



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUAN MACHADO PONTES

STARBUS - UMA PLATAFORMA DE DADOS SOBRE O TRANSPORTE PÚBLICO

TERESINA, PIAUÍ

2020

LUAN MACHADO PONTES

STARBUS - UMA PLATAFORMA DE DADOS SOBRE O TRANSPORTE PÚBLICO

Projeto apresentado à Banca Examinadora como requisito para aprovação na disciplina de Trabalho de Conclusão de Curso II do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí.

Orientador: Prof. Dr. Adalton Sena

TERESINA, PIAUÍ

2020

LUAN MACHADO PONTES

STARBUS - UMA PLATAFORMA DE DADOS SOBRE O TRANSPORTE PÚBLICO

Projeto apresentado à Banca Examinadora como requisito para aprovação na disciplina de Trabalho de Conclusão de Curso II do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí.

Aprovado pela banca examinadora em _____.

BANCA EXAMINADORA:

Prof. Dr. Adalton Sena

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

Prof. M^º. TBD

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

Prof. M^º. TBD

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

TERESINA, PIAUÍ

2020

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama UML de Casos de Uso	11
Figura 2 – Diagrama entidade-relacionamento	13
Figura 3 – Tela de Autenticação	14
Figura 4 – Tela Principal	15
Figura 5 – Tela Principal	16
Figura 6 – Tela de Histórico	17
Figura 7 – Tela de Ranking	18
Figura 8 – Tela de Negociações	19

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
WWW	World Wide Web
REST	Representational State Transfer
HTML5	Hypertext Markup Language, versão 5
JSON	JavaScript Object Notation
CSS	Cascading Style Sheets
MVC	Model View Controller
CDI	Contexts and Dependency Injection
ORM	Object/Relational Mapping
JDBC	Java DataBase Connection
SGBD	Sistema de Gerenciamento de Banco de Dados
UUID	Universally Unique Identifier

SUMÁRIO

1	INTRODUÇÃO	6
1.1	Objetivos	7
1.1.1	Objetivo Geral	7
1.1.2	Objetivos Específicos	7
2	TECNOLOGIAS ENVOLVIDAS	8
3	MODELAGEM DO PROJETO	10
3.1	Levantamento de Requisitos	10
3.1.1	Requisitos Funcionais	10
3.1.2	Requisitos Não Funcionais	10
3.2	Diagramas de casos de uso	10
3.3	Diagramas de classe	12
3.4	Arquitetura do Sistema	12
3.5	Diagrama de Entidades-Relacionamentos	12
3.6	Interface	13
3.6.1	Tela de Autenticação	13
3.6.2	Tela Principal	14
3.6.3	Tela de Perfil	15
3.6.4	Tela de Histórico	16
3.6.5	Tela de Ranking	17
3.6.6	Tela de Negociações	18
4	SOFTWARE	20
4.1	Implantação	20
4.2	Testes	20
5	CONSIDERAÇÕES FINAIS	21
5.1	Trabalhos Futuros	21
	REFERÊNCIAS	22

1 INTRODUÇÃO

Após a Segunda Guerra Mundial, os processos de urbanização e industrialização do Brasil, levaram milhões de pessoas do campo à cidade. Segundo o IBGE, nos anos 60, mais da metade da população brasileira era considerada rural, pouco mais de 50 anos depois, 84,4% da população brasileira reside em áreas urbanas [1]. Os investimentos em infraestruturas não acompanharam o ritmo acelerado dessa mudança, o que levou ao agravamento dos problemas urbanos, incluindo os relacionados à mobilidade e mais diretamente o transporte público.

Diversos problemas afetam os passageiros em seu dia a dia, no entanto a superlotação provavelmente é o que mais interfere na qualidade de vida dos usuários. Algumas empresas de ônibus, a fim de solucionar tal problema, colocam diversos veículos na mesma linha em um curto intervalo de tempo durante os horários de pico, mas a ansiedade do usuários de chegar em casa, a instabilidade do horário dos ônibus e a incerteza se o ônibus extra já passou ou não, faz com que os passageiros normalmente peguem o primeiro veículo disponível, causando o mau aproveitamento da frota. Nessa caso a falta de informação sobre os veículos disponíveis tem impacto direto sobre qualidade de vida dos usuário.

