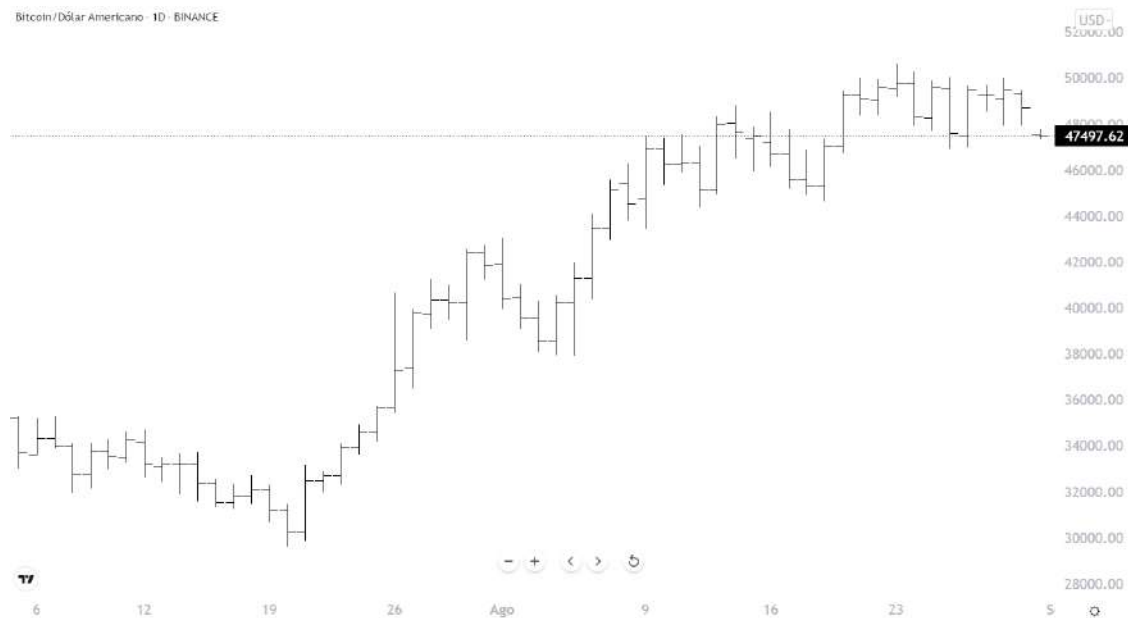


Figura 2.6: Gráfico de Barras Bitcoin (Gerado com o tradingview).

Gráfico de Candlestick

Antes de falarmos sobre o gráfico de *candlestick* é essencial demonstrar a estrutura de um *candlestick*. Um *candlestick* assim como no gráfico de barras, trás as informações dos preços de abertura, de fechamento, a máxima e a mínima de um período. Como ilustrado na Figura 2.7, o *candlestick* de alta geralmente possui um corpo em verde e representa uma variação positiva, enquanto o *candlestick* de baixa usualmente possui um corpo vermelho e representa uma variação negativa. Na imagem 2.8 é possível visualizar os *candlesticks* verdes que possuem variação positiva, e os vermelhos variação negativa.

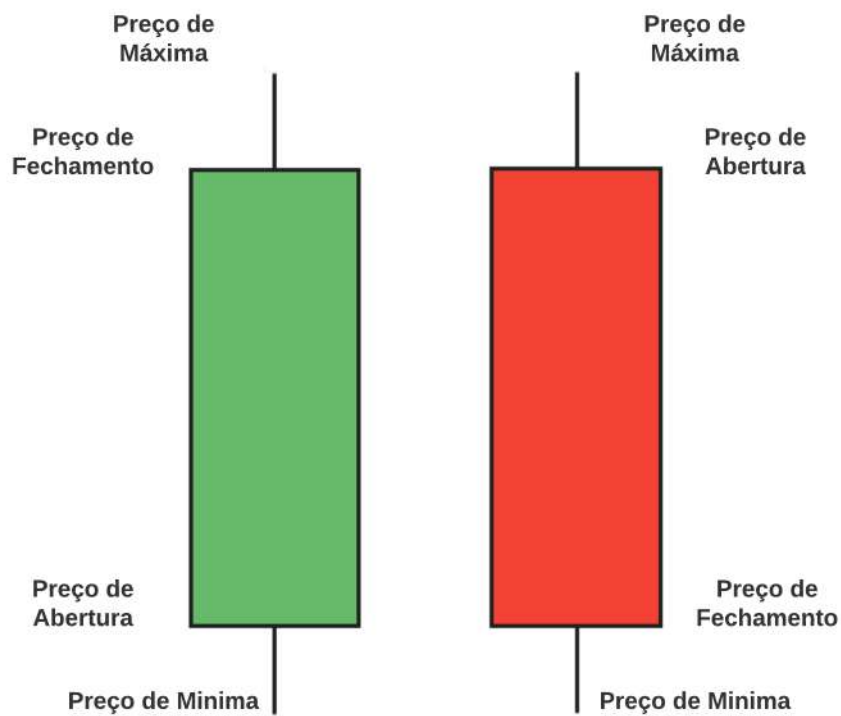


Figura 2.7: *Estrutura de um Candlestick*



Figura 2.8: Gráfico de *Candlestick* do Bitcoin (Gerado com o *tradingview*).

2.3.2 Indicadores

Os indicadores da análise técnica, geralmente consideram combinação de critérios, a partir dos valores de um *candlestick*: abertura; fechamento; máxima; mínima. Um dos principais indicadores, são: Médias Móveis (MM), Índice de Força relativa (IFR) e \mathbb{R} Bandas de Bollinger (BB).

Médias Móveis

Conforme Debastiani (2008) as médias móveis pertencem a uma classe de indicadores chamados seguidores de tendências. Esses indicadores possuem uma inércia natural, ou seja, não foram projetados para apontar reversões rapidamente. Uma média, mostra o valor médio de uma amostra de determinado dado.

- **Média Móvel Aritmética** — A Média Móvel Aritmética (MMA) também conhecida como Média Móvel Simples (MMS), representa o valor médio, normalmente dos preços de fechamento. É calculada da mesma forma que uma média aritmética, no entanto, por ser móvel os seus valores são substituídos a cada novo preço em um período conforme a equação 2.1.

$$MMA = \frac{\sum_{i=1}^N P_f(i)}{N} \quad (2.1)$$

Onde:

- i representa o período atual;
- P_f representa o preço de fechamento;
- e N caracteriza o número de períodos usado para o cálculo da média móvel.

- **Média Móvel Exponencial** — Outro tipo de média móvel bastante utilizado por analistas técnicos são as médias móveis exponenciais (MME). Essa média é calculada através da equação 2.2 onde é atribuído um peso maior para os últimos preços, sendo mais sensível aos últimos movimentos do mercado.

$$MME(i) = (P_f(i) * \frac{F_s}{1 + N}) + (MME(i - 1) * (1 - \frac{F_s}{1 + N})) \quad (2.2)$$

Em que:

- i representa o período atual;
- P_f é o preço de fechamento;
- N caracteriza o número de períodos usado para o cálculo da média móvel;
- F_s representa o fator de suavização, se esse fator for aumentado, observações mais recentes terão mais influência na MME. O que torna esse indicador bastante flexível para a análise.

2.3.3 Bandas de Bollinger

Este é um indicador que surgiu na década de 80 sendo criado por John Bollinger. A ideia por trás desse indicador é mostrar a volatilidade do ativo.

Conforme Bollinger (2002) este indicador é composto por três linhas: uma Média Móvel, a Banda Bollinger Superior (BBS) e a Banda Bollinger Inferior (BBI). É geralmente usada a MMS, que fica no centro, enquanto as bandas atuam baseadas nos desvios padrões dos preços de fechamento. Por padrão são usados 2 desvios para a BBS e -2 desvios para a BBI. Nas equações abaixo são descritas as equações das bandas.

$$BBS = MMS_N + (Q * DP_N) \quad (2.3)$$

$$BBI = MMS_N - (Q * DP_N) \quad (2.4)$$

Onde:

- N caracteriza o número de períodos usado para o cálculo;
- MMS é a média móvel simples de N períodos;
- DP_N é o desvio padrão de N períodos;
- Q é o número de desvios considerado.

2.3.4 Índice de Força Relativa

O Índice de Força Relativa (IFR), funciona como uma espécie de termômetro do mercado, ele busca medir se o mercado já subiu ou caiu muito. Ao contrário das médias móveis, o IFR é considerado um oscilador de momento, ou seja, aponta com que velocidade os preços mudam (Wilder, 1978). Esse indicador possui valores entre 0 e 100 conforme a equação abaixo.

$$IFR = 100 - \frac{100}{1 + \frac{Up}{Down}} \quad (2.5)$$

EM que:

- Up representa a média das variações positivas das cotações nos últimos n períodos;
- $Down$ é a média das variações negativas das cotações também desses n períodos.

Capítulo 3

Desenvolvimento

Neste capítulo, descreveremos o sistema proposto apresentando seus processos, módulos e as tecnologias utilizadas para seu desenvolvimento. Também mostraremos, com auxílio de diagramas, uma visão geral da arquitetura definida e seu funcionamento, desde a interface com o usuário até as estratégias de negociação utilizadas pelos robôs.

3.1 Sistema Proposto

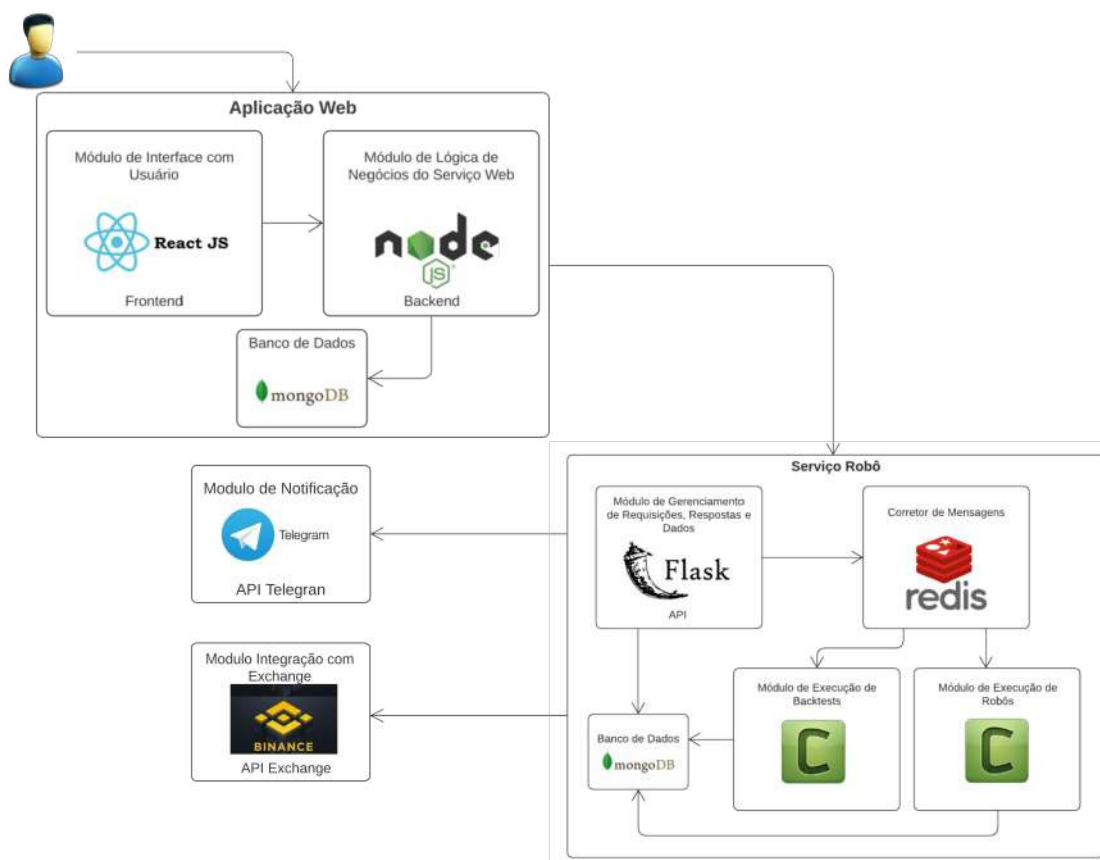


Figura 3.1: Estrutura do Sistema Proposto

Pode-se ver na Figura 3.1 que a plataforma é dividida em duas grandes partes, sendo a Aplicação Web onde funciona toda a lógica de negócio relacionada ao cadastro de usuários e a interface para o monitoramento dos robôs. E o Serviço Robô, o qual recebe requisições da Aplicação Web, realiza a integração com a *exchange*, gerencia a execução dos robôs em tempo real assim como em *backtests*, usa estratégias já conhecidas da análise técnica, como também uma estratégia otimizada através de aprendizado de máquina. Por fim notifica os usuário através do Telegram.

3.2 Serviço Robô

O serviço robô é hospedado em servidores na internet, e oferece várias funcionalidades relacionadas ao *backtest* e o uso dos robôs em tempo real. Pode-se ver na Figura 3.2 que a API implementada em Flask (FLASK, 2021), envia tarefas para uma fila, para serem posteriormente executadas em *workers* Celery (CELERY, 2021), representados pelos módulos de execução de *backtests* e execução de robôs, os quais serão detalhados a seguir.

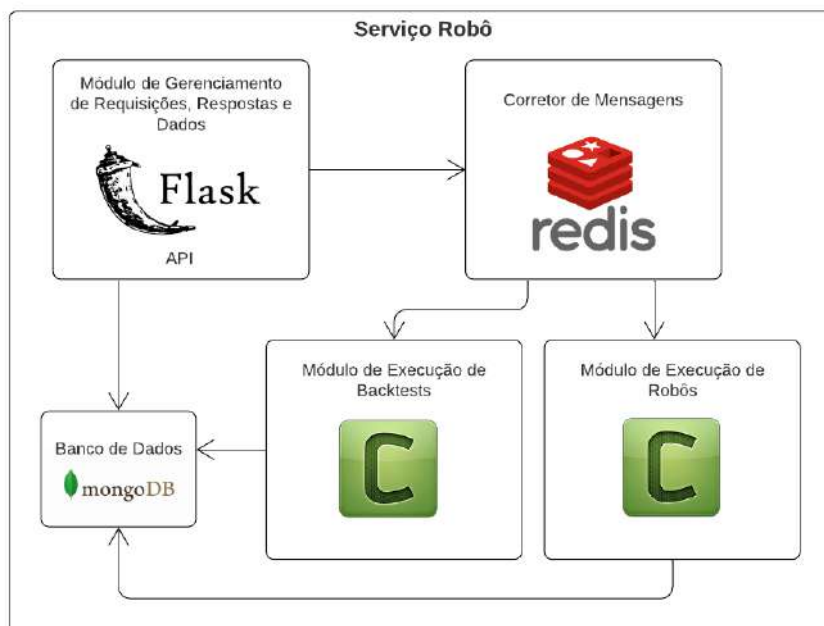


Figura 3.2: Visão geral do Serviço Robô

3.2.1 Módulo de Gerenciamento de Requisições, Respostas e Dados

O Módulo de Gerenciamento de Requisições, Respostas e Dados fornece recursos relacionados ao gerenciamento dos robôs através de uma API que está dividida em 4 camadas. São elas:

- **Controllers** — Essa é o ponto de entrada/saída de informações e comunicação com os clientes. No contexto desse trabalho, o Módulo de Lógica de Negócios da Aplicação Web, que acessa diretamente os métodos da lógica de negócios relacionados ao gerenciamento de robôs.
- **Service** — Na camada de serviço está contida toda a lógica de negócio. Ela está imediatamente abaixo da camada de *Controller*, servindo a camada de *Controller* para prover os recursos requisitados pelos clientes;
- **Schema** — A camada de *schema* é responsável por toda a lógica de definição de um documento de entidade. Ela também gerencia as operações básicas para um CRUD, utilizando MongoEngine que é um *Document-Object Mapper* que permite trabalhar com MongoDB a partir do Python (MONGOENGINE, 2021).
- **DTO** — A camada de *Data Transfer Object* (DTO) é responsável pelo tráfego de informações entre qualquer camada acima da camada Schema. Assim, não está acessível qualquer uso de entidades acima da camada Service, pois o tráfego direto das entidades mostra ser um risco considerável de segurança.

A API foi desenvolvida utilizando a linguagem de programação Python, com o Flask, um *microframework* que utiliza a linguagem Python para criar aplicativos web. Embora seja classificada como um *microframework* o Flask é poderoso e ideal para aplicações robustas, já que é totalmente personalizável, permitindo, caso necessário, a criação de uma arquitetura mais definida (FLASK, 2021).

