

GABRIEL AKIO TAKANO RIGOLI
LUCAS NORIYUKI HOMI DOS SANTOS

**AUMENTO DA LIQUIDEZ DE ATIVOS IMOBILIÁRIOS EM APLICAÇÕES DE
EMPRÉSTIMOS E ALUGUÉIS POR MEIO DA TOKENIZAÇÃO**

São Paulo
2022

**GABRIEL AKIO TAKANO RIGOLI
LUCAS NORIYUKI HOMI DOS SANTOS**

**AUMENTO DA LIQUIDEZ DE ATIVOS IMOBILIÁRIOS EM APLICAÇÕES DE
EMPRÉSTIMOS E ALUGUÉIS POR MEIO DA TOKENIZAÇÃO**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para
obtenção do Título de Engenheiro Eletricista
com Ênfase em Computação.

**São Paulo
2022**

**GABRIEL AKIO TAKANO RIGOLI
LUCAS NORIYUKI HOMI DOS SANTOS**

**AUMENTO DA LIQUIDEZ DE ATIVOS IMOBILIÁRIOS EM APLICAÇÕES DE
EMPRÉSTIMOS E ALUGUÉIS POR MEIO DA TOKENIZAÇÃO**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para
obtenção do Título de Engenheiro Eletricista
com Ênfase em Computação.

Área de Concentração:
Engenharia da Computação

Orientador:
Prof. Dr. Marcos Antonio Simplicio Junior

**São Paulo
2022**

RESUMO

Atualmente, existem diversos tipos de ativos, sejam eles ativos reais, como máquinas, veículos, imóveis, ou ativos intangíveis, como ações, títulos públicos, títulos privados, entre outros. Os ativos intangíveis possuem, em geral, uma maior liquidez, ou seja, uma maior facilidade em serem comprados, vendidos, e até mesmo convertidos em dinheiro. Com isso em mente, e observando-se o mercado imobiliário, surge a ideia da utilização do conceito de tokenização para este mercado. A tokenização consiste essencialmente na conversão de ativos reais em tokens digitais, comumente usando uma rede blockchain como infraestrutura subjacente para garantir a integridade e transparência das transações realizadas. Os tokens são gerados por um smart contract (ou contrato inteligente) capaz de definir regras e funções a serem executadas na blockchain, e podem, por exemplo, representar diferentes porcentagens do direito da posse do ativo real. Dessa forma, gera-se uma maior liquidez deste ativo, uma vez que é possível comercializar frações dele por valores proporcionalmente menores. Além da maior facilidade de comercialização dos ativos imobiliários, a tokenização permite utilizar um mesmo ativo imobiliário como garantia de mais de um empréstimo, já que a utilização de apenas uma porcentagem do ativo como garantia evita o cenário em que um imóvel de valor muito superior ao do empréstimo é “bloqueado” como garantia em uma só transação dessa natureza. Finalmente, pode-se tirar proveito de plataformas de economia compartilhada (e.g. Airbnb) para remunerar os múltiplos donos do ativo de forma proporcional à porcentagem do ativo que cada um deles possui. O presente trabalho tem como objetivo a criação de uma plataforma digital que permita a tokenização de ativos imobiliários, habilitando os casos de uso aqui descritos.

Palavras-chave: Ativos Imobiliários. Tokenização. Blockchain. Plataforma Digital. Smart Contract.

ABSTRACT

Nowadays, there are several types of assets, whether real assets, such as machines, vehicles, properties, or intangible assets, such as stocks, government bonds, private bonds, etc. The intangible assets, in general, have a greater liquidity, in other words, a better chance of being purchased, sold and converted into cash. With this in mind and observing the real estate market, comes up the idea of using tokenization concepts in this market. The tokenization essentially consists of converting real assets into digital tokens, commonly using a blockchain network as underlying infrastructure to ensure integrity and transparency of the transactions. The tokens are generated by a smart contract capable of defining rules and functions to be executed in the blockchain, and can represent, for instance, different percentages of the real asset ownership. Thus, a greater liquidity of this kind of asset is created once it enables the commercialization of its fractions for proportionally smaller amounts. Furthermore, tokenization allows using the same real estate asset as a guarantee of multiple loans since using just a percentage of the asset avoids cases where a property with higher value than the loan is blocked as guarantee in just one transaction of this nature. Finally, we can benefit from shared economy platforms (e.g. Airbnb) to pay multiple owners of the asset according to their ownership percentage. The current study has the objective of creating a digital platform to allow tokenization of real estate assets and enable the use cases described above.

Keywords: Real estate asset. Tokenization. Blockchain. Digital platform. Smart Contract.

LISTA DE FIGURAS

x	Arquitetura Proposta	
x	Diagrama de sequência para registro de novo usuário na plataforma	
x	Diagrama de sequência para autenticação de usuário na plataforma	
x	Diagrama de sequência para cadastro de endereço de carteira de usuário na plataforma	
x	Diagrama de sequência para a tokenização de um imóvel	
x	Diagrama de sequência para a validação dos documentos de um imóvel para o processo de tokenização	
x	Diagrama de sequência para a criação de oferta de venda de porcentagem de posse de ativo tokenizado	
x	Diagrama de sequência para aceite de oferta de ativo imobiliário tokenizado	
x	Diagrama de sequência para a confirmação de compra de ativo imobiliário	
x	Diagrama de sequência para criação de empréstimo	
x	Diagrama de sequência para confirmação de empréstimo	
x	Diagrama de sequência para registro de quitação de empréstimo	
x	Diagrama de sequência para confirmação de quitação de empréstimo	
x	Diagrama de sequência para tomada de garantia de empréstimo	
x	Diagrama de sequência para registro de pagamento de aluguéis	
x	Diagrama de sequência para visualização dos pagamentos de aluguéis recebidos por um proprietário	
x	Diagrama de sequência para auditoria dos dados do imóvel na blockchain	
x	Modelagem do banco de dados do projeto	
x	Estrutura de Controladores e serviços do servidor	
x	Documentação do Swagger para as rotas de Smart Contracts, Users, Tokenized Assets, Proposals e Ownerships	
x	Documentação do Swagger para as rotas de Collaterals, Offers, Rent Payments e Authentication	

x	Tela de login e cadastro de usuário
x	Aba de dados gerais da conta
x	Aba de imóveis tokenizados do usuário
x	Registro dos aluguéis recebidos enquanto proprietário do imóvel
x	Aba de garantias do usuário
x	Aba de ofertas do usuário abertas no Marketplace
x	Aba de compras realizadas no Marketplace pelo usuário
x	Formulário de envio dos dados do imóvel para tokenização
x	Tela de listagem dos imóveis disponíveis para criação de oferta
x	Formulário de criação de oferta no Marketplace
x	Marketplace de imóveis da plataforma
x	Tela de detalhes da oferta do imóvel
x	Tela de listagem dos imóveis disponíveis para criação de garantia
x	Formulário de criação de garantia de empréstimo
x	Aba listando todas as propostas de tokenização da plataforma
x	Aba listando todos os pagamentos de ofertas pendentes de confirmação
x	Aba listando todos os cadastros de garantia de empréstimos pendentes na plataforma
x	Aba listando todas as garantias de empréstimo registradas na plataforma
x	Aba listando todas os imóveis tokenizados da plataforma
x	Formulário de registro de pagamento de aluguel
x	Modal de auditoria dos dados do imóvel tokenizado