A espera de ônibus faz os usuário se exporem muitas vezes a uma situação de risco e desconforto. Cada minuto em um ponto de ônibus o passageiro fica exposto a intempéries do tempo, como chuva e sol e não raro, assaltos são registrados em tais locais. Muitas vezes o usuário obriga-se a estar na parada muito tempo antes do ônibus passar, por vezes, deixando de fazer suas atividades, perdendo sono ou deixando de passar mais tempo com a família, caracterizando outro impacto negativo que falta de informação sobre o transporte público pode ter sobre a qualidade de vida das pessoas. Aumentar a frota de veículos para diminuir o tempo de espera pode ser economicamente inviável, mas oferecer a informação da quantidade de veículo disponíveis, bem como suas respectivas localizações e horários, pode ajudar a diminuir a quantidade de tempo que o usuário precisará ficar na parada sem elevar consideravelmente os custos operacionais, possibilitando até mesmo a diminuição dos preços para o usuário final por tornar possível a diminuição da frota.

Do ponto de vista das autoridades que cuidam do sistema como um todo, existe a dificuldade de levantar problemas em pontos específicos do sistema. Por exemplo, como o órgão responsável pelos os pontos de ônibus da cidade pode saber se os tais atendem as expectativas dos usuários? Ou como saber a visão do usuário a respeito da frota de veículos de uma determinada linha ou empresa? Atualmente cada órgão possui seu canal de contato com o usuário, que poderiam colher tais informações, no entanto tais canais requerem um esforço considerável do usuário, que pode desestimular o engajamento e a perda do sentimento que levaria ele a apontar problemas. Por outro lado as atuais plataformas digitais permitem o aproveitamento do sentimento de engajamento de forma imediata e com um esforço consideravelmente menor.

O presente trabalho descreve o desenvolvimento de uma plataforma que ofereça informações sobre o transporte público a desenvolvedores de aplicativos móveis, além oferecer um repositório centralizado de avaliações realizadas por usuário sobre diversos aspectos do transporte público. Com tal ferramenta os usuários poderão ter acesso fácil a informações sobre os horários, itinerários, localização de veículos e linhas por paradas, que podem ajudar a resolver os problemas citados anteriormente. Por outro lado, os mesmos poderão conceber uma visão sobre sistema transporte público local, por avaliarem pontos de ônibus ou estado de conservação dos veículos, informações essas que podem ser úteis as entidades que cuidam do transporte público da cidade.

1.1 OBJETIVOS

Abaixo são listados os objetivos gerais e específicos do presente trabalho.

1.1.1 Objetivo Geral

O objectivo geral deste trabalho é descrever o desenvolvimento de uma plataforma que centralize os dados do transporte público de Teresina e que facilite o desenvolvimento de novas aplicações voltadas para a melhoria da mobilidade urbana.

1.1.2 Objetivos Específicos

- Descrever os processo de desenvolvimento de uma REST API web, incluindo as decisões técnicas tomadas para a mesma.
- Demonstrar o funcionamento da plataforma criada.
- Expor a documentação da API exposta.

2 TECNOLOGIAS ENVOLVIDAS

Pode-se dividir as tecnologias utilizadas na construção da plataforma Starbus em três grupos de ferramentas: auxílio ao desenvolvimento, desenvolvimento e suporte. O primeiro grupo abrange as ferramentas utilizadas para a construção da plataforma em si, o segundo auxiliou a utilização das ferramentas do primeiro grupo, por fim o último grupo que permitiu colocar a solução em produção para colhermos os resultados do trabalho.

Software é desenvolvido utilizando uma linguagem de programação, que deve obedecer alguns critérios relativos ao problema a ser solucionado. No caso deste trabalho os critérios foram: uma linguagem de curta curva de aprendizagem, dispondo de um conjunto de ferramentas maduras e com abordagem pragmática para desenvolvimento. A linguagem Ruby em sua versão 2.2, apresentou excelente desempenho nesses critérios, principalmente no que diz respeito a seu ferramental que tem influenciado toda a comunidade de desenvolvimento de software com sua praticidade e boas práticas. Um exemplo disso, é o framework Grape que disponibiliza uma série de soluções específicas para o desenvolvimento de RESTs APIs, que chama a atenção por sua praticidade e simplicidade. As ferramentas citadas acima juntamente com o banco de dados Postgres na versão 10 fazem parte do grupo de ferramentas de desenvolvimento utilizadas para implementar a plataforma Starbus.