No contexto desse trabalho, foram identificados desafios relacionados à realização de tarefas em segundo plano, como a execução dos robôs em tempo real, e a efetuação de *backtests* que são tarefas de uma longa duração.

Desse modo, optamos por utilizar o Celery, que é uma fila de tarefas assíncronas de código aberto que se baseia na passagem de mensagens distribuídas. As filas de tarefas são usadas como um mecanismo para distribuir o trabalho entre *threads* ou máquinas. A entrada de uma fila de tarefas é uma unidade de trabalho chamada tarefa. *Workers* dedicados monitoram constantemente essas filas para novos trabalhos serem executados (CELERY, 2021).

O Celery se comunica através de mensagens, geralmente usando um corretor de mensagens para fazer a mediação entre clientes e *workers*. Para iniciar uma tarefa, o cliente adiciona uma mensagem à fila e o corretor de mensagens entrega essa mensagem a um *worker*. Um sistema baseado em Celery pode consistir em vários *workers* e corretores de mensagem, possibilitando uma alta disponibilidade e escala horizontal.

As tarefas definidas neste trabalho são encaminhadas para dois *workers* distintos. O *worker* Módulo de Execução dos Robôs que executa as tarefas ativar e desativar um robô. E o *worker* Módulo de Execução de *backtests* que realiza as tarefas de execução de um

backtest personalizado para um robô específico, e a criação de um *ranking* de robôs a partir da realização de diversos *backtests*, estes serão explicados adiante.

3.2.2 Módulo de Execução dos Robôs

Esse módulo consiste em um *worker*, onde os robôs são executados em tempo real. Foi realizada a integração com a *exchange* da Binance, onde se obtêm dados de mercado, e são efetuadas as transações de compra e venda. Também é realizada a comunicação com a API do Telegram para a notificação com os usuários. Na imagem 3.3 é possível visualizar a estrutura deste *worker*.



Figura 3.3: *Estrutura do Worker de Execução de Robôs*

O diagrama da figura 3.4 ilustra a classe abstrata *Robot*, que serve de modelo para os robôs implementados nesse trabalho, atuando como uma superclasse.

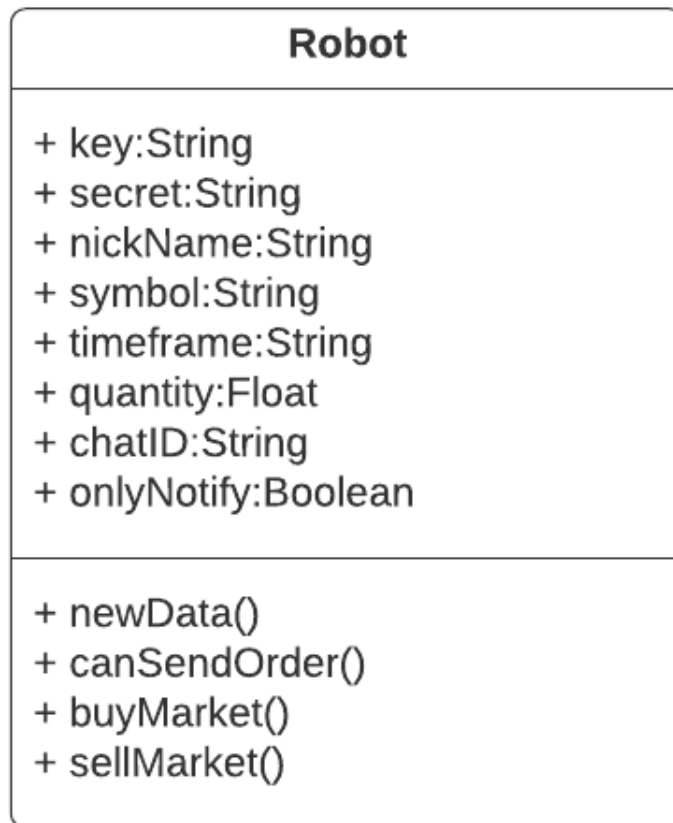


Figura 3.4: Diagrama de classe Robot

Os principais atributos desta classe são:

- **key** — a chave de API da conta do usuário dentro da *exchange* Binance.
- **secret** — o segredo de API da conta do usuário dentro da *exchange* Binance.
- **nickName** — Nome do robô dado pelo criador do mesmo, o qual vai servir como identificador para o usuário manter o controle de seus robôs.
- **timeframe** — esse atributo define o tempo gráfico em que o robô vai se basear para a leitura dos dados de *candlesticks*.
- **symbol** — a criptomoeda usada para as negociações.
- **quantity** — esse atributo define quanto em dólar o robô irá considerar para as negociações.
- **onlyNotify** — define se o robô deve operar apenas no modo notificações, nesse modo não são realizadas negociações, no entanto o robô continua analisando o mercado, e apenas envia notificações para o Telegram do usuário.

- **chatID** — o identificar do Telegram do criador do robô.

Os principais métodos desta classe são:

- **newData()** — é nesse método onde são recebidos os novos dados de mercado, e acontecem toda a lógica necessária para análise da estratégia.
- **canSendOrder()** — esse método fica encarregado de verificar se o robô pode enviar uma ordem, verificando se o mesmo já tem uma operação em andamento, e se está não está apenas no modo notificações.
- **buyMarket()** — realiza o envio de uma ordem de compra considerando a criptomoeda e quantidade configurada.
- **sellMarket()** — realiza o envio de uma ordem de venda considerando a criptomoeda e quantidade configurada.

Dados de Mercado

Foi realizada a integração com a *exchange* Binance que é a maior *exchange* de criptomoedas do mundo, e dispõe de uma excelente documentação. Foi utilizado a biblioteca Binance Public API Connector Python fornecida pela própria *exchange*, onde através de um modelo dirigido por eventos é possível fazer a assinatura de cotações de criptomoedas (BINANCE-API, 2021).

Na Binance Public API são disponibilizados *streams* de dados, nessa etapa utilizamos o *stream* Kline/Candlestick, onde ocorre a notificação de novos dados a cada 2s. Na imagem 3.5 é possível verificar a estrutura recebida nesse *stream*. Os dados mais relevantes para esse trabalho são o (s) sendo a criptomoeda, o (c) que caracteriza o preço de fechamento de um *candlestick* e o parâmetro (x) que representa se o *candlestick* está fechado ou não.


```

{
  "e": "kline",      // Event type
  "E": 123456789,   // Event time
  "s": "BNBBTC",   // Symbol
  "k": {
    "t": 123400000, // Kline start time
    "T": 123460000, // Kline close time
    "s": "BNBBTC", // Symbol
    "i": "1m",     // Interval
    "f": 100,      // First trade ID
    "L": 200,      // Last trade ID
    "o": "0.0010", // Open price
    "c": "0.0020", // Close price
    "h": "0.0025", // High price
    "l": "0.0015", // Low price
    "v": "1000",   // Base asset volume
    "n": 100,      // Number of trades
    "x": false,    // Is this kline closed?
    "q": "1.0000", // Quote asset volume
    "V": "500",    // Taker buy base asset volume
    "Q": "0.500",  // Taker buy quote asset volume
    "B": "123456"  // Ignore
  }
}

```

Figura 3.5: Estrutura de um stream Kline/Candlestick

Roteamento de Ordens

As estratégias implementadas no AllBot utilizam ordens do tipo à mercado, ou seja, quando uma estratégia gera um sinal de compra, é enviado uma requisição de compra à *exchange*, especificando apenas a quantidade e a criptomoeda. Ela é efetuada no momento de seu recebimento e seu preço é determinado somente no momento de sua execução.

No trecho de código abaixo, pode-se ver a implementação do método `buyMarket()` que faz a criação do objeto `client` proveniente da biblioteca `Binance Public API Connector Python` que define a conexão com a *exchange* `Binance` a partir da `API Key` e `Secret Key`, dados estes referentes ao acesso à conta de negociação do usuário. Para o envio de ordens, é feita a chamada do método `new_order()`, passando os parâmetros para a requisição, que são: o nome da criptomoeda (`symbol`), o lado da operação (`side`), o tipo de ordem (`type`) e quantidade (`quoteOrderQty`) sendo esta a quantia em dólar que deseja ser comprada da referente criptomoeda.

Listing 3.1: Exemplo

```

1 def buyMarket(robot):
2     params = {
3         "symbol": robot.symbol,
4         "side": "BUY",

```

```

5     "type": "MARKET",
6     "quoteOrderQty": robot.quantity,
7 }
8 client = Client(robot.key, robot.secret, base_url=BASE_URL)
9 try:
10     response = client.new_order(**params)
11     logging.info(response)
12     updateEntryPosition(robot, "BUY", response)
13     message = 'robot: ' + robot.nickName + '\n' +
14              'symbol: ' + robot.symbol + '\n' +
15              'side: BUY' + '\n' +
16              'quoteOrderQty: ' + str(response['executedQty'])
17     sendTelegramMessage(message, robot.chatID)
18     return response['orderId']
19
20 except ClientError as error:
21     logging.error(
22         "Found error. status: {}, error code: {}, error message: {}"
23         .format(error.status_code, error.error_code, error.error_message)
24     )
25 )

```

Notificações

De modo a notificar as ações tomadas pelo robô ao longo de sua execução foi desenvolvido um módulo de notificação com o Telegram, usando `python-telegram-bot`, uma biblioteca que proporciona uma integração entre *softwares*, desenvolvidos em Python, com a API do Telegram Bot.

Esse módulo consiste em um *bot* para o Telegram. Os *bots* do Telegram são contas especiais que não exigem um número de telefone adicional para serem configuradas, os usuários podem interagir com ele diretamente ou até mesmo coloca-los em grupos. Para a criação do *bot* usamos o BotFather (um *bot* oficial do Telegram) e após seguir alguns passos simples, recebemos um *token* de autorização, que nos permite enviar mensagens aos usuários (TELEGRAM-API, 2021).

3.2.3 Módulo de Execução de Backtests

Nesse módulo encontra-se o *worker* de *backtest*, responsável pela execução dos robôs em dados históricos, encarregado por calcular todas as estatísticas dos robôs.

Nesse worker, além de possibilitar a execução de um *backtest* personalizado por robô, ele realiza diariamente um conjunto de simulações envolvendo todos os robôs disponíveis,

combinando criptomoedas e *timeframes* diferenciados. Proporcionando como resultado um *ranking*.

Base de Dados

A API da *binance*, possui uma limitação no que se refere aos dados históricos de mercado, fornecendo no máximo os últimos 1000 *candlesticks* por requisição, dessa forma seria inviável a realização de alguns *backtests*, por exemplo, o caso do *timeframe* de 1 minuto, uma base de dados de apenas 1000 *candlesticks* seria equivalente a apenas 16 horas.

Para resolver esse problema, foi criado um *script* com a finalidade de obter os dados divididos em *slots* mensais disponibilizados pela *exchange* de forma compactada, esses foram concatenados para popular uma única base de dados, viabilizando a realização de *backtests* em um maior período.

Modelagem do Backtest

A modelagem do *backtest* foi realizada, usando o *framework* *Backtesting.py* para inferir a viabilidade de estratégias de negociação em dados históricos.

O *framework* *Backtesting.py* é de código aberto e permite definir a estrutura de uma estratégia conforme os objetivos necessários. Após escrever todo o código que envolve a lógica da estratégia de negociação é possível usar recursos do *framework*, que recebem como parâmetros a base de dados, para retornar uma estrutura com os resultados da estratégia (BACKTESTING, 2021).

Na imagem 3.6 é possível ver a estrutura básica das estatísticas geradas, dentre as quais destacam-se o *Return [%]* que caracteriza o lucro percentual obtido, e o *Win Rate [%]* que representa a taxa de acerto por operação.

Start	2004-08-19 00:00:00
End	2013-03-01 00:00:00
Duration	3116 days 00:00:00
Exposure Time [%]	94.27
Equity Final [\$]	68935.12
Equity Peak [\$]	68991.22
Return [%]	589.35
Buy & Hold Return [%]	703.46
Return (Ann.) [%]	25.42
Volatility (Ann.) [%]	38.43
Sharpe Ratio	0.66
Sortino Ratio	1.30
Calmar Ratio	0.77
Max. Drawdown [%]	-33.08
Avg. Drawdown [%]	-5.58
Max. Drawdown Duration	688 days 00:00:00
Avg. Drawdown Duration	41 days 00:00:00
# Trades	93
Win Rate [%]	53.76
Best Trade [%]	57.12
Worst Trade [%]	-16.63
Avg. Trade [%]	1.96
Max. Trade Duration	121 days 00:00:00
Avg. Trade Duration	32 days 00:00:00
Profit Factor	2.13
Expectancy [%]	6.91
SQN	1.78

Figura 3.6: Resultados de um backtest

3.3 Estratégias

Foram implementadas estratégias bastante conhecidas da análise técnica, sendo estas classificadas entre: seguidora de tendência (que é quando o mercado tem uma direção bem definida) e estratégias de reversão à média (que é quando o mercado se encontra em um estado estacionário, ou seja, não possui uma tendência definida).

3.3.1 Mean Reversion

Essa estratégia utiliza o indicador de análise técnica \textcircled{R} Bandas de Bollinger. Em conformidade a Bollinger (2002), esse indicador parte do princípio de que uma série de preços possui um comportamento estacionário, assim é assumido que em cerca de 95% do tempo os preços estariam entre 2 e -2 desvios-padrão. Dessa forma, a estratégia propõe que um sinal

de compra seja gerado no momento em que for observado um preço abaixo da banda de -2 desvios. E de forma oposta, um sinal de venda quando o preço estiver acima da banda de 2 desvios. Na imagem 3.7 é possível verificar os sinais gerados em um gráfico do bitcoin.



Figura 3.7: Gráfico de bitcoin, com o indicador bandas de bollinger (Gerado com o tradingview).

3.3.2 Cross Average

Nessa estratégia é empregado um conceito estatístico bastante comum: as médias móveis. O sinal de entrada dessa estratégia consiste no cruzamento de duas médias móveis. Uma rápida e outra lenta. Na imagem 3.8 é utilizado uma média móvel rápida de 5 períodos, e uma média móvel lenta de 12 períodos. O sinal de compra é gerado no momento em que a média rápida cruza a média lenta, de baixo para cima. E o sinal de venda ocorre no momento em que a média rápida cruza a média lenta, de cima para baixo.



Figura 3.8: Gráfico de Bitcoin, com o indicador médias móveis (Gerado com o tradingview).

3.3.3 IFR2

Segundo Connors and Alvarez (2009), na estratégia IFR2 os sinais de negociação são gerados a partir do indicador IFR e do indicador média móvel simples. Nessa estratégia, o sinal de compra acontece quando o IFR estiver com valor abaixo de 10 e a saída da operação (venda) ocorre quando o preço fecha acima da média móvel de 5 períodos. Na imagem 3.9 é possível observar os sinais de entrada e saída gerados a partir do IFR e da média.



Figura 3.9: Gráfico de bitcoin, com o indicador médias móveis e IFR (Gerado com o tradingview).

3.4 Otimização de Estratégias

O processo de otimização das estratégias seguiu uma adaptação do modelo proposto no livro "Advances in Financial Machine Learning", Prado (2018), onde um modelo secundário é treinado de modo a melhorar os sinais de entrada de um modelo primário.

É realizado o *backtest* de uma estratégia, e com os resultados desse *backtest* é desenvolvido um modelo usando Aprendizado de Máquina para identificar padrões que indiquem quando a estratégia irá ser lucrativa ou não. A ideia desse modelo secundário é funcionar como uma espécie de filtro, definindo quando um robô deve ou não operar, dadas as circunstâncias do mercado. Na figura 3.10 é possível observar o diagrama de atividades de um robô otimizado, onde o modelo primário é a própria estratégia utilizada pelo robô, e ao gerar um sinal de compra, passa por um filtro, caracterizado pelo modelo secundário que vai definir se o robô vai ou não executar a operação.

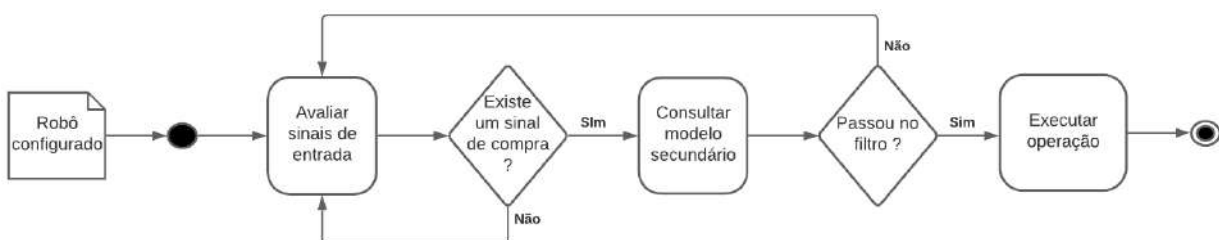


Figura 3.10: Diagrama de atividades de um robô otimizado.

