



FACULDADE DE TECNOLOGIA DE COTIA
TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE
MULTIPLATAFORMA

THARSEO

Automatizador de Estratégias de Negociação em Criptomoedas

ALBERTO RIBEIRO DE OLIVEIRA, ALESSANDRA SANCHES,
CAIO RIBEIRO, FLAVIO DOMINGUES VIEIRA, RICHARD
GUEDES RIBEIRO DA SILVA

Cotia – SP
Junho/2024

FACULDADE DE TECNOLOGIA DE COTIA

THARSEO

Automatizador de Estratégias de Negociação em Criptomoedas

**ALBERTO RIBEIRO DE OLIVEIRA, ALESSANDRA SANCHES,
CAIO RIBEIRO, FLAVIO DOMINGUES VIEIRA, RICHARD
GUEDES RIBEIRO DA SILVA**

Projeto Integrador (A1) entregue à
disciplina de Experiência do Usuário
para o Trabalho de Graduação do curso
de Tecnologia em Desenvolvimento de
Software Multiplataforma da
Faculdade de Tecnologia de Cotia.

**Cotia – SP
Junho/2024**

Sumário

1.Introdução.....	5
2. Desenvolvimento	5
2.1. Objetivos.....	5
2.2 Justificativa	6
2.3. Fundamentação Teórica.....	8
2.4. Elicitação de requisitos	10
2.4.1. Brainstorm.....	10
2.4.2. Gestão do projeto	11
2.4.2.1. Trello (Kanban)	11
2.4.2.2. Scrum	12
2.4.2.3. Sprint (Reuniões semanais)	12
2.4.3. Requisitos dos usuários	13
2.4.4. Requisitos funcionais.....	13
2.4.5 Requisitos não funcionais	15
2.4.6. Regras de Negócios	16
2.5. Modelagem do banco de dados.....	17
2.5.1. Modelo do Banco relacional	18
2.6. Desenvolvimento do sistema	19
2.6.1. Tecnologias utilizadas	19
2.6.1.1. FIGMA	19
2.6.1.2. Javascript	23
2.6.1.3. React.....	24
2.6.1.4. Java	25
2.6.1.5. Spring.....	27
2.6.1.6. API Binance	29
2.6.1.7. MySQL.....	30
2.6.1.8.Websocket	31
2.6.1.9.Git.....	31
2.6.1.10.Docker	32
2.6.1.11.SonarCloud	32
2.6.1.12.Azure.....	34
2.6.1.13. Trello	36
2.6.1.14. Kanban	37
2.6.1.15.Scrum	38
2.6.1.16.Clarity	40

2.6.1.17.Owasp.....	41
2.6.2. Testes de UI e UX realizados	44
2.6.2.1. Desenvolvimento de Persona	44
2.6.2.2. Jornada do Usuário	45
2.6.2.3. Desenvolvimento de Questionário	45
2.6.2.3. Teste qualitativo em relação a tela desenvolvida	50
2.6.3. Pipeline e Testes automatizados.....	57
2.6.4. Conclusão	60
Bibliografia	61

1.Introdução

O Tharseo, nossa proposta de aplicação para automatizar estratégias de negociação em criptomoedas, foi criado com o objetivo primordial de automatizar negociações no mercado de criptomoedas, oferecendo aos usuários uma abordagem estratégica e eficiente permitindo que o usuário possa seguir sua rotina enquanto investe no mercado financeiro no segmento de criptomoedas.

A crescente popularidade e volatilidade do mercado de criptomoedas têm suscitado a necessidade de ferramentas inovadoras que possam auxiliar os traders na otimização de suas operações comerciais, maximizando os ganhos potenciais e gerenciando os riscos de forma inteligente. Nesse contexto, o Tharseo surge como uma solução poderosa, proporcionando uma plataforma robusta e resiliente para a automação de negociações.

Através dessa, buscamos fornecer uma visão abrangente das características, funcionalidades e processos envolvidos na utilização do Tharseo. Serão abordados aspectos como a necessidade do usuário. Buscamos dar ênfase a experiência única do usuário, explorando através de ferramentas de UX e UI, o desenvolvimento em que possamos trazer uma melhor usabilidade.

Adicionalmente, abordaremos a problemática que motivou a criação do aplicativo, destacando os desafios enfrentados pelos traders no mercado de criptomoedas e a necessidade de soluções inovadoras para enfrentá-los. Através das personas e pesquisas de campo, mostraremos os resultados alcançados e a definição de estratégias, funcionalidades e prototipação definidas através de pesquisas de campo.

2. Desenvolvimento

2.1. Objetivos

O objetivo do presente trabalho consiste na concepção de uma aplicação destinada a viabilizar e incentivar o usuário a empreender e investir no mercado de Crypto moedas, visando a automação do processo de compra e venda, gerando lucro e satisfação ao cliente, otimizando o seu tempo para que o mesmo tempo que seria gasto avaliando e analisando o mercado para que esse tempo seja aproveitado com outras atividades, gerando satisfação e lucratividade ao usuário.

A aplicação encontra sua motivação primordial no público em geral pois muitas das pessoas tem como dificuldade para investimento o tempo e a falta de conhecimento de como operar, o Tharseo tem como objetivo automatizar o processo para os que não tem tempo para dedicar ao mercado financeiro, mas também dando suporte aos leigos que não conhecem do mercado, gerando lucro e satisfação a todo tipo de usuário, desde o que domina o mercado até o mais leigo.

A proposta do grupo consiste no desenvolvimento de um aplicativo que compre uma fração das criptos à partir de um pré-set na aplicação e após determinada valorização o sistema faça a sua venda de forma automatizada rendendo lucro à partir da diferença do valor de compra para o valor da venda, sendo destinado do lucro um percentual simbólico a empresa Tharseo, e o demais ao seu usuário.

Com uma ampla adesão e participação, almejamos obter destaque não apenas nacional, mas também internacional, contribuindo significativamente para que as pessoas se tornem independentes no seu planejamento de investimentos e consequentemente promovendo a marca Tharseo.

2.2 Justificativa

Segundo a ANBIMA(Associação Brasileira das Entidades dos Mercados Financeiro e de Capitais), no ano de 2023/2024 os investimentos dos brasileiros (pessoas físicas), somaram sucessivamente no mês de Setembro 5,5 trilhões, que é equivalente a um aumento de 9,7% em relação aos anos anteriores, a observação feita engloba os segmentos de varejos tradicionais, varejo de alta renda e private.

Foi observado que mais da metade dos investimentos dos brasileiros estavam aplicados em títulos e valores mobiliários, além de CDB's, e fundos de renda fixa.

Como podemos observar o público brasileiro está em crescente busca por investimentos e formas alternativas de renda além das atuais usuais, um dos mercados com maior liquidez e potencial de rendimentos que temos hoje são as crypto moedas, estas que tem valor variável, porém não gera a mesma insegurança que ativos e ações que podem despencar com mal planejamento das pessoas que dirigem a empresa ou por dividas que vem a tona e pegam a todos desprevenidos, Olhando para o atual potencial e para a crescente busca por investimentos dos brasileiros desenvolvemos a idéia do Tharseo de modo a facilitar a vida dos que não tem o tempo e nem o Knowhow no mercado de compra e venda de crypto moedas, onde a aplicação faz a compra e venda de forma automatizada a partir de um pré-set, e realiza a venda a partir do mínimo lucro também setado no pré set automatizando o processo e otimizando o tempo dos seus usuários.

Público-alvo:

População em geral, grandes investidores, leigos, jovens investidores, investidores experientes etc.

Problema:

Conforme matéria ANBIMA e Datafolha:

“A 7ª edição do Raio X do Investidor Brasileiro, pesquisa que realizamos em parceria com o Datafolha, mostra que a caderneta de poupança mantém o posto de investimento mais conhecido e mais utilizado do país. Um quarto da população afirma ter recursos aplicados no produto. Os títulos privados e as moedas digitais, entretanto, vêm ganhando espaço e cresceram na preferência das pessoas pelo segundo ano consecutivo: já fazem parte das carteiras de 5% e de 4% dos brasileiros, respectivamente.

“Apesar de a poupança ainda ser a opção mais lembrada e acessada, vemos ao longo dos anos uma proporção cada vez maior de pessoas mencionando e investindo em outros produtos. É, inclusive, uma tendência maior entre a população mais jovem. Muitos fatores têm contribuído para isso, como a procura de investidores por aplicações mais rentáveis em momentos de queda de juros; a sofisticação do nosso mercado financeiro; e o fenômeno dos influenciadores digitais, que popularizaram a conversa sobre investimentos nas redes sociais”, afirma Marcelo Billi, nosso superintendente de Sustentabilidade, Inovação e Educação.”

Como podemos observar através da matéria, a poupança é o investimento que tem maior índice de investimento até porque é o mais conhecido, estudos também apontam que a maior porcentagem dos que investem em crypto são os jovens que optam por investir em plataformas digitais, além de que a categoria de pessoas que mais investem são os de categoria A/B, tal índice tem relação com a falta de conhecimento dos mais velhos, ou da classe inferior a A/B, além da dificuldade de interação com os sistemas digitais. Abaixo relatórios onde tais observações e conclusões foram possíveis:



Conhece algum tipo de instituição financeira:



	População	Classe A/B	Classe C	Classe C/D
Conhece algum tipo de instituição financeira	89%	95%	91%	81%
Conhece bancos tradicionais	78%	83%	79%	72%
Conhece bancos digitais	33%	46%	34%	19%
Possui conta em instituição financeira	85%	96%	89%	70%
Possui conta em banco tradicional	69%	79%	72%	57%
Possui conta em banco digital	40%	54%	42%	25%

*Tabela completa está no relatório da pesquisa



14%

da população usa apps de apostas (bets)



40%

dessas pessoas encaram as apostas como uma chance de ganhar dinheiro rápido em momentos de necessidade

22%

dessas pessoas consideram as apostas online um tipo de investimento financeiro

Olhando para o cenário apresentado na pesquisa desenvolvemos o Tharseo, na busca de gerar acessibilidade ao investimento para diversas classes e idades, gerando o conhecimento a moedas digitais para investimentos.

2.3. Fundamentação Teórica

O “inverno crypto” pode até assustar os que acham que a criptoeconomia se resume a criptomoedas, mas nos bastidores há um mercado trilionário de DeFi crescendo com apoio das tecnologias de blockchain, que pode representar 10% do PIB global em 2030 - US\$ 16,1 trilhões, para sermos mais precisos - mesmo com as diferenças regulatórias globais (ou a ausência delas) e a redução de investimentos de risco nessas iniciativas.

Esse pode ser um bom resumo para o estudo recém-publicado pelo Boston Consulting Group (BCG), em parceria com a exchange ADDX, de Singapura, que aprofunda

o entendimento das implicações da tokenização de ativos on-chain (gravadas na blockchain) para o crescimento das Finanças Descentralizadas, sua diferenciação do fracionamento de ativos na economia tradicional, as oportunidades abertas para a tokenização de ativos “ilíquidos” e o que é preciso fazer para alavancar seu crescimento.

O relatório - “Relevance of on-chain asset tokenization in ‘crypto winter’” - é assinado pelo diretor administrativo do BCG, Sumit Kumar, o cofundador do ADDX, Darius Liu, e pelos consultores Rajaram Suresh, Bernhard Kronfellner e Aaditya Kaul. Segundo os autores, “uma grande parte da riqueza do mundo hoje está bloqueada em ativos ilíquidos”. A tokenização (transformação em ativos digitais fracionáveis) pode desbloquear esses ativos, contornando as barreiras tradicionais de negociação, gerando alternativas de comercialização fora das práticas tradicionais, baixando o ticket de entrada e abrindo espaço para novos investidores e modalidades de negócio.

Em 2021, a tokenização de ativos representou US\$ 2,3 bilhões, e deve chegar a US\$ 5,6 bilhões em 2026, segundo o estudo. Ao incluir bens ilíquidos, daria saltos trilionários nas DeFi (veja gráfico acima). Na lista de ativos ilíquidos, o relatório menciona imóveis, recursos naturais, terrenos, commodities, infra-estrutura pública como minas e portos, arte, carros antigos e bebidas raras, que se somam a ações pré-IPO, dívidas privadas, receitas de pequenas e médias empresas, fundos privados e títulos de atacado, entre outros.

As razões para a falta de liquidez desses ativos incluem ticket médio alto para a entrada de investidores em massa, acesso restrito a grupos de elite (no caso de obras de arte e carros antigos), obstáculos regulatórios, entre outros, que dificultam a sua compra ou negociação.

O fracionamento de ativos é uma prática adotada no mercado tradicional, principalmente nos bens patrimoniais e imobiliários - por exemplo, fundos de investimento imobiliário (FIIs) e ETFs. Estão presentes nos mercados públicos, eficientes e com alta tecnologia, mas não nos mercados privados, muitas vezes manuais, lentos, e com altos custos gerais, aponta o relatório.

A tokenização de ativos on-chain nas DeFi (gráfico acima) contorna as barreiras para os ativos ilíquidos “ajudando a reimaginar o processo de encontrar e combinar investidores com oportunidades de investimento”, aponta o estudo. O “minting” (emissão) de tokens representando frações digitais de ativos físicos gravadas em um livro razão distribuído (DLT ou blockchain) oferece garantia de propriedade transparente e imutável.

Baseado no potencial da tendência de digitalização, e o aumento a cada dia da aderência da população em investimentos e carteiras digitais de crypto que o projeto Tharseo ganhou força e engajamento, pois o seu potencial de aderência com o futuro dos investimentos é grande.

2.4. Elicitação de requisitos

“Elicitação de requisitos é uma fase do projeto onde são extraídas informações do cliente sobre o que ele deseja que seja construído. É a fase em que o profissional de TI entende a necessidade do cliente e o orienta. É o momento de conversa com o usuário, de sentimento sobre o que este espera que seja entregue a ele.”

(DEVMEDIA, 2022).

Adentramos em uma das fases fundamentais do projeto, explorando a importância significativa e a facilitação do seu desenvolvimento, juntamente com o potencial de investimento bem-sucedido, através de uma elicitação de requisitos devidamente estruturada.

A dificuldade inicial identificada reside na adesão e motivação dos usuários em aderirem ao Tharseo seria a incerteza e insegurança a princípio de confiarem o seu dinheiro em um sistema totalmente novo, será preciso gerar pontos de identificação do projeto com o usuário e a sociedade de modo a gerarem confiança e segurança a todos que aderirem a aplicação.

Para superar essa barreira, implementamos a técnica de Brainstorming. Essa abordagem permitiu a coleta de uma variedade de opções provenientes dos diversos membros do grupo, sem avaliações críticas nesta fase inicial. O objetivo foi obter um conjunto abrangente de possíveis requisitos para subsidiar as etapas subsequentes

2.4.1. Brainstorm

“O Brainstorming é uma dinâmica de grupo que é usada em várias empresas como uma técnica para resolver problemas específicos, para desenvolver novas ideias ou projetos, para juntar informação e para estimular o pensamento criativo.” (Significados,2022).

BRAINSTORMING		
19/09/2023 à 23/09/2023	Entre os intervalos das aulas	Presencial
Participantes		
Alberto Ribeiro		
Alessandra Sanches		
Caio Ribeiro		
Flávio Domingues		
Richard Guedes		
Assunto: Elicitação de Requisitos Funcionais, Não-Funcionais, Regras de Negócios		
O que foi feito:	Em pequenas reuniões, fizemos o levantamento dos requisitos funcionais, não	

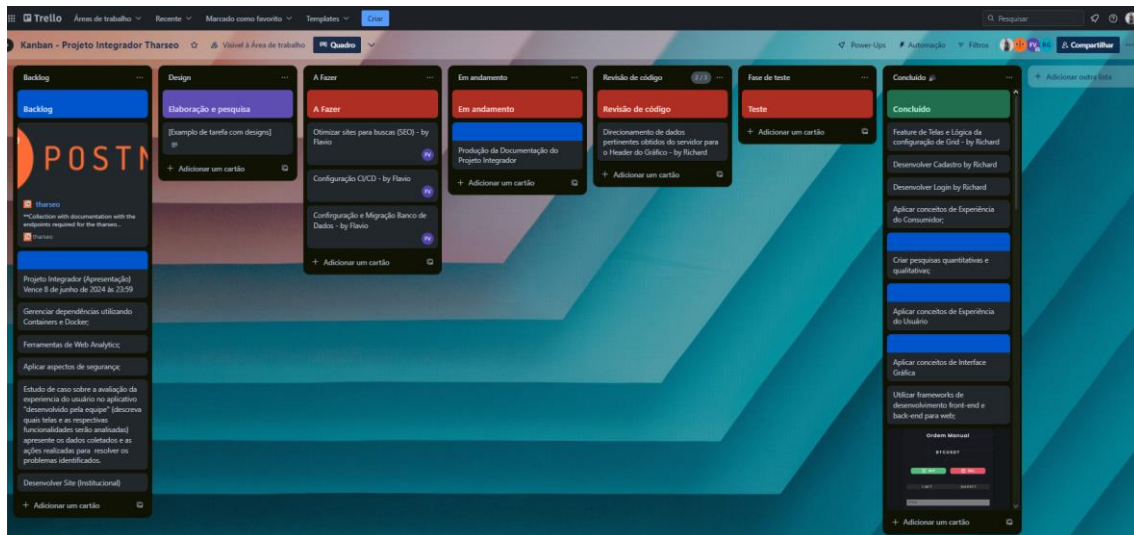
	funcionais e regras de negócio baseadas na necessidade da demanda estudada.
Como foi feito:	Foi feito de modo a pensar em tornar atrativa a aplicação para a sociedade, independente da idade ou classe social.
Porque foi feito:	Devido ao potencial atual e futuro de lucratividade com as crypto moedas, sendo um assunto muito tratado entre diversos públicos e mídias como tenência futura com a digitalização das moedas e bens.
Onde foi feito:	Polo da Fatec em Cotia/SP.
Quem fará:	Os alunos integrantes do grupo.