O desenvolvimento utilizando as tecnologias acima pode ser agilizado utilizando um segundo grupo de ferramentas que auxiliam no desenvolvimento e na entrega com qualidade. Esse segundo grupo é composto por ferramentas de auxílio de desenvolvimento, que podemos destacar das utilizadas no desenvolvimento desse projeto o editor de texto Vim que possui uma série de funcionalidades para manipulação de código bem como o suporte a linguagem Ruby por meio highlight e autocomplete. Outra funcionalidade interessante desse editor é o fato de roda direto na linha de comando, o que permite alternar entre a edição de código e a execução de tasks em terminal, tão frequentes no desenvolvimento utilizando Ruby. A biblioteca Rspec foi fundamental para a escrita de testes automatizados, pela sua facilidade, maturidade e principalmente expressividade. Outras ferramentas utilizadas para suporte ao desenvolvimento foi o Git que permite o versionamento de código, que juntamente com o site Github, permitiu armazenar de forma segura e compartilhada o código da aplicação durante e pós desenvolvimento. Para acesso e manipulação de dados no Postgres foi utilizada a IDE DBeaver na versão 5.2 que prover uma interface amigável e completa para validação de estrutura de banco de dados bem execução de queries SQL.

Por fim, para executar a plataforma Starbus em ambiente produtivo e assim atingir os objetivos específicos do trabalho, procurou-se uma ferramenta que pudesse trazer a menor carga de trabalho relacionada a infra estrutura de aplicação, visto que o foco do trabalho deveria ser no desenvolvimento da solução. Assim sendo, a plataforma Heroku foi a solução mais viável,

visto que oferece um ambiente extremamente flexível e escalável com apenas poucas linha de configuração. Por exemplo, para se colocar no ar uma nova versão da aplicação basta realizar um git push para um determinado repositório da ferramenta.

3 MODELAGEM DO PROJETO

3.1 LEVANTAMENTO DE REQUISITOS

3.1.1 Requisitos Funcionais

Abaixo são listados os requisitos funcionais do sistema:

1. **Fornecimento de localização de veículos:** Fornecer informações de localização dos veículos do sistema no momento solicitado.
2. **Catálogo de linhas disponíveis no Sistema:** Listar as linhas disponíveis no sistema.
3. **Catálogo das paradas de ônibus:** Listar as paradas disponíveis no sistema e as linhas que estão disponíveis em cada uma delas.
4. **Avaliação de entidades do sistema:** Registrar avaliações dos usuários sobre paradas e veículos do sistema de transporte público.
5. **Gerenciamento de usuários e autenticação:** Listar histórico de operações do usuário e ranking de usuários cadastrados.

3.1.2 Requisitos Não Funcionais

Abaixo são listados os requisitos não funcionais do sistema:

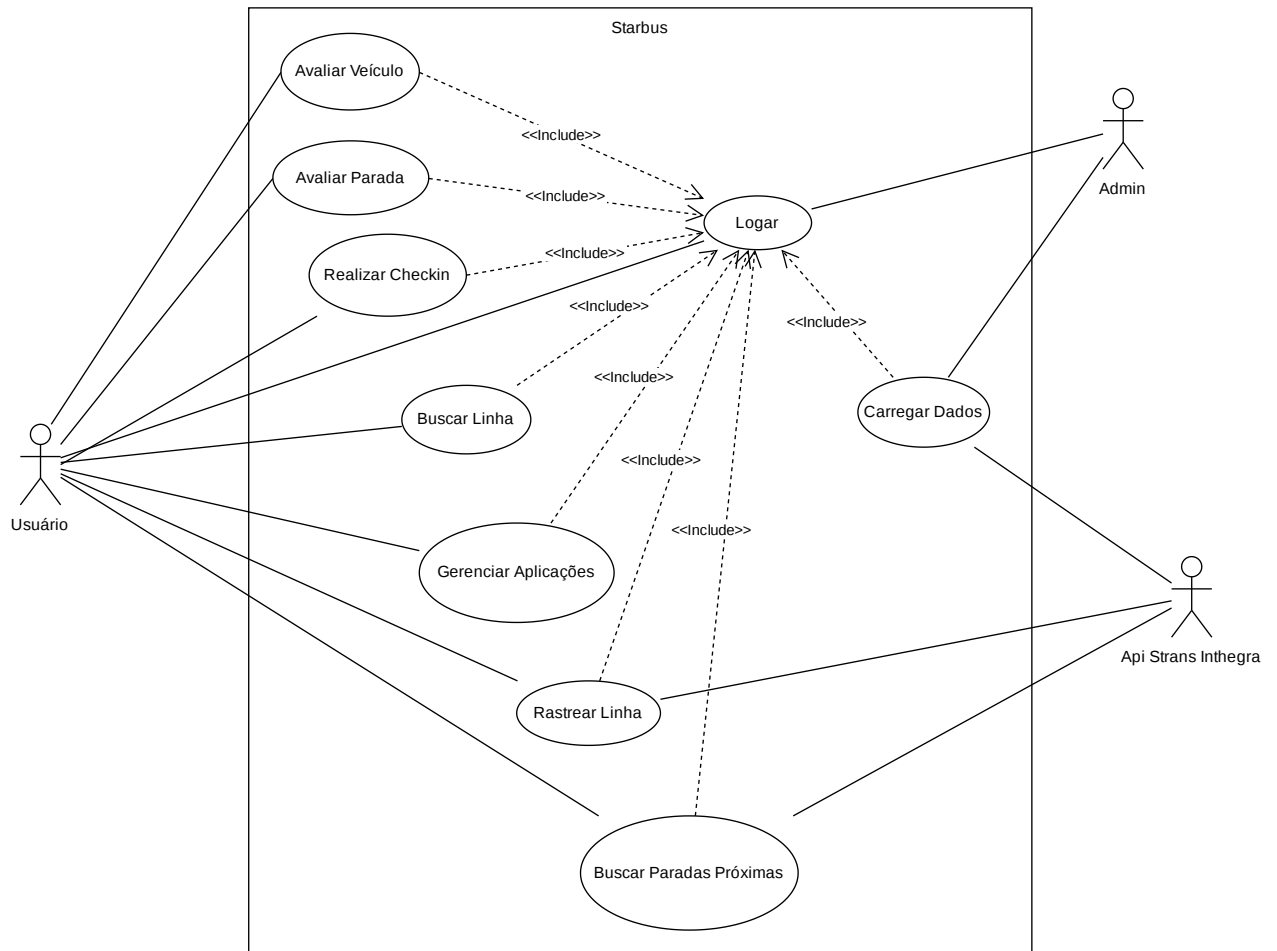
1. **Escalável e de alta disponibilidade:** Deve ser facilmente escalável para atender o aumento repentino de demanda ao mesmo tempo que mantém-se sempre disponível.
2. **Baixa carga de trabalho de Infra:** A baixa carga de trabalho de infra pode ajudar na evolução da plataforma.
3. **API deve ser simples e flexível:** Para ser atrativa ao uso de desenvolvedores a api deve ser bem documentada e flexível.