Na etapa de otimização de estratégias, realizamos *backtests* usando as criptomoedas: Bitcoin, Ethereum e Ripple, combinando as três estratégias disponíveis, e os *timeframes*: 5 minutos, 15 minutos, 30 minutos, 1 hora, 2 horas, 4 horas, 6 horas, 8 horas, 12 horas, 1 dia. A fim de se obter um resultado mais alinhado com a realidade, foi considerada uma taxa operacional de 0.075 %. Para cada *backtest*, foi realizado o treinamento do modelo e a análise de desempenho. Como variáveis para o treinamento foram usados a volatilidade dos últimos 10 dias, a volatilidade acumulada dos últimos 5 dias e a distância do preço atual para uma média móvel simples de 80 períodos.

3.5 Aplicação Web

A Aplicação Web assim como o Serviço Robô é hospedada em um servidor na internet e abriga a interface para utilização da plataforma por parte do usuário e a lógica de negócio relacionada às funcionalidades disponibilizadas, funcionalidades estas que em partes usam o Serviço Robô. A mesma segue uma arquitetura semelhante à arquitetura em 3 camadas. Comumente, arquiteturas em três camadas são arquiteturas distribuídas, ou seja, a camada de interface executa na máquina dos clientes, a camada de lógica de negócio executa em um servidor, normalmente chamado servidor de aplicação, e por fim, temos o banco de dados (VALENTE, 2020). Dessa forma, as camadas são distribuídas da seguinte maneira:

- **Interface com o Usuário** —

"Também chamada camada de apresentação, é responsável por toda interação com o usuário. Ela trata tanto da exibição de informação, como da coleta e processamento de entradas e eventos de interfaces, tais como cliques em botões, marcação de texto, etc. A camada de interface pode ser uma aplicação *desktop*, em Windows ou outro sistema operacional com interface gráfica, como também Web." (VALENTE, 2020).

- **Lógica de Negócio** — "Também conhecida como camada de aplicação, implementa as regras de negócio do sistema." (VALENTE, 2020);
- **Banco de Dados** — "Que armazena os dados manipulados pelo sistema." (VALENTE, 2020)

3.5.1 Módulo de Lógica de Negócios da Aplicação Web

A camada de lógica de negócio, que chamamos Módulo de Lógica de Negócios da Aplicação Web, por motivos de organização foi dividida em módulos menores, que serão citados e explicados posteriormente, estes, ficaram subdivididos em:

- **Controllers** — Responsável por receber as requisições da camada de apresentação e as enviar para os *Services*, também é responsável pelo fluxo contrário, que consiste em receber as respostas dos *Services* e enviá-las para a Interface com o Usuário;
- **Services** — Implementa as regras de negócio do sistema como validações, consultas e inserções à base de dados, assim como chamadas as integrações;
- **Models** — Arquivos mapeados para uma coleção MongoDB e definem a forma dos documentos na mesma;
- **Integrações** — Podendo ou não estar presentes, são responsáveis por se comunicarem com o Serviço Robô, enviando requisições para o mesmo e as recebendo respostas, dado que ele é um serviço separado da Aplicação Web.

Na Figura 3.11 podemos observar um diagrama de classe genérico que representa a comunicação entre as partes que estruturam os módulos internos ao Módulo de Lógica de Negócios da Aplicação Web.

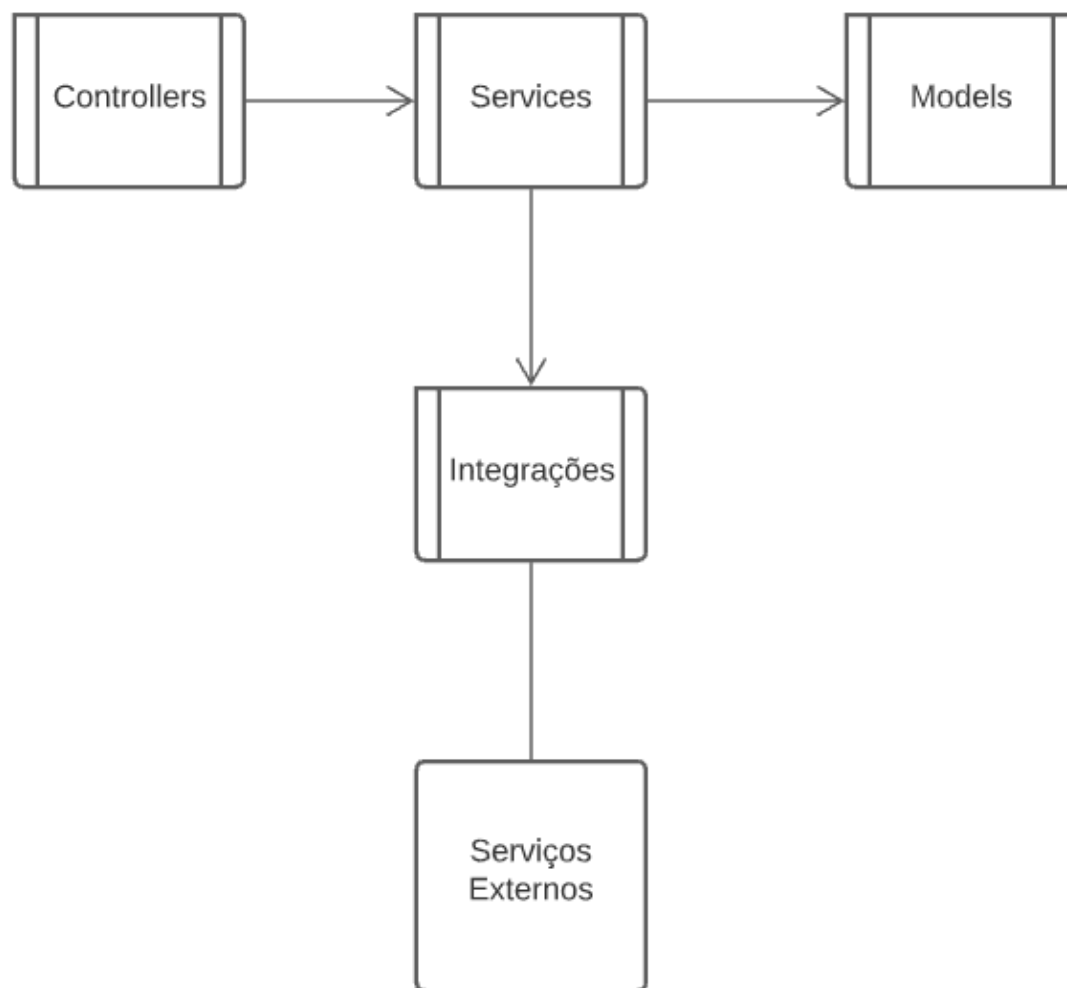


Figura 3.11: Diagrama de classe genérico para subdivisões do Módulo de Lógica de Negócios da Aplicação Web

O Módulo de Lógica de Negócios da Aplicação Web foi desenvolvido utilizando a linguagem de programação Javascript, o Node.js, uma *tecnologia* assíncrona de código aberto, multiplataforma, interpretada pela máquina virtual V8, a mesma utilizada pelo Google no navegador Google Chrome ao executar códigos JavaScript (DUARTE, 2021), o Axios, este é um cliente HTTP baseado em *promises* simples para navegador e Node.js (AXIOS, 2021), e o Mongoose, uma ferramenta que fornece uma solução direta e baseada em esquema para modelar os dados do seu aplicativo. Inclui conversão de tipo embutida, validação, construção de consulta, ganchos de lógica de negócios e muito mais, prontos para uso (MONGOOSE, 2021).

Módulo Login

Este módulo é responsável pelo acesso do usuário aos seus dados pessoais e públicos exibidos no sistema, nele são utilizadas todas as classes relacionadas ao usuário, são elas: LoginController; LoginService e UserModel.

Aprofundando um pouco sobre o funcionamento deste módulo, o LoginController possui um método chamado *login*, este é responsável por retirar da requisição os dados recebidos da camada de apresentação e passá-los para o LoginService, que possui um método cujo nome também é *login*, porém, que tem a finalidade de realizar toda a regra de negócio necessária para realização do *login*, desde a validação até a consulta ao banco de dados para devolução dos mesmos ao LoginController que, em caso de sucesso, retorna à camada de Interface com o Usuário permitindo o acesso a todas as outras funcionalidades do sistema. Para melhor entendimento do Módulo Login temos sua representação através de um diagrama de classes na Figura 3.12.

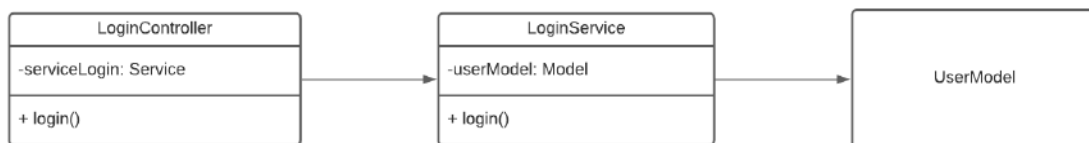


Figura 3.12: Diagrama de classe do Módulo Login

Módulo Usuário

Este módulo é responsável pelo gerenciamento (cadastro, consulta, atualização e remoção) dos dados pessoais e públicos dos usuários no sistema, nele são utilizadas todas as classes relacionadas ao usuário, são elas: UserController; UserService e UserModel. Podendo também instanciar outras classes, até mesmo de outros módulos, para acessar alguns métodos específicos como, por exemplo, o sendConfirmEmail do *service* SendEmailService.

Adentrando um pouco mais nos fluxos desse módulo, podemos tomar como exemplo de funcionamento do próprio, o fluxo de cadastro de usuário. Este começa pelo `UserController`, que possui vários métodos, entre eles, o `createUser`, responsável por repassar o corpo da requisição para um método de mesmo nome contido dentro do `UserService`. Tal método realiza todas as validações de dados necessárias para criação de um novo usuário, como a validação de CPF e a verificação da existência do usuário na base de dados, posteriormente utiliza o método `sendConfirmEmail` do `SendEmailService` para enviar um email de confirmação de cadastro, além de retornar as informações necessárias para que o *controller* envie a resposta para a camada de apresentação. Na figura 3.13 podemos observar o diagrama de classes do Módulo Usuário e classes acessadas por ele.

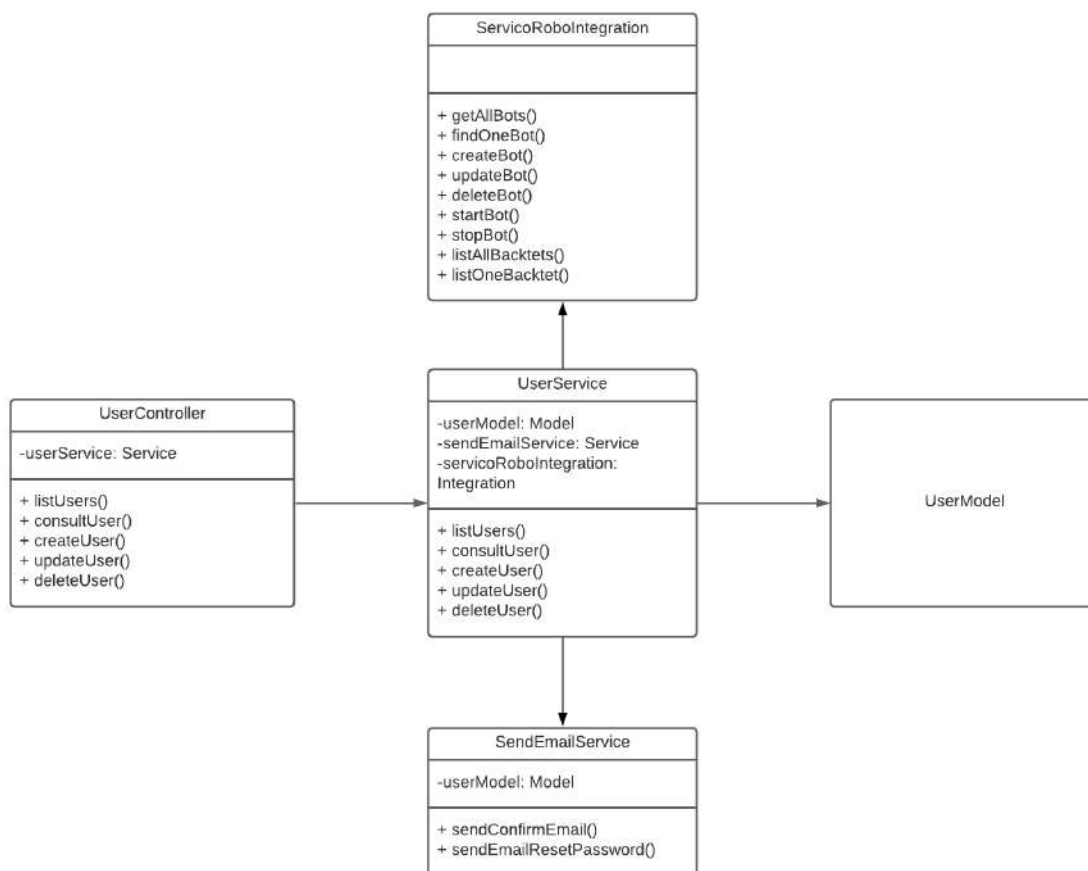


Figura 3.13: Diagrama de classes do Módulo Usuário

Módulo Robô

Este módulo é responsável pelo gerenciamento (cadastro, consulta, atualização e remoção) dos dados públicos e privados dos robôs para isso utilizando o Serviço Robô, nele são usadas todas as classes relacionadas ao robô, são elas: `RoboController`; `RoboService` e `ServicoRoboIntegration`.

Para explicar um pouco o funcionamento desse módulo, podemos citar o fluxo de criação de robôs, começamos o fluxo a partir do `RoboController` que assim como os *controllers* anteriores, possui vários métodos responsáveis por retirar da requisição a informação necessária para prosseguir até a etapa de aplicação das regras de negócios.

Após informar os dados para o método do `RoboService`, são realizadas validações, por exemplo, averiguar a existência do usuário no banco de dados da Aplicação Web, após tais validações, é criado um corpo de requisição, unindo os dados provenientes da camada de apresentação e os consultados na base da Aplicação Web, para serem passados ao devido método da integração `ServicoRoboIntegration`, esta contém vários métodos, os quais têm como função realizar requisições para o Serviço Robô, no caso do fluxo em questão o método chamado seria o `createBot`, que assim como os demais métodos dessa integração informa a resposta obtida de volta para o *service*, que por sua vez retorna ao controller para ele enviar uma resposta de sucesso ou de erro à Interface com o Usuário. Na Figura 3.14 temos o diagrama de classes do Módulo Robô.

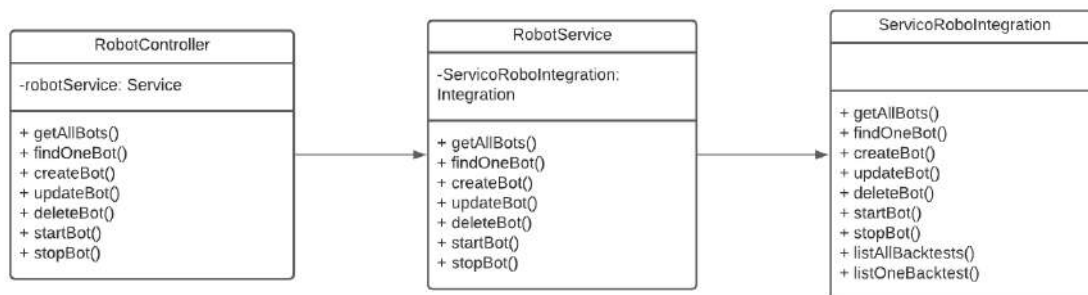


Figura 3.14: Diagrama de classes do Módulo Robô

Módulo Backtest

Este módulo é responsável pelas consultas dos dados dos *backtests* no Serviço Robô, nele são usadas todas as classes necessárias para obtenção das informações sobre os *backtests*, são elas: `BacktestController`; `BacktestService` e `ServicoRoboIntegration`.