LISTA DE ABREVIATURAS E SIGLAS

HTTP
API REST
LARC
JWT
ORM
SQL
USP
FT
NFT
IDE
ERC
ETH
CPF

SUMÁRIO

1 INTRODUÇÃO	11
1.1 MOTIVAÇÃO	11
1.2 OBJETIVO	12
1.3 JUSTIFICATIVA	13
1.4 SOLUÇÕES DE MERCADO	13
1.5 ORGANIZAÇÃO DO TRABALHO	14
2 ASPECTOS CONCEITUAIS	14
2.1 BLOCKCHAIN	15
2.2 SMART CONTRACTS	15
2.3 TOKEN	16
2.3.1 Token Fungível (FT)	16
2.3.2 Token Não Fungível (NFT)	17
3 MÉTODO DO TRABALHO	17
4 ESPECIFICAÇÃO DO SISTEMA	18
4.1 AGENTES DO SISTEMA	18
4.2 REQUISITOS FUNCIONAIS	18
4.3 REQUISITOS NÃO-FUNCIONAIS	19
4.3.1 Escalabilidade	19
4.3.2 Segurança	20
4.4 ARQUITETURA PROPOSTA	21
4.4.1 Interface da plataforma (Front-End)	21
4.4.2 Servidor da plataforma (Back-End)	22
4.4.3 Banco de dados relacional (Back-end)	22
4.4.4 Blockchain e Smart Contracts	22
4.5 CASOS DE USO PRINCIPAIS	23
4.5.1 Registro e login de usuários na plataforma	23
4.5.2 Cadastro de carteira do usuário na plataforma	25
4.5.3 Validação de documentos e tokenização de um imóvel	26
4.5.4 Compra e venda de porcentagem de posse de ativo imobiliário	27
4.5.5 Registro de garantia de empréstimo	30
4.5.6 Quitação de empréstimo e tomada de garantia	31
4.5.7 Distribuição de retornos financeiros provenientes de aluguel	34
4.5.8 Auditoria de dados registrados na blockchain	35
5 DESENVOLVIMENTO DO TRABALHO	36
5.1 TECNOLOGIAS UTILIZADAS	37
5.1.1 Node.js	38
5.1.2 React	38

5.1.3 PostgreSQL	38
5.1.4 Solidity	38
5.1.5 Remix IDE	39
5.1.6 Web3.js	39
5.1.7 Ethereum Goerli Testnet	39
5.1.8 TypeORM	40
5.2 PROJETO E IMPLEMENTAÇÃO	40
 5.2.1 Desenvolvimento do Smart Contract	40
5.2.1.1 Atributos armazenados no smart contract desenvolvido	41
5.2.1.2 Métodos desenvolvidos no smart contract	42
5.2.1.3 Publicação e implantação do smart contract no blockchain Goerli	44
 5.2.2 Estrutura do banco de dados relacional	44
 5.2.3 Desenvolvimento do servidor	47
 5.2.4 Desenvolvimento do Front-End	51
5.2.4.1 Login e cadastro	52
5.2.4.2 Perfil de usuário	52
5.2.4.3 Tokenização de imóvel	56
5.2.4.4 Criar oferta no Marketplace	57
5.2.4.5 Marketplace	58
5.2.4.6 Criar garantia de empréstimo	59
5.2.4.7 Portal do administrador	60
5.3 TESTES E AVALIAÇÃO	65
 5.3.1 Tokenização de imóvel	65
 5.3.2 Oferta e compra de percentual de imóvel	66
 5.3.3 Garantias de empréstimos	66
 5.3.4 Distribuição de retornos financeiros provenientes de aluguel	66
6 CONSIDERAÇÕES FINAIS	66
REFERÊNCIAS	68

1 INTRODUÇÃO

Existem diversos tipos de ativos no mercado atual, podendo ser classificados em duas categorias: os ativos reais ou tangíveis, e os ativos financeiros [1]. Os ativos reais englobam bens materiais que podem ser tocados, como máquinas, veículos, imóveis, moedas antigas, vinhos e entre outros. Já os ativos financeiros são bens intangíveis como ações, títulos públicos, títulos privados e entre outros.

Os ativos financeiros possuem, em geral, uma maior liquidez, ou seja, uma maior facilidade em serem convertidos em dinheiro. Enquanto isso, os ativos reais possuem baixa liquidez, devido à dificuldade de transferência entre os proprietários por conta de diversos procedimentos legais no processo, além de, eventualmente, altos valores envolvidos.

Além disso, os ativos reais são, muitas vezes, utilizados para a obtenção de empréstimos com bancos, que, na maioria das vezes, exigem uma garantia para que possam realizar o empréstimo requisitado. Dessa forma, os ativos reais (principalmente carros e imóveis) são utilizados frequentemente por pessoas físicas e jurídicas para cobrir esta garantia exigida pelos bancos em diferentes modalidades de empréstimo [2].

Porém, atualmente esta prática não vem sem suas limitações, tendo como uma das principais o fato de, no caso de imóveis, ser permitida a utilização de apenas um imóvel para um único empréstimo, sendo ainda proibida a utilização do mesmo imóvel como garantia para dois ou mais empréstimos diferentes [3]. Além disso, é recorrente a utilização de imóveis como garantia para empréstimos cujo valor é menor que o valor do próprio imóvel utilizado na garantia, paralisando um valor patrimonial muito grande do indivíduo, e que poderia estar sendo utilizado, por exemplo, para garantir outro empréstimo.

1.1 MOTIVAÇÃO

A motivação do presente trabalho vem da observação da baixa liquidez inerente aos ativos imobiliários, que os torna menos acessíveis em termos de comercialização, sem que haja formas popularizadas de transacioná-los de modo fracionado, com exceção de por fundos de investimentos. Esta dinâmica atual é também responsável por dificultar a distribuição de retornos financeiros de aluguéis

para indivíduos com direitos sobre o imóvel e por impedir que um mesmo imóvel seja utilizado como garantia de mais de um empréstimo. Neste contexto, surge uma proposta para ganho de liquidez ainda pouco explorada no mercado, de utilizar a recente tecnologia de blockchain para associar imóveis a tokens digitais, num processo chamado de tokenização, de tal forma que estes tokens representam a propriedade do imóvel e podem ser comercializados entre diferentes partes de forma fracionada ou completa, garantindo segurança, auditabilidade e transparência das operações.