3.2 DIAGRAMAS DE CASOS DE USO

O diagrama de caso de uso desenvolvido mostrado na figura 1 descreve as possíveis interações do usuário através da ferramenta.

Figura 1 – Diagrama UML de Casos de Uso

Visual Paradigm Online Diagrams Express Edition



Visual Paradigm Online Diagrams Express Edition

Caso de uso Starbus

O diagrama acima expressa os seguintes casos de uso:

1. **Logar:** Ação de identificar-se para fornecer informação para a plataforma, ação necessária para a realização dos casos de uso a seguinte;
2. **Carregar Dados:** Usuário pode carregar as informações de linhas e paradas a partir da API Strans Inthegra;
3. **Avaliar Veículo:** Usuário devidamente identificado publica sua opinião sobre a parada de ônibus em aspectos como acessibilidade, estado de conservação, lotação e conforto;
4. **Avaliar Parada:** Usuário devidamente identificado publica sua opinião sobre a parada de ônibus em aspectos como acessibilidade, estado de conservação, movimentação e conforto;

5. **Realizar Checkin:** Usuário devidamente identificado indica que no momento encontra-se em uma devida parada de ônibus ou veículo;
6. **Buscar Linha:** Usuário fornece um código ou um termo para pesquisa que é usado para encontrar uma linha que ele deseja detalhes, como itinerário e localização dos veículos;
7. **Gerenciar Aplicações:** Um usuário pode ter várias aplicações, aqui ele pode gerenciá-las;
8. **Rastrear Linha:** Tendo um identificador de uma linha, que pode ser recuperado no caso de uso anterior, consulta a localização dos veículos dessa linha em tempo real com dados fornecidos pela Api Strans Inthebra;
9. **Buscar Paradas Próximas:** Usuário fornece uma localização, que pode ser a sua atual ou de um ponto específico que pode quer chegar, a plataforma retorna todas as paradas próximas aquele ponto, com os dados da parada inclusive a avaliação dos usuários sobre a parada;

3.3 DIAGRAMAS DE CLASSE

O modelo de dados da aplicação apresentado na Figura 2, onde é apresentado um diagrama UML de classes do projeto model.