Esse módulo é extremamente semelhante ao módulo anterior, utilizando da mesma integração e do mesmo serviço externo à Aplicação Web, o que os diferenciam são seus *controller* e *service* (`BacktestController`, `BacktestService`). Tomando o fluxo de listagem de um *backtest* específico para explicar como o módulo se comporta, temos a retirada dos dados vindos da requisição, por parte do método `listOneBacktest` do `BacktestController`, os quais serão necessários para que o *service* caminhe com as regras de negócio, que em maior parte consiste em validações simples dos dados de maneira que possam seguir para o método `listOneBot` da classe `ServicoRoboIntegration` que realiza uma requisição para o Serviço Robô e aguarda a resposta para posteriormente ser enviada de volta no fluxo até chegar ao *controller*

que a transmitirá como resposta para o Módulo de Interface com Usuário. A Figura 3.15 representa o diagrama de classes do Módulo *backtest*.

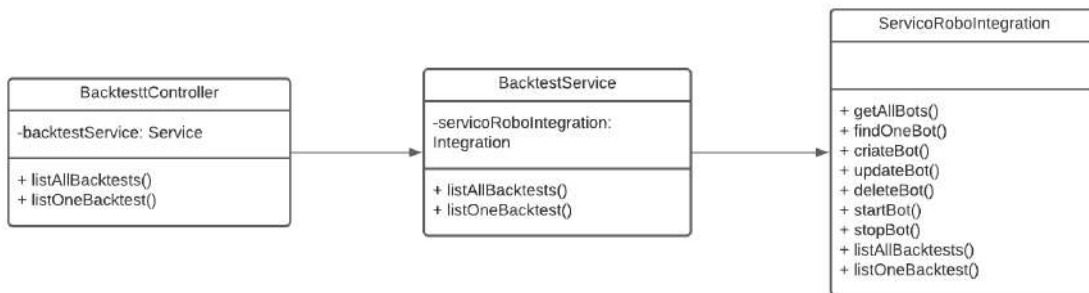


Figura 3.15: Diagrama de classes do Módulo *Backtest*

Módulo Envio de Email

Como o próprio nome sugere ele é responsável executar os fluxos de envio de email, nele são usadas todas as classes necessárias para receber uma requisição, consultar informações na base de dados de usuários e realizar o envio do email específico de acordo com fluxo em execução, são elas: *SendEmailController*; *SendEmailService*.

Esse módulo possui dois fluxos realizados de maneiras diferentes, o primeiro fluxo é o de envio de email de confirmação de cadastro, que diferentemente dos demais módulos é iniciado e finalizado em outro módulo a partir de outro fluxo, o de cadastro de usuário, que se passa no Módulo Usuário, mais especificamente no *UserService*, neste *service* é instanciado o *SendEmailService* para possibilitar o uso do método *sendConfirmEmail*, este responsável por enviar definitivamente o email de confirmação ao usuário, o segundo fluxo, é o de redefinição de senha, semelhante aos explicados anteriormente, inicia a partir do *controller* *SendEmailController* com os dados recebido por uma requisição, que seguem até o *service*, onde serão executadas as validações e logo depois o envio de email, de maneira que ao final é retornada uma mensagem de sucesso para a camada de interface com o usuário. A seguir na figura 3.16 podemos visualizar o diagrama do Módulo de Envio de Email e classe que o acessam.

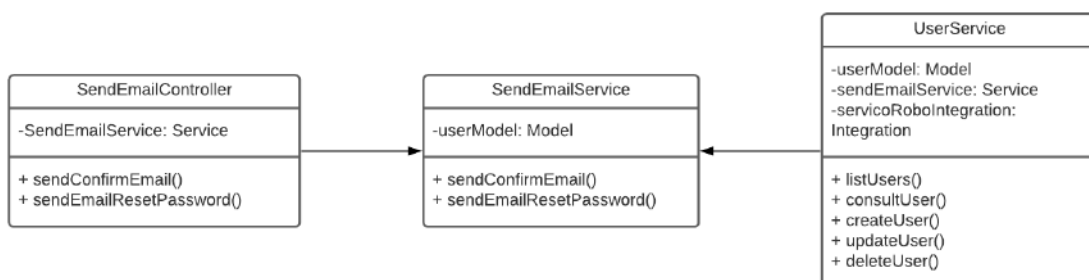


Figura 3.16: Diagrama de classes do Módulo *Envio de Email*

3.5.2 Módulo de Interface com Usuário

Esse módulo é a camada de Interface com o usuário, e com isso, voltado para exibição e comunicação entre o usuário e o sistema, de forma que todo o sistema é controlado a partir deste, para definirmos como o módulo foi arquitetado podemos ver adiante os principais componentes que integram essa camada:

- **Services** — Componente que tem como principal função reger a comunicação com a camada de lógica de negócios, definindo a implementação das requisições e os *endpoints* de cada requisição.
- **Routers** — Componente que possui responsabilidade de gerenciar a navegação dentro desse serviço, fazendo algumas restrições, como, por exemplo, usuários não autenticados acessar páginas restritas a usuários autenticados.
- **Layout** — Componente em que define a estrutura de *design* para visualização dos usuários, como *menu* de exibição de usuários autenticados, na finalidade de evitar recompilação de componentes já previamente compilados.
- **Componentes** — Componente que define apresentação dos componentes do framework.
- **Common** — Componente que define os parâmetros de estrutura de apresentação dos componentes tal como parâmetros de temas, configurações, dados de estruturas (robôs, usuários, ativos, *menus* e testes).

Para o desenvolvimento desse serviço, foi utilizado a linguagem de programação JavaScript, mais especificamente o framework React, sendo uma biblioteca com foco em criar interfaces de usuário em páginas web, o Axios assim como no Módulo de Lógica de Negócios para estabelecer a comunicação com este que irá fornecer as informações necessárias para a amostragem das informações destinadas ao usuário.

Segurança

Para garantir a segurança do acesso ao Módulo de Lógica de Negócios da Aplicação Web, uma vez que o sistema está hospedado em um servidor na *internet*, é utilizado o protocolo HTTPS para que o mesmo utilize o padrão de comunicação criptografado já validado pelo mercado, além disso, é utilizado o padrão JWT para fazer autenticação via *tokens* da comunicação realizadas pelas requisições com destino a camada de lógica de negócios.

Comunicação

A comunicação com a camada de lógica de negócios é feita utilizando o Axios (como já explicado anteriormente) que realiza as requisições (*get*, *post*, *put*, *delete*) para os *endpoints* do Módulo de Lógica de Negócios da Aplicação Web.

Módulo Apresentação

Esse módulo é designado a apresentar o que é o sistema e possibilitar a criação ou autenticação de usuário.

Módulo Usuário

Este módulo é responsável pelo gerenciamento (cadastro, consulta, atualização) dos dados pessoais dos usuários no sistema, para acesso a esses dados só é necessário acessar o perfil, e caso haja necessidade de alteração, após as alterações, clicar em salvar como segue a imagem.

Também falando do módulo usuário, temos as configurações de conta do usuário, onde é definido a API Key e o Secret Key para acessar a *exchange* Binance (*definida* para realizar as operações da plataforma) e é possível ativar notificações de execução do robô via email (é utilizado o email cadastrado na plataforma) e via Telegram (para que essa funcionalidade seja ativada é necessário informar o *chat-id*).

Módulo Robô

Este módulo é responsável pelo gerenciamento (cadastro, consulta, atualização) dos dados pessoais dos robôs no sistema, para criação de um novo robô, possui o acesso a essa página pelo menu de navegação lateral, preencher as informações de criação do robô, as seguintes:

- **Nome do robô** — Nome pessoal do robô dado pelo criador do mesmo, o qual vai servir como identificador para o usuário manter o controle de seus robôs.
- **Estratégia** — Definir qual das estratégias predefinidas o robô vai utilizar para fazer as operações, para saber mais sobre as estratégias olhar seção 3.3. Ao selecionar a estratégia do robô novos campos se abrirão para preenchimento das configurações de estratégia.
- **Timeframe** — Determinar o tempo gráfico em que o robô vai se basear na para realizar a estratégia de operação.
- **Criptomoeda** — Escolher qual criptomoeda será operada pelo robô, sabendo que todas as opções fazem o par com Tether (USDT), pois é uma moeda mais estável para operações de trade já que possui equivalência com o dólar.

- **Quantidade** — Decidir a quantidade em dólar disponível para o robô fazer operações, caso a carteira da Binance não possua limite suficiente nenhuma operação será realizada por esse robô em específico.

Módulo Backtest

Esse módulo é destinado a realizar operações de testes de robôs, este faz-se testes baseado em toda a base de dados disponível, trazendo assim informações detalhadas de funcionamento no período definido.

Para realizar o *backtest* é possível ao editar um robô em que a opção realizar teste ficará disponível, ao abrir este módulo é possível editar as informações na qual se deseja realizar o *backtest* e confirmar. Será realizado o teste e as informações do retorno serão preenchidas destacado na tela.

Módulo Ranking

Esse módulo traz para o usuário as informações dos melhores robôs baseadas nos *backtests* realizado do cruzamento de todas as configurações possíveis dos robôs.

Ao acessar essa página pelo *menu* de navegação lateral é visualizado uma lista com a classificação de robôs com maiores índices de retorno em ordem decrescente com as principais informações de configuração para que o usuário possa replicar o robô e aproveitar as informações do ranking para obter os melhores resultados possível com o sistema. Ao abrir a bandeja de mais informações, é possível acessar as informações avançadas que possui dados de todos os estados do robô no tempo também especificado e poder investigar a fundo os dados de desempenho, como está na figura 3.17 a seguir.

Estratégia robô	Ativo	Timeframe	Quantidade operações	Retorno (%)					
^ CROSSAVERAGE	ETHUSD	4h	247	12515.46					
Informações adicionais									
Dia Inicio	Dia Fim	Duração	Media de duração por trade	Taxa de acerto (%)	Trade de maior duração	Melhor trade (%)	Pior trade (%)	Retorno de Buy & Hold (%)	Média por trade (%)
Thu, 17 Aug 2017 04:00:00 GMT	Wed, 08 Sep 2021 00:00:00 GMT	1482 days 20:00:00	3 days 06:00:00	38.87	17 days 04:00:00	74.22	-12.73	1011.70	1.98

Figura 3.17: Informações de ranking robô

Capítulo 4

Resultados

Ao final deste trabalho foram obtidos os resultados esperados no planejamento inicial. Foram necessários ajustes, como em todo processo de desenvolvimento de *software*: mudanças de lógicas de negócio, adições de funcionalidades e refatorações. Devido ao uso de tecnologias populares, foi possível contornar esses imprevistos e realizar a entrega do que foi estipulado.

4.1 Resultados do Serviço Robô

Dentre os *backtests* realizados, apenas em duas combinações foram identificadas melhorias consideráveis na otimização. Foram elas a estratégia Mean Reversion no Ripple com *timeframe* de 5 minutos, e Mean Reversion no Ethereum com *timeframe* de 30 minutos. Nos dados abaixo temos os resultados em dados *out-of-sample* (dados que não foram usados para treinamento) dessas duas estratégias otimizadas.

Na figura 4.1 é possível observar um maior retorno acumulado obtido, e além disso no Quadro 2 podemos perceber diversos outros benefícios, como um menor número de negociações o que implica diretamente na redução do valor gasto com taxas de negociação, assim como um menor tempo de exposição ao mercado.

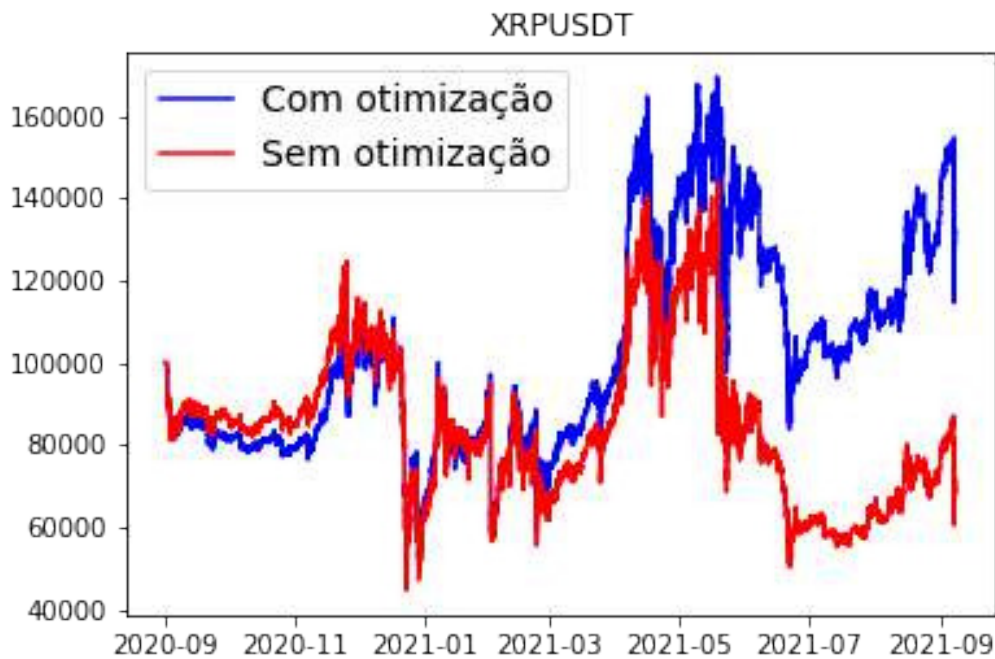


Figura 4.1: Resultado acumulado do Robô Mean Reversion com e sem otimização.

Quadro 2: Comparativo do Robô Mean Reversion na Ripple com e sem otimização.

	Sem Otimização	Com Otimização
Retorno Acumulado [%]	-31,43	32,02
Tempo de Exposição [%]	49,93	41,71
Taxa de Acerto [%]	66,89	68,49
Negociações	1178	987
Rebaixamento Máximo [%]	-65,15	-61,65

Semelhante às melhorias observadas no Ripple, no Ethereum tivemos diversas melhorias. Na figura 4.2 temos o retorno acumulado, e no Quadro 3 temos um comparativo dos resultados.

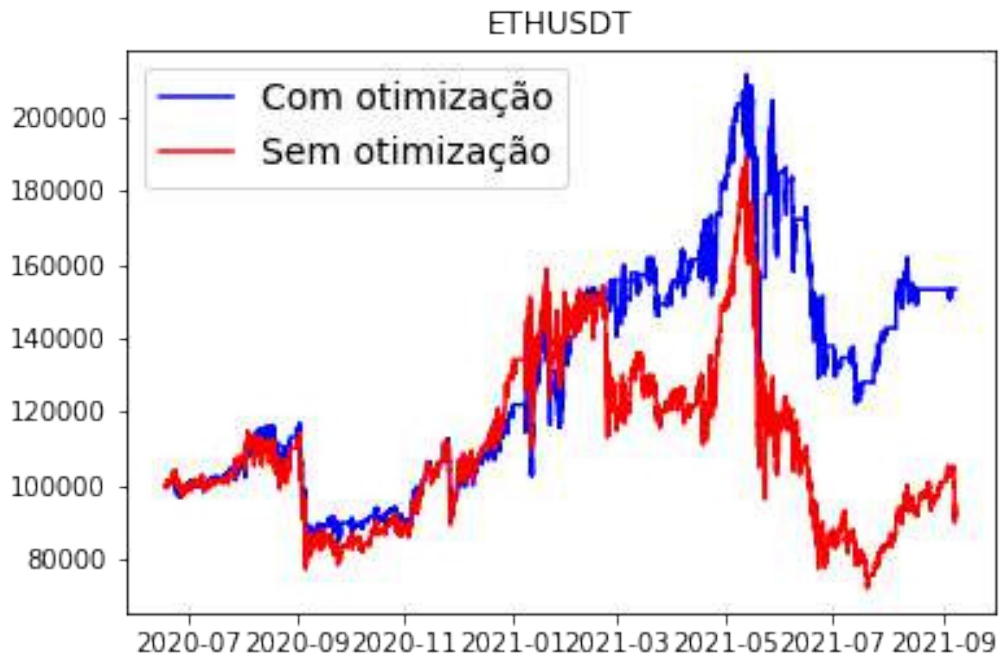


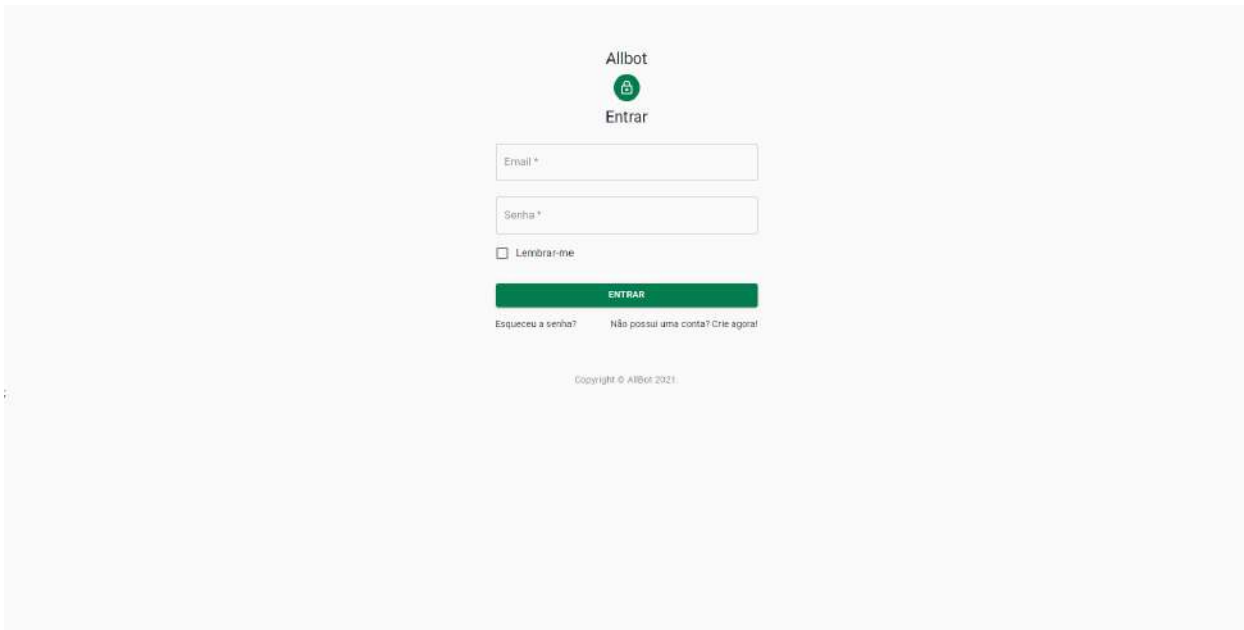
Figura 4.2: Resultado acumulado do Robô Mean Reversion com e sem otimização. (Resultado acumulado por tempo de operação)

Quadro 3: Comparativo do Robô Mean Reversion na Ethereum com e sem otimização.