2.4.2. Gestão do projeto

2.4.2.1. Trello (Kanban)

O Trello é uma plataforma de gerenciamento de projetos que utiliza um sistema de quadros, listas e cartões para organizar tarefas e colaborar em equipes. Lançado em 2011, o Trello rapidamente se tornou uma ferramenta popular devido à sua simplicidade e eficácia. Seu design intuitivo baseado em quadros permite que os usuários visualizem e controlem o progresso de suas tarefas de forma clara e interativa. Com a capacidade de personalizar os quadros, adicionar listas e atribuir cartões a membros da equipe, o Trello se destaca como uma solução versátil para gerenciamento de projetos em diversos contextos.

O Kanban é um método visual de gerenciamento de processos originado no Japão e popularizado pela Toyota nas décadas de 1940 e 1950. Essa abordagem se baseia em cartões (kanbans) que representam tarefas ou itens em um quadro visual. O objetivo principal do Kanban é otimizar a eficiência, minimizar o desperdício e melhorar a visibilidade do processo de produção ou gestão de projetos. As etapas do fluxo de trabalho são representadas em colunas no quadro, permitindo que os membros da equipe visualizem facilmente o status de cada tarefa. O Kanban é conhecido por sua flexibilidade e adaptabilidade, tornando-se uma ferramenta valiosa em ambientes ágeis e colaborativos.



2.4.2.2. Scrum

O Scrum é um dos frameworks mais populares no mundo ágil, oferecendo uma abordagem flexível e colaborativa para o desenvolvimento de projetos complexos. Desenvolvido inicialmente para a gestão de projetos de software, o Scrum tem se expandido para diversas áreas, tornando-se uma metodologia amplamente adotada em diversos setores.

Central para o Scrum está o conceito de iteração e colaboração contínua. Os projetos são divididos em ciclos de desenvolvimento chamados "sprints", geralmente com duração de duas a quatro semanas. Cada sprint envolve um planejamento detalhado, seguido pela execução e revisão, permitindo adaptações rápidas às mudanças e feedback contínuo.

O Scrum inclui papéis claramente definidos, como o Scrum Master, responsável por garantir a aplicação correta do Scrum, o Product Owner, encarregado de definir e priorizar os requisitos do projeto, e a equipe de desenvolvimento, responsável por entregar as tarefas definidas durante o sprint.

As cerimônias do Scrum, como as reuniões diárias ("daily stand-ups") e as revisões de sprint, promovem a comunicação eficaz e a transparência, essenciais para o sucesso do projeto. Ao adotar princípios como a entrega incremental de valor, o Scrum oferece uma estrutura adaptável e eficiente para equipes que buscam maior flexibilidade e capacidade de resposta em um ambiente em constante evolução.

Definimos como Product Owner o integrante Richard e o Scrum Master o integrante Alessandra, os quais direcionaram o andamento do trabalho e dirigiam as reuniões semanais verificando em que ponto o trabalho estava e as tarefas necessárias a serem feitas para o bom andamento do trabalho.

2.4.2.3. Sprint (Reuniões semanais)

O termo "sprint" é uma peça fundamental na metodologia ágil, especialmente no contexto do Scrum, um framework popular para o desenvolvimento de software e gerenciamento de projetos. Uma sprint é um período definido, tipicamente curto, que varia de uma a quatro semanas, durante o qual uma equipe se compromete a completar um conjunto específico de tarefas ou entregáveis.

Essas sprints são marcadas por uma intensa colaboração e foco, visando a criação de incrementos tangíveis e funcionais do produto ao final de cada ciclo. Durante a sprint, a equipe realiza reuniões diárias conhecidas como "daily stand-ups" para discutir o progresso, identificar obstáculos e ajustar estratégias conforme necessário.

O conceito de sprint proporciona uma abordagem iterativa e incremental ao desenvolvimento de projetos, permitindo que as equipes respondam rapidamente às mudanças nos requisitos ou nas prioridades do cliente. Essa prática enraizada na metodologia ágil promove a transparência, a adaptação contínua e a entrega de valor de forma consistente ao longo do ciclo de vida do projeto.

Nossas Sprints tinham como objetivo monitorar o andamento do projeto distribuindo funções e tarefas para os mais adiantados, prestando suporte aos que estavam com dificuldades, implementando as partes prontas e realizando testes, e claro documentando cada parte.

2.4.3. Requisitos dos usuários

Sendo assim estruturamos e definimos os requisitos de usuários listados abaixo:

“Requisitos de usuários são requisitos de mais alto nível, escritos por usuários, normalmente em linguagem natural e sem entrar em detalhes técnicos.” (Engenharia de Software Moderna, 2022)

REQUISITOS DE USUÁRIOS		
Identificador	Descrição	Prioridade
UR01	Eu como usuário deverei ter acesso a cotação de cada Crypto em tempo real.	Must Have
UR02	Eu como usuário deverei ter a minha carteira digital com os meus ativos	Must Have
UR03	Eu como Usuário deverei ter acesso as moedas disponíveis para compra e venda.	Must Have
UR04	Eu como usuário deverei ter acesso ao histórico de transações do dia com meus lucros.	Must Have

2.4.4. Requisitos funcionais

“Os requisitos funcionais são todos os problemas e necessidades que devem ser atendidos e resolvidos pelo software por meio de funções ou serviços”. (CUNHA, 2022).

Baseados nos requisitos de usuário, gera-se através de filtros e estudos os requisitos funcionais que irão dar base ao desenho e arquitetura do sistema e aplicativo. A importância da estruturação, reforça que o planejamento de qualquer projeto evita perda de tempo, foco e desgaste durante a sua execução. Utilizando o método Moscow para definir a prioridade dos itens dos requisitos funcionais em questão, com o complemento

da ferramenta 5wh1 para que possamos definir de forma estruturada o papel e tarefa de cada requisito. A seguir a documentação dos requisitos funcionais:

REQUISITOS FUNCIONAIS		
Identificador	Nome	Prioridade
#RF01	Lista de criptos e seus valores de compra	Must Have
Descrição	O sistema deverá apresentar cada Crypto disponível para negociação e seus valores atuais	
Entradas e Pré-condições	O Usuário para poder operar deverá ter o seu cadastro efetuado e deverá estar logado no sistema.	
Saídas e Pós-Condições	O usuário conseguirá setar o sistema com o valor de compra e após quanto deverá ser vendido após alcançado determinado valor	
Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário irá fazer login no sistema; 2. O sistema retornará o menu principal com a carteira e principais moedas do usuário; 3. O usuário selecionará qual a crypto que ele vai negociar, seu valor de compra e a qual movimentação deverá ser feita a venda. 4. O sistema vai executar com as compras e vendas de acordo com o que foi pré-setado. 	
Identificador	Nome	Prioridade
#RF02	Carteira com valores disponíveis para compra e venda das Cryptos	Must Have
Descrição	O Usuário devera a partir de determinada tela ter acesso a como se fosse um perfil, e nele deve estar apresentado a sua carteira com o valor disponível para negociar em compra e venda de Cryptos.	
Entradas e Pré-condições	O usuário deve cadastrar seus dados; O usuário deve depositar uma quantia mínima para conseguir operar com a compra e venda das cryptos; O usuário deve estar conectado na internet para atualização do mercado de cryptos em tempo real.	
Saídas e Pós-Condições	O aplicativo retornará uma atualização de valores em carteira e das cryptos antes, durante e após as transações;	
Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário deve logar no sistema; 2. O usuário deverá acessar a sua carteira, caso haja valor disponível em carteira, se não o usuário também deverá realizar um depósito; 3. O sistema deve retornar os valores em carteira do usuário. 	
Identificador	Nome	Prioridade

#RF03	Acesso a moedas para compra e venda	Must Have
Descrição	O usuário ao entrar na aba de cryptos, a princípio ela deverá disponibilizar uma aba de pesquisa e as principais cryptos para serem negociadas.	
Entradas e Pré-condições	O sistema deverá captar em tempo real as cryptos e as suas cotações a partir da API da binance.	
Saídas e Pós-Condições	Após a busca o sistema deverá apresentar a crypto e a opção de favoritá-las, para não gerar mais a necessidade de pesquisa e para que as mesmas apareçam logo que logado.	
Fluxo de Eventos	<ol style="list-style-type: none"> 1. Após processo de login o usuário deverá acessar a zona de cryptos. 2. O sistema irá retornar a lista de cryptos, seus valores e uma barra para busca de diversas cryptos. 	
Identificador	Nome	Prioridade
#RF04	Histórico de transações	Must Have
Descrição	O sistema deverá apresentar o histórico de transações, com a possibilidade de filtrar por período das transações.	
Entradas e Pré-condições	O usuário deverá acessar o aplicativo; O usuário deverá se logar no sistema; O usuário ao acessar o menu principal deverá acessar a aba de históricos das transações.	
Saídas e Pós-Condições	Ao acessar a aba de históricos o sistema deverá retornar as transações com data e valor de lucro.	
Fluxo de Eventos	<ol style="list-style-type: none"> 1. O usuário deve acessar o aplicativo. 2. O usuário deve logar no aplicativo; 3. O usuário deverá acessar a aba de históricos. 	

2.4.5 Requisitos não funcionais

Após definidos os requisitos funcionais, daremos início ao levantamento dos requisitos não funcionais, partindo da mesma forma de estruturação:

“Os requisitos não funcionais são todos aqueles relacionados à forma como o software tornará realidade os que está sendo planejado. Ou seja, enquanto os requisitos funcionais estão focados no que será feito, os não funcionais descrevem como serão feitos. ” (CUNHA, 2022).

REQUISITOS NÃO FUNCIONAIS	
Identificador	Descrição
RNF01	O aplicativo terá como padrão as cores.
RNF02	O aplicativo terá um logotipo.
RNF03	O aplicativo possuirá tela de login e cadastro
RNF04	O aplicativo terá como público-alvo o público em geral, investidores ou não, iniciantes ou experientes.
RNF05	O sistema deverá ter a possibilidade de compartilhamento do aplicativo para indicar a mais usuários

2.4.6. Regras de Negócios

“Como o próprio termo sugere, são regras que servem para definir ou restringir alguma ação nos processos de sua empresa. São declarações que irão descrever como determinadas operações devem ser realizadas e se há algum limite que precisa ser aplicado.” Heflo, 2022

A seguir listaremos as definições das regras de negócio, são elas que manterão o projeto em uma linha de raciocínio para que possamos desenvolver o projeto com o foco nos Requisitos fornecidos pelos usuários.

REGRAS DE NEGÓCIO	
Identificador	Descrição
RN01	O Aplicativo apresenta todas as possíveis Cryptos para compra.
RN02	O aplicativo apresenta os valores das Cryptos e as suas variações gráficas.
RN03	O aplicativo apresenta o histórico de valores de compra e venda das transações
RN04	O aplicativo apresenta a entrada e saída de valores.
RN05	O aplicativo deduz o valor de Impostos e a taxa percentual da Tharseo.
RN06	O aplicativo deve ser intuitivo e ter um fluxo que o usuário consiga atingir o seu objetivo.
RN07	O aplicativo deve gerar informe de rendimentos para se necessário o seu usuário declarar no imposto de renda.

2.5. Modelagem do banco de dados

“Modelagem de banco de dados é o processo de levantamento, análise, categorização e exploração de todos os dados e tipos de informações que irão sustentar uma aplicação. “ (Escola Superior de Redes, 2020)

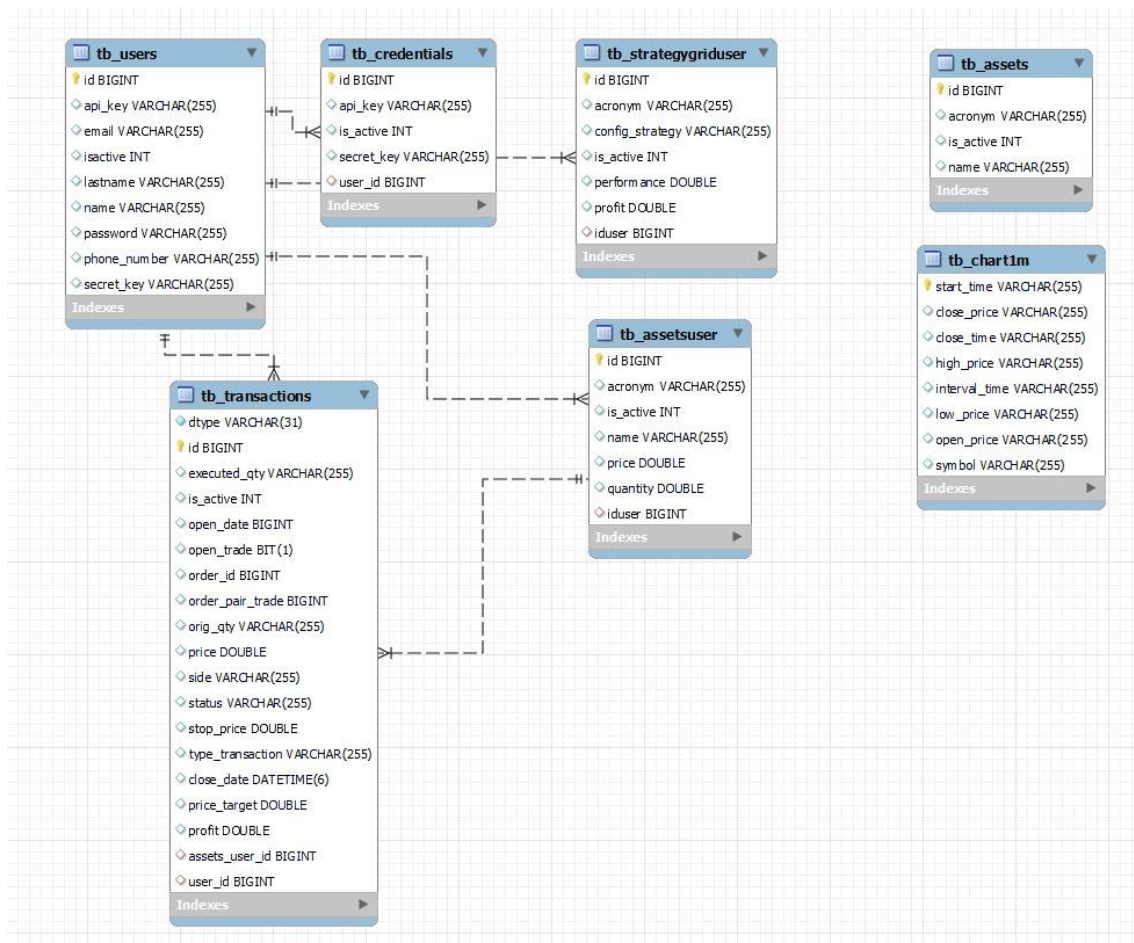
Após a estruturação e estudo dos requisitos, iniciamos uma etapa muito importante na construção de qualquer aplicação: A Modelagem do Banco de Dados.

Listamos os requerimentos e através de análises, listamos as entidades e atributos que seriam pertinentes a base da aplicação. Após formalização das entidades e atributos, suas devidas cardinalidades, foram formados os relacionamentos de forma a agilizar a gestão de dados do banco de dados.

Com a estrutura do BD formada, foram iniciadas as normalizações baseadas nas prioridades de seus conceitos:

1. Primeira Forma Normal de Codd:
 - Possuir somente dados atômicos;
 - Não há grupos de atributos repetidos;
 - Todas as entidades possuem uma chave primária;
 - Relação não possui atributos multivalorados.
2. Segunda Forma Normal de Codd:
 - Estar na 1FN;
 - Todos os atributos não-chaves são funcionalmente dependentes de todas as partes da chave primária;
 - Não existem dependências parciais.
3. Terceira Forma Normal de Codd:
 - Estar na 2FN;
 - Nenhuma coluna não-chave depende de outra coluna não-chave.

2.5.1. Modelo do Banco relacional



2.6. Desenvolvimento do sistema

2.6.1. Tecnologias utilizadas

2.6.1.1. FIGMA

O Figma é uma ferramenta de design colaborativo baseada em nuvem, amplamente reconhecida por sua capacidade de permitir que equipes trabalhem simultaneamente em projetos de design de interface de usuário (UI) e experiência do usuário (UX). Lançado em 2016 pela Figma Inc., o Figma revolucionou a maneira como designers e equipes de design colaboram e criam interfaces digitais.

Design em Tempo Real: Uma das principais vantagens do Figma é a capacidade de editar designs em tempo real. Vários membros da equipe podem trabalhar simultaneamente no mesmo projeto, visualizando as alterações conforme são feitas. Isso elimina a necessidade de enviar arquivos e versões atualizadas manualmente.

Interface Intuitiva e Acessível: O Figma possui uma interface amigável e intuitiva, que facilita o aprendizado e a navegação para designers de diferentes níveis de habilidade. As ferramentas de design são acessíveis através de menus contextuais e barras de ferramentas bem-organizadas.

Funcionalidades de Prototipagem: Além de criar designs estáticos, o Figma suporta a criação de protótipos interativos. Isso permite que os designers criem fluxos de usuário, simulem interações e testem a usabilidade de seus designs diretamente na plataforma.

Componentes Reutilizáveis: O Figma utiliza o conceito de componentes, que são elementos de design reutilizáveis. Os componentes permitem que os designers criem bibliotecas de design consistente e atualizem automaticamente todos os elementos vinculados sempre que são modificados.

Colaboração e Comentários: A colaboração é central no Figma. Os usuários podem compartilhar seus designs com outras pessoas, permitindo que revisem, editem e comentem diretamente no projeto. Isso facilita o feedback em tempo real e melhora a eficiência da equipe.

Integração com Outras Ferramentas: O Figma oferece integrações com uma variedade de ferramentas de prototipagem, gerenciamento de projetos e desenvolvimento, como Slack, Jira, Trello e GitHub. Isso permite que equipes integrem facilmente o fluxo de trabalho de design com outros processos de desenvolvimento de software.

Eficiência de Colaboração: Equipes distribuídas ou colaboradores remotos podem trabalhar de forma síncrona no mesmo projeto, aumentando a produtividade e reduzindo o tempo de desenvolvimento.