Dadas as recentes discussões do governo brasileiro a respeito da utilização de um imóvel como garantia em diferentes empréstimos [4] [5], bem como a crescente presença de tokenização e criptomoedas no mercado imobiliário [6], se entende que a discussão levantada neste trabalho é de grande relevância. É de se notar que a tokenização depende de aspectos jurídicos que podem variar de acordo com a região do imóvel, porém, este assunto tem evoluído significativamente com o intuito de que os registros de imóveis como propriedade digital possam caminhar em sintonia com a atuação de cartórios regionais [7].

Tendo em vista o cenário em que o projeto aqui descrito se insere, entende-se que os âmbitos financeiros e jurídicos são relevantes para a discussão e, por conta disso, serão devidamente contextualizados, porém o foco principal do trabalho será o estudo e a implementação das tecnologias envolvidas.

1.2 OBJETIVO

O objetivo do projeto é desenvolver uma plataforma digital, mais especificamente um *web app*, em que o usuário fornece informações de seus ativos imobiliários e torna-se capaz de convertê-los em *tokens*. Para isso, o *web app* se conectará a uma rede *blockchain*, na qual estarão contidos contratos inteligentes responsáveis pela geração dos *tokens*. Uma vez obtidos, os ativos imobiliários tokenizados poderão ser utilizados dentro da plataforma como garantia para múltiplos empréstimos, e poderão ser transacionados de modo fracionado, o que viabiliza a distribuição de retornos financeiros derivados destes ativos de forma transparente, segura e auditável.

1.3 JUSTIFICATIVA

Tendo em vista o cenário descrito nos capítulos anteriores, é possível observar que o objetivo do projeto está alinhado com necessidades do mercado imobiliário atual, implementando uma plataforma que permitirá o aumento da liquidez de ativos imobiliários (atualmente com baixa liquidez), bem como permitindo a utilização de um mesmo ativo imobiliário como garantia de múltiplos empréstimos e a distribuição de retornos financeiros provenientes do imóvel (cenários bastante discutidos atualmente, com algumas soluções propostas e implementadas, listadas na seção 1.4).

O objetivo deste projeto também é bastante relevante no cenário atual devido à sua implementação utilizando a tecnologia do *blockchain*, que permite tornar as dinâmicas envolvendo imóveis, atualmente realizadas por cartórios de forma muitas vezes manual, mais transparentes, auditáveis, seguras, descentralizadas e menos burocráticas. Esta tecnologia é bastante estudada atualmente, se tornando cada vez mais uma tendência no mercado [8], e portanto o presente trabalho também serve de referência para futuras pesquisas sobre o mercado de ativos imobiliários, podendo incentivar novas utilizações da tecnologia do *blockchain* e *smart contracts* em cima das funcionalidades aqui propostas.

1.4 SOLUÇÕES DE MERCADO

No mercado imobiliário, algumas soluções de tokenização de imóveis para ganho de liquidez já têm ocupado um posto relevante.

A *Netspaces* [9] é uma empresa brasileira e possui uma plataforma de venda fracionada de imóveis por meio de *tokens*, para a distribuição de recebimentos de aluguel, financiamento, compra, venda e apreciação de imóveis. Em aspectos jurídicos, o proprietário do imóvel a ser tokenizado assina uma escritura de permuta e paga as taxas de cartório para registrar a *Netspaces* como proprietária do imóvel, para que, assim, a tokenização aconteça e a propriedade digital na *blockchain* passe a fazer parte da carteira digital do proprietário. No caso de transações de propriedades digitais, o comprador informa e confirma os dados da transação e o vendedor recebe para análise, podendo aceitar e transferir o imóvel para a carteira digital do comprador ou recusar [10].

Além desta solução, outras brasileiras são a *Vanhall* [11], cuja principal função é fornecer toda a infraestrutura necessária para tokenização de ativos que vão além dos imobiliários; e o *Sync* [12], que propõe o fornecimento de tokens limitados com lastro em ativos imobiliários.

Fora do Brasil, várias soluções propõem a tokenização de imóveis como uma ferramenta para tornar investimentos no mercado imobiliário mais acessíveis. Empresas como *Realt* [13], *Tokenyzed* [14], *Digishares* [15], *Redswan* [16] e *Tokeny* [17] oferecem tokens de ativos imobiliários num formato similar a cotas de fundos de investimento, com expectativa de valorização e pagamento de dividendos de acordo com os recebimentos de aluguéis do imóvel.

O trabalho desenvolvido, por sua vez, busca implementar a funcionalidade de fracionamento da posse do ativo imobiliário para distribuição dos retornos financeiros, já bem explorada pelo mercado, e integrá-la ao contexto de imóveis dados em múltiplas garantias.

1.5 ORGANIZAÇÃO DO TRABALHO

Neste projeto, o trabalho está dividido da seguinte forma:

O Capítulo 2 apresenta breves introduções aos principais conceitos abordados no trabalho.

O Capítulo 3 consiste na metodologia de trabalho empregada, explicitando planejamento dos trabalhos e métodos de desenvolvimento empregados.

O Capítulo 4 versa sobre as especificações e os requisitos que o sistema desenvolvido deve cumprir.

O Capítulo 5 detalha as tecnologias utilizadas, o desenvolvimento do projeto, os testes e as considerações a respeito do trabalho desenvolvido.

O Capítulo 6 traz as considerações finais sobre o trabalho desenvolvido, dificuldades encontradas, conclusões e perspectivas sobre a continuidade do projeto.

2 ASPECTOS CONCEITUAIS

Para o entendimento completo deste trabalho, é necessário o conhecimento de alguns conceitos importantes que são citados diversas vezes no presente

trabalho. Esta seção contém uma breve descrição teórica desses conceitos que atravessam o projeto.

2.1 BLOCKCHAIN

Blockchain é o termo utilizado para definir uma estrutura de dados composta por blocos que registram informações sobre transações do sistema sequencialmente, de tal forma que cada bloco é associado ao anterior por meio de criptografia baseada em *hash*. Em geral, os principais elementos presentes num bloco são o *hash* do bloco anterior, que proporciona a organização de dados em cadeia; dados sobre transações que ocorreram no sistema; e o *Timestamp*, que registra e prova que as transações contidas no bloco existiam no momento em que o bloco foi adicionado à cadeia [18].