	Sem Otimização	Com Otimização
Retorno Acumulado [%]	-8,17	53,36
Tempo de Exposição [%]	48,23	34,66
Taxa de Acerto [%]	66,66	68,82
Negociações	237	170
Rebaixamento Máximo [%]	-61,89	-42,23

4.2 Resultados do Serviço Web

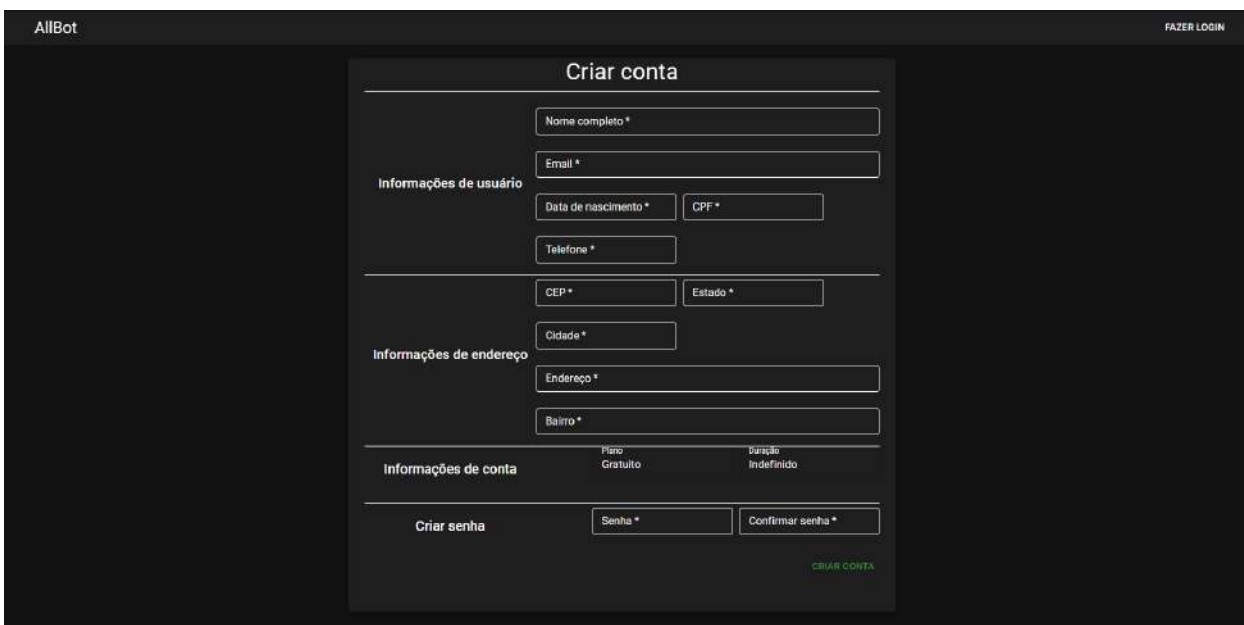
A seguir, na figura 4.3 é ilustrado a tela de login da plataforma.



A screenshot of the Allbot login page. At the top center, the Allbot logo is displayed above a green lock icon and the word "Entrar". Below this, there are two input fields: "Email *" and "Senha *". A checkbox labeled "Lembrar-me" is positioned below the password field. A prominent green button labeled "ENTRAR" is centered below the inputs. At the bottom of the form area, there are two links: "Esqueceu a senha?" and "Não possui uma conta? Crie agora!". The footer contains the text "Copyright © AllBot 2021".

Figura 4.3: *Tela de Login.*

Caso o Usuário ainda não possua um cadastro na plataforma, ele pode acessar a tela de cadastro clicando no *link* "Não possui uma conta? Crie agora!" que está na parte inferior direita da tela de login, assim ele será redirecionado para uma página semelhante à Figura 4.4.



A screenshot of the AllBot user registration page. The page has a dark theme. At the top left is the "AllBot" logo and at the top right is the text "FAZER LOGIN". The main heading is "Criar conta". The form is divided into several sections: "Informações de usuário" with fields for "Nome completo *", "Email *", "Data de nascimento *", and "CPF *"; "Informações de endereço" with fields for "Telefone *", "CEP *", "Estado *", "Cidade *", "Endereço *", and "Bairro *"; and "Informações de conta" with radio buttons for "Plano Gratuito" and "Plano Indefinido", and fields for "Criar senha", "Senha *", and "Confirmar senha *". A green "CRIAR CONTA" button is located at the bottom right of the form.

Figura 4.4: *Tela de Cadastro de Usuário.*

Após a criação de conta, com finalidade de autenticação do usuário, é enviado um email para o email cadastrado com intuito de realizar uma verificação, apenas um *link* de verificação em que o usuário garanta que seja um email verdadeiro.



Figura 4.5: *Email Enviado ao Usuário.*

Ao acessar o *link* informado no email, o usuário é levado a uma tela do sistema para que possa autenticar sua conta. veja na Figura 4.6 abaixo

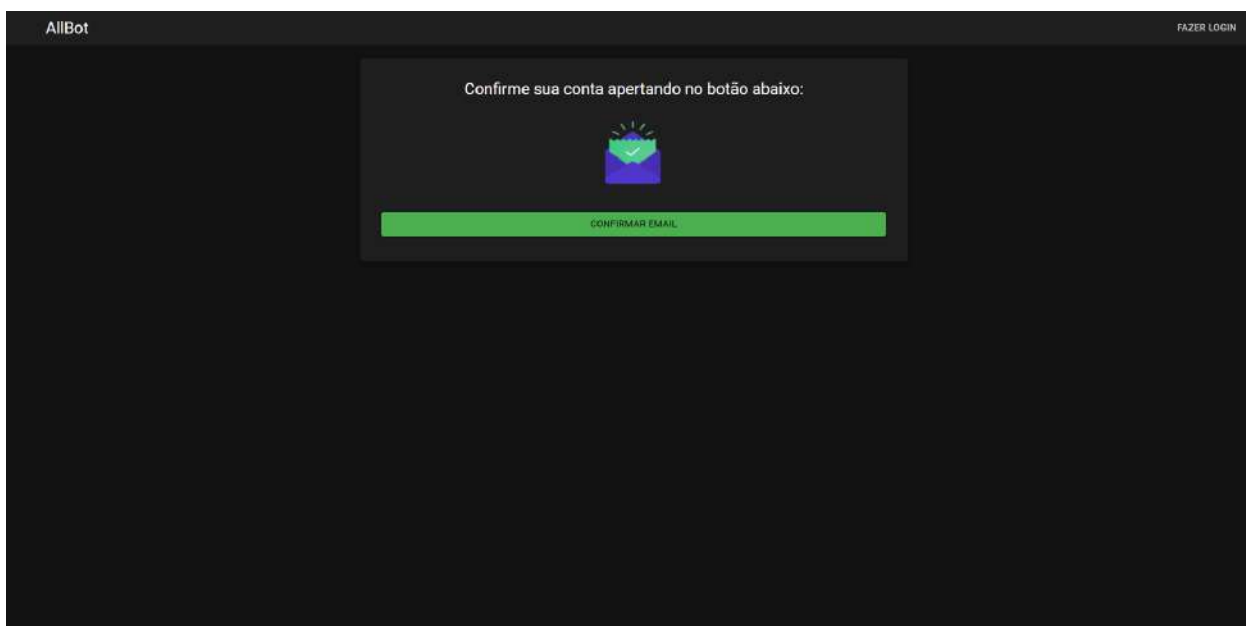


Figura 4.6: *Tela Confirmação de Email.*

Caso o usuário perca ou esqueça a senha, também é possível recuperá-la, na página de login, acessando o *link* "Esqueceu a senha?" onde será redirecionado para a tela de recuperação de senha como mostrada na Figura 4.7.

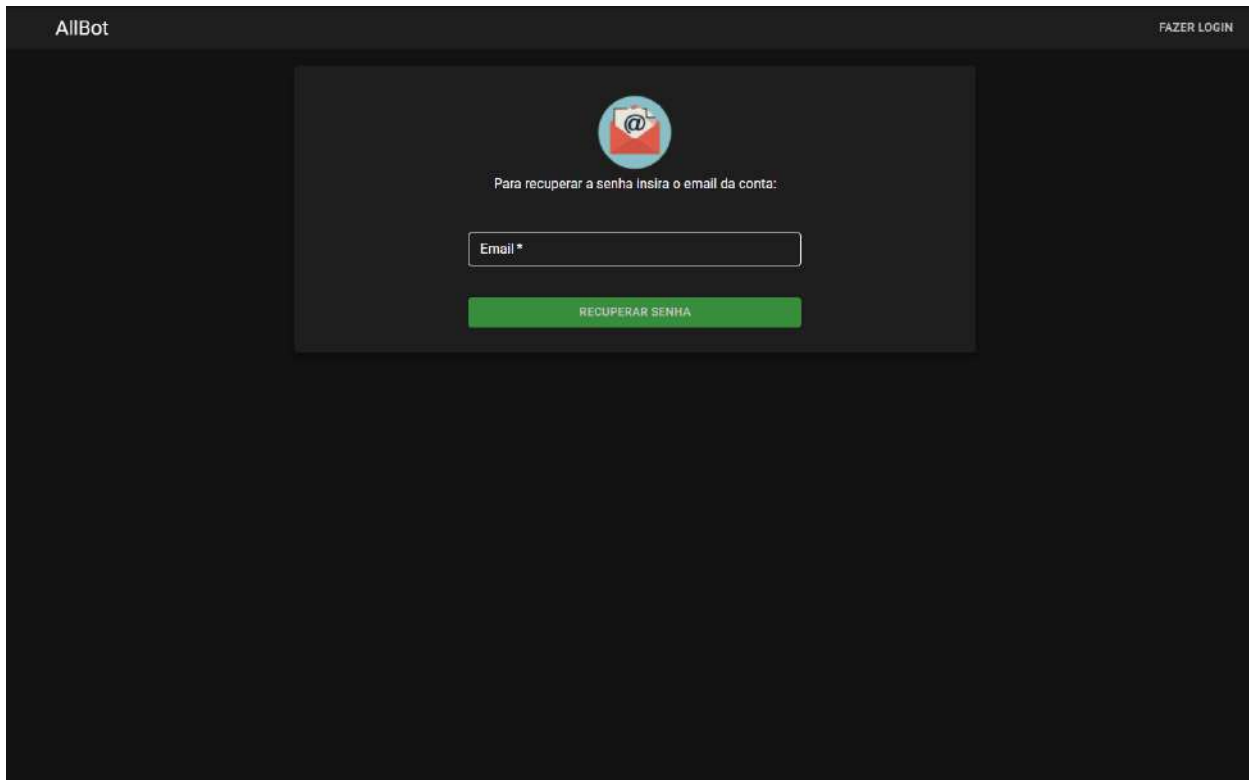


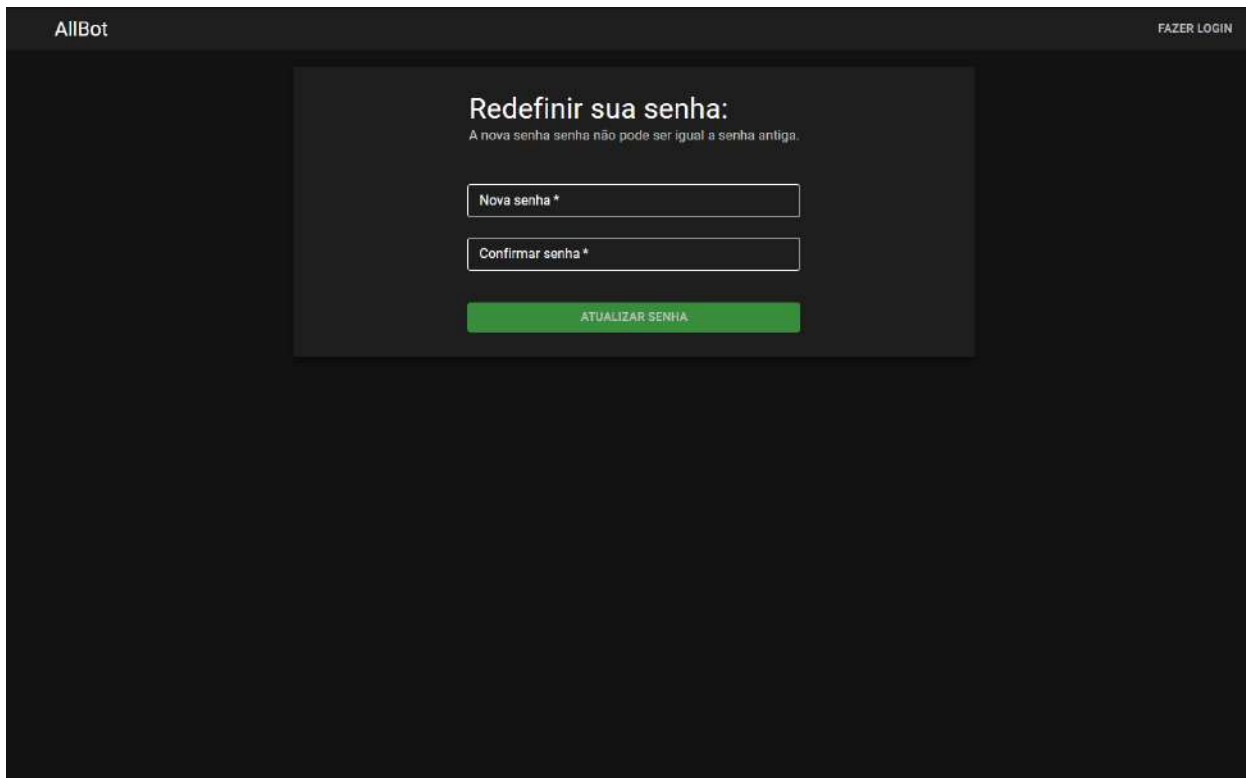
Figura 4.7: Tela de Recuperação de Senha.

Ao informar um email cadastrado no AllBot, o sistema enviará um email de troca de senha, onde o usuário pode acessar o *link* para redefinição de senha. A Figura 4.8 mostra o exemplo desse passo.



Figura 4.8: Email de Redefinição de senha.

Acessando o *link* informado por email, é redirecionado a uma página onde é capaz de inserir uma nova senha para voltar a ter acesso a conta do sistema. como é observado na Figura 4.9.



The screenshot shows a dark-themed web interface for password reset. At the top left, the text "AllBot" is visible, and at the top right, "FAZER LOGIN" is displayed. The main content area is a dark gray box with the following elements:

- Redefinir sua senha:** The main heading.
- A nova senha não pode ser igual a senha antiga.: A note below the heading.
- : A text input field for the new password.
- : A text input field for confirming the new password.
- : A green button to submit the form.