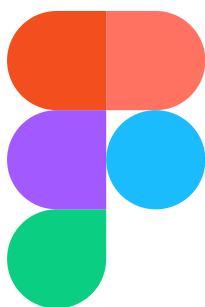
Consistência de Design: Com o uso de componentes e estilos compartilhados, o Figma ajuda a manter a consistência visual em todos os projetos, garantindo uma experiência de usuário coesa.

Feedback Iterativo: A capacidade de receber feedback instantâneo e iterar designs rapidamente melhora a qualidade final do produto e a satisfação do cliente.

Redução de Custos: Ao eliminar a necessidade de várias ferramentas separadas e simplificar o processo de design e desenvolvimento, o Figma pode ajudar a reduzir custos operacionais.

O Figma é amplamente utilizado por empresas de todos os tamanhos, desde startups até grandes corporações, devido à sua acessibilidade e recursos poderosos de colaboração. Ele se tornou uma ferramenta indispensável para equipes de design que buscam eficiência, agilidade e qualidade em seus processos de criação de interfaces digitais.

O Figma transformou a maneira como equipes de design colaboram e criam interfaces digitais, proporcionando uma plataforma robusta e intuitiva para o design de UI/UX. Com funcionalidades avançadas de colaboração, prototipagem e reutilização de componentes, o Figma continua a ser uma escolha popular entre designers e equipes de desenvolvimento que buscam eficiência, qualidade e inovação em seus projetos de design.



Através da ferramenta Figma foi feito todo o nosso processo de prototipagem, ela iniciou-se através de um estudo sobre as cores, onde utilizamos para inspiração de Layouts aplicações voltadas para o mesmo propósito da ideia inicial.

Em relação as cores utilizadas foram utilizadas tons frios para passar mais sobriedade e elegância, com a maior paleta partindo do azul para o roxo.





Abaixo imagens do protótipo gerado como ideia inicial do que tínhamos como proposta para posterior desenvolvimento do front end:

T H A R S E O

E-mail ou celular*




 Digite aqui...

 Login com o Google

 Login com a Apple

[Criar conta](#)


[Acessar](#)

-  **Informações pessoais**
Crie seu acesso
-  **Sobre a sua conta**
Informações sobre a conta
-  **Revisão e confirmação**
Revise suas informações

T H A R S E O

Cadastro:

E-mail ou celular*

 Digite aqui...

Senha*:

 Digite aqui...

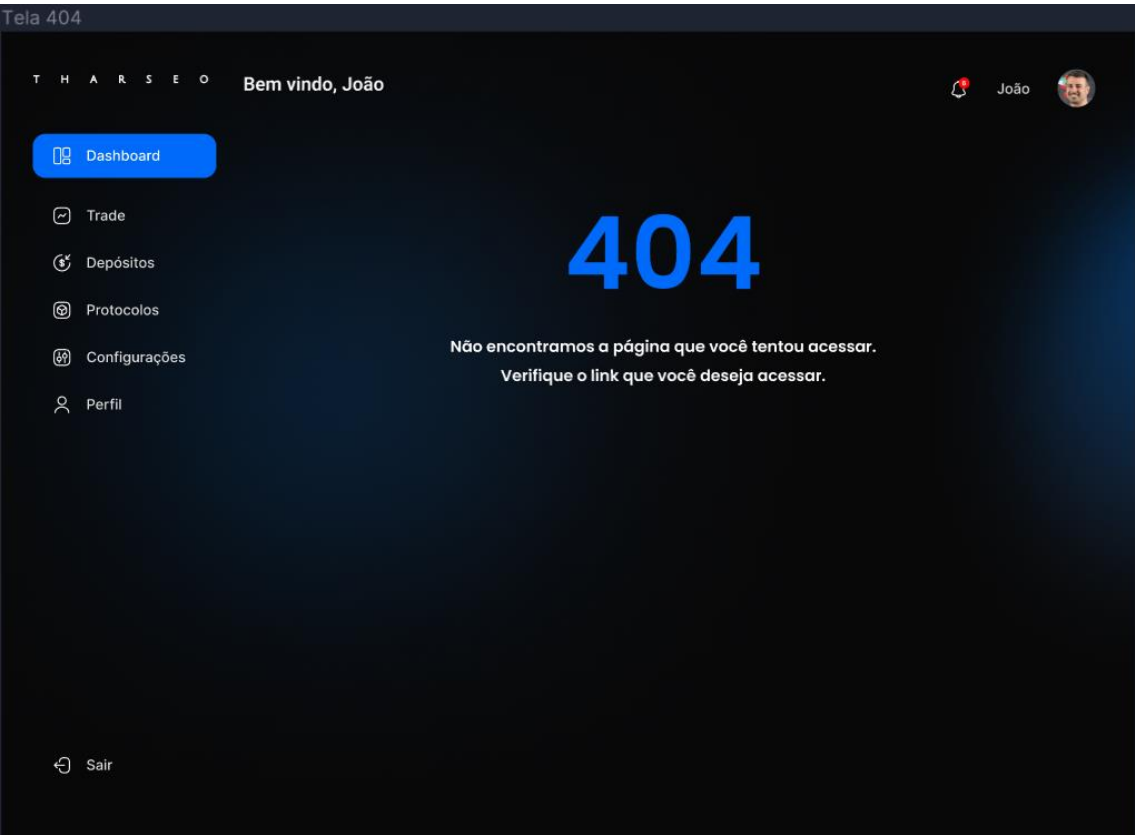
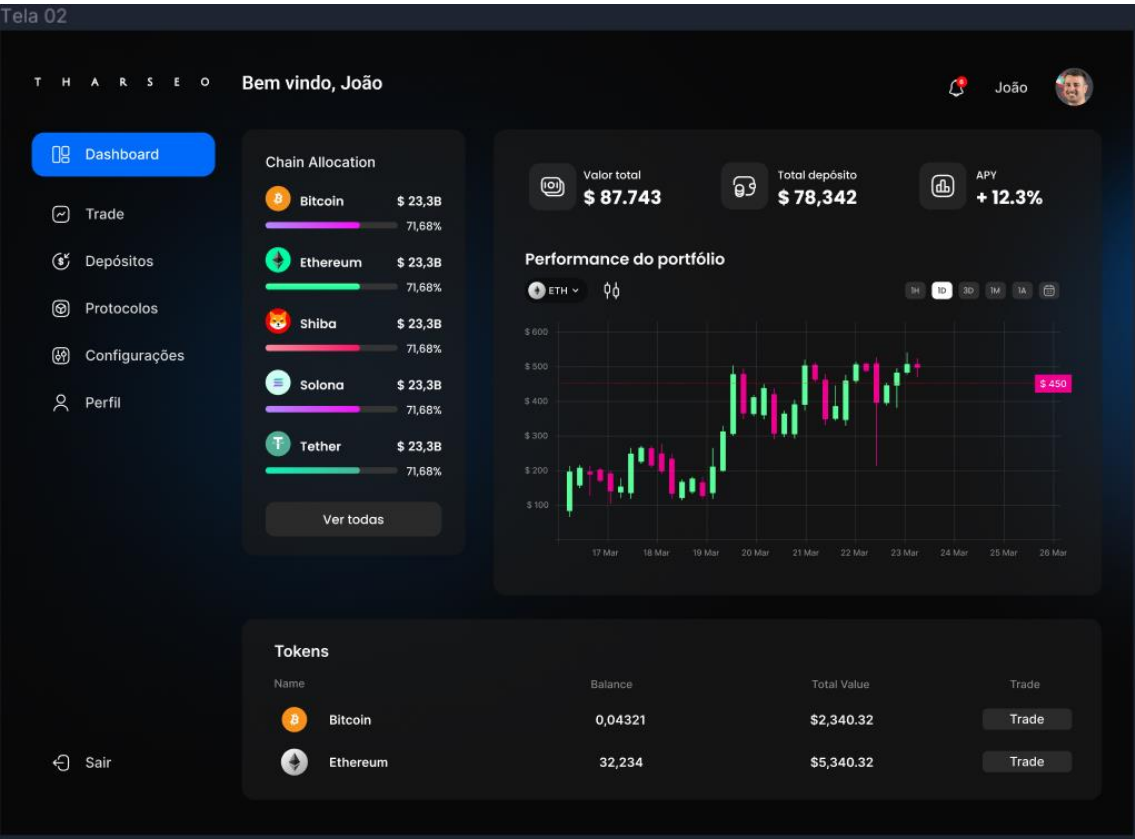
Confirmar senha*:

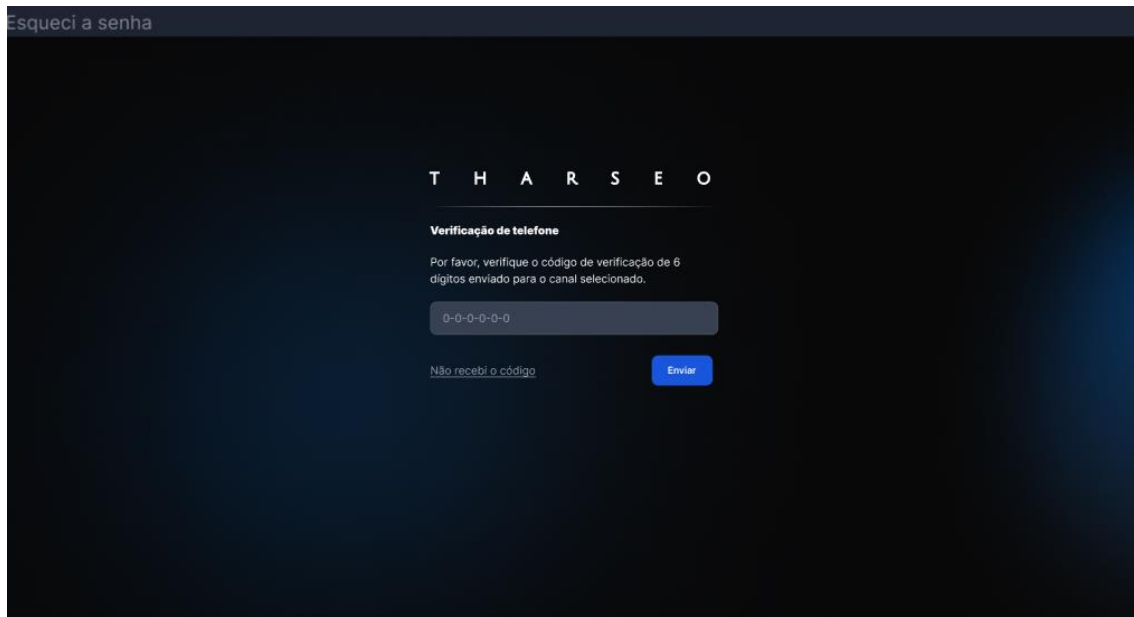
 Digite aqui...

Nós nunca iremos compartilhar seus dados.
Confira nossa [Política de privacidade](#).

[Já tenho uma conta](#)

[Próximo](#)





2.6.1.2. Javascript

JavaScript é uma linguagem de programação de alto nível, interpretada e dinâmica, amplamente utilizada para criar interatividade em páginas web. Desde sua criação em 1995 por Brendan Eich, na Netscape, JavaScript evoluiu de uma linguagem simples de script para uma poderosa ferramenta de desenvolvimento, presente em praticamente todos os navegadores modernos.

Uma das principais características do JavaScript é sua capacidade de manipular o DOM (Document Object Model), permitindo que os desenvolvedores alterem o conteúdo, estrutura e estilo das páginas web de forma dinâmica. Isso possibilita a criação de interfaces ricas e interativas, respondendo a ações do usuário em tempo real.

Além de sua forte presença no desenvolvimento web, JavaScript também se expandiu para outras áreas, incluindo o desenvolvimento de aplicativos móveis (por meio de frameworks como React Native e Ionic), desenvolvimento de servidores (com o Node.js), desenvolvimento de jogos (usando bibliotecas como Phaser e Three.js) e até mesmo para a construção de aplicações de desktop (com ferramentas como Electron).

JavaScript é uma linguagem versátil que suporta vários paradigmas de programação, incluindo programação funcional e orientada a objetos. Sua sintaxe é inspirada em outras linguagens de programação, como Java e C, tornando-a relativamente fácil de aprender para aqueles familiarizados com essas linguagens.

No entanto, JavaScript também tem suas peculiaridades, como a natureza assíncrona de suas operações, que podem ser desafiadoras para os iniciantes. No entanto, com o uso de recursos como promessas e `async/await`, esses desafios podem ser superados.

Com uma vasta comunidade de desenvolvedores, uma infinidade de bibliotecas e frameworks, e sua presença onipresente na web moderna, JavaScript continua a ser uma ferramenta indispensável para qualquer desenvolvedor web. Sua capacidade de criar

experiências interativas e dinâmicas faz com que seja uma escolha popular para uma ampla gama de projetos de desenvolvimento de software.

Linguagem que utilizamos para desenvolver a parte do nosso front end utilizando da técnica de componentização atrelado ao framework que falaremos na sequência que foi o react.



2.6.1.3. React

React é uma biblioteca JavaScript de código aberto para a criação de interfaces de usuário, desenvolvida pelo Facebook e mantida por uma comunidade de desenvolvedores e empresas. Lançado em 2013, o React revolucionou o desenvolvimento front-end ao introduzir conceitos inovadores e eficientes para a construção de componentes de interface de usuário reutilizáveis e de alta performance.

Um dos principais conceitos do React é a componentização. Em vez de pensar em páginas web como um único bloco, o React permite que os desenvolvedores dividam a interface em componentes pequenos e isolados. Cada componente pode ser considerado como um bloco de construção que encapsula sua própria lógica e estilo, e pode ser reutilizado em diferentes partes da aplicação. Essa abordagem modular facilita a manutenção e a escalabilidade de aplicações complexas.

Outra inovação fundamental do React é o Virtual DOM. O Virtual DOM é uma representação leve da interface de usuário na memória. Quando o estado de um componente muda, o React cria uma nova árvore de Virtual DOM e a compara com a versão anterior (processo conhecido como "reconciliation"). Apenas as mudanças mínimas necessárias são aplicadas ao DOM real do navegador, resultando em atualizações mais rápidas e eficientes. Isso melhora significativamente o desempenho, especialmente em aplicações grandes e dinâmicas.

React também introduz o JSX (JavaScript XML), uma extensão de sintaxe que permite escrever HTML dentro do código JavaScript. JSX torna o código mais legível e expressivo, aproximando o desenvolvimento da interface de usuário ao seu layout final. Embora possa parecer estranho inicialmente, JSX se torna intuitivo e poderoso com o uso contínuo.

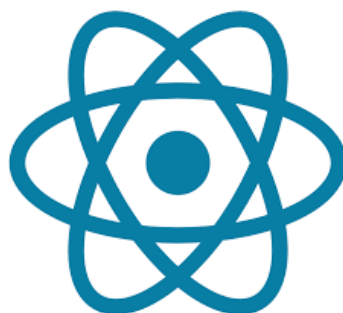
Os componentes do React podem gerenciar seu próprio estado interno, tornando possível criar interfaces dinâmicas e interativas. Além disso, o React fornece métodos de ciclo de vida que permitem aos desenvolvedores controlar o comportamento dos componentes em diferentes fases de sua existência, como montagem, atualização e desmontagem.

O ecossistema do React é vasto e inclui uma ampla variedade de bibliotecas e ferramentas que complementam a biblioteca principal. O React Router permite a criação de navegação em uma aplicação de página única (SPA). O Redux e o Context API são usados para gerenciamento de estado global. Ferramentas como Create React App simplificam o setup de novos projetos, enquanto Next.js e Gatsby oferecem soluções para renderização do lado do servidor e geração de sites estáticos, respectivamente.

A popularidade do React se reflete em sua ampla adoção por empresas de todos os tamanhos, desde startups até gigantes da tecnologia. Facebook, Instagram, Airbnb, e Netflix são alguns exemplos de empresas que utilizam React em suas aplicações. A comunidade ativa de desenvolvedores contribui constantemente com novas ideias, plugins e melhorias para a biblioteca.

O React trouxe uma nova abordagem para o desenvolvimento de interfaces de usuário, focando em desempenho, modularidade e facilidade de manutenção. Sua filosofia de componentes, juntamente com o Virtual DOM e o JSX, oferece uma maneira eficiente e intuitiva de construir aplicações web modernas. Com um ecossistema rico e uma comunidade ativa, React continua a evoluir e a moldar o futuro do desenvolvimento front-end.

A importância do React no nosso trabalho está atrelada ao front end através da componentização conseguimos reaproveitar boa parte do nosso código componetizado que nos otimizou tempo e para futuras manutenções irá facilitar o processo e o entendimento de quem for fazer a manutenção.



2.6.1.4. Java

Java é uma linguagem de programação de alto nível, orientada a objetos e projetada para ser independente de plataforma, o que significa que o código Java pode ser executado em qualquer dispositivo que possua uma máquina virtual Java (JVM). Criada por James Gosling e sua equipe na Sun Microsystems em 1995, a linguagem foi inicialmente concebida para dispositivos eletrônicos, mas rapidamente se tornou uma das linguagens de programação mais utilizadas no mundo.

Java foi desenvolvida com base em cinco objetivos principais:

Simples, Orientada a Objetos e Familiar: Java foi projetada para ser fácil de aprender e usar, com uma sintaxe que lembra C++ mas simplificada, removendo características complexas como herança múltipla e ponteiros explícitos.

Robusta e Segura: A linguagem inclui várias funcionalidades para garantir a robustez e a segurança, como a gestão automática de memória (garbage collection), controle rigoroso de tipos e a ausência de ponteiros diretos, o que minimiza erros comuns de programação.

Arquitetura-Neutra e Portável: A independência de plataforma é alcançada através do uso da JVM, que permite que o mesmo código Java seja executado em qualquer sistema operacional que possua uma JVM correspondente.

Desempenho Elevado: Embora interpretada, a JVM inclui um compilador Just-In-Time (JIT) que traduz bytecode em código nativo em tempo de execução, otimizando a performance.

Multithreaded e Dinâmica: Java foi projetada para ser utilizada em ambientes de rede e suporta a criação de aplicações multithreaded, o que permite a execução simultânea de várias threads, melhorando a eficiência e a reatividade das aplicações.