A dependência de informações contidas em blocos anteriores é explorada no funcionamento de um mecanismo de consenso presente nas *blockchains* [19], que garante a existência de uma versão única da sequência de dados de forma descentralizada e resistente a modificações. Além disso, todos os dados sobre transações são expostos de forma transparente, garantindo auditabilidade, o que torna esta tecnologia extremamente útil em cenários que demandam alta confiabilidade.

2.2 SMART CONTRACTS

Um conceito que vem ganhando cada vez mais importância em diversas tecnologias de blockchain é o conceito de *smart contracts*, ou contratos inteligentes, que podem ser definidos como programas ou trechos de programas que podem ser executados diretamente na rede blockchain onde são implantados.

Esses programas possuem funções e estados internos (dados internos ou até mesmo um saldo de criptomoedas) que estão associados a um endereço específico no blockchain, determinado automaticamente durante a sua implantação inicial e que através dele é possível que usuários do blockchain invoquem as diversas funções implementadas no *smart contract*, que poderá fazer cálculos, verificar condições e realizar transações automaticamente para a rede dependendo de seu estado interno e das condições verificadas [20].

Os *smart contracts* são responsáveis pela difusão dos *tokens* no universo do blockchain, pois é através deles que são criados os diversos *tokens*, sejam eles fungíveis ou não fungíveis, e que possuem aplicações das mais diversas, abrangendo o universo dos jogos, arte e até mesmo o mercado imobiliário [21].

2.3 TOKEN

No contexto do blockchain, um conceito bastante importante é o de *token*. No contexto geral, um *token* é um objeto que representa outra entidade, seja ela um objeto físico ou virtual, ou até mesmo um conceito abstrato. Aplicando este conceito no contexto de blockchain, temos que um *token* é um objeto digital criado através de um smart contract existente num blockchain e que representa um ativo físico ou digital qualquer, seja ele dinheiro, um bem, uma propriedade ou uma arte.

Um *token* no contexto de blockchain pode possuir atributos diversos, como por exemplo, informações sobre seu(s) dono(s), informações sobre o(s) ativo(s) que representa (no caso de imóveis, por exemplo, seu endereço, área útil e etc.) e também pode ser dividido em uma ou mais unidades, cada uma delas podendo ter seu respectivo dono.

Tradicionalmente, os *tokens* são classificados em duas categorias, dependendo de sua natureza fungível ou não fungível. Essas categorias serão descritas nas subseções a seguir [22] [23] [24].

2.3.1 Token Fungível (FT)

A primeira categoria de *tokens* é a de *tokens* fungíveis, ou seja, que são passíveis de substituição.

Um *token* ser passível de substituição implica que qualquer unidade do token é igual a todas as outras existentes, assim como uma nota de um real válida é igual a qualquer nota de um real válida, sendo possível que, se duas pessoas possuem uma unidade de um *token* cada, caso uma delas envie seu *token* à outra e então esta pessoa envie um *token* de volta, ao final do processo não haverá ocorrido mudança de valor.

Além disso, *tokens* fungíveis podem ser divididos ou repartidos, possibilitando que possam ser comercializados em frações menores caso seja conveniente para os compradores e vendedores.

2.3.2 Token Não Fungível (NFT)

Ao contrário dos *tokens* fungíveis, os *tokens* não fungíveis ou NFTs são um tipo de *token* único, geralmente gerado para representar algum ativo específico e colecionável, como por exemplo uma arte digital.

Os NFTs são indivisíveis e, como diz o nome, não são fungíveis, o que significa que duas unidades de NFTs sempre serão diferentes entre si, e portanto não são intercambiáveis, da mesma forma que duas obras de arte diferentes não intercambiáveis.

Esta categoria de *tokens* vem ganhando bastante espaço no universo do blockchain devido ao interesse crescente na monetização e comercialização de artes digitais, sejam elas desenhos, músicas ou objetos pertencentes a jogos virtuais [25].

3 MÉTODO DO TRABALHO

O método de trabalho empregado neste projeto consiste num desenvolvimento iterativo e adaptativo de acordo com os cenários em cada uma das etapas da execução, tendo como objetivo desenvolver uma solução que reúne funcionalidades de projetos já existentes e novas funcionalidades, propostas pelo grupo em conjunto com o orientador.

Dessa forma, foram realizadas reuniões e consultas com grupos do Laboratório de Arquitetura e Redes de Computadores - USP (LARC-USP) e profissionais inseridos no mercado financeiro para direcionar e entender a viabilidade dos requisitos funcionais e não funcionais propostos.

Em paralelo, foram feitas diversas pesquisas sobre tecnologias e soluções já existentes, com o intuito de embasar a escolha subsequente das tecnologias e dos casos de uso a serem desenvolvidos.

Feitas essas escolhas, entra-se na etapa de desenvolvimento, iniciada pela implementação das funcionalidades do *smart contract* e sua subida para a

blockchain no servidor escolhido. Em seguida, será implementado o *web app*, composto pela interface (*Front-End*) na qual usuários poderão interagir e enviar solicitações de serviços ao servidor (*Back-End*) que estará integrado ao banco de dados do projeto.

No decorrer do projeto e na finalização, serão realizados testes e revisões dos casos de uso em desenvolvimento, a fim de evitar retrabalhos e validar a implementação.

4 ESPECIFICAÇÃO DO SISTEMA

Neste Capítulo, são abordados os requisitos que devem ser atendidos pelo sistema desenvolvido, bem como a arquitetura proposta para a solução, tomando como base o levantamento de soluções de mercado feito na seção 1.4 e as funcionalidades desejadas para o projeto, de fracionamento da posse de ativos imobiliários para distribuição de retornos financeiros e de garantia de imóveis para mais de um empréstimo.

4.1 AGENTES DO SISTEMA

Na plataforma, haverá três tipos de agentes, sendo eles:

- *Owner*, detentor de parte ou de todo o ativo imobiliário *tokenizado*, capaz de disponibilizar o ativo para transações percentuais ou total da posse;
- *Buyer*, potencial comprador de porcentagem de um ativo *tokenizado* disponível na plataforma;
- *Administrador*, cuja função é controlar fluxos de validação e controle da plataforma; e
- *Bank*, credor de um empréstimo, entidade que irá realizar o empréstimo e receber os ativos imobiliários como garantia.

Para cada um destes agentes, foram estruturados casos de uso que pautam as funcionalidades implementadas no sistema, descritos mais adiante.