Figura 4.9: Tela Redefinição de Senha.

Após realizar o login, o usuário tem a possibilidade de utilizar das demais funcionalidades da plataforma. Para isso, ele inicia o acesso visualizando a tela inicial, que chamamos painel de controle, exemplificada na Figura 4.10 e na Figura 4.11.

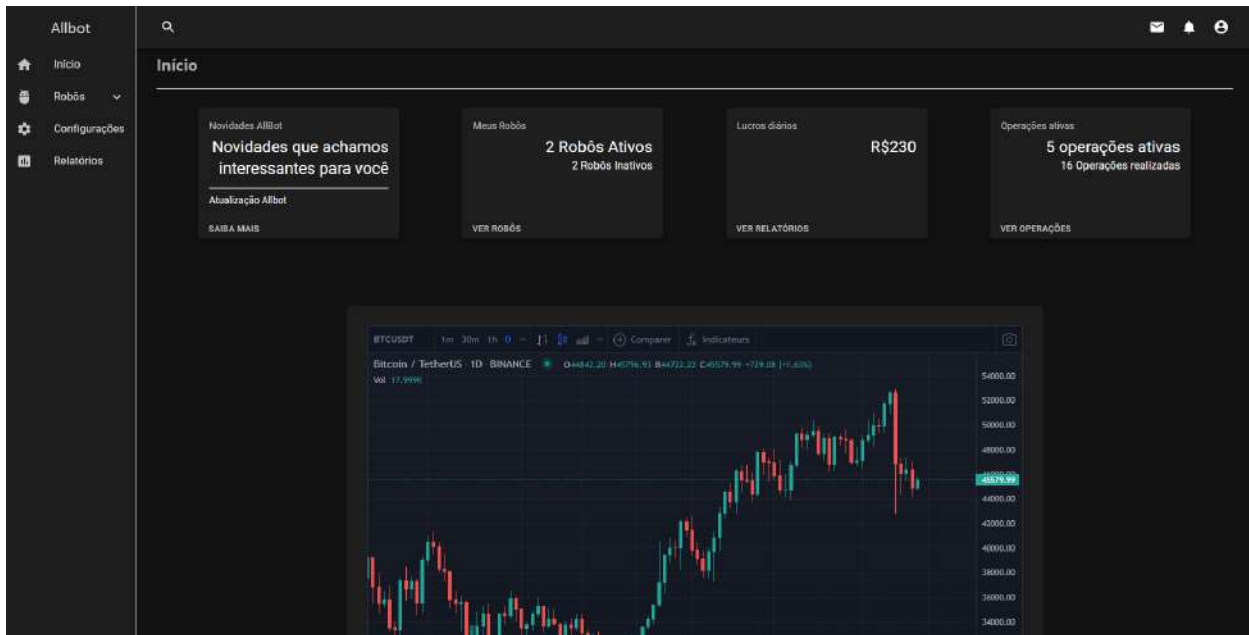


Figura 4.10: Tela do Painel de Controle (parte 1).

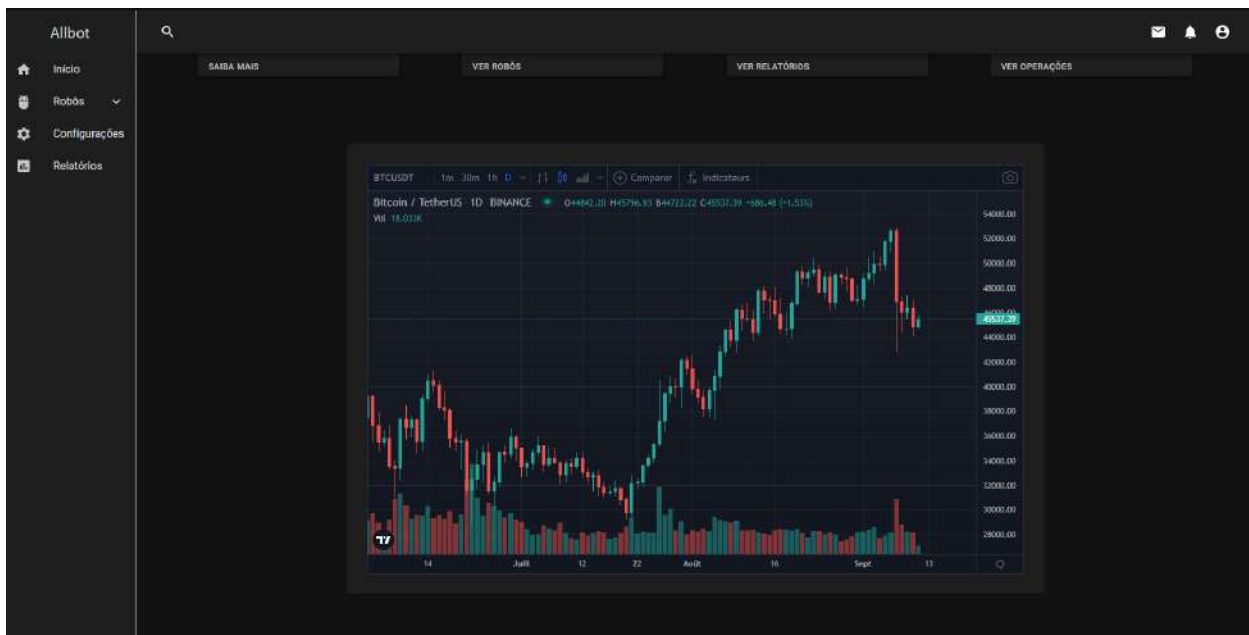


Figura 4.11: Tela do Painel de Controle (parte 2).

O sistema também foi desenvolvido para que haja interações via dispositivos móveis, como é possível verificar nas figuras a seguir.

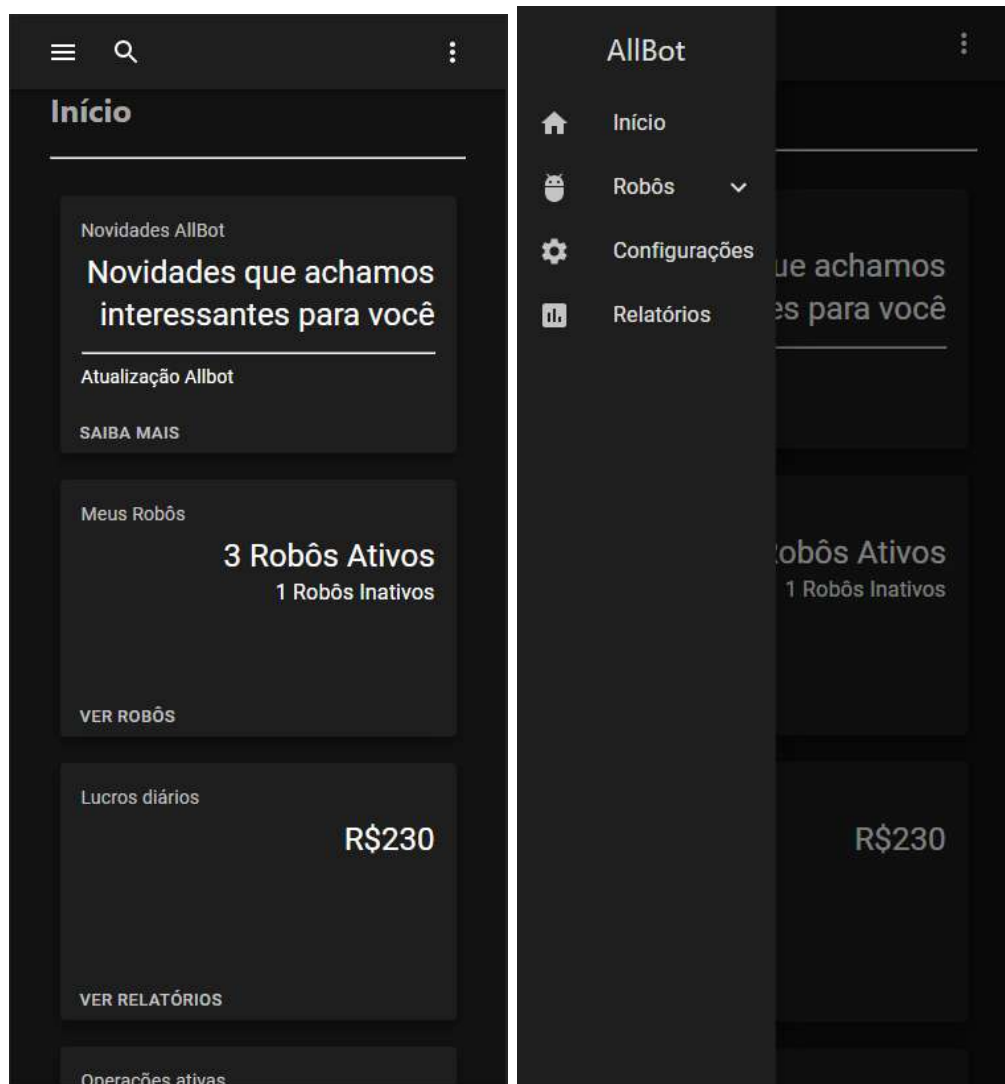


Figura 4.12: Tela do Painel de Controle Versão Mobile.

A plataforma também proporciona para o usuário a possibilidade do mesmo visualizar e alterar os seus dados. Para que essas ações sejam realizadas, o usuário deve acessar a tela de perfil do usuário evidenciada na Figura 4.13.

The screenshot shows the 'Perfil' page of the Allbot application. The page is dark-themed and contains several sections for user information. The 'Informações de usuário' section includes fields for 'Nome completo*' (Ranzeus Narsison Muniz), 'Email*' (zeusmuniz@hotmail.com), 'Data de nascimento' (14-03-1998), 'CPF' (999.999.999-99), and 'Telefone*'. The 'Informações de endereço' section includes fields for 'CEP*', 'Estado*' (Campina Grande), 'Endereço*', and 'Bairro*'. The 'Informações de conta' section shows 'Plano' (Padrão) and 'Duração' (Indefinido). At the bottom, there is an 'Alterar senha' section with 'Senha*' and 'Confirmar senha*' fields, and 'CANCEL' and 'SALVAR' buttons.

Figura 4.13: Tela do Perfil do Usuário.

Na plataforma existe também a funcionalidade de configuração do Usuário, esta, é semelhante à funcionalidade citada anteriormente, porém ela está voltada a dados específicos da configuração de conta do usuário, dados estes chamados, API key e Secret Key, ligados a sua conta na *exchange* (Binance) e de suma importância para que os robôs do usuário possam realizar as negociações no mercado de criptomoedas. Por motivos de segurança, esses dados não são disponibilizados ao usuário em momento algum. Ao informar e salvar os dados, estes irão criptografados para o banco de dados, sendo acessado apenas pelo Serviço robô no servidor. Na figura 4.14 podemos visualizar a tela que permite a utilização da funcionalidade por parte do usuário.

Allbot

Início

Robôs

Configurações

Relatórios

Configurações

Binance

Para que haja uma comunicação com a Binance, nós precisamos de algumas informações.

API Key *

Secret Key *

SALVAR

Notificações

Ative notificações Para ficar por dentro de tudo que o Allbot faz por você.

Telegram

Chat ID Telegram *

SALVAR

Figura 4.14: Tela de Configuração do Usuário.

Também é possível alterar as configurações via mobile:

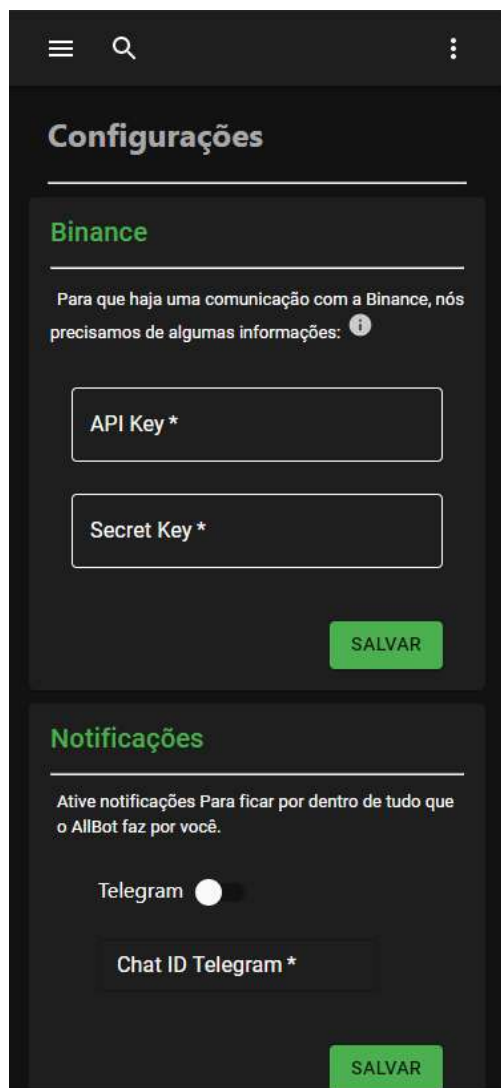
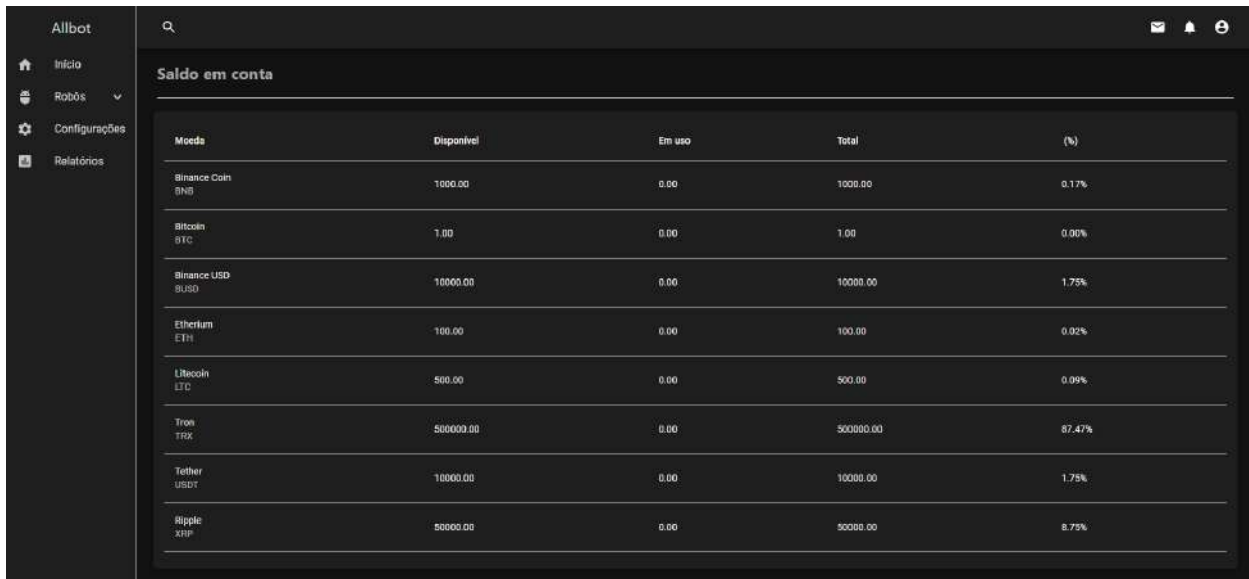


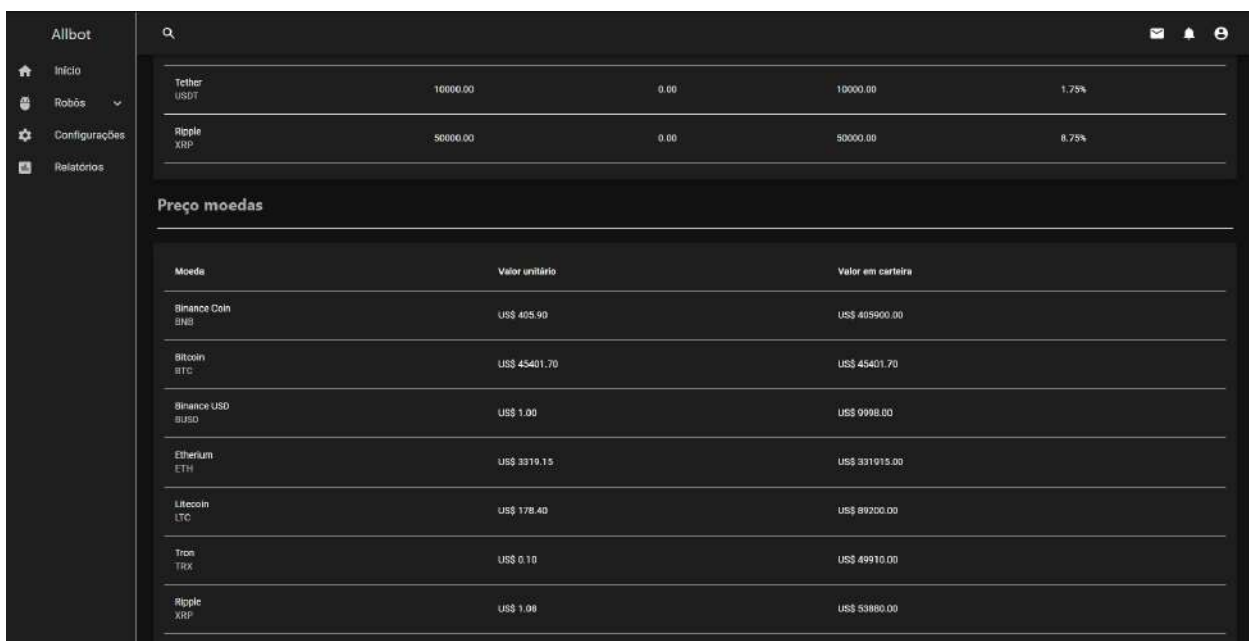
Figura 4.15: Tela de Configuração do Usuário Versão Mobile.