Java é uma linguagem fortemente tipada, o que significa que todas as variáveis devem ser declaradas com um tipo específico. A estrutura básica de um programa Java consiste em classes e objetos. Cada programa Java deve ter uma classe com um método main, que é o ponto de entrada da aplicação.

A sintaxe de Java é rigorosa e segue convenções claras, o que contribui para a legibilidade e manutenção do código. A herança, polimorfismo, encapsulamento e abstração são pilares fundamentais do paradigma orientado a objetos em Java.

A independência de plataforma de Java é conseguida através da compilação do código fonte em bytecode, que é uma forma intermediária de código executável pela JVM. Cada máquina virtual Java é adaptada ao sistema operacional subjacente, permitindo que o bytecode seja executado de maneira consistente em qualquer ambiente.

Java possui um ecossistema vasto e diversificado, com uma enorme quantidade de bibliotecas, frameworks e ferramentas de desenvolvimento. Alguns dos frameworks mais populares incluem:

Spring: Um framework poderoso para desenvolvimento de aplicações empresariais, com suporte abrangente para injeção de dependência, AOP (Programação Orientada a Aspectos) e muito mais.

Hibernate: Uma biblioteca para mapeamento objeto-relacional (ORM) que facilita a interação com bancos de dados relacionais.

JavaServer Faces (JSF): Um framework para desenvolvimento de interfaces web baseadas em componentes.

Ferramentas de desenvolvimento como o Eclipse, IntelliJ IDEA e NetBeans oferecem ambientes robustos para codificação, depuração e gerenciamento de projetos Java.

Java é amplamente utilizada em uma variedade de aplicações:

Desenvolvimento Web: Java é frequentemente usada para criar aplicativos web escaláveis e robustos.

Aplicações Empresariais: Java é uma escolha popular para sistemas empresariais de grande escala devido à sua segurança, robustez e portabilidade.

Desenvolvimento Móvel: Android, o sistema operacional móvel mais popular, utiliza Java como sua principal linguagem de programação.

Aplicações Científicas e Financeiras: A confiabilidade e desempenho de Java fazem dela uma escolha comum em aplicações científicas e financeiras.

Java possui uma das comunidades de desenvolvedores mais ativas e uma vasta quantidade de recursos de aprendizado, desde tutoriais e cursos online até fóruns e conferências. A linguagem é constantemente atualizada e aprimorada pela Oracle, que mantém o desenvolvimento do Java Development Kit (JDK).

Java é uma linguagem madura, poderosa e versátil, adequada para uma ampla gama de aplicações. Sua filosofia de independência de plataforma, robustez e facilidade de uso, combinada com um ecossistema rico e uma comunidade ativa, assegura que Java continue sendo uma escolha popular para desenvolvedores em todo o mundo.

O Java teve grande importância no nosso projeto, sendo ele a linguagem que constituiu o nosso back end, estruturando e dando forma a parte de execução do nosso projeto, a mesma foi utilizada atrelada a um dos seus principais frameworks o Spring Boot.

2.6.1.5. Spring

O Spring Framework é um dos frameworks mais amplamente utilizados para o desenvolvimento de aplicações empresariais em Java. Criado por Rod Johnson em 2003, o Spring se destaca por sua abrangente funcionalidade, flexibilidade e capacidade de simplificar o desenvolvimento de aplicações robustas e escaláveis.

Uma das características fundamentais do Spring é o suporte à Injeção de Dependência. DI permite que os componentes da aplicação sejam desacoplados e facilmente testáveis. Em vez de criar dependências diretamente dentro das classes, estas são injetadas pelo contêiner Spring, promovendo um design mais modular e limpo.

Spring também suporta a Programação Orientada a Aspectos, que permite a separação de preocupações transversais (como logging, segurança, transações) do código de negócios principal. Isso resulta em um código mais limpo e de fácil manutenção.

O Spring facilita o gerenciamento de transações declarativas, permitindo que os desenvolvedores definam transações em um nível de configuração ou anotação, sem a necessidade de lidar diretamente com a API de transações.

Spring integra-se facilmente com outros frameworks e tecnologias populares, como Hibernate, JPA, JMS, e mais. Isso permite que os desenvolvedores escolham as melhores ferramentas para cada parte da aplicação.

Um dos módulos mais populares do Spring é o Spring MVC (Model-View-Controller), que fornece uma maneira estruturada de desenvolver aplicações web. Ele suporta a construção de APIs RESTful, além de facilitar o desenvolvimento de interfaces de usuário ricas.

O Spring Framework é altamente modular, composto por uma série de projetos e módulos que podem ser usados de forma independente ou conjunta, dependendo das necessidades do desenvolvedor:

Spring Core: O núcleo do framework, que fornece funcionalidades básicas como Injeção de Dependência e controle de inversão.

Spring AOP: Suporte para Programação Orientada a Aspectos.

Spring Data: Facilita a integração com várias tecnologias de armazenamento de dados, como bancos de dados relacionais, NoSQL e mais.

Spring Batch: Suporte para processamento em lote, ideal para tarefas como ETL (Extract, Transform, Load).

Spring Security: Um módulo poderoso para segurança de aplicações, que inclui autenticação e autorização.

Spring Boot: Um projeto que simplifica a configuração e a inicialização de novos projetos Spring, reduzindo a necessidade de configuração manual.

Lançado em 2014, o Spring Boot é uma extensão do Spring Framework que visa simplificar o processo de configuração e desenvolvimento de novas aplicações Spring. Ele fornece convenções padrão para configuração, permitindo que os desenvolvedores iniciem projetos rapidamente com configuração mínima.

Autoconfiguração: Spring Boot automaticamente configura componentes e bibliotecas com base no que está disponível no classpath.

Starters: POMs (Project Object Models) predefinidos que agregam dependências necessárias para diferentes tipos de projetos (web, dados, segurança etc.), facilitando a inclusão de funcionalidades.

Spring Boot CLI: Uma interface de linha de comando que permite a criação rápida de aplicações Spring Boot usando Groovy.

O Spring é amplamente utilizado em diversos tipos de aplicações:

Aplicações Web: Spring MVC e Spring Boot são populares para o desenvolvimento de APIs RESTful e aplicações web completas.

Serviços Empresariais: O suporte robusto a transações e integração com tecnologias de banco de dados fazem do Spring uma escolha ideal para aplicações empresariais.

Microserviços: Spring Cloud, uma extensão do Spring, oferece ferramentas para construir e gerenciar arquiteturas de microserviços, incluindo configuração distribuída, descoberta de serviços e gerenciamento de circuitos.

A comunidade Spring é vasta e ativa, com inúmeros recursos disponíveis, incluindo documentação oficial, tutoriais, fóruns e conferências. A Pivotal, agora parte da VMware, lidera o desenvolvimento do Spring Framework e oferece suporte comercial para empresas que precisam de garantias adicionais de suporte e manutenção.

O Spring Framework é uma ferramenta poderosa e flexível para o desenvolvimento de aplicações em Java. Com suas características robustas de injeção de dependência, programação orientada a aspectos, e modularidade, combinado com a simplicidade e conveniência do Spring Boot, o Spring continua a ser a escolha preferida para desenvolvedores que buscam construir aplicações escaláveis, seguras e de alto desempenho.



Spring Boot

2.6.1.6. API Binance

Podemos definir API como Interface de Programação de Aplicações, é um conjunto de definições e protocolos que permite a comunicação entre diferentes softwares. APIs especificam como os componentes de software devem interagir, facilitando a integração de novas funcionalidades e serviços. Elas são amplamente utilizadas para permitir que diferentes sistemas e aplicações troquem informações e funcionem em conjunto de maneira eficiente.

Tipos de API:

APIs Web: Permitem a comunicação entre servidores e clientes através da internet, utilizando protocolos como HTTP/HTTPS. São comuns em serviços web e aplicativos móveis.

APIs de Sistema Operacional: Fornecem acesso a funcionalidades do sistema operacional, permitindo que aplicações interajam com hardware e recursos do sistema.

APIs de Biblioteca: Oferecem funções e métodos pré-definidos para realizar tarefas específicas dentro de uma linguagem de programação.

APIs de Banco de Dados: Facilitam a interação entre aplicações e sistemas de gerenciamento de bancos de dados.

Componentes de uma API:

End Points: URLs específicas onde os recursos da API podem ser acessados.

Métodos HTTP: Definem a ação a ser realizada, como GET (recuperar dados), POST (enviar dados), PUT (atualizar dados) e DELETE (remover dados).

Cabeçalhos de Requisição: Contêm informações adicionais sobre a solicitação, como autenticação e formato de dados.

Corpo da Requisição: Contém os dados que são enviados para a API (usado principalmente com métodos POST e PUT).

Respostas da API: Incluem códigos de status HTTP e, frequentemente, dados no formato JSON ou XML.

Vantagens das APIs:

Interoperabilidade: Permitem que diferentes sistemas e plataformas trabalhem juntos.

Reutilização de Código: Promovem a reutilização de funcionalidades existentes, economizando tempo e esforço.

Desenvolvimento Modular: Facilitam o desenvolvimento de aplicações modulares e escaláveis.

Exemplos de Uso de APIs:

Serviços de Pagamento: APIs como PayPal e Stripe permitem que lojas online processem pagamentos.

Redes Sociais: APIs do Facebook, Twitter e Instagram permitem a integração de funcionalidades dessas plataformas em outras aplicações.

Mapas e Geolocalização: APIs do Google Maps permitem a inclusão de mapas e serviços de localização em aplicações.

APIs são ferramentas essenciais no desenvolvimento de software moderno, permitindo a criação de aplicações mais integradas, eficientes e fáceis de manter. Elas possibilitam que diferentes sistemas e aplicações se comuniquem de maneira padronizada, promovendo a interoperabilidade e a reutilização de código.

Foi utilizada a API da Binance para que fosse possível importar os valores e cotações das cryptos, utilizada também para acesso a carteira do cliente e para possibilitar a compra e a venda das cryptos, possibilitando a visualização da cotação além das transações de compras e vendas conforme setado pelo usuário.

2.6.1.7. MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto que usa a linguagem de consulta estruturada para gerenciar e manipular dados. O MySQL foi desenvolvido originalmente pela MySQL AB, uma empresa sueca; em 2008 foi adquirido pela Sun Microsystems e depois pela Oracle Corporation em 2010. O MySQL é usado amplamente em aplicações empresariais e na web devido à robustez, eficiência e facilidade de uso. Alguns dos principais recursos do MySQL são: código aberto, desempenho e escalabilidade, segurança, compatibilidade, alta disponibilidade etc. Portanto, sendo um software de código aberto, seu código-fonte é disponibilizado voluntariamente, permitindo que qualquer pessoa possa estudá-lo, modificar e distribuí-lo sem restrição.

MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto que utiliza a linguagem SQL (Structured Query Language) para gerenciar e manipular dados. Desenvolvido inicialmente por MySQL AB, uma empresa sueca, o MySQL foi adquirido pela Sun Microsystems em 2008, e posteriormente pela Oracle Corporation em 2010. MySQL é amplamente utilizado em aplicações web e empresariais devido à sua robustez, eficiência e facilidade de uso.

Componentes e Arquitetura

Servidor MySQL: O núcleo do MySQL, responsável pelo gerenciamento de bases de dados e processamento de consultas.

Cientes MySQL: Programas que se conectam ao servidor MySQL para realizar operações de administração e manipulação de dados.

Conectores e APIs: Ferramentas que permitem que aplicações em várias linguagens de programação interajam com o banco de dados MySQL.

Armazenamento de Mecanismos: MySQL suporta vários mecanismos de armazenamento, como InnoDB (transacional e com suporte a ACID) e MyISAM (mais rápido e eficiente para leituras).

Benefícios do MySQL

Facilidade de Uso: MySQL é conhecido por sua facilidade de instalação e configuração. Ele possui uma interface intuitiva e bem documentada, que facilita o gerenciamento de bancos de dados.

Comunidade Ativa: Como um projeto de código aberto, MySQL tem uma comunidade global ativa que contribui para seu desenvolvimento, fornece suporte e compartilha recursos.

Flexibilidade: MySQL pode ser usado para uma ampla variedade de aplicações, desde pequenos sites e blogs até grandes sistemas corporativos e plataformas de e-commerce.

2.6.1.8.Websocket

WebSocket é uma tecnologia de comunicação bidirecional que permite a transmissão de dados entre um navegador web e um servidor de maneira eficiente e em tempo real. Diferente do modelo tradicional de requisições HTTP, onde o navegador faz uma solicitação e espera por uma resposta do servidor, com WebSockets, uma conexão persistente é mantida, permitindo a troca de mensagens de ambos os lados a qualquer momento.

Como Funciona

Estabelecendo a Conexão: Um WebSocket começa como uma conexão HTTP normal, com um pedido de "upgrade" sendo enviado do cliente para o servidor, solicitando a troca para o protocolo WebSocket. Uma vez aceito, a comunicação é mantida de forma persistente, e mensagens podem fluir em ambas as direções.

Bidirecionalidade: Uma vez estabelecida a conexão, tanto o cliente quanto o servidor podem enviar mensagens a qualquer momento sem a necessidade de uma solicitação explícita.

Eventos e Callbacks: A comunicação WebSocket é geralmente baseada em eventos. O cliente e o servidor podem definir manipuladores para eventos como a abertura da conexão, recebimento de uma mensagem, ou fechamento da conexão.

2.6.1.9.Git

O git é um sistema de controle de versão de códigos. Gerencia ramificações feitas por diferentes pessoas de um time de modo a manter a ordem e integração de todas as partes. Ou seja, mesmo que várias pessoas estejam trabalhando em um programa ao mesmo tempo, ainda é possível controlar as mudanças para evitar erros.

Sem um controle de versionamento, como o que o git propõe, pode haver o risco de pessoas acessarem diferentes versões e suas mudanças sumirem por conta da desorganização.

Além disso, o git é um sistema que permite rastrear alterações e manter um histórico dessas alterações para possibilitar voltar a uma versão estável sempre que ocorrer um problema. Ou seja, caso algum erro quebre o sistema em desenvolvimento, é viável voltar para a versão funcional saudável.

Em complemento ao que falamos, o git ajuda também na transparência e no controle das mudanças feitas. Viabiliza o conhecimento sobre quem fez o quê e a frequência dessas modificações.

Para prosseguirmos, no entanto, é importante destacar a diferença do git para o github. O github é uma plataforma de gestão de códigos que se baseia no git e adiciona um fator social, de compartilhamento e interação interpessoal. O hub é crucial para a gestão de projetos, pois ajuda a visualizar as responsabilidades e as entregas, com um controle de prazos e de detalhes.

O hub é um site aberto para todos. Então, é comum que pessoas colaborem com projetos de outras pessoas dev que nem conhecem, de outros estados ou países. Projetos open-source são disponibilizados lá e podem ser alterados por qualquer pessoa desenvolvedora. Isso só reforça o aspecto comunitário e colaborativo da área de programação.

Ou seja, enquanto o git é o sistema por trás, que cuida dos códigos e comandos e gerência as ramificações, o hub é a plataforma onde se postam os códigos.

O github é uma das plataformas baseadas no git, inclusive. Existem várias outras que tentam implementar as mesmas funções e nelas se usam os mesmos comandos git. Alguns exemplos delas: GitLab e Apache Allura.

2.6.1.10.Docker

O Docker é um software de código aberto usado para implantar aplicativos dentro de containers virtuais. A containerização permite que vários aplicativos funcionem em diferentes ambientes complexos. Por exemplo: o Docker permite executar o WordPress em sistemas Windows, Linux e macOS, sem problemas.

Docker vs Máquina Virtual

Embora o Docker e as máquinas virtuais tenham um propósito semelhante, seu desempenho, portabilidade e suporte a sistemas operacionais diferem significativamente.

A principal diferença é que os containers do Docker compartilham o sistema operacional do host, enquanto as máquinas virtuais também têm um sistema operacional convidado sendo executado no sistema host. Esse método de operação afeta o desempenho, as necessidades de hardware e o suporte do SO.

2.6.1.11.SonarCloud

SonarCloud é uma ferramenta de análise de código baseada em nuvem desenvolvida pela SonarSource. Ela oferece serviços de inspeção de código para identificar problemas relacionados à qualidade, segurança e desempenho em projetos de software. SonarCloud é amplamente utilizado por equipes de desenvolvimento para melhorar a qualidade do

código, garantir padrões de codificação e detectar vulnerabilidades de segurança em fases iniciais do desenvolvimento.

Características Principais:

Análise de Qualidade do Código: SonarCloud analisa o código-fonte para identificar problemas de qualidade, como bugs, vulnerabilidades, duplicação de código, cobertura de testes e violações de padrões de codificação.

Integração com CI/CD: A ferramenta se integra facilmente com sistemas de Integração Contínua/Entrega Contínua (CI/CD) como Jenkins, Travis CI, Azure DevOps, GitHub Actions e Bitbucket Pipelines. Isso permite a análise automática do código em cada fase do desenvolvimento.

Suporte a Várias Linguagens: SonarCloud suporta mais de 25 linguagens de programação, incluindo Java, JavaScript, TypeScript, Python, C#, PHP, C++, Ruby e Go. Essa ampla compatibilidade facilita a análise de projetos políglotas.

Feedback em Tempo Real: Fornece feedback contínuo aos desenvolvedores durante o processo de desenvolvimento, permitindo que problemas de qualidade e segurança sejam corrigidos imediatamente.

Dashboard e Relatórios: SonarCloud oferece um dashboard intuitivo que fornece uma visão abrangente do estado da qualidade do código. Relatórios detalhados ajudam as equipes a identificar áreas problemáticas e acompanhar o progresso ao longo do tempo.

Regra de Qualidade Personalizável: Permite que as equipes personalizem regras de qualidade de acordo com suas necessidades específicas, garantindo que o código atenda aos padrões internos e requisitos do projeto.

Análise de Ramificações e Pull Requests: Analisa ramificações e pull requests para detectar problemas antes de serem mesclados com a linha principal do código, prevenindo a introdução de novos defeitos.

Benefícios do SonarCloud:

Melhoria Contínua da Qualidade do Código: A análise contínua e o feedback em tempo real ajudam os desenvolvedores a manter um alto padrão de qualidade do código ao longo do ciclo de desenvolvimento.