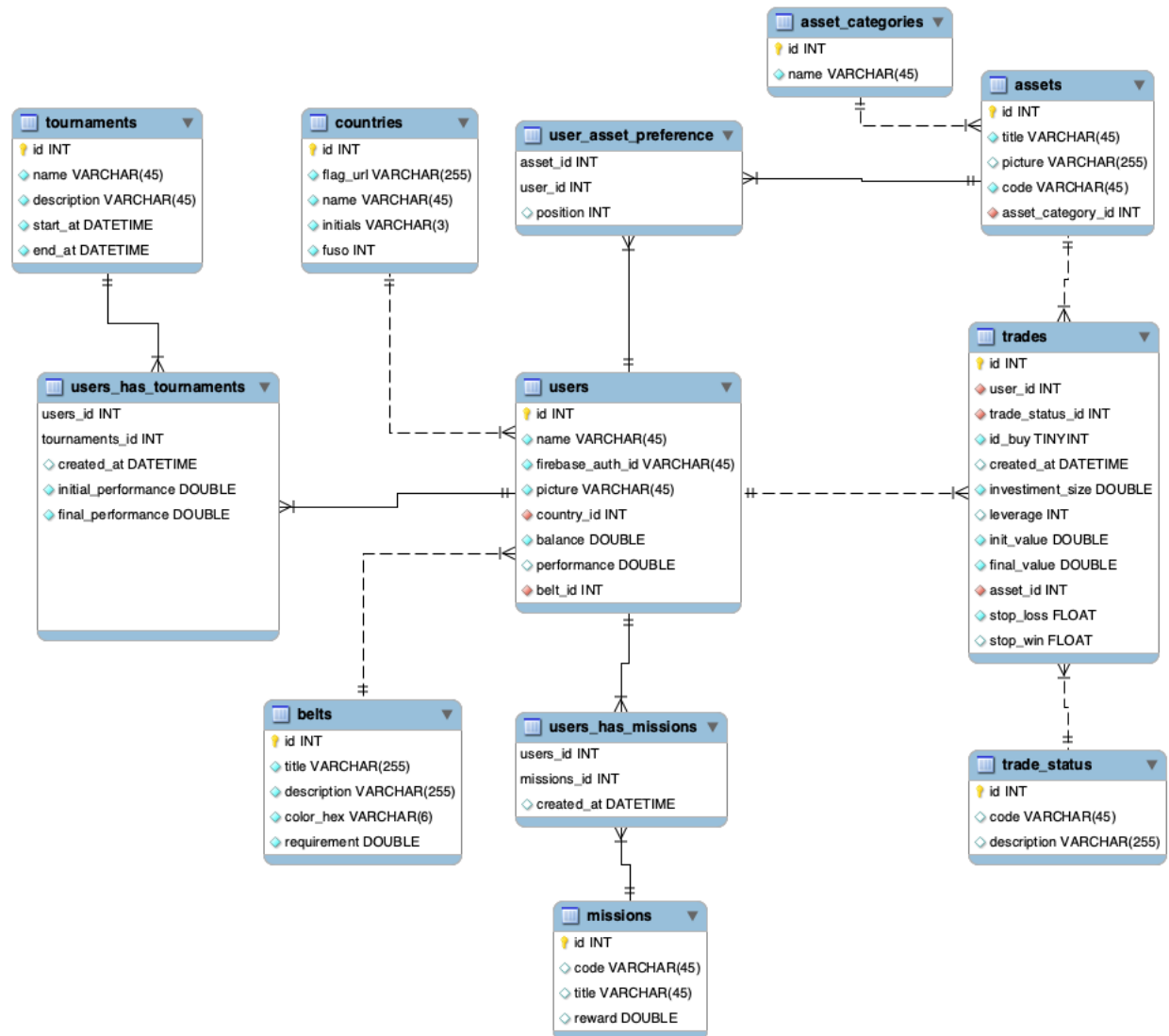
- **User:** Representa um usuário cadastrado do sistema;
- **Asset:** Representa um ativo financeiro;

3.4 ARQUITETURA DO SISTEMA

3.5 DIAGRAMA DE ENTIDADES-RELACIONAMENTOS

Na figura 2 podemos ver a estrutura lógica usada como base para o banco de dados.