Para que o usuário possa verificar o saldo em sua conta, é disponibilizado a tela de relatórios, onde é capaz de verificar quais moedas estão em sua conta Binance e a quantidade de cada uma delas, como está sendo exemplificado nas Figuras 4.16 e 4.17



Moeda	Disponível	Em uso	Total	(%)
Binance Coin BNB	1000.00	0.00	1000.00	0.17%
Bitcoin BTC	1.00	0.00	1.00	0.00%
Binance USD BUSD	10000.00	0.00	10000.00	1.75%
Etherium ETH	100.00	0.00	100.00	0.02%
Litecoin LTC	500.00	0.00	500.00	0.09%
Tron TRX	500000.00	0.00	500000.00	87.47%
Tether USDT	10000.00	0.00	10000.00	1.75%
Ripple XRP	50000.00	0.00	50000.00	8.75%

Figura 4.16: Tela de Relatórios (parte 1).



Moeda	Valor unitário	Valor em carteira
Tether USDT	10000.00	0.00
Ripple XRP	50000.00	0.00
Binance Coin BNB	US\$ 405.90	US\$ 405900.00
Bitcoin BTC	US\$ 45401.70	US\$ 45401.70
Binance USD BUSD	US\$ 1.00	US\$ 9998.00
Etherium ETH	US\$ 3310.15	US\$ 331015.00
Litecoin LTC	US\$ 178.40	US\$ 89200.00
Tron TRX	US\$ 0.10	US\$ 49910.00
Ripple XRP	US\$ 1.08	US\$ 53880.00

Figura 4.17: Tela de Relatórios (parte 2).

Para visualização dos relatórios em mobile, foi desenvolvido as telas a seguir:

The image displays two side-by-side mobile application screens. The left screen, titled 'Saldo em conta', shows a table of account balances for various cryptocurrencies. The right screen, titled 'Preço moedas', shows a table of current market prices for the same cryptocurrencies. Both screens feature a dark theme and a top navigation bar with a menu icon, a search icon, and a settings icon.

Moeda	Disponível	Em uso	Total
Binance Coin BNB	1000.00	0.00	1000.00
Bitcoin BTC	1.00	0.00	1.00
Binance USD BUSD	10000.00	0.00	10000.00
Etherium ETH	100.00	0.00	100.00
Litecoin LTC	500.00	0.00	500.00
Tron TRX	500000.00	0.00	500000.00
Tether USDT	10000.00	0.00	10000.00

Moeda	Valor unitário	Valor em carteira
Binance Coin BNB	US\$ 405.90	US\$ 405900.00
Bitcoin BTC	US\$ 45401.70	US\$ 45401.70
Binance USD BUSD	US\$ 1.00	US\$ 9998.00
Etherium ETH	US\$ 3319.15	US\$ 331915.00
Litecoin LTC	US\$ 178.40	US\$ 89200.00
Tron TRX	US\$ 0.10	US\$ 49910.00
Ripple XRP	US\$ 1.08	US\$ 53880.00

Figura 4.18: Tela de Relatórios Versão Mobile.

Passando agora para o conjunto de funcionalidades relacionadas aos robôs, temos uma funcionalidade que permite que o usuário visualize seus robôs e o status dos mesmos, esta está exemplificada na Figura 4.19 que é a tela Meus Robôs.

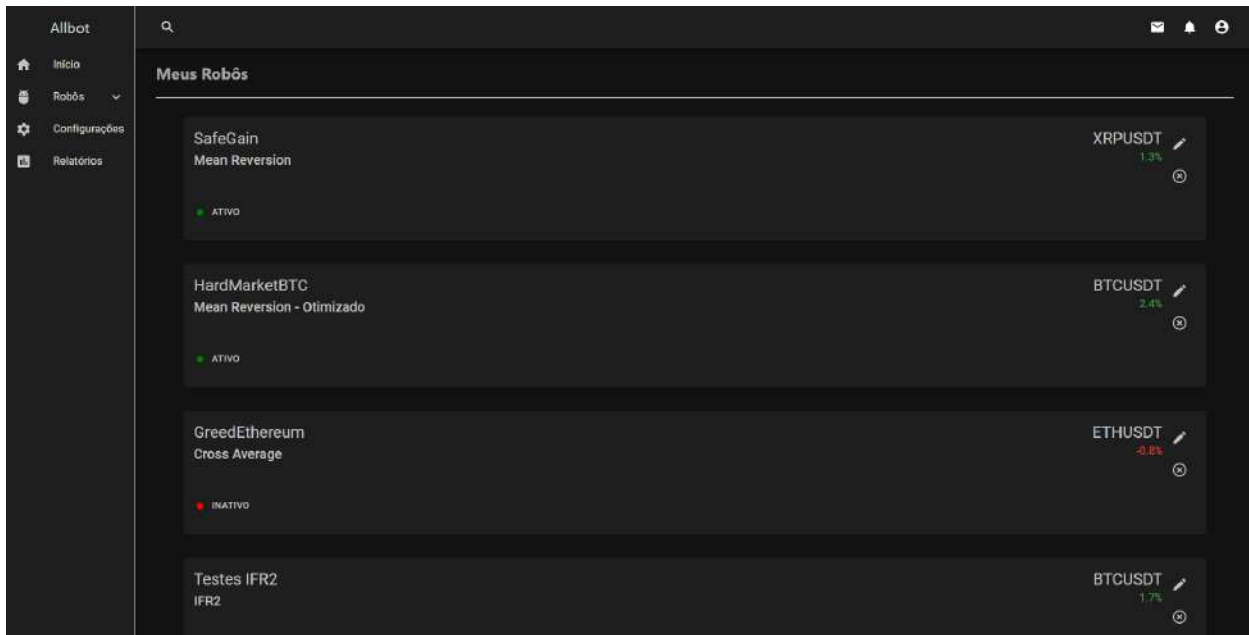


Figura 4.19: Tela dos Robôs do Usuário.

Assim como as outras telas, também é possível gerenciar os robôs via dispositivos móveis.

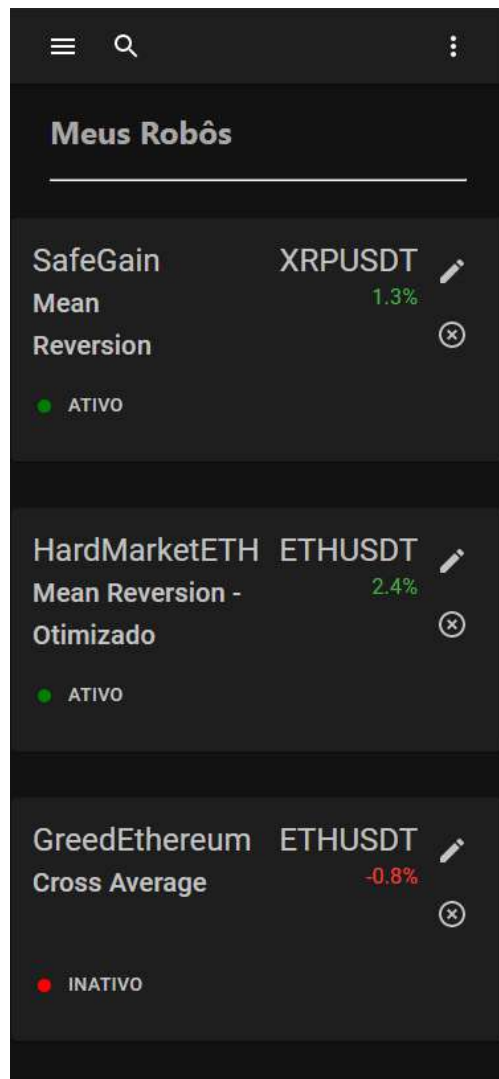


Figura 4.20: Tela dos Robôs do Usuário Versão Mobile.

Continuando no conjunto de funcionalidades citado anteriormente temos a funcionalidade de criação de robôs, essa permite ao usuário criar um robô ao seu modo escolhendo entre as estratégias disponibilizadas pela plataforma, escolhendo o tempo de operação, o ativo e a quantidade (em dólar) do valor no qual o robô tem acesso a fazer operações. Para representar esta funcionalidade, nós temos a Figura 4.21.

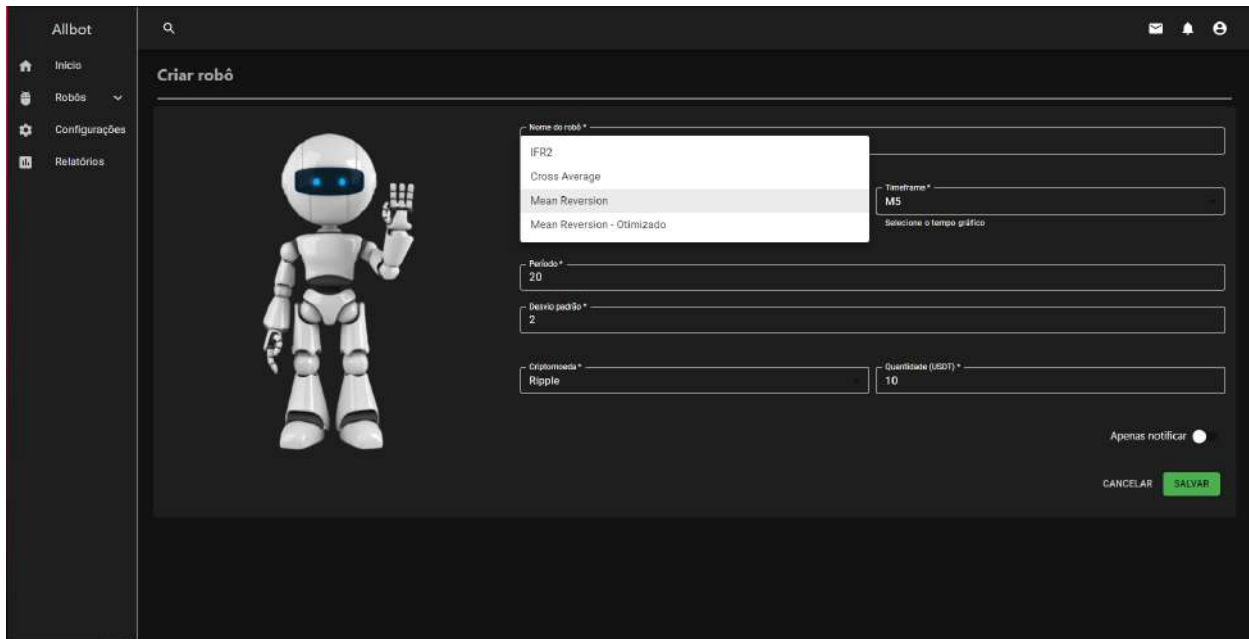


Figura 4.21: Tela de Criação de Robôs.

Para finalizar o conjunto de funcionalidades ligadas aos robôs, essa permite ao usuário editá-lo mantendo sua principal característica, que é a estratégia. Mas, além disso, fornece a possibilidade de realizar os *backtestes* individuais de cada um. Na Figura 4.22 é ilustrada a tela de edição de robô.

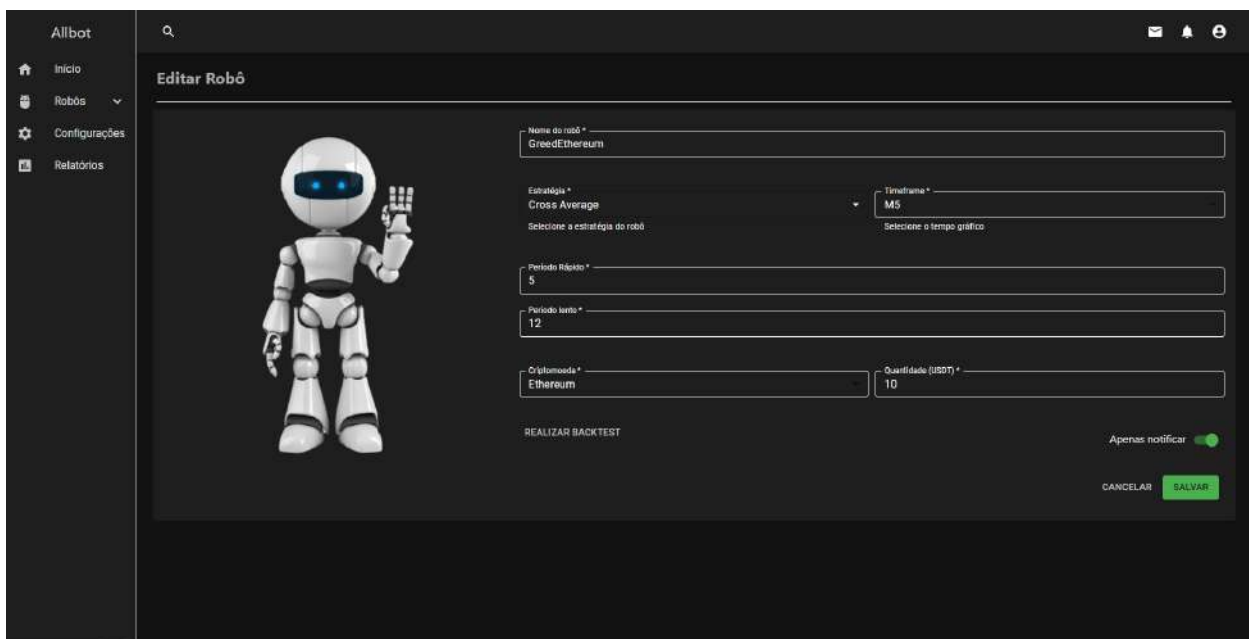


Figura 4.22: Tela de Edição de Robô.

Partindo para as funcionalidades relacionadas a *backtests* de robôs, temos a funcionalidade que exibe o *ranking* de robôs segundo os *backtests* realizados, testes esses que consideram vários robôs, ativos e tempos gráficos. Para acessar esta funcionalidade o usuário pode ir até à tela de Ranking de Robôs, que podemos observar na figura 4.23.

Estratégia robô	Ativo	Timeframe	Quantidade operações	Retorno (%)
CROSSAVERAGE	ETHUSD	4h	247	12515.46
CROSSAVERAGE	ETHUSD	1h	1084	4274.00
CROSSAVERAGE	BTCUSD	4h	242	3640.90
CROSSAVERAGE	ETHUSD	1d	38	3405.07
CROSSAVERAGE	ETHUSD	2h	545	2774.11
CROSSAVERAGE	ETHUSD	8h	132	2639.76
CROSSAVERAGE	ETHUSD	12h	92	2099.55
CROSSAVERAGE	ETHUSD	6h	171	1770.71
CROSSAVERAGE	BTCUSD	5h	168	1662.86
CROSSAVERAGE	BTCUSD	1d	36	1381.58
CROSSAVERAGE	BTCUSD	2h	554	1157.66

Figura 4.23: Ranking de Robôs.

Onde na tabela podemos ver informações básicas dos testes, porém é possível expandir essas informações clicando no botão de expansão definido pelo botão de seta, podemos ver a tabela expandida na figura 4.24.