Redução de Custos com Correção de Bugs: Detectar e corrigir problemas de qualidade e segurança nas fases iniciais do desenvolvimento reduz significativamente os custos associados à correção de bugs em estágios posteriores.

Maior Segurança: A identificação precoce de vulnerabilidades de segurança ajuda a proteger o software contra ameaças potenciais e a cumprir normas de segurança.

Facilidade de Uso: A interface intuitiva e a fácil integração com ferramentas de CI/CD existentes tornam o SonarCloud acessível e fácil de usar para equipes de todos os tamanhos.

Visibilidade e Transparência: Os dashboards e relatórios fornecem visibilidade sobre a qualidade do código, permitindo que todos os membros da equipe, incluindo gerentes de projeto e stakeholders, acompanhem o progresso e identifiquem áreas de melhoria.

2.6.1.12.Azure

O Microsoft Azure é uma plataforma destinada à execução de aplicativos e serviços, baseada nos conceitos da computação em nuvem da Microsoft. Anteriormente conhecido como Windows Azure, ele é, atualmente, um dos principais pontos de ação da empresa e de seu CEO, Satya Nadella.

Atualmente, adotado cada vez mais pelas grandes corporações, a plataforma é um importante passo dentro da jornada de transformação digital. No entanto, dúvidas e inseguranças geradas pela falta de conhecimento a respeito da computação em nuvem atrasam, em muitos casos esse processo de modernização.

Antes de entendermos como se dá a atuação do Microsoft Azure, é necessário dar um passo para trás e compreender o que é a nuvem.

A definição mais direta possível afirma que a nuvem é uma rede global de servidores virtuais que contam com as mais variadas funções. Por não ser física, essa rede consegue conectar servidores remotos espalhados globalmente, possibilitando que eles operem em conjunto.

Mas, afinal, quais são essas funções? Como já dissemos aqui, elas são múltiplas – e, por isso, a nuvem atende tão bem as necessidades do mundo corporativo moderno. Dentre elas, podemos citar o armazenamento e gerenciamento de dados e o desenvolvimento e execução de aplicativos e serviços como webmail e softwares.

A virtualidade da nuvem também permite que seus usuários consigam acessar todos os seus recursos por meio dos mais variados dispositivos. Na prática, isso traz um grande incentivo à produtividade.

Existem quatro tipos de nuvem:

Nuvem pública: modelo padrão. Os serviços são oferecidos a todos que querem comprá-los (ou simplesmente usá-los) na internet pública.

Nuvem privada: também conhecida como nuvem corporativa. Nela, os serviços são disponibilizados somente a usuários selecionados.

Nuvem de comunidade: os recursos são compartilhados entre organizações com interesses convergentes, como instituições ligadas a governos, por exemplo.

Nuvem híbrida: combinação da nuvem pública e privada; os dados e aplicativos podem ser compartilhados entre elas de acordo com a demanda. É a opção mais flexível e a classificação do Microsoft Azure.

O Microsoft Azure é a plataforma de nuvem híbrida da Microsoft. Por definição, ele concentra toda a infraestrutura de cloud computing da empresa, incluindo os seus produtos, serviços e aplicativos.

Do ponto de vista prático, há dois modelos de serviços na nuvem: IaaS (Infrastructure as a Service, ou Infraestrutura como um Serviço, em português) e PaaS (Platform as a Service, ou Plataforma como um Serviço, em português).

Microsoft Azure e IaaS

O IaaS é voltado para a infraestrutura de computação, que inclui itens como servidores e backups. O formato prioriza a diminuição de gastos com a compra e administração de servidores físicos e demais estruturas de processamento de dados. Nesse sentido, o Microsoft Azure gerencia a infraestrutura enquanto o usuário cuida de toda a configuração (incluindo a compra e instalação) do software.

Dessa maneira, além de otimizar os gastos, o IaaS elimina preocupações com a infraestrutura de TI da empresa, incentivando a inovação e garantindo a segurança dos dados e aplicativos mesmo em situações de desastre e demais interrupções.

O Microsoft Azure também funciona no modelo PaaS, que oferece ainda mais tranquilidade ao usuário. E o motivo é simples! Além de incluir todos os recursos de IaaS, ele também engloba todo o ambiente necessário para o desenvolvimento e administração dos aplicativos na nuvem.

Ou seja, o usuário só precisa se preocupar em gerenciar os apps que desenvolveu, deixando todo o resto por conta do provedor. Além disso, PaaS também oferece serviços de business intelligence (BI), gerenciamento de bancos de dados e middleware (uma camada de mediação entre diferentes tecnologias e aplicativos).

Quando tratamos de suas vantagens, podemos dizer que o Microsoft Azure facilita consideravelmente a vida dos desenvolvedores. O modelo PaaS tem os mesmos benefícios do IaaS – e mais.

Nesse formato, os destaques são as ferramentas sofisticadas de desenvolvimento que permitem o gerenciamento de todas as etapas de criação de aplicativos. Elas garantem a diminuição do tempo de codificação de novos apps e permitem que os desenvolvedores trabalhem não somente nos computadores tradicionais, mas também em dispositivos móveis. A colaboração também entra em evidência aqui: em um ambiente de desenvolvimento online, profissionais conseguem trabalhar juntos, mesmo à distância.

Graças a sua variedade de recursos, o Microsoft Azure pode ser utilizado por empresas dos mais diferentes setores – desde a saúde até o varejo, passando pela manufatura e logística.

Levando em consideração todas as possibilidades proporcionadas pela computação em nuvem, não é uma surpresa o fato de que o Microsoft Azure tem muitos serviços diferentes.

Com o Azure, é possível desenvolver e implantar apps diversos, inclusive para dispositivos móveis. Os usuários também podem migrar aplicativos, servidores e bancos de dados para a nuvem da Microsoft.

Há, ainda, a opção de trabalhar sem servidores no Microsoft Azure. Nela, os desenvolvedores conseguem produzir com mais rapidez e os custos com a infraestrutura são otimizados. Vale notar que os aplicativos sem servidor continuam contando com a possibilidade de incluir recursos de Inteligência Artificial e algoritmos de machine learning.

A computação de alto desempenho (HPC) também tem um grande aliado no Microsoft Azure. Com a nuvem da Microsoft, empresas de diferentes setores conseguem gerenciar e executar facilmente os seus fluxos de trabalho. Aqui, versatilidade e escalabilidade são palavras-chave.

Outro uso muito relevante para o Microsoft Azure é o seu papel no backup e recuperação de desastres. O mundo corporativo lida com um número crescente de dados; ter espaço

para armazená-los com segurança é imprescindível. O Azure permite que as companhias aumentem a sua capacidade de armazenamento de acordo com a demanda, otimizando gastos.

Além disso, é importante notar que todas as soluções do Microsoft Azure foram projetadas para facilitar a conformidade com as leis de proteção de dados – isto é, LGPD e GDPR.

Microsoft Azure e segurança

Como citamos aqui, a segurança é um item muito importante no Microsoft Azure. A empresa investe massivamente nesse aspecto – no total, US\$ 1 bilhão são empreendidos anualmente em soluções de cibersegurança na nuvem da Microsoft.

O Microsoft Azure tem camadas espessas de segurança que protegem os usuários das mais variadas ameaças. Exemplo disso são os ataques DDoS, que não são classificados exatamente como uma invasão ao sistema, mas como uma forma de impedir que seus recursos sejam acessados.

No dia a dia, todas as operações realizadas no Microsoft Azure também são altamente protegidas. No Azure Active Directory (AD), por exemplo, é possível gerenciar identidades e controlar os acessos, adotando as melhores práticas do setor. Dessa maneira, garante-se que somente as pessoas certas, sob condições específicas, consigam acessar os dados da empresa.

Além disso, o Microsoft Azure também garante a segurança da rede por meio de controles fortes e nativos da plataforma, como o Firewall do Azure. Outra ferramenta importante é o Azure Key Vault, cuja função é armazenar e gerenciar senhas. É interessante notar que, antes de realizar qualquer ação nele, é preciso passar por um processo de autenticação.

2.6.1.13. Trello

Trello é uma ferramenta de gerenciamento de projetos baseada em nuvem, desenvolvida pela Atlassian, que utiliza um sistema visual de quadros, listas e cartões para ajudar as equipes a organizar e priorizar projetos de forma colaborativa. Lançado em 2011, o Trello se destaca por sua simplicidade, flexibilidade e facilidade de uso, tornando-se uma escolha popular para indivíduos, pequenas equipes e grandes organizações.

Características Principais

Quadros (Boards): Os quadros são a base do Trello, representando projetos ou fluxos de trabalho distintos. Cada quadro pode ser configurado de acordo com as necessidades do projeto ou da equipe.

Listas: Dentro de cada quadro, os usuários podem criar listas que representam diferentes estágios de um projeto, tarefas agrupadas por categorias, ou quaisquer outras divisões que façam sentido para o fluxo de trabalho.

Cartões: Os cartões são os blocos de construção básicos dentro de uma lista. Eles podem representar tarefas, ideias, itens a fazer, ou qualquer outro componente do projeto. Os cartões podem conter descrições, checklists, datas de vencimento, anexos, etiquetas e comentários.

Arrastar e Soltar: Os cartões podem ser facilmente movidos entre listas através de uma interface intuitiva de arrastar e soltar, permitindo que os usuários atualizem o status das tarefas de forma visual e rápida.

Colaboração em Tempo Real: Trello permite a colaboração em tempo real, onde os membros da equipe podem ver instantaneamente as atualizações feitas por outros membros, facilitando a comunicação e a coordenação.

Etiquetas e Filtros: Os cartões podem ser organizados e categorizados usando etiquetas coloridas. Filtros permitem que os usuários visualizem apenas os cartões que atendem a certos critérios, facilitando a gestão de grandes quantidades de tarefas.

Benefícios do Trello

Facilidade de Uso: Trello é conhecido por sua interface simples e intuitiva, que facilita a adoção rápida por parte de novos usuários sem a necessidade de treinamento extensivo.

Flexibilidade: A estrutura de quadros, listas e cartões permite que o Trello seja adaptado a uma variedade de metodologias de gerenciamento de projetos, como Kanban, Scrum, ou workflows personalizados.

Visualização Clara: O sistema visual do Trello ajuda as equipes a verem o progresso dos projetos de maneira clara e organizada, identificando facilmente o status de cada tarefa.

Colaboração Efetiva: As funcionalidades de colaboração em tempo real e comunicação integrada melhoram a coordenação e a transparência dentro das equipes, independentemente de onde os membros estejam localizados.

Escalabilidade: Trello é adequado tanto para pequenas equipes quanto para grandes organizações, oferecendo planos gratuitos e pagos que atendem a diferentes necessidades e tamanhos de equipe.

2.6.1.14. Kanban

Kanban é uma metodologia de gerenciamento de projetos e processos que utiliza cartões visuais e quadros para gerenciar o fluxo de trabalho. Originário do sistema de produção da Toyota na década de 1940, o Kanban foi desenvolvido para melhorar a eficiência da produção e minimizar desperdícios. Hoje, é amplamente utilizado em várias indústrias, incluindo desenvolvimento de software, fabricação e serviços profissionais.

Princípios do Kanban:

Visualização do Trabalho: Um dos princípios fundamentais do Kanban é a visualização do trabalho. As tarefas são representadas por cartões em um quadro, geralmente dividido em colunas que representam diferentes etapas do processo de trabalho. Isso permite que todos os membros da equipe vejam o estado atual de cada tarefa e identifiquem rapidamente gargalos e áreas de melhoria.

Limitação do Trabalho em Progresso (WIP): Para evitar sobrecarga e melhorar o fluxo de trabalho, o Kanban limita a quantidade de trabalho em progresso. Isso significa que apenas um número definido de tarefas pode estar em cada estágio do processo ao mesmo tempo. Limitar o WIP ajuda a garantir que as tarefas sejam concluídas antes de novas serem iniciadas, promovendo um fluxo mais estável e previsível.

Gestão do Fluxo: O Kanban se concentra na gestão eficiente do fluxo de trabalho. Isso envolve monitorar o movimento dos cartões através das colunas, identificar e eliminar gargalos, e garantir que o trabalho flua de maneira contínua e suave através do processo.

Feedback Contínuo: O Kanban promove uma cultura de feedback contínuo. Reuniões regulares, como revisões de fluxo e sessões de retrospectiva, são realizadas para discutir o progresso, identificar problemas e implementar melhorias. Esse ciclo constante de feedback ajuda a equipe a se adaptar e melhorar continuamente.

Melhoria Contínua: Kaizen, ou melhoria contínua, é um princípio central do Kanban. As equipes são encorajadas a procurar constantemente maneiras de melhorar seus processos e aumentar a eficiência. Isso pode incluir ajustes nas políticas do quadro, mudanças na estrutura da equipe ou a introdução de novas práticas de trabalho.

Benefícios do Kanban:

Transparência: A visualização clara do trabalho e do progresso melhora a transparência e a comunicação dentro da equipe.

Flexibilidade: O Kanban é altamente flexível e pode ser adaptado a diferentes tipos de projetos e processos. Ele permite ajustes rápidos e respostas ágeis a mudanças.

Eficiência: Limitar o WIP e focar na melhoria contínua ajuda a aumentar a eficiência do processo e reduzir desperdícios.

Foco na Entrega: O Kanban incentiva a conclusão de tarefas antes de iniciar novas, promovendo um foco maior na entrega de valor.

Kanban é uma metodologia eficaz para gerenciar o fluxo de trabalho e promover a eficiência através da visualização, limitação de trabalho em progresso e melhoria contínua. Sua flexibilidade e foco em feedback contínuo o tornam uma escolha popular para equipes em diversas indústrias que buscam melhorar seus processos e aumentar a produtividade.

2.6.1.15.Scrum

Scrum é uma metodologia ágil de gerenciamento de projetos que se concentra na entrega incremental e iterativa de produtos de alta qualidade. Desenvolvido originalmente para a indústria de software, Scrum pode ser aplicado a uma ampla gama de projetos e setores. A metodologia enfatiza a colaboração, a flexibilidade e a melhoria contínua, permitindo que as equipes respondam rapidamente a mudanças e entreguem valor de forma contínua.

Princípios e Valores do Scrum

Transparência: Todos os aspectos significativos do processo devem ser visíveis para aqueles responsáveis pelo resultado. A transparência permite uma comunicação clara e eficaz dentro da equipe.

Inspeção: O progresso em direção aos objetivos do projeto deve ser inspecionado regularmente para detectar variações indesejadas. As revisões regulares ajudam a identificar problemas rapidamente.

Adaptação: Se uma inspeção revela que um ou mais aspectos do processo estão fora dos limites aceitáveis, o processo ou o material sendo produzido deve ser ajustado. A capacidade de adaptação é essencial para a melhoria contínua.

Papéis no Scrum:

Product Owner: Responsável por maximizar o valor do produto e gerenciar o backlog do produto. O Product Owner define e prioriza os requisitos do produto, garantindo que a equipe de desenvolvimento trabalhe nas tarefas mais importantes.

Scrum Master: Serve como facilitador e líder da equipe Scrum. O Scrum Master ajuda a equipe a seguir as práticas e valores do Scrum, remove impedimentos e facilita a comunicação.

Equipe de Desenvolvimento: Consiste em profissionais que trabalham juntos para entregar incrementos do produto. A equipe de desenvolvimento é auto-organizada e multifuncional, possuindo todas as habilidades necessárias para completar o trabalho.

Artefatos do Scrum:

Product Backlog: Uma lista priorizada de tudo o que pode ser necessário no produto, gerenciada pelo Product Owner. O Product Backlog evolui constantemente à medida que o produto e o ambiente de negócios mudam.

Sprint Backlog: Um subconjunto do Product Backlog que a equipe de desenvolvimento se compromete a concluir durante um Sprint. Inclui as tarefas necessárias para transformar os itens do Product Backlog em incrementos de valor.

Incremento: O conjunto de todos os itens do Product Backlog concluídos durante um Sprint e todos os Sprints anteriores, representando um aumento do valor do produto.

Eventos do Scrum:

Sprint: Um ciclo de desenvolvimento fixo, geralmente de duas a quatro semanas, durante o qual um incremento do produto é criado. Cada Sprint deve resultar em um incremento utilizável e potencialmente lançável.

Sprint Planning: Uma reunião no início do Sprint onde a equipe planeja o trabalho a ser feito. O que será entregue no incremento e como o trabalho será realizado são decididos nesta reunião.

Daily Scrum: Uma reunião diária de 15 minutos onde a equipe de desenvolvimento sincroniza atividades e cria um plano para as próximas 24 horas. O objetivo é inspecionar o progresso e ajustar o plano conforme necessário.

Sprint Review: Uma reunião no final do Sprint para inspecionar o incremento e adaptar o Product Backlog, se necessário. A equipe e os stakeholders discutem o que foi feito e o que precisa ser feito a seguir.

Sprint Retrospective: Uma reunião após o Sprint Review onde a equipe reflete sobre o Sprint passado e planeja melhorias para o próximo Sprint. O foco é na melhoria contínua do processo.

Benefícios do Scrum:

Flexibilidade e Adaptação: Scrum permite que as equipes respondam rapidamente a mudanças nos requisitos e prioridades do projeto.

Entrega Contínua de Valor: A abordagem incremental e iterativa garante que incrementos utilizáveis do produto sejam entregues regularmente.

Melhoria Contínua: As retrospectivas incentivam a equipe a refletir sobre seu desempenho e identificar oportunidades de melhoria.

Transparência e Comunicação: As práticas e eventos do Scrum promovem a transparência e a comunicação aberta entre os membros da equipe e stakeholders.

Foco no Cliente: O Product Owner representa os interesses do cliente e garante que a equipe esteja sempre trabalhando nas tarefas de maior valor.

2.6.1.16. Clarity

Clarity é uma ferramenta de análise de comportamento de usuários desenvolvida pela Microsoft, focada em fornecer insights detalhados sobre como os visitantes interagem com um site. Lançada em 2020, Clarity é uma solução gratuita que oferece gravações de sessões, mapas de calor e diversas métricas para ajudar os proprietários de sites a entender melhor a experiência do usuário e identificar áreas para melhorias.