4.2 REQUISITOS FUNCIONAIS

Os requisitos funcionais do sistema denotam as principais funcionalidades implementadas no projeto. Tendo isso em vista, são eles:

- Registrar, autenticar e autorizar usuários;
- Exibir dados de um usuário;
- Permitir ao usuário a entrada de dados de um imóvel via formulário, para que possa ser associado a *tokens*;
- Permitir transações de porcentagens de ativos tokenizados;
- Listar todas as ofertas de ativos da plataforma;
- Transferir porcentagens de ativos tokenizados entre usuários;
- Associar porcentagem de ativos tokenizados a garantias de empréstimos;
- Registrar quitação de empréstimo e desalienação subsequente do ativo tokenizado;
- Permitir tomada da garantia pelo credor, caso não haja quitação até a data de vencimento do empréstimo;
- Registrar a distribuição de retornos financeiros provenientes dos ativos tokenizados proporcional a porcentagem de posse de cada proprietário;
- Permitir auditoria de dados registrados na blockchain

4.3 REQUISITOS NÃO-FUNCIONAIS

Os requisitos não-funcionais correspondem àqueles que definem as qualidades que o sistema deve apresentar e como ele implementa as funcionalidades. Abaixo, estão listados os principais requisitos não-funcionais do projeto.

4.3.1 Escalabilidade

A escalabilidade é a capacidade de um sistema atender a demandas crescentes sem que haja perdas de desempenho. Este requisito é de extrema importância para a motivação deste projeto, dado que a plataforma deve ser capaz de gerar transações de *tokens* de ativos imobiliários de modo compatível ao volume crescente de operações de lançamentos e vendas de imóveis no mercado, que,

apenas em 2021, totalizaram 153,7 e 126,7 mil unidades no Brasil, respectivamente [26].

Como a proposta deste projeto é a de implementar uma prova de conceito, a arquitetura foi simplificada e as tecnologias escolhidas priorizam a agilidade do desenvolvimento, porém, é de se esperar que, numa eventual continuação do projeto, sejam incluídas camadas de complexidade adicionais à implementação da plataforma para que seja possível garantir-se a escalabilidade adequada.

4.3.2 Segurança

Tratando-se do mercado de ativos imobiliários, é fundamental que a plataforma desenvolvida seja capaz de realizar as funcionalidades propostas nos capítulos anteriores ao mesmo tempo que garante a segurança dos dados que irão permear a aplicação.

Para isso, um dos requisitos não-funcionais do projeto diz respeito ao processo de *login* dos usuários da plataforma. Neste processo, será necessário garantir a segurança das credenciais dos usuários através de técnicas de segurança como o *password hashing* durante o armazenamento de senhas no banco de dados de aplicação, garantindo o nível de segurança necessário para a proteção dos dados do usuário.

Ademais, também será necessário garantir confiabilidade, imutabilidade, auditabilidade e transparência dos dados dos ativos armazenados na plataforma. Estas propriedades serão concedidas à plataforma através da tecnologia do *blockchain* que será utilizada para a tokenização, empréstimo e distribuição de aluguéis, pois são propriedades intrínsecas do *blockchain* Ethereum utilizado.

Outro fator importante para a segurança dos dados que irão trafegar pela plataforma desenvolvida se refere à comunicação entre o Front-End e o Back-End da aplicação, bem como a comunicação do Back-End com o banco de dados. Para garantir-se a confidencialidade dos dados enquanto trafegam entre os componentes da aplicação, deverá ser utilizado o HTTPS ao invés do padrão HTTP, que não possui confidencialidade dos dados como uma de suas características.

4.4 ARQUITETURA PROPOSTA

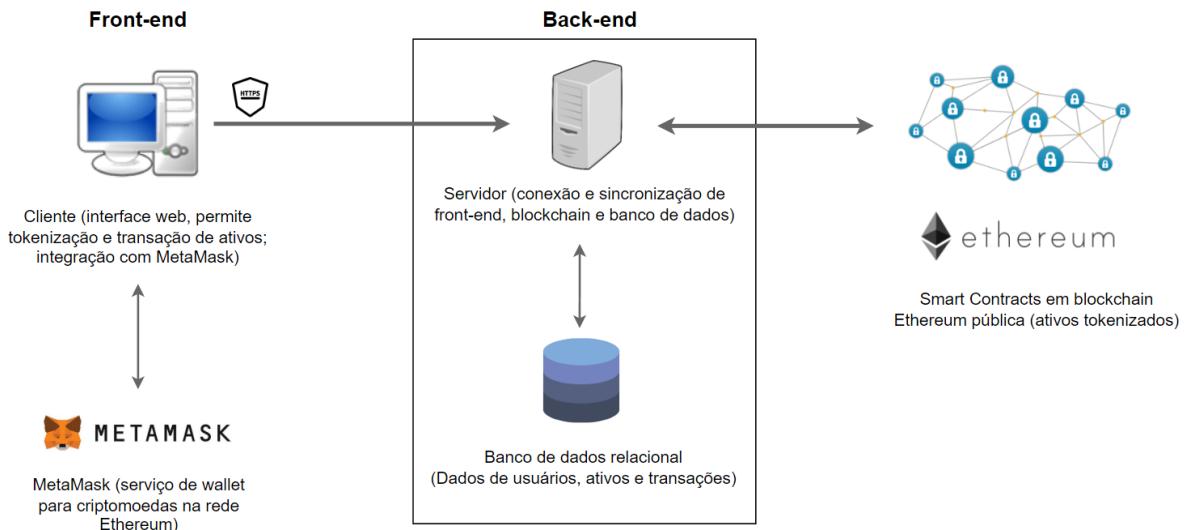


Figura X - Arquitetura Proposta

A arquitetura presente na Figura X dispõe de *Front-End*, *Back-End*, *Database* e *Blockchain*, de modo que cada um destes componentes desempenha os papéis descritos a seguir.

4.4.1 Interface da plataforma (Front-End)

O primeiro componente da arquitetura do projeto trata-se da interface da plataforma, comumente denominada de *Front-End*.

Este componente será responsável por viabilizar ao usuário a utilização das diversas funcionalidades que atendem aos requisitos funcionais, descritos na seção 4.3, de maneira fácil e intuitiva.

Trata-se de uma aplicação *web* em que o usuário pode acessar por meio da internet e que contém telas (por exemplo, de cadastro, de tokenização dos ativos do usuário, entre outras) nas quais o usuário pode selecionar quais operações quer realizar e o *Front-End* envia as informações para o servidor nos formatos e com as informações necessárias para o correto funcionamento.

Além disso, o *Front-End* conta com integração com o MetaMask [27], um serviço de carteira digital que permite gerenciamento de chaves, conexão segura com redes *blockchain* e transações de criptomoedas por meio de uma extensão no

navegador. Neste projeto, o MetaMask será utilizado para obter o endereço na rede *blockchain* da carteira digital do usuário logado.