Estratégia robô	Ativo	Timeframe	Quantidade operações	Retorno (%)					
CROSSAVERAGE	ETHUSD	4h	247	12515.46					
Informações adicionais									
Dia Início	Dia Fim	Duração	Média de duração por trade	Taxa de acerto (%)	Trade de maior duração	Melhor trade (%)	Pior trade (%)	Retorno de Buy & Hold (%)	Média por trade (%)
Thu, 17 Aug 2017 04:00:00 GMT	Wed, 08 Sep 2021 00:00:00 GMT	1482 days 20:00:00	3 days 06:00:00	38.87	17 days 04:00:00	74.22	-12.73	1011.70	1.98
CROSSAVERAGE	ETHUSD	1h	1084	4274.00					
Informações adicionais									
Dia Início	Dia Fim	Duração	Média de duração por trade	Taxa de acerto (%)	Trade de maior duração	Melhor trade (%)	Pior trade (%)	Retorno de Buy & Hold (%)	Média por trade (%)
Thu, 17 Aug 2017 04:00:00 GMT	Wed, 08 Sep 2021 03:00:00 GMT	1482 days 23:00:00	0 days 18:00:00	35.42	3 days 14:00:00	44.80	-12.34	1025.11	0.35
CROSSAVERAGE	BTCUSD	4h	242	3640.90					
CROSSAVERAGE	ETHUSD	1d	38	3405.07					
CROSSAVERAGE	ETHUSD	2h	545	2774.11					
CROSSAVERAGE	ETHUSD	8h	132	2639.76					

Figura 4.24: Ranking de Robôs expandido.

Também está disponibilizado a forma de visualizar o *ranking* de robôs via dispositivos móveis, porém, devido à falta de espaço, as informações foram simplificadas para continuarem legíveis. como é demonstrado pela figura 4.25

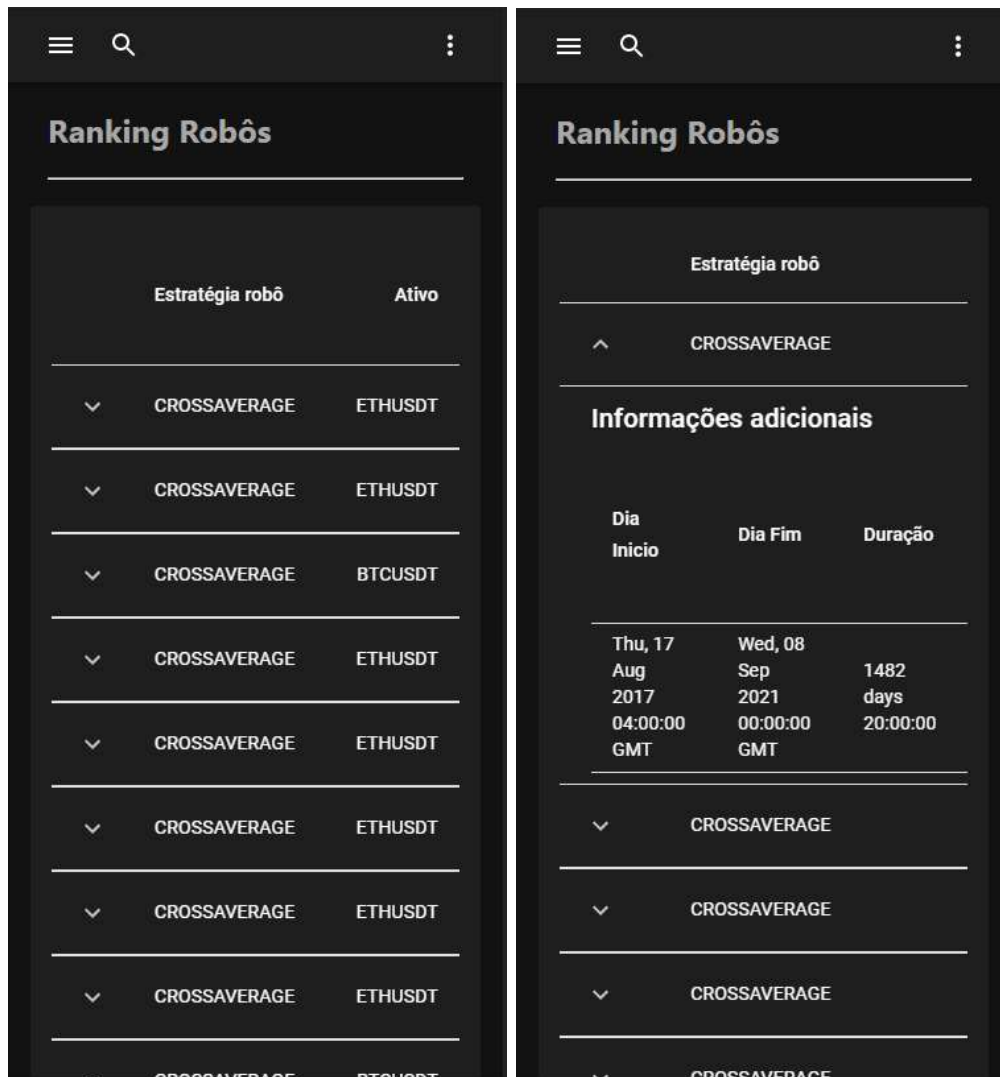


Figura 4.25: Tela de Relatórios Versão Mobile.

Finalizando o conjunto de funcionalidades referentes a *backtests* de robôs, temos a funcionalidade que exibe o *backtest* para um robô específico e para acessar esta funcionalidade o usuário pode utilizar a tela de Editar Robô e clicar no botão "REALIZAR TESTES", cujo resultado está exemplificado na figura 4.26



Figura 4.26: Tela de Backtest de um Único Robô.

Na imagem abaixo é possível visualizar o *bot* de Telegram notificando ao usuário, as ações tomadas pelo robô.



Figura 4.27: *Captura de tela do aplicativo telegram.*

Como resultados do trabalho realizado, evidenciamos a plataforma desenvolvida, em que parte dessa consiste nos dados demonstrados, através de um gráfico e uma tabela, a melhora de um dos robôs após aplicarmos nossa estratégia de otimização utilizando aprendizado de máquina. Consiste também na interface com o usuário, que compõe o serviço Web, mostrando assim as telas nas quais o usuário poderá acessar o sistema e interagir com o serviço robô, como, por exemplo: tela de login, de cadastro de usuário, de criação de robôs e tela de robôs do usuário.

Capítulo 5

Considerações Finais e Sugestões para Trabalhos Futuros

5.1 Considerações Finais

Ao iniciar esse desenvolvimento foi observado haver necessidade de criação de uma plataforma que disponibilize uma forma de testes de atuação dos robôs de investimento, dos quais se utilizam estratégias já consolidadas do mercado, tendo em vista que os estudos realizados não proporcionavam formas de realizar os testes de estratégias desenvolvidas pelos mesmos. É necessário a disposição para que haja a melhoria de estratégias e visualização dos resultados em tempo real, isso até mesmo possibilitaria a entrada de novos usuários ao mercado financeiro.

Assim como foi discutido anteriormente sobre a finalidade desse trabalho, o desenvolvimento de uma plataforma de robôs de investimento foi concluída, de modo que é possível a criação, utilização e realização de *backtests* como visto nos resultados obtidos, além dos resultados dos testes que são realizados em tempo real para obter os resultados dos robôs, possibilitando ao usuário verificar a funcionalidade com dados históricos, configurados pelo próprio usuário.

A plataforma foi desenvolvida com acesso web para possibilitar o uso de todas as suas funcionalidades, o sistema foi desenvolvido de uma forma que permite uma possibilidade de expansão do mesmo, de tal maneira que alterações para melhorias são realizadas com facilidade.

A integração com a Binance foi feita de forma a que todo o desenvolvimento teve como foco a otimização das operações entre os robôs e a *exchange*, assim realizadas de forma rápida proporcionando que nenhum sinal de compra seja perdido.

Os robôs desenvolvidos são baseados em estratégias de análise técnica, e as operações realizadas por eles foram definidas por estratégias previamente validadas, onde os mesmos podem ser totalmente gerenciados pelo usuário através da plataforma, a qual está integrada ao módulo de notificações que mantém o usuário ciente de todas as atuações de seus robôs.

O módulo de *backtest* discutido teve sua principal importância para verificar todos os dados acerca do resultado dos relatórios de execução de um robô, o qual pode ser utilizado com qualquer robô configurado pelo usuário de modo a obter resultados sobre seu desempenho, usando uma ampla base histórica. A plataforma também disponibiliza um ranking de robôs com os resultados de *backtests* dos principais robôs.

Com a finalização do desenvolvimento do AllBot conclui-se que a plataforma atende aos objetivos propostos, proporcionando aos usuários diversos robôs com características distintas que atendem aos interesses dos investidores, dispondo de um conjunto de funcionalidades relacionadas a validação de estratégias através de *backtests* e proporcionando uma ampla capacidade de personalização.

5.2 Sugestões para Trabalhos Futuros

Este trabalho apresenta uma plataforma de robôs para investimento no mercado de criptomoedas. Através do modelo secundário implementado, foi possível perceber uma melhora nos resultados das estratégias.

- **Integração com outras exchanges de criptomoedas:** atualmente a plataforma está integrada apenas a Binance, seria interessante, uma integração com outras gigantes, como Coinbase e NovaDax.
- **Implementação de mais estratégias:** com um maior número de estratégias é possível criar um portfólio de robôs mais diversificado.
- **Criação de módulo de gerenciamento de risco:** Nesse módulo poderia ser implementado funcionalidades referentes a gestão de risco como definir uma perda máxima diária, máximo número de operações por dia e meta de lucro diária.
- **Implementações de funcionalidades para mobile:** Para o uso completo do *mobile*, seria necessário a implementação de algumas funcionalidades para o uso do mesmo, como a utilização do gráfico da tela inicial, que não é possível pelo *mobile* e a realização do *backtest* que também não é possível por dispositivos móveis.
- **Realização de testes de desempenho:** Para avaliar o uso de uma grande quantidade de robôs, e verificar a viabilidade de implementação de estratégias de alta frequência de operações.

Referências Bibliográficas

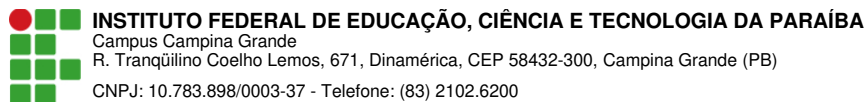
- AXIOS. *Getting Started*. 2021. Available on: <<https://axios-http.com/docs/intro>>. 34
- BACKTESTING. *Backtesting.py*. 2021. Available on: <<https://kernc.github.io/backtesting.py>>. 27
- BINANCE-API. *Binance Public API Connector Python*. 2021. Available on: <<https://binance-connector.readthedocs.io/en/stable/>>. 24
- BitcoinTrade. *Câmbio de Bitcoin*. 2021. Available on: <<https://blog.bitcointrade.com.br/cambio-de-bitcoin/>>. 9
- Bollinger, J. *Bollinger on Bollinger bands*. : McGraw-Hill New York, 2002. 17, 28
- CARVALHO, C. E.; PIRES, D. A.; ARTIOLI, M.; CONTENTO, G. Bitcoin, criptomoedas, blockchain: desafios analíticos, reação dos bancos, implicações regulatórias1. *Anais... Fórum Liberdade Econômica, São Paulo, SP, Brasil*, 2017. 7
- CELERY. *Introduction to Celery*. 2021. Available on: <<https://docs.celeryproject.org/en/stable/getting-started/introduction.html>>. 20, 21
- COINMARKETCAP. *Bitcoin price today*. 2021. Available on: <<https://coinmarketcap.com/currencies/bitcoin/>>. 8
- Connors, L. A.; Alvarez, C. *Short Term Trading Strategies that Work: A Quantified Guide to Trading Stocks and ETFs*. : TradingMarkets Publishing Group, 2009. 30
- Debastiani, C. A. *Análise Técnica de Ações: identificando oportunidades de compra e venda*. : Novatec Editora, 2008. 16
- DUARTE, L. *O que é Node.js e outras 5 dúvidas fundamentais*. 2021. Available on: <<https://www.luiztools.com.br/post/o-que-e-nodejs-e-outras-5-duvidas-fundamentais/#1>>. 34
- Duarte, M. A.; GOMES, S. S.; LIMA, V.; FERRETO, V.; AMBRÓSIO, E. Estudo da estatística de análise temporal com inteligência artificial para aplicação em robôs de investimento no auxílio dos investidores para otimização de ganhos na bolsa de valores de são paulo. *Revista Computação Aplicada-UNG-Ser*, v. 7, n. 1, p. 12–18, 2019. 4
- EDWARDS, R. D.; MAGEE, J.; BASSETTI, W. C. *Technical analysis of stock trends*. : CRC press, 2018. 12
- FLASK. *User's Guide*. 2021. Available on: <<https://flask.palletsprojects.com/en/2.0.x/>>. 20, 21
- Garcia, V. d. S. F. Séries temporais para predição de finanças no contexto de criptomoedas. 2021. 4

- KHAN, M. E.; KHAN, F. et al. A comparative study of white box, black box and grey box testing techniques. *Int. J. Adv. Comput. Sci. Appl*, Citeseer, v. 3, n. 6, 2012. 11
- LEBEUF, C.; STOREY, M.-A.; ZAGALSKY, A. Software bots. *IEEE Software*, IEEE, v. 35, n. 1, p. 18–23, 2017. 10
- LEBEUF, C.; ZAGALSKY, A.; FOUCAULT, M.; STOREY, M.-A. Defining and classifying software bots: A faceted taxonomy. p. 1–6, 2019. 10
- Lichtman, J. K. Automação de investimentos. 2019. 3, 4
- Liu, Y.; Tsyvinski, A. Risks and returns of cryptocurrency. *The Review of Financial Studies*, Oxford University Press, v. 34, n. 6, p. 2689–2727, 2021. 2
- MONGOENGINE. *Mongoengine tutorial*. 2021. Available on: <<http://mongoengine.org/#home>>. 21
- MONGOOSE. *Elegant mongodb object modeling for node.js*. 2021. Available on: <<https://mongoosejs.com/>>. 34
- NAKAMOTO, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. Available on: <<http://www.bitcoin.org/bitcoin.pdf>>. 1
- NUNES, A. d. S.; GARROTE, A. F. M. d. L.; ARAÚJO, M. A. A. d. et al. Utilização de robôs na bolsa de valores. Pontifícia Universidade Católica de Goiás, 2021. 2
- Pardo, R. *The evaluation and optimization of trading strategies*. : John Wiley & Sons, 2011. v. 314. 11
- Prado, M. L. D. *Advances in financial machine learning*. : John Wiley & Sons, 2018. 31
- Raimundo, M. S. *Desenvolvimento de um modelo adaptativo baseado em um sistema SVR-Wavelet híbrido para previsão de séries temporais financeiras*. Tese (Doutorado) — Universidade de São Paulo, 2018. 3
- Rodrigues, D. C. A. *Os efeitos do Algorithmic Trading na previsibilidade dos mercados financeiros*. Tese (Doutorado), 2019. 11
- SICHEL, R. L.; CALIXTO, S. R. Criptomoedas: impactos na economia global. perspectivas. *Revista de Direito da Cidade*, v. 10, n. 3, p. 1622–1641, 2018. 7
- Silva, L. et al. Criptomoedas, moedas digitais e sistemas de pagamentos: desafios e oportunidades para os bancos centrais frente ao surgimento do bitcoin. [sn], 2021. Available on: <<http://rdu.unicesumar.edu.br/handle/123456789/2015>>. 2
- Silva, P.; Vieira, P. Automatização de arbitragem de criptomoedas. *Ciência da Computação-Tubarão*, 2021. 4
- TELEGRAM-API. *Telegram Bot API*. 2021. Available on: <<https://core.telegram.org/bots/api>>. 26
- TRADINGSTRATEGYGUIDES. *Best Crypto Trading Bots 2021 - Automate Your Trades*. 2019. Available on: <<https://tradingstrategyguides.com/crypto-trading-bot/>>. 3
- ULRICH, F. *Bitcoin: A moeda na era digital*. : LVM Editora, 2017. 7

VALENTE, M. T. Engenharia de software moderna (livro digital). 2020. Available on: <<https://engsoftmoderna.info/cap7.html>>. 32

Wilder, J. W. *New concepts in technical trading systems.* : Trend Research, 1978. 18

Wuzu. *Hot wallet ou cold wallet.* 2021. Available on: <<https://wuzu.io/blog/hot-wallet-ou-cold-wallet/>>. 8



Documento Digitalizado Ostensivo (Público)

Trabalho de conclusão de curso

Assunto: Trabalho de conclusão de curso
Assinado por: Guilherme Silva
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Ostensivo (Público)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Guilherme Pereira da Silva, ALUNO (201611250017) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE**, em 23/09/2022 13:46:25.

Este documento foi armazenado no SUAP em 23/09/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 633576
Código de Autenticação: 68e4eecda4