Figura 2 – Diagrama entidade-relacionamento



Fonte: Elaborado pelo Autor

3.6 INTERFACE

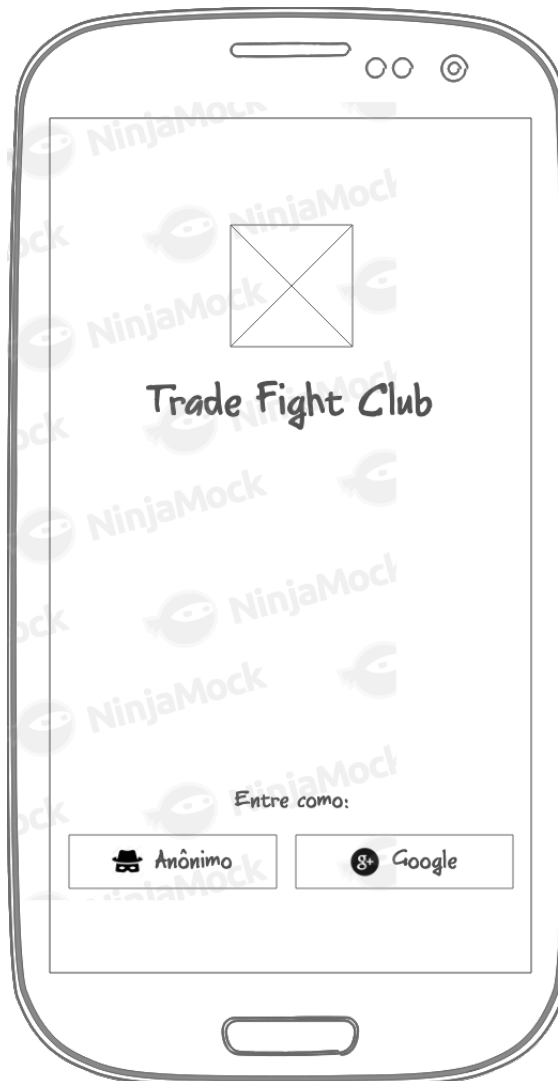
3.6.1 Tela de Autenticação

Logo que a aplicação é iniciada é verificado se o usuário já possui sessão ativa, caso não é exibido a tela de autenticação, onde o usuário pode:

- Acessar utilizando suas credenciais da conta Google;
- Acessar como anônimo;

Em qualquer um dos fluxos escolhidos é criado um registro de usuário, caso não o tenha.

Figura 3 – Tela de Autenticação



Fonte: Elaborado pelo Autor

3.6.2 Tela Principal

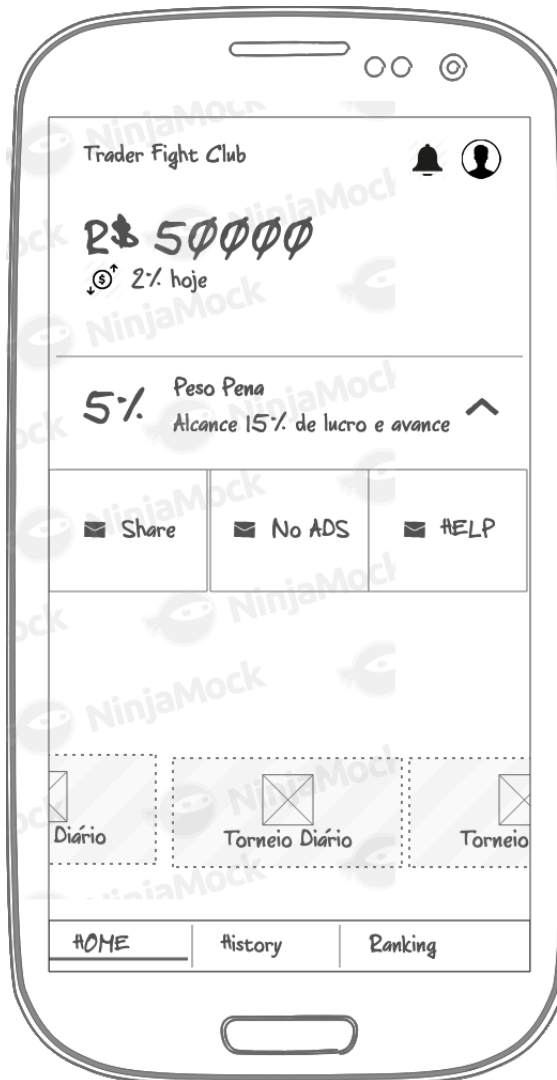
A primeira tela que um usuário autenticado tem acesso é a tela *home* ou tela principal. Nela são exibidos os seguintes elementos:

- Um avatar do usuário que leva ao perfil do mesmo;
- Saldo atual;
- Porcentagem de lucro das últimas 24 horas;
- Uma seção que informa a categorial atual da conta;
- Um *card* com missões que além de desafiar e guiar o usuário, dão recompensas;
- Um *card* para compartilhamento do aplicativo nas redes sociais;
- Um *card* para compra do plano sem propagandas;

- Um *card* para um módulo de ajuda ao usuário;
- *Cards* de acesso aos torneios disponíveis;

Nessa mesma tela, também podemos ver um menu em *tabs* na parte inferior do aplicativo que auxilia na navegação entre as telas: principal, histórico de *trades* e ranking. E na parte superior que dá acesso a tela de perfil do usuário.