4.4.2 Servidor da plataforma (Back-End)

O segundo componente da arquitetura do projeto se trata do servidor, que recebe mensagens do *front-end* e realiza tratamentos e validações das informações recebidas e, então, as envia para os *smart contracts* e/ou armazena informações pertinentes no banco de dados da plataforma. As aplicações do *Front-End* e do *Back-End* foram dispostas de modo apartado para garantir maior escalabilidade do sistema, uma vez que, dessa forma, podem ser desenvolvidas de forma independente e mais generalizada. Neste mesmo contexto, é importante mencionar que ambas podem ter várias instâncias executando em paralelo, de acordo com a demanda.

4.4.3 Banco de dados relacional (Back-end)

A plataforma utiliza um banco de dados relacional para armazenar informações dos usuários e dos imóveis tokenizados na plataforma, além de outras informações relevantes para atender os requisitos funcionais descritos.

Para garantir a eficiência do sistema, bem como os requisitos não-funcionais de segurança e consistência dos dados, as informações relacionadas a transações são salvas no banco de dados de modo sincronizado com os dados registrados na *blockchain*. Na etapa de desenvolvimento, será discutida com mais detalhes a modelagem de entidades e relacionamentos do banco de dados.

4.4.4 Blockchain e Smart Contracts

A plataforma terá que se comunicar com *smart contracts*, que são contratos inteligentes desenvolvidos pelo grupo e que implementam as funcionalidades necessárias para a tokenização de ativos imobiliários, bem como a atribuição de ativos a garantias de empréstimos e a compra e venda de porcentagens dos ativos

de modo a viabilizar a distribuição de retornos financeiros no cenário de aluguéis de imóveis.

Os *smart contracts* estarão numa *blockchain* de *Ethereum*, onde serão registradas todas as operações e informações necessárias sobre as transações realizadas de ativos na plataforma, de modo que cada *smart contract* enviado para a rede representa um ativo tokenizado.

Devido à natureza imutável das *blockchains*, o processo de posse e registro de transações com ativos imobiliários na plataforma se torna confiável e facilmente auditável, em que todas as transações registradas na *blockchain* são validadas pela plataforma e apenas registradas se forem legítimas.

4.5 CASOS DE USO PRINCIPAIS

Com base nos requisitos funcionais e não funcionais e na arquitetura propostos acima, foram idealizados os principais casos de uso da plataforma.

4.5.1 Registro e login de usuários na plataforma

Para que seja possível a utilização da plataforma, para todos os demais casos de uso, é necessário que o usuário realize seu cadastro na plataforma através de informações como seu nome, usuário, CPF e senha.

Para realizar este cadastro, o usuário deverá acessar a tela correspondente a esta funcionalidade no *Front-End* da plataforma e então será apresentado um formulário *online* onde deverá inserir os dados necessários para o registro e a utilização da plataforma. Após inserir seus dados para o cadastro, o usuário irá submeter o formulário pelo *Front-End* que irá enviar estas informações para o *back-end*, que irá então realizar o cadastro do usuário no banco de dados da plataforma e irá retornar ao usuário seus dados de perfil criado na plataforma.

A Figura X mostra o diagrama de sequência deste caso de uso envolvendo o usuário da plataforma, a tela utilizada e o servidor, além das devidas mensagens trocadas entre os elementos para que seja possível o registro seguro de um usuário na plataforma.

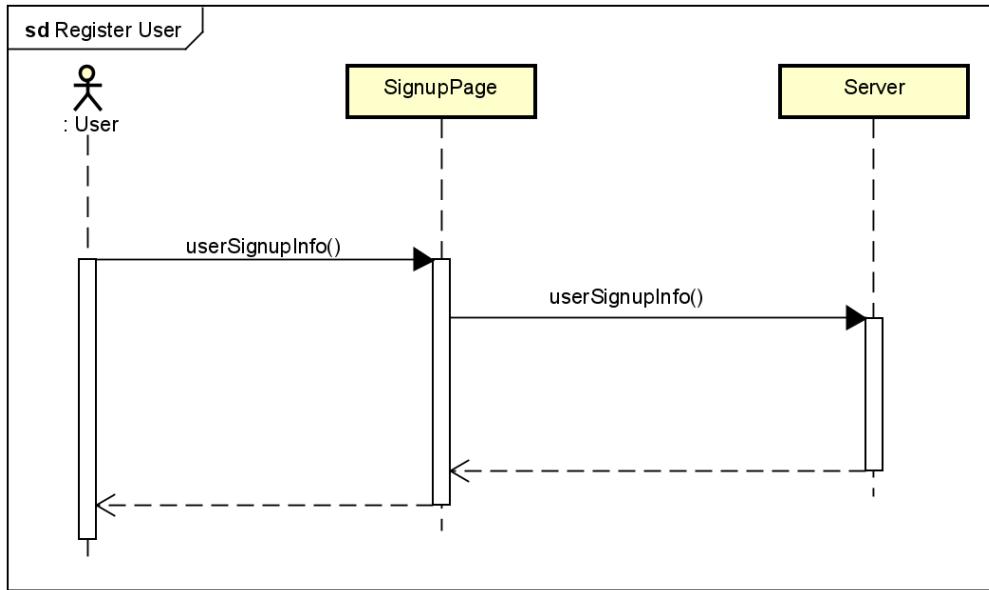


Figura X - Diagrama de sequência para registro de novo usuário na plataforma

Após a realização do cadastro do usuário na plataforma, é necessário que este se autentique através da realização de um *login* na plataforma, que será realizado através do usuário único e da senha cadastrada durante o caso de uso de registro de usuário.

Para realizar esta autenticação, o usuário deverá acessar a tela correspondente à esta funcionalidade no *Front-End* da plataforma e então será apresentado um formulário *online* onde deverá inserir os dados necessários para o *login*. Após inserir seus dados, o usuário irá submeter o formulário pelo *Front-End* que irá enviar estas informações para o *back-end*, que irá então verificar a autenticidade do usuário através da validação do nome de usuário e da senha enviados pelo *Front-End*, e, por fim, caso os dados estejam corretos, o *back-end* irá retornar ao *Front-End* uma mensagem dizendo que o usuário passa a estar autenticado durante aquela sessão, concedendo-lhe as devidas autorizações. Este fluxo caso de uso é esboçado pelo diagrama de sequência na Figura X.