Características Principais:

Gravações de Sessões: Clarity permite que os proprietários de sites assistam gravações de sessões dos usuários, mostrando exatamente como eles navegam pelo site, onde clicam, quanto tempo passam em cada página e outros comportamentos detalhados.

Mapas de Calor: Clarity gera mapas de calor que visualizam onde os usuários clicam, movem o mouse e rolam. Esses mapas ajudam a identificar padrões de interação e áreas de maior interesse ou frustração.

Métricas e Análises: A ferramenta fornece uma variedade de métricas de engajamento, como tempo na página, taxa de rolagem, e cliques mortos (cliques em áreas não interativas). Essas métricas ajudam a avaliar a eficácia do design do site e a usabilidade.

Deteção de Problemas: Clarity pode identificar automaticamente problemas de usabilidade, como cliques rápidos ou hesitações, que podem indicar confusão ou frustração dos usuários.

Filtros e Segmentação: Os usuários podem filtrar e segmentar dados de sessões e mapas de calor por diversos critérios, como dispositivo, navegador, origem de tráfego e comportamento específico, para obter insights mais detalhados e específicos.

Privacidade e Segurança: Clarity é projetado para ser compatível com regulamentos de privacidade, como o GDPR. Ele anonimiza automaticamente dados sensíveis e permite a exclusão de dados pessoais dos usuários conforme necessário.

Benefícios do Clarity:

Insights Detalhados: Com gravações de sessões e mapas de calor, Clarity oferece uma visão detalhada de como os usuários interagem com o site, permitindo a identificação de problemas e oportunidades de melhoria.

Facilidade de Uso: A interface intuitiva de Clarity facilita a configuração e o uso, tornando-a acessível para desenvolvedores, designers e proprietários de sites sem necessidade de treinamento extensivo.

Gratuito: Clarity é uma ferramenta gratuita, tornando-a uma opção atrativa para pequenas empresas e sites pessoais que desejam obter insights sobre a experiência do usuário sem incorrer em custos adicionais.

Compatibilidade com Outras Ferramentas: Clarity pode ser usado em conjunto com outras ferramentas de análise, como Google Analytics, para proporcionar uma visão mais abrangente do comportamento do usuário.

Melhoria da Experiência do Usuário: Ao entender melhor como os usuários interagem com um site, os proprietários podem fazer ajustes informados no design e na funcionalidade para melhorar a experiência do usuário e aumentar a satisfação e conversão.

2.6.1.17.Owasp

OWASP, que significa "Open Web Application Security Project" (Projeto Aberto de Segurança em Aplicações Web), é uma organização sem fins lucrativos focada em melhorar a segurança de software. Fundada em 2001, a OWASP oferece uma ampla gama de recursos, ferramentas, e documentação para ajudar desenvolvedores, arquitetos de software e organizações a entender e mitigar os riscos de segurança em aplicações web.

Principais Objetivos e Atividades da OWASP

Educação e Conscientização:

OWASP Top Ten: Uma lista das dez vulnerabilidades mais críticas em aplicações web. Esta lista é amplamente reconhecida e utilizada como um padrão de referência para segurança de aplicações.

Documentação: Diversos guias, livros e artigos sobre melhores práticas em segurança de software.

Treinamentos e Conferências: OWASP organiza eventos e conferências globais para promover a troca de conhecimento e treinamento em segurança de software.

Ferramentas e Projetos:

OWASP ZAP (Zed Attack Proxy): Uma das ferramentas de segurança de aplicações mais populares, usada para encontrar vulnerabilidades em aplicações web.

OWASP Dependency-Check: Ferramenta que verifica componentes e bibliotecas de software para identificar vulnerabilidades conhecidas.

OWASP ASVS (Application Security Verification Standard): Um framework de verificação de segurança para aplicações, que fornece um conjunto de requisitos para o desenvolvimento seguro.

Comunidade e Colaboração:

Capítulos Locais: OWASP tem capítulos regionais ao redor do mundo que organizam encontros, palestras e workshops para promover a segurança de software em nível local.

Projetos Open Source: A OWASP mantém uma série de projetos de código aberto focados em diferentes aspectos da segurança de software.

Importância da OWASP

A OWASP é uma referência crucial para profissionais de segurança da informação, desenvolvedores e gestores de TI. Suas publicações e ferramentas são amplamente utilizadas para ajudar a identificar, entender e remediar vulnerabilidades em aplicações web. A OWASP também promove a importância da segurança no ciclo de desenvolvimento de software (SDLC) e incentiva a adoção de práticas de desenvolvimento seguro desde as fases iniciais de um projeto.

Recursos Populares da OWASP

OWASP Top Ten:

Fornecer uma visão geral das falhas de segurança mais críticas que afetam as aplicações web.

Atualizado periodicamente para refletir as ameaças emergentes.

OWASP ZAP:

Uma ferramenta de proxy interceptador que ajuda a encontrar vulnerabilidades em aplicações web durante o teste de segurança.

OWASP ASVS:

Um conjunto de normas e requisitos que ajudam a definir e alcançar níveis de segurança apropriados para aplicações web.

Auto Release: O número de versão em auto release refere-se à prática de incrementar e gerenciar automaticamente os números de versão de um software sempre que uma nova release é feita. Isso é crucial para rastrear mudanças, melhorias e correções em um projeto de software de forma sistemática e compreensível. A numeração de versão segue frequentemente o padrão de Versionamento Semântico (SemVer), que é amplamente adotado na indústria de software.

Versionamento Semântico (SemVer)

O Versionamento Semântico (SemVer) é um sistema de numeração de versão que utiliza um formato MAJOR.MINOR.PATCH, onde:

MAJOR: Incrementado quando há mudanças incompatíveis com versões anteriores.

MINOR: Incrementado quando novas funcionalidades são adicionadas de maneira compatível com versões anteriores.

PATCH: Incrementado quando são feitas correções de bugs de maneira compatível com versões anteriores.

Exemplo de Números de Versão

1.0.0: Primeira versão estável.

1.1.0: Nova funcionalidade adicionada de forma compatível.

1.1.1: Correção de bug.

2.0.0: Mudança incompatível com versões anteriores.

Como Funciona no Auto Release

Quando você implementa auto release, o número de versão do seu projeto é automaticamente incrementado com base nas mudanças detectadas. Isso é feito através da análise de commits, mensagens de commit ou tags para determinar se a mudança é uma correção de bug, uma nova funcionalidade ou uma alteração significativa que quebra a compatibilidade com versões anteriores.

Ferramentas para Auto Release

Ferramentas como semantic-release podem automatizar todo o processo de versionamento e lançamento. Elas analisam as mensagens de commit para determinar o tipo de mudança e incrementam o número de versão apropriado. Além disso, essas ferramentas podem gerar changelogs, criar tags de versão no repositório e até publicar pacotes em registries (como npm, PyPI, etc.).

2.6.2. Testes de UI e UX realizados

2.6.2.1. Desenvolvimento de Persona

CRIAÇÃO DA PROTO-PERSONA:

Nome: Jackson Five

Profissão: Motorista

Idade: 35 anos

Gênero: Masculino

Nível de ensino: Ensino médio incompleto

Midas Objetivo: Seu objetivo é encontrar investimentos que ofereçam retorno modesto em alvos grandes de investimentos, diminuindo sua exposição a riscos de perda.

Desafios: Jackson tem uma rotina pesada entre trabalho e criar seus filhos, tendo pouco tempo em análises de investimento de retorno modesto, minimização de riscos e seleção de alvos de investimentos.

Como minha empresa pode ajudá-lo: Criaremos um sistema de investimento automático com base nos filtros utilizados por Jackson, buscando retorno baixo sem exposição ao risco. Sem que ele precisasse ficar boa parte do seu tempo no computador, esse sistema facilitaria seu dia a dia.

Nome: Maria Fatecana

Profissão: Auxiliar de escritório

Idade: 25 anos

Gênero: Feminino

Nível de ensino: Ensino médio completo

Midas Objetivo: Maria pretende conciliar suas atividades de investimento com sua rotina agitada de recém-casada e mãe.

Desafios: O mercado financeiro é complexo e bem volátil, isso exige análise cautelosas e um gerenciamento de risco de acordo com seu perfil, porém Maria tem seu tempo escasso devido sua rotina.

Como minha empresa pode ajudá-la: Com nosso automatizador, Maria poderia configurar suas estratégias de investimentos sem se preocupar com volatilidade do mercado e sem que precise passar boa parte do seu tempo fazendo análises. Dessa forma Maria poderia conciliar tranquilamente sua rotina pesada com seus investimentos.

Nome: Fred Mercurie

Profissão: Médico

Idade: 37 anos

Gênero: Masculino

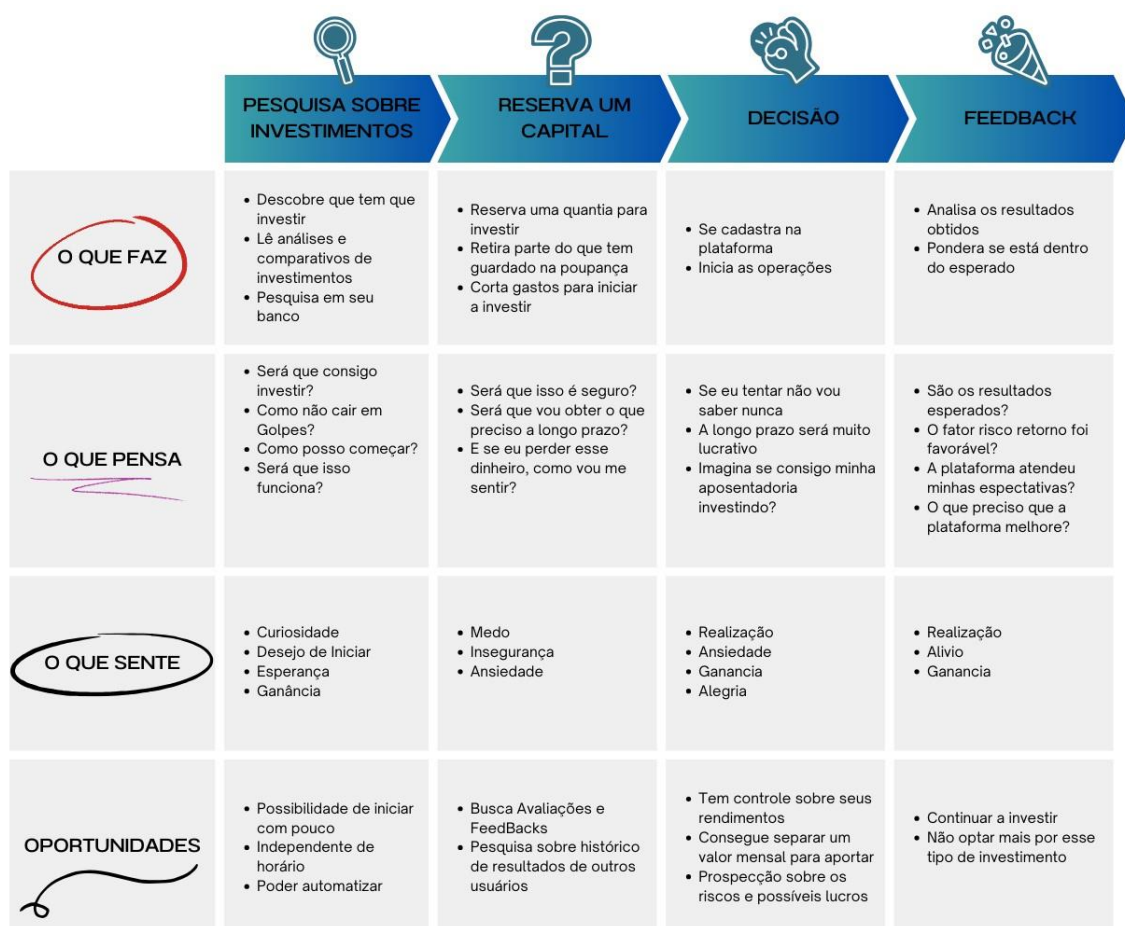
Nível de ensino: Ensino superior completo

Midas Objetivo: Fred pretende alavancar sua reserva financeira por meio de operações de investimento de alta frequência, buscando alvos curtos com potencial de gerar retorno significativo sobre seu capital.

Desafios: Fred divide seu trabalho com seus compromissos do dia a dia, limitando seu tempo de dedicação a análise e execução de suas estratégias de investimento, que requer entendimento detalhado do mercado financeiro.

Como minha empresa pode ajudá-la: Sem que Fred tenha que passar horas do seu dia buscando informações do mercado, nosso sistema buscaria de forma automática os melhores investimentos de acordo com sua estratégia e tolerância ao risco, permitindo-lhe que possa realizar suas tarefas pessoais e profissionais sem preocupações

2.6.2.2. Jornada do Usuário



2.6.2.3. Desenvolvimento de Questionário

01. Sexo:

- Masculino;
- Feminino;

02. Idade:

- entre 16 e 22 anos;
- entre 23 e 29 anos;
- entre 30 e 36 anos;
- 37 anos ou mais;

03. Qual a região de São Paulo que você mora atualmente?

- Centro;

- Zona Norte;
- Zona Sul;
- Zona Leste;
- Zona Oeste

04. Qual o seu grau de escolaridade?

- Ensino fundamental completo;
- Ensino médio completo;
- Ensino superior completo;
- Ensino incompleto;

05. Você já investiu em cripto, ações, fundos imobiliários etc.?

- sim;
- não;

06. Se sim, você se considera um investidor:

- Experiente;
- Iniciante;
- Intermediário;

07. Se não, o que te impediu ou te deixou inseguro?

- Falta de tempo;
 - Inexperiência;
 - Falta de conhecimento;
 - Outros;
- Especifique:

08. Qual o seu perfil, ou qual seria o seu perfil como investidor?

- Conservador - O investidor considerado conservador é aquele que prefere não correr riscos na valorização de perder o dinheiro que decidiu aplicar;
- Moderado - O investidor moderado é aquele que aplica uma parte dos seus recursos em investimentos com menor estabilidade e liquidez, com riscos médios em suas variações tendo retorno de médio a longo prazo;
- Arrojado/agressivo - Arrojado é aquele investidor que aceita os riscos de variação em seus rendimentos ou até mesmo alterações em seu capital investido inicialmente, acreditando que toda movimentação trará resultado positivo ou de pelo menos reequilíbrio;

09. Como o seu tempo é administrado para avaliar seus investimentos e o mercado?

- Observo esporadicamente, ou quando as pessoas indicam algum investimento em potencial;
- Observo semanalmente, acompanho noticiários para estar por dentro de eventos que potencialize algum investimento;
- Não observo, não tenho tempo;

10. Como você avalia uma ferramenta de automação para investimento em Cryptomoedas?

- Atrativo;

- Indiferente;
- Não atrativo;

Aplicação da pesquisa e coleta de resultados:

Experiência do Usuário | Plataforma para investimento em cryptomoedas

32
Respostas

02:48
Tempo médio para concluir

Ativo
Status

Revisar respostas

Lançar pontuação

Abrir no Excel ...

1. Selecione seu sexo (0 ponto)

[Mais Detalhes](#)

Insights

Masculino	13
Feminino	19
Outro	0



2. Selecione sua idade (0 ponto)

[Mais Detalhes](#)

Insights

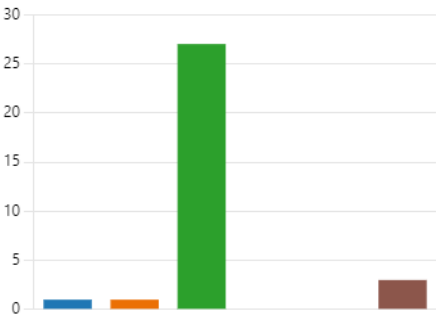
Entre 16 e 22 anos	13
Entre 23 e 29 anos	8
Entre 30 e 36 anos	8
37 anos ou mais	3



3. Qual região de São Paulo você mora atualmente? (0 ponto)

[Mais Detalhes](#)

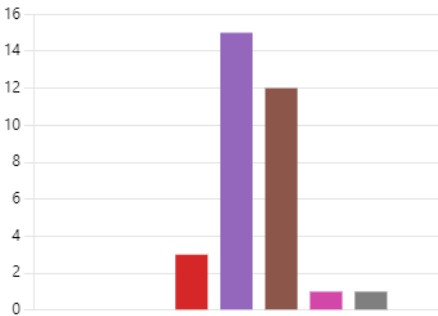
Zona Sul	1
Zona Leste	1
Zona Oeste	27
Zona Norte	0
Centro	0
Não moro em São Paulo	3



4. Qual seu grau de escolaridade? (0 ponto)

[Mais Detalhes](#)

Ensino Fundamental Incompleto	0
Ensino Fundamental Completo	0
Ensino Médio Incompleto	0
Ensino Médio Completo	3
Ensino Superior Incompleto	15
Ensino Superior Completo	12
Ensino Técnico	1
Ensino Técnico incompleto	1
Outro	0



5. Você já investiu em crypto, ações fundos imobiliários, etc? (0 ponto)

[Mais Detalhes](#)

[Insights](#)

Sim	12
Não	20



6. Você se considera um investidor: (0 ponto)

[Mais Detalhes](#)

Experiente	0
Iniciante	7
Intermediário	5



7. O que te impediu ou deixou inseguro? (0 ponto)

[Mais Detalhes](#)

[Insights](#)

Falta de tempo	2
Inexperiência	6
Falta de conhecimento	20
Outros	4



8. Qual seu perfil ou qual seria o seu perfil como investidor? (0 ponto)

[Mais Detalhes](#)

- Conservador - O investidor cons... 22
- Moderado - O investidor moder... 6
- Arrojado/agressivo - Arrojado é ... 4



9. Como o seu tempo é administrado para avaliar seu investimentos e o mercado? (0 ponto)

[Mais Detalhes](#)

[Insights](#)

- Observo esporadicamente, ou q... 7
- Observo semanalmente, acomp... 5
- Não observo, não tenho tempo; 20



10. Como você avalia uma ferramenta de automação para investimento em criptomoedas? (0 ponto)

[Mais Detalhes](#)

- Atrativo 17
- Indiferente 12
- Não atrativo 3



Revisão e alteração de todos os artefatos criados anteriormente:

Jackson Five:

Antes: Seu objetivo era encontrar investimentos com retorno modesto e minimizar riscos.