Figura 4 – Tela Principal



Fonte: Elaborado pelo Autor

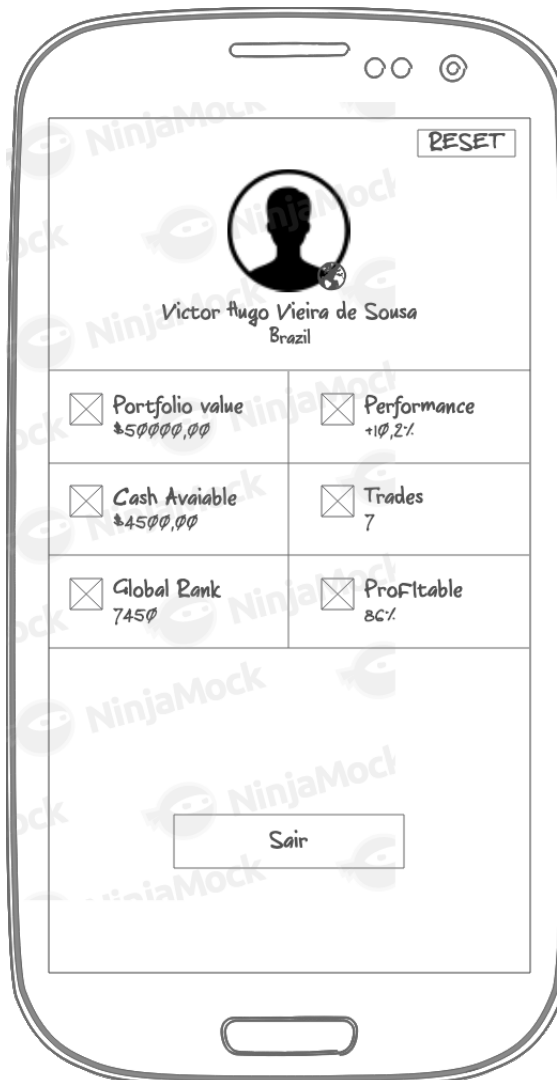
3.6.3 Tela de Perfil

Na tela de perfil o usuário tem a possibilidade de restaurar seus dados aos valores iniciais, bem como sair da aplicação e tem acesso alguns indicadores, sendo eles:

- Portfólio: Valor total;
- Dinheiro disponível: Valor total disponível para operação de compra e venda;

- **Performance:** Porcentagem comparativa do saldo atual com o saldo inicial;
- **Trades:** Total de negociações realizadas;
- **Ranking Global:** Posição referente a performance realizada e comparada com todos os usuários da plataforma;
- **Profitable:** Razão de acertos e total de operações;

Figura 5 – Tela Principal



Fonte: Elaborado pelo Autor

3.6.4 Tela de Histórico

Na tela de histórico o usuário visualiza todas as negociações já realizadas.