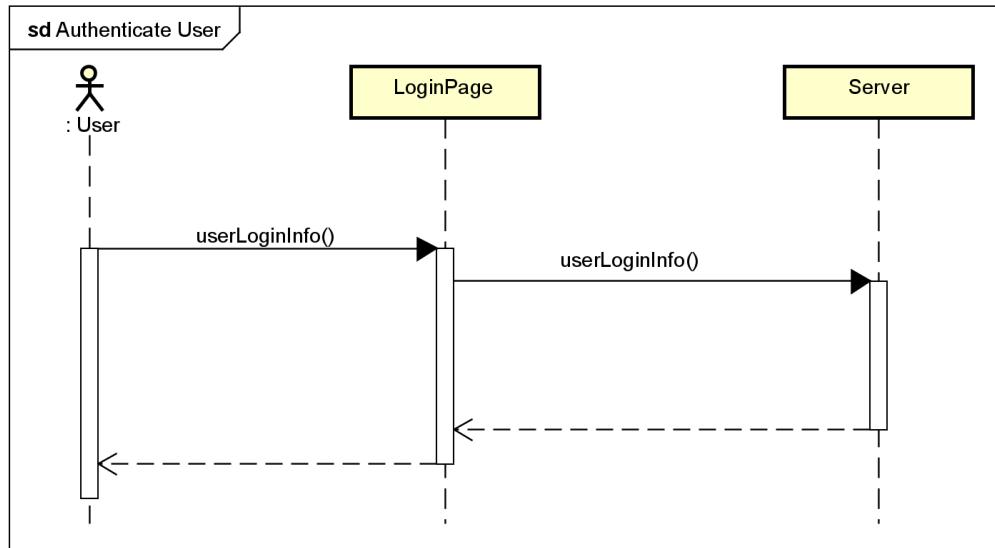


Figura X - Diagrama de sequência para autenticação de usuário na plataforma

4.5.2 Cadastro de carteira do usuário na plataforma

Após o usuário se cadastrar e autenticar na plataforma, para que seja possível utilizar as funcionalidades da plataforma que envolvem transações utilizando *smart-contracts* e *blockchain*, ele deverá registrar a chave pública de sua carteira de *Ethereum* na plataforma, necessária para os casos de uso que serão descritos a seguir, envolvendo tokenização, compra e venda de posse de imóveis, além da utilização de ativos tokenizados em empréstimos e a distribuição de retornos financeiros associados a ativos tokenizados na plataforma.

Para isso, o usuário autenticado irá acessar sua página de perfil no *Front-End*, onde poderá cadastrar a chave pública (endereço da carteira do usuário na rede Ethereum) de sua carteira por meio de um botão, que busca as carteiras do usuário disponíveis na MetaMask. Em o usuário selecionando a carteira escolhida, o *Front-End* irá enviar esta informação para o *back-end*, que atualizará o cadastro do usuário com o endereço de sua carteira, tornando-o apto a utilizar as funcionalidades da plataforma. A Figura X mostra o diagrama de sequência com todas as etapas descritas.

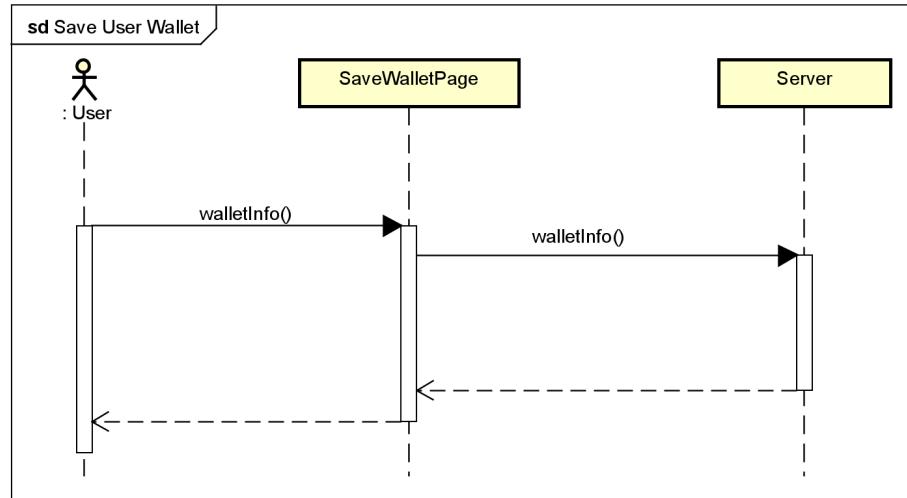


Figura X - Diagrama de sequência para cadastro de endereço de carteira de usuário na plataforma

4.5.3 Validação de documentos e tokenização de um imóvel

O caso de uso base da plataforma trata-se da tokenização de um ativo imobiliário.

Neste caso de uso, o proprietário do imóvel irá interagir com uma tela no *Front-End* e irá inserir os dados de endereço, área útil, número de registro em cartório e um documento identificando a posse do ativo imobiliário físico. Estas informações serão enviadas para o servidor e ficarão disponíveis para validação por um usuário administrador. Caso a validação ocorra, os dados do imóvel serão enviados para a *blockchain* por meio de um *smart contract* que representará os tokens do imóvel, sendo inicialmente 100% deles de posse do proprietário do ativo físico. Por fim, as informações do imóvel também são registradas no banco de dados da plataforma.

A Figuras X e X mostram os diagramas de sequência deste caso de uso envolvendo o dono e o administrador, as telas utilizadas, o servidor e o *smart contract*, além das devidas mensagens trocadas entre os elementos para que seja possível a tokenização.