Mudança: Os desafios agora destacam seu tempo limitado com à rotina pesada entre trabalho e cuidar dos filhos, o que dificulta a realização de análises de investimento.

Maria Fatecana:

Antes: Seu objetivo era conciliar investimentos com sua rotina agitada.

Mudança: Os desafios agora enfatizam seu tempo escasso devido à rotina agitada e a dificuldade do mercado financeiro, isso requer análises minuciosas e um gerenciamento de risco bem elaborado e estudado.

Fred Mercurie:

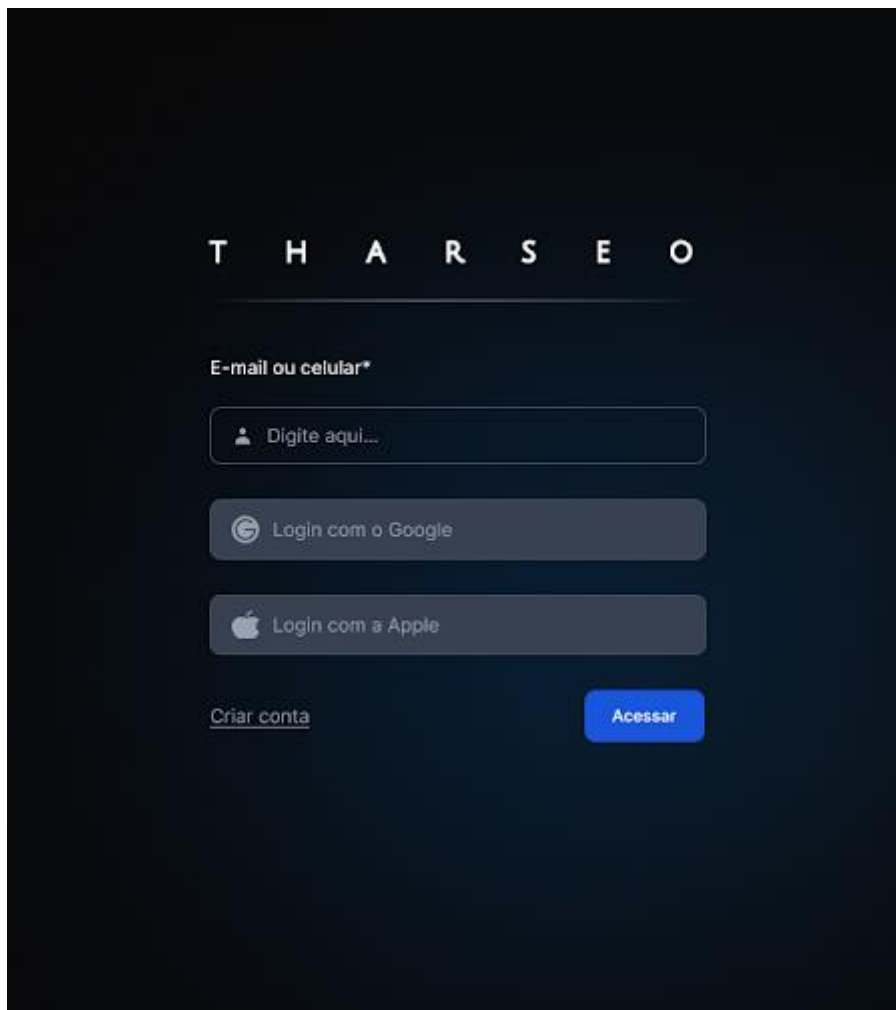
Antes: Seu objetivo era alavancar sua reserva financeira por meio investimentos mesmo que arriscasse mais.

Mudança: Os desafios agora ressaltam seu tempo limitado devido aos compromissos profissionais e pessoais, dificultando o acompanhamento detalhado do mercado financeiro.

2.6.2.3. Teste qualitativo em relação a tela desenvolvida

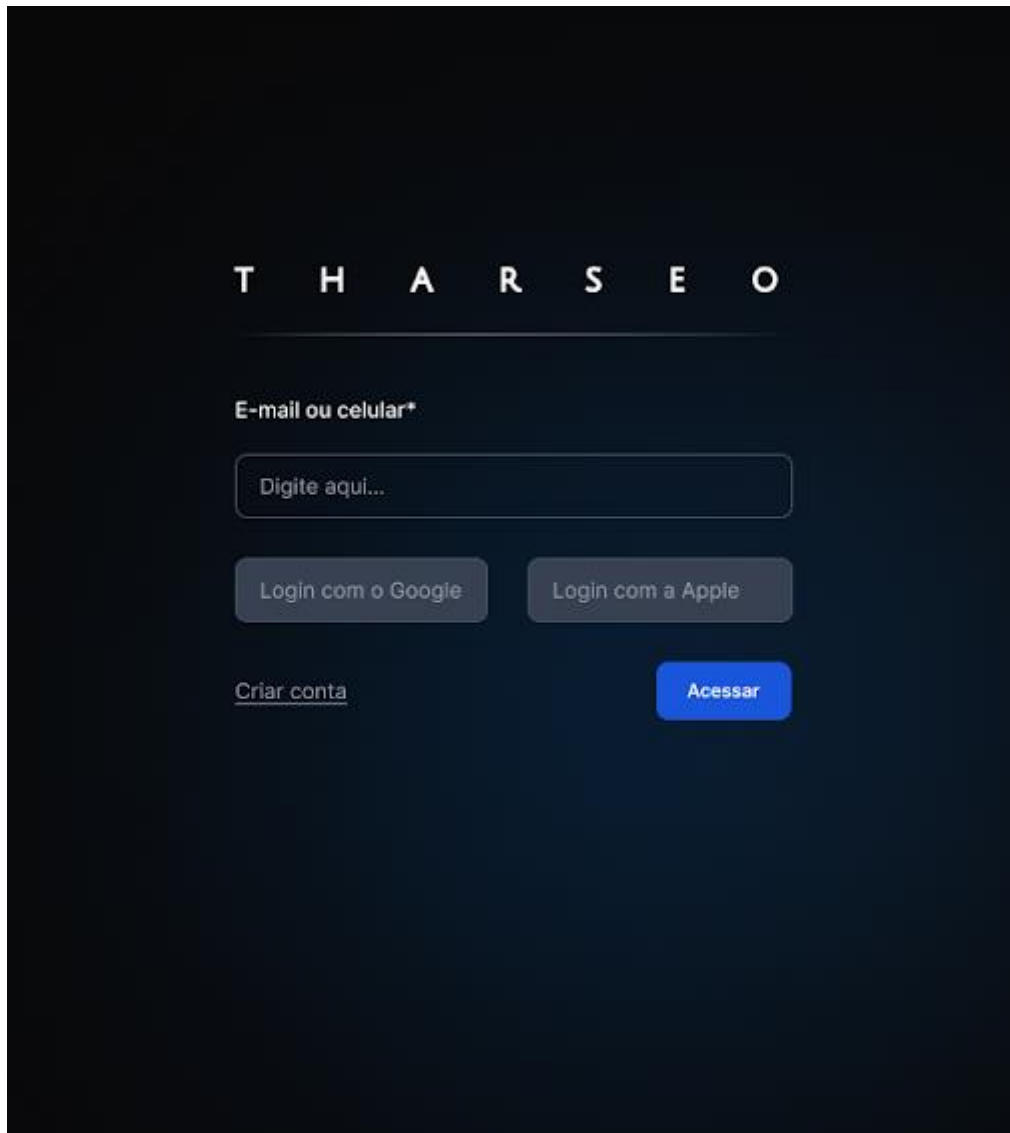
Versão Original

Partindo desse princípio, temos o protótipo da tela original onde contém todas as opções necessária para o usuário efetuar login. Temos o logo na parte superior com uma paleta de cores voltada para tecnologia e elementos que condizem com o tema. Temos um layout em coluna com os elementos com aspecto de transparência, outro fator, são os elementos internos dos botões que dão opção de login com integração com google e apple, apresentam também os ícones. Opção para criar conta discreta na parte inferior e um botão em destaque sugerindo ao usuário o próximo passo.



Versão Modificada

Na Versão modificada, trazemos o mesmo tema e cores, porém com menos detalhes de elementos que a tela anterior. Retiramos os ícones internos dos botões e alteramos o layout para disposição linear.



- i. Definição de métricas para avaliar o desempenho das interfaces.

Uma vez com nossas telas, escolhemos a métrica:

- Feedback Qualitativo do Usuário
- ii. Execução do teste com um grupo de usuários, onde cada um é direcionado aleatoriamente para uma das versões.

Qual interface para login é mais agradável?

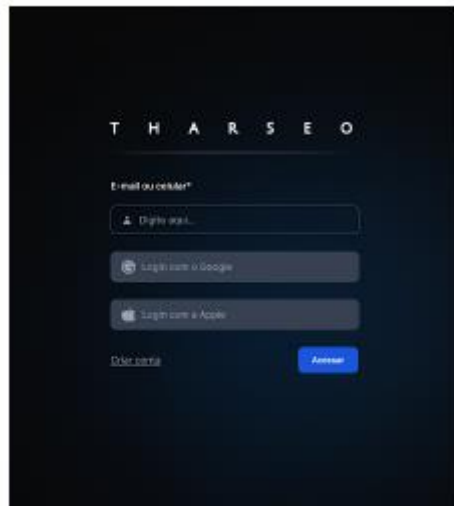
richard.franguedes@hotmail.com [Mudar de conta](#)



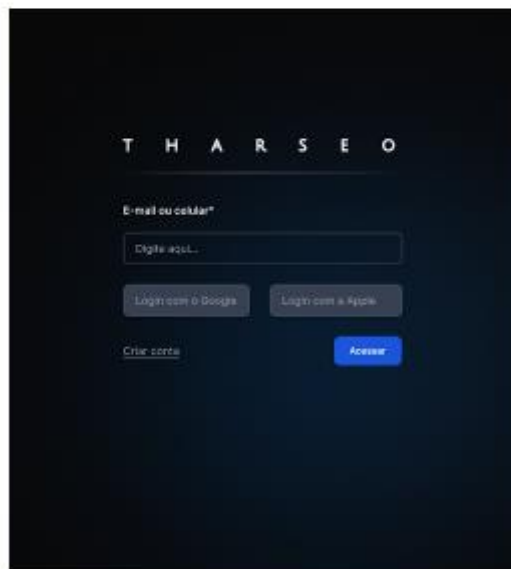
Não compartilhado

* Indica uma pergunta obrigatória

Opção A



Opção B



Selecione *

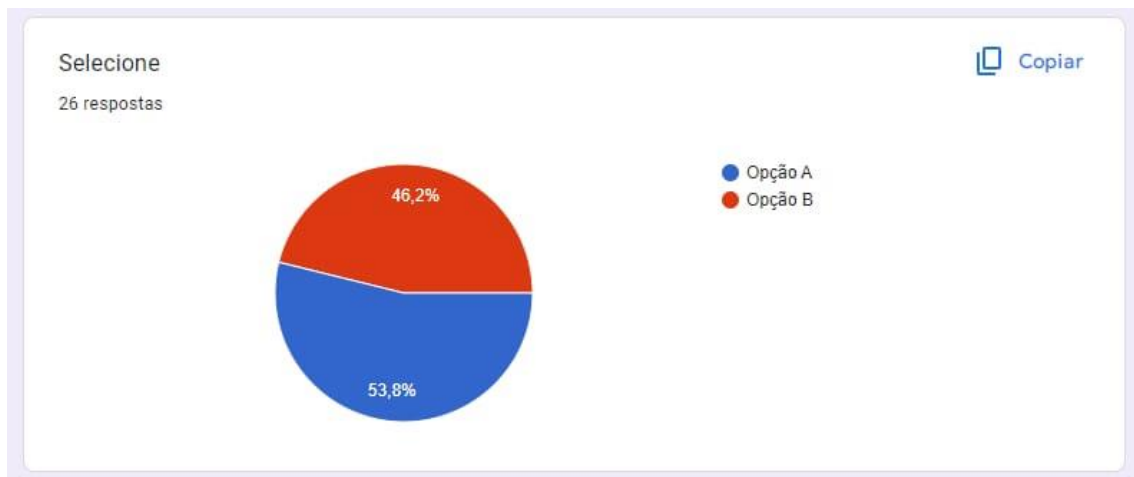
☐ Opção A

☐ Opção B

Enviar

[Limpar formulário](#)

Análise dos resultados e comparação das métricas entre as versões.



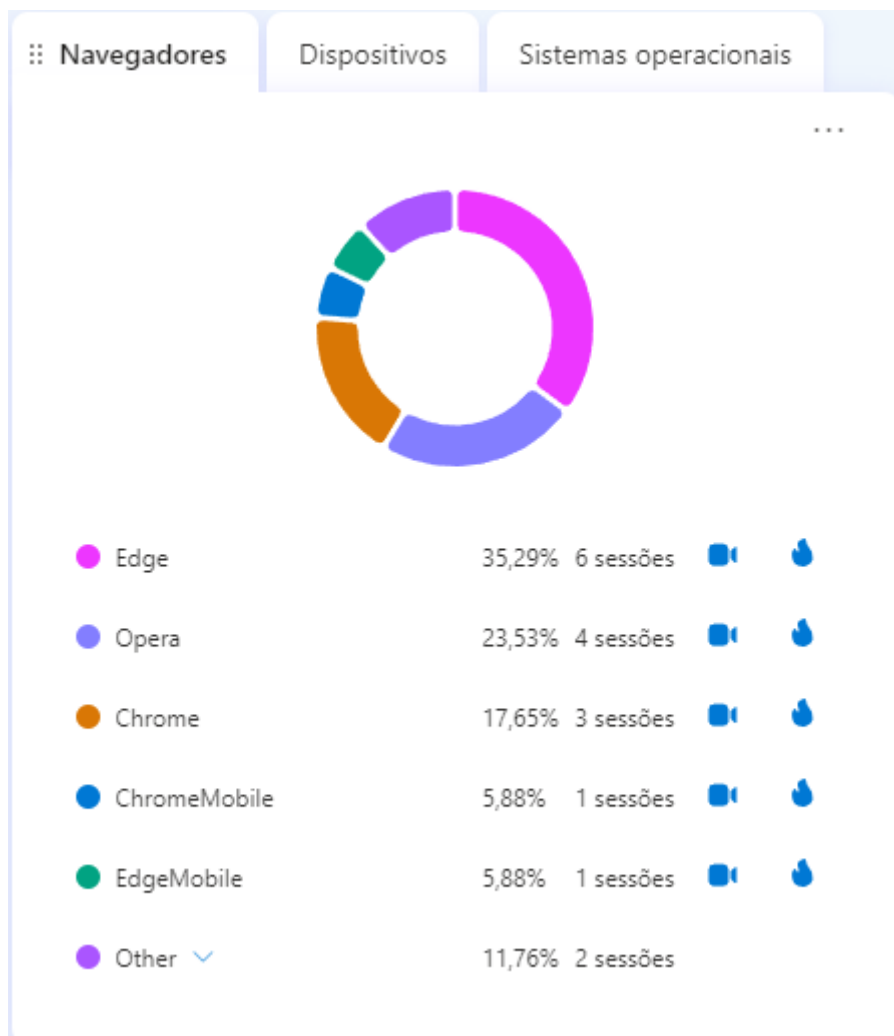
Com base no resultado da pesquisa com foco em um feedback qualitativo do usuário, temos um resultado de 53,80% para a interface original contra 46,20% para a interface modificada. A partir desses dados podemos concluir que a preferência, baseado na amostra de dados coletados, permanece com a interface original para a maioria.

Fizemos nossa análise utilizando o Microsoft Clarity que é gratuito. Subimos um servidor em nosso desktop particular e disponibilizamos através do ngrok para acesso externo.

Através do alcance da campanha, tivemos o acesso de 12 usuários sendo feitas em 17 sessões.