Figura 6 – Tela de Histórico

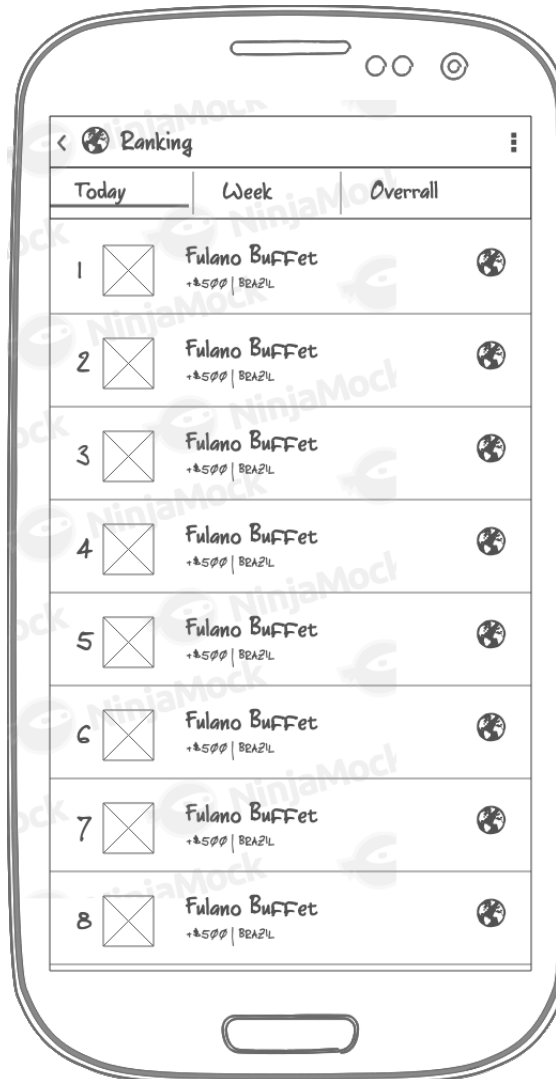


Fonte: Elaborado pelo Autor

3.6.5 Tela de Ranking

A tela de ranking dos usuários é ordenada pelo valor total do portfólio, seguido da porcentagem de lucro da conta, nome, país de origem e avatar.

Figura 7 – Tela de Ranking



Fonte: Elaborado pelo Autor

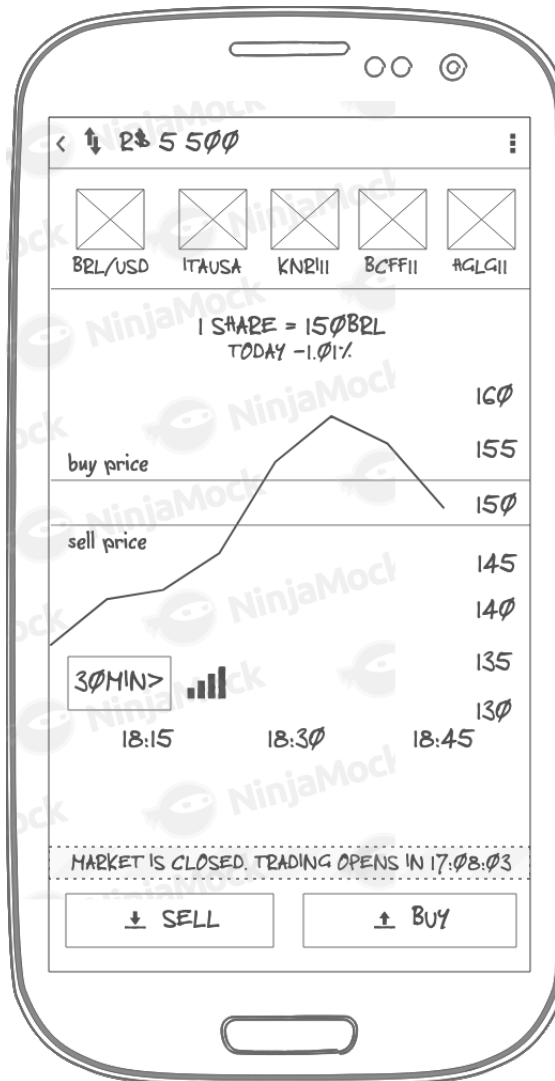
3.6.6 Tela de Negociações

Nessa tela, o usuário pode fazer praticamente as mesmas coisas da tela principal, podendo clicar no nome do usuário que publicou a receita e ir para o seu perfil. Com um clique, o usuário também pode curtir, comentar e até mesmo favoritar as receitas. A parte mais importante dessa tela são as informações sobre a receita, onde o usuário pode ver, por exemplo, o tempo de preparo e cozimento, o rendimento e a dificuldade, além de ter todas as informações de ingredientes necessários e modo de preparo para fazer a receita.

Na tela de negociações o usuário acompanha a flutuação do preço dos ativos, cria ordens de compra ou venda, a depender do horário de negociações do mesmo. Na barra superior é exibido saldo, sendo atualizado de acordo com as negociações em andamento. Ainda na barra superior, o usuário pode editar quais ativos de sua preferência devem ser exibidos na listagem horizontal que fica no topo da tela. No gráfico temos informações referente ao valor do ativo, a

valorização do mesmo em relação ao fechamento do dia, o preço médio entre compra e venda, bem como a opção de alterar o tempo do gráfico e dois modos de visualização: em linha ou em barras.

Figura 8 – Tela de Negociações



Fonte: Elaborado pelo Autor

4 SOFTWARE

4.1 IMPLANTAÇÃO

4.2 TESTES

5 CONSIDERAÇÕES FINAIS

5.1 TRABALHOS FUTUROS

REFERÊNCIAS

- 1 GOLÁN, M. L. Crowdfunding: A funding model that succeeds in times of revolt of the masses. p. 1–6, June 2015. ISSN 2166-0727. Citado na página 6.