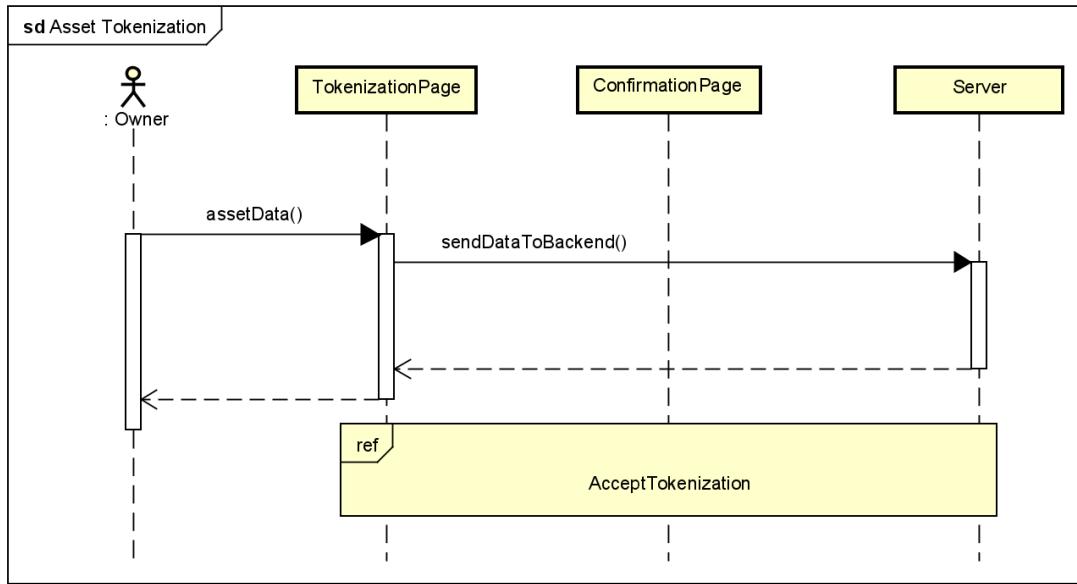


Figura X - Diagrama de sequência para a tokenização de um imóvel

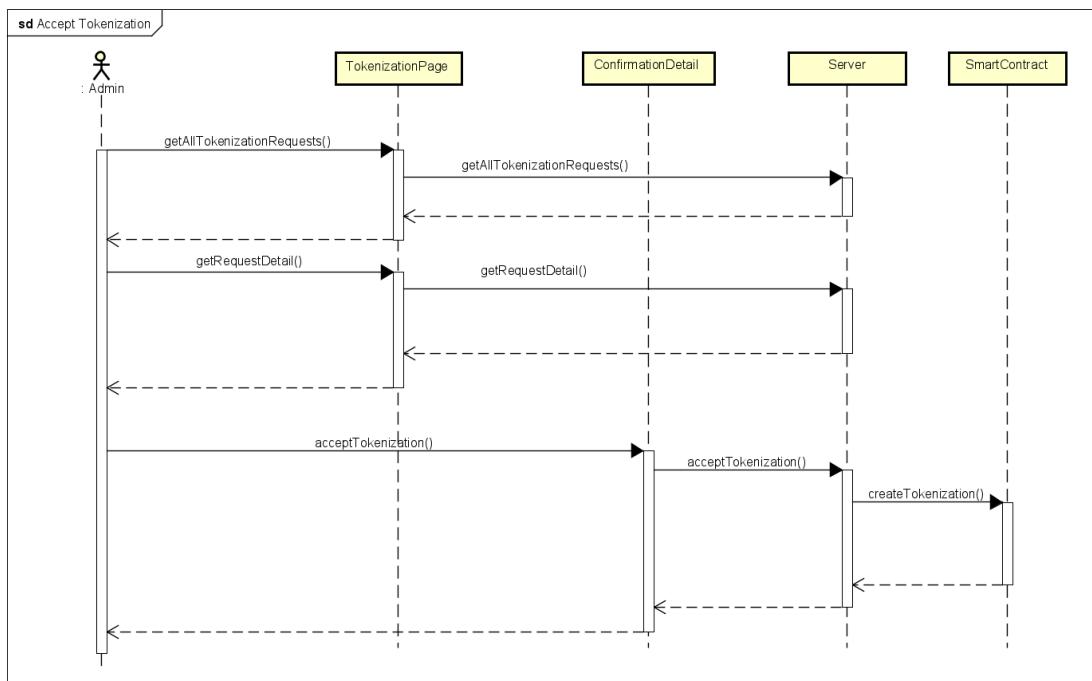


Figura X - Diagrama de sequência para a validação dos documentos de um imóvel para o processo de tokenização

4.5.4 Compra e venda de porcentagem de posse de ativo imobiliário

Para viabilizar que diversos usuários possuam porcentagens de imóveis tokenizados, foi proposta a possibilidade de compra e venda destas porcentagens.

Para isso, o comprador deverá acessar uma tela no *Front-End* com uma lista de todas as ofertas de porcentagem de ativos tokenizados disponíveis (que deverão ser colocadas à venda pelos respectivos proprietários) e informações básicas de cada uma delas. Ao se interessar por uma oferta, o comprador poderá interagir com essa tela e ser redirecionado a uma tela com informações detalhadas da oferta, onde poderá aceitá-la.

Após o aceite da oferta, o comprador deverá então realizar o pagamento para a plataforma que, ao custo de uma taxa, irá então repassar o valor pago ao dono do imóvel (fluxo de pagamento fora do escopo do projeto). Opcionalmente, o comprador também pode adicionar o comprovante de pagamento em uma tela do *Front-End* que lista todas as suas compras realizadas na plataforma.

Em seguida, um administrador da plataforma deverá registrar a confirmação do pagamento, para que, por fim, a plataforma realize a transferência de posse na *blockchain*, bem como atualizar as informações de posse em seu banco de dados.

Seguem abaixo, nas Figuras X, X e X, respectivamente, os diagramas de sequência deste caso de uso envolvendo o dono do imóvel, o comprador, as telas utilizadas, o servidor e o *smart-contract*, além das devidas mensagens trocadas entre os elementos para que seja possível a compra e venda de porcentagens de ativos imobiliários tokenizados.

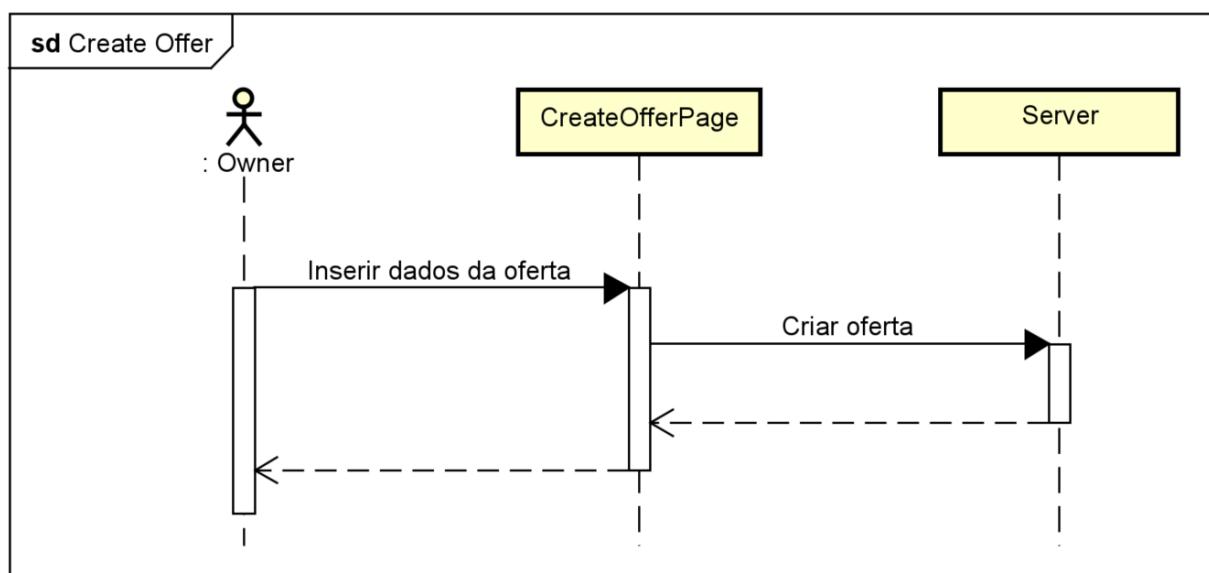


Figura X - Diagrama de sequência para a criação de oferta de venda de porcentagem de posse de ativo tokenizado