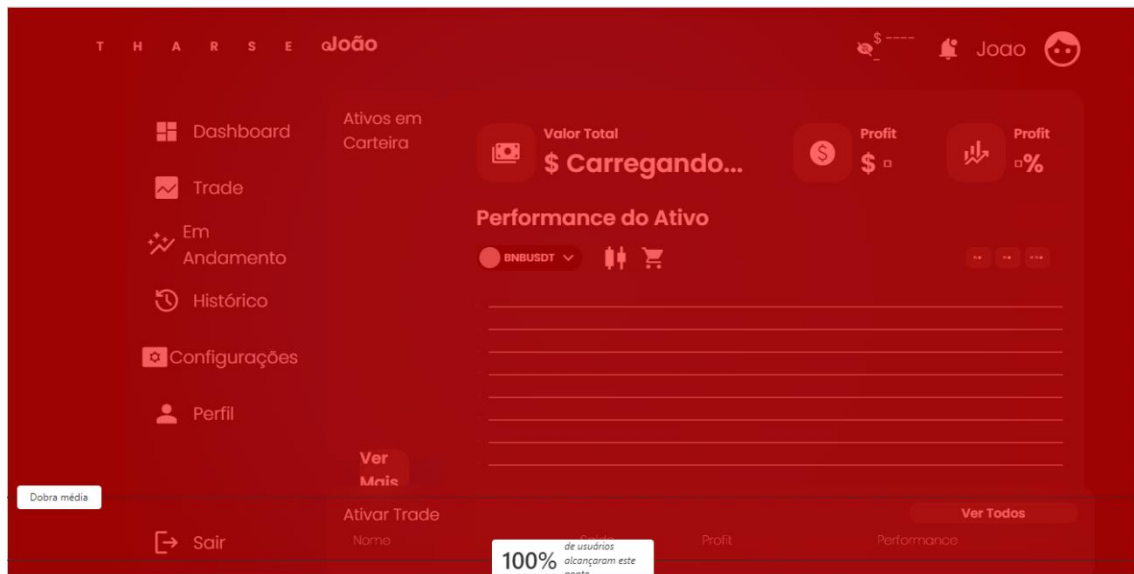
Dos navegadores utilizados, abaixo temos a relação de todas as visitas do site:



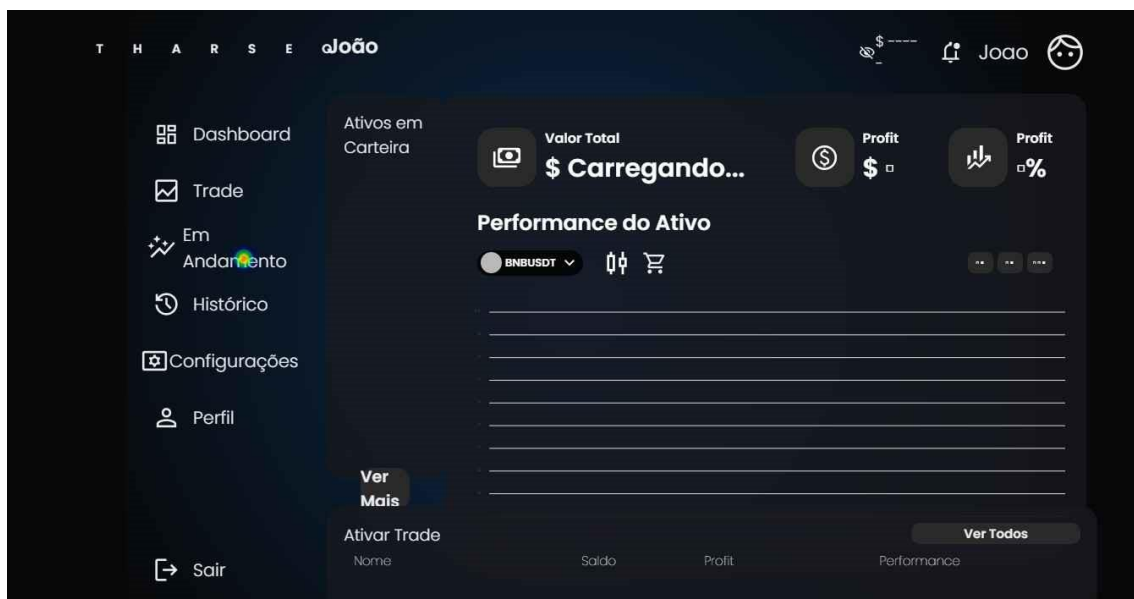
Rolagem de Página

Das 17 sessões, obtemos uma rolagem de 95,35%. O que já era esperado devido ao nosso layout ter foco para ocupar apenas uma página. Dessa forma, todas as informações já estariam categorizadas e disponíveis para o usuário dentro da própria página sem o problema de estar rolando a página.

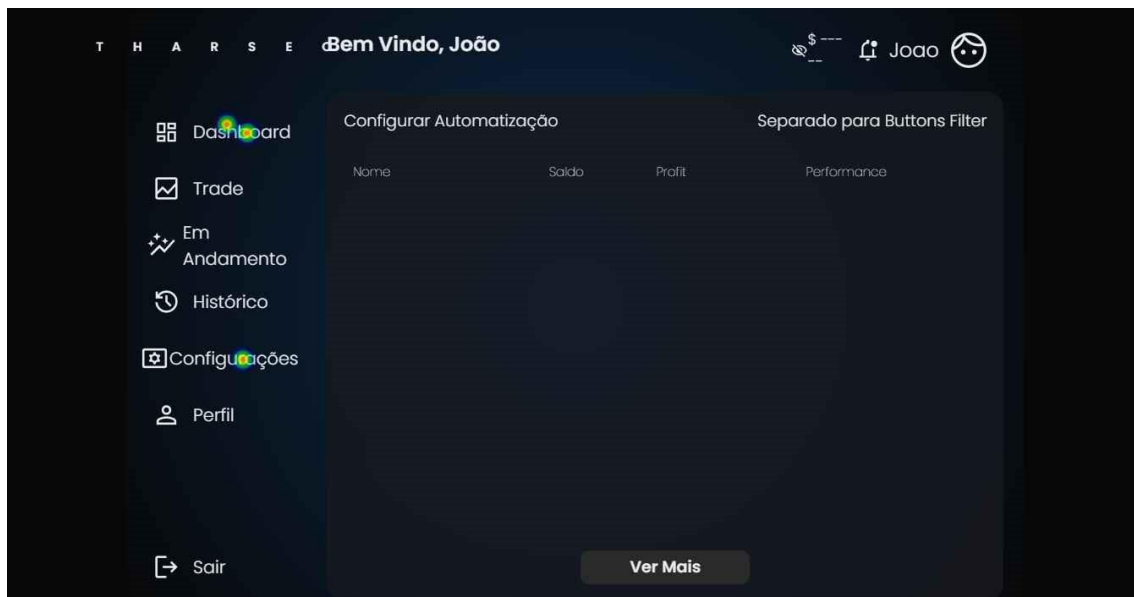




Quais informações importantes não estão sendo visualizadas



Nossa Dashboard foi a página mais acessada, porém ainda temos implantações a realizar, por exemplo, quando o usuário acessa a aplicação, assim como toda aplicação que consome dados em um servidor, temos um pequeno delay para o retorno dos mesmos, as informações que não estão sendo visualizadas são os ativos e gráficos devido ao usuário já clicar em outra tela.



Em nossa tela de trades, acontece a mesma situação, como os dados não foram carregados existe a desistência do usuário em aguardar o recebimento dos mesmos.

A solução para o problema é a tela de um pre-load para a aplicação, dessa forma em 2 segundos que os dados forem carregados do servidor, a tela com todos elementos e informações serão disponibilizados ao usuário.

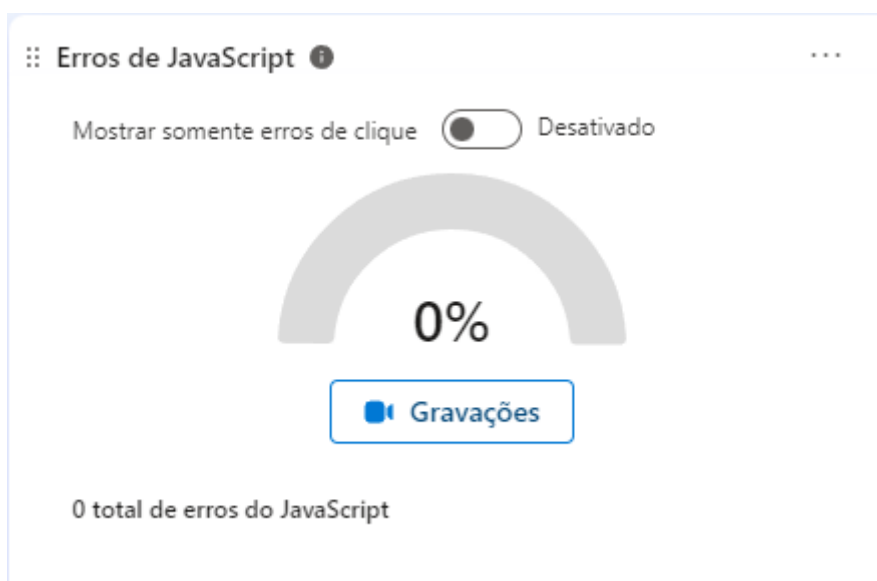
Que outros pontos estão “roubando a atenção” e podem ser retirados

Não existem pontos que estão roubando a atenção, de todas as telas os elementos mais acessados são os botões do menu, o que confirma a navegação do usuário entre as telas.

Quais áreas do site estão funcionando como deveriam

O menu está funcionando da forma que deveria levando o usuário para as páginas requisitadas sem apresentar bugs.

Outro fator obtido pelo Clarity é a quantidade de erros observados durante a navegação:



Tivemos um resultado positivo, obtendo 0% de erros ao navegar entre as páginas.

Que outros pontos estão “roubando a atenção” e podem ser retirados.

Temos um elemento que está oculto no gráfico sem a devida atenção: Comprar ou Vender Manualmente. Como não há sinalização, o acesso a função acaba sendo deixado de lado e muitas vezes não visto pelo usuário:



2.6.3. Pipeline e Testes automatizados

Um pipeline de desenvolvimento é um conjunto automatizado de processos que facilita a construção, teste e implantação de software. Ele segue os princípios de Integração Contínua (CI) e Entrega Contínua (CD) para garantir que as alterações no código sejam integradas, testadas e entregues de maneira rápida e eficiente. Um pipeline típico de desenvolvimento pode incluir as seguintes etapas:

1. Compilação (Build): O código-fonte é compilado em binários executáveis.
2. Testes Automatizados: O código é submetido a uma série de testes automatizados, como testes unitários, de integração e de aceitação, para garantir sua qualidade e funcionalidade.
3. Implantação (Deploy): O software é implantado em ambientes de teste e, eventualmente, em produção.
4. Monitoramento: Após a implantação, o software é monitorado para detectar quaisquer problemas em tempo real.

Automação de Testes

A automação de testes é o uso de ferramentas e scripts para executar testes de software automaticamente, sem intervenção manual. O objetivo é aumentar a eficiência e a cobertura dos testes, reduzindo o tempo e os custos associados ao processo de testes manual. Existem vários tipos de testes que podem ser automatizados:

Testes Unitários: Testam unidades individuais de código, como funções ou métodos, para garantir que funcionem corretamente isoladamente.

Testes de Integração: Verificam se diferentes módulos ou componentes do sistema funcionam corretamente juntos.

Testes Funcionais: Avaliam se o sistema cumpre seus requisitos funcionais, focando na interface do usuário e na interação do sistema.

Testes de Regressão: Garantem que novas alterações no código não introduzam novos bugs em funcionalidades previamente testadas.

Testes de Desempenho: Avaliam a performance do sistema sob diversas condições, como carga e estresse.

Benefícios

Qualidade e Confiabilidade: A automação de testes e os pipelines de desenvolvimento garantem que o software seja testado de maneira consistente e abrangente, aumentando a qualidade e a confiabilidade do produto.

Rapidez e Eficiência: Automatizar o pipeline de desenvolvimento e os testes permite ciclos de desenvolvimento mais rápidos, reduzindo o tempo de entrega do software.

Deteção Precoce de Erros: Erros e bugs são identificados e corrigidos mais cedo no ciclo de desenvolvimento, antes que se tornem problemas críticos.

Redução de Custos: Embora a configuração inicial possa exigir investimento, a automação reduz significativamente os custos de testes e manutenção a longo prazo.

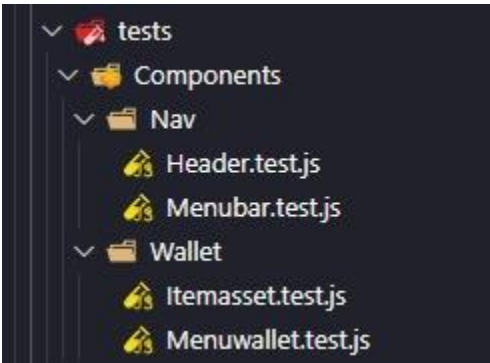
Fizemos os testes unitários do React através do próprio sistema do React, onde testamos o Header, Menu, Menu Wallet e o Item asset, abaixo imagens do teste executado de forma automatizada:

```
PS D:\Faculdade\OneDrive - Fatec Centro Paula Souza\Projetos\Tharseo\frontend\tharseo> npm test

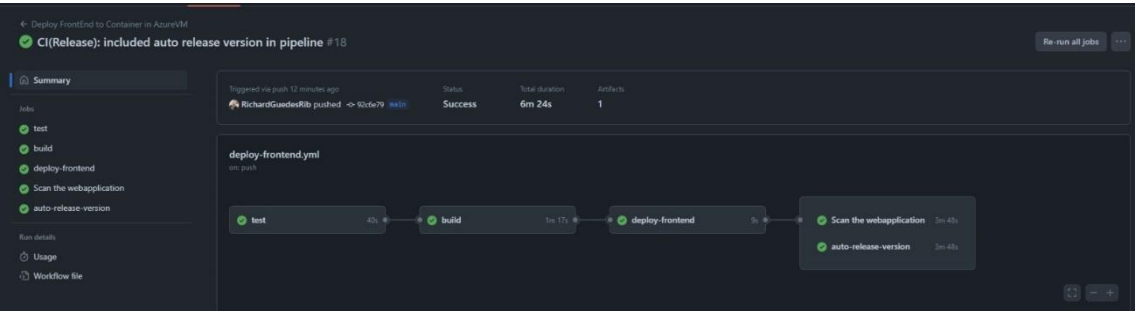
> tharseo@0.1.0 test
> jest

PASS src/tests/Components/Wallet/Itemasset.test.js
PASS src/tests/Components/Nav/Header.test.js
PASS src/tests/Components/Nav/Menuabar.test.js
PASS src/tests/Components/Wallet/Menuwallet.test.js

Test Suites: 4 passed, 4 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        6.719 s, estimated 16 s
Ran all test suites.
```



Fizemos também o teste automatizado com test, build e deploy no Vercel (imagem abaixo):



Como adicional fizemos o teste de segurança através do OWASP para verificar os pontos de vulnerabilidades para futuras melhoras, buscando assim sempre a melhoria contínua do nosso projeto em todos os aspectos:

ZAP Scanning Report

Sites: <https://104.45.141.47> <http://104.45.141.47:8080> <http://104.45.141.47>

Generated on Sun, 26 May 2024 21:53:51

ZAP Version: 2.15.0

ZAP is supported by the [Crash Override Open Source Fellowship](#)

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	4
Low	2
Informational	2
False Positives	0

Alerts

Name	Risk Level	Number of Instances
Cloud Metadata Potentially Exposed	High	1
Content-Security-Policy (CSP) Header Not Set	Medium	9
HTTP Only Site	Medium	1
Missing Anti-clickjacking Header	Medium	9
Relative Path Confusion	Medium	12
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	11
X-Content-Type-Options Header Missing	Low	11
Information Disclosure - Suspicious Comments	Informational	1
Modern Web Application	Informational	9

Alert Detail

High	Cloud Metadata Potentially Exposed
Description	The Cloud Metadata Attack attempts to abuse a misconfigured NGINX server in order to access the instance metadata maintained by cloud service providers such as AWS, GCP and Azure. All of these providers provide metadata via an internal unrouteable IP address '169.254.169.254' - this can be exposed by incorrectly configured NGINX servers and accessed by using this IP address in the Host header field.
URL	http://104.45.141.47/latest/meta-data/

2.6.4. Conclusão

Através das ferramentas utilizadas conseguimos desenvolver boa parte de uma potencial aplicação com rapidez e buscando sempre a maior assertividade possível, tanto em relação ao código (front, back, banco de dados) quanto em relação a expectativa do usuário/ cliente, buscando o aperfeiçoamento e segurança quanto a proteção de dados e segurança da carteira e investimentos dos futuros usuários e tudo da forma mais automatizada possível gerando produtividade e entrega de forma continua sempre com o foco no usuário buscando um sistema intuitivo e de fácil entendimento simplificando o máximo possível as principais funcionalidades e objetivos da aplicação.

Tudo isso foi possível através dos conhecimentos obtidos de forma conjunta com as disciplinas lecionadas no semestre e através dos orientadores das disciplinas de Integração e entrega continua, Laboratório Web e Experiência do Usuário.

Bibliografia

https://www.anbima.com.br/pt_br/imprensa/volume-investido-por-brasileiros-cresce-9-7-em-2023-e-chega-a-r-5-5-trilhoes.html. Acesso em: 03/05/2024.

https://www.anbima.com.br/pt_br/noticias/raio-x-do-investidor-brasileiro-chega-a-setima-edicao-com-novos-levantamentos-sobre-estresse-financeiro-e-apostas-online-8A2AB2AE8F2C5D7F018F2CFA8A016764-00.html. Acesso em: 03/05/2024.

https://www.anbima.com.br/pt_br/especial/raio-x-do-investidor-brasileiro.html. Acesso em: 03/05/2024.

CUNHA, Fernando. Disponível em: <https://mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>. Acesso em: 03/05/2024.

DEVMEDIA Disponível em: <https://www.devmedia.com.br/elicitacao-de-requisitos-levantamento-de-requisitos-e-tecnicas-de-elicitacao>. Acesso em: 03/05/2024.

Engenharia de Software Moderna, Disponível em: engsoftmoderna.info/cap3.html. acesso em: 04/05/2024.

Java Script o Guia definitivo, Bookman. David Flanagan

MySQL: Comece com o principal banco de dados, casa do código, Vinicius Carvalho

<https://blog.rocketseat.com.br/websocket-o-que-e-e-quando-usar/#:~:text=WebSocket%20%C3%A9%20uma%20tecnologia%20de,eficiente%20e%20em%20tempo%20real>. Acesso em:03/05/2024

<https://blog.somostera.com/desenvolvimento-web/comandos-git#:~:text=O%20git%20%C3%A9%20um%20sistema,as%20mudan%C3%A7as%20para%20evitar%20erros>. Acesso em:10/05/2024

<https://www.hostinger.com.br/tutoriais/o-que-e-docker#:~:text=O%20Docker%20%C3%A9%20um%20software,Linux%20e%20macOS%20C%20sem%20problemas>. Acesso em 12/05/2024.

https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-azure/?&ef_id=_k_Cj0KCQjw0_WyBhDMARIsAL1Vz8v2KJnUBU9uMLb2OLjgZfrMdjCoRiw5_LaxFhbQ1OEu3vvfsuya0voaAq6xEALw_wcB_k_&OCID=AIDcmmzmnb0182_SEM__k_Cj0KCQjw0_WyBhDMARIsAL1Vz8v2KJnUBU9uMLb2OLjgZfrMdjCoRiw5_LaxFhbQ1OEu3vvfsuya0voaAq6xEALw_wcB_k_&gad_source=1&gclid=Cj0KCQjw0_WyBhDMARIsAL1Vz8v2KJnUBU9uMLb2OLjgZfrMdjCoRiw5_LaxFhbQ1OEu3vvfsuya0voaAq6xEALw_wcB. Acesso em:15/05/2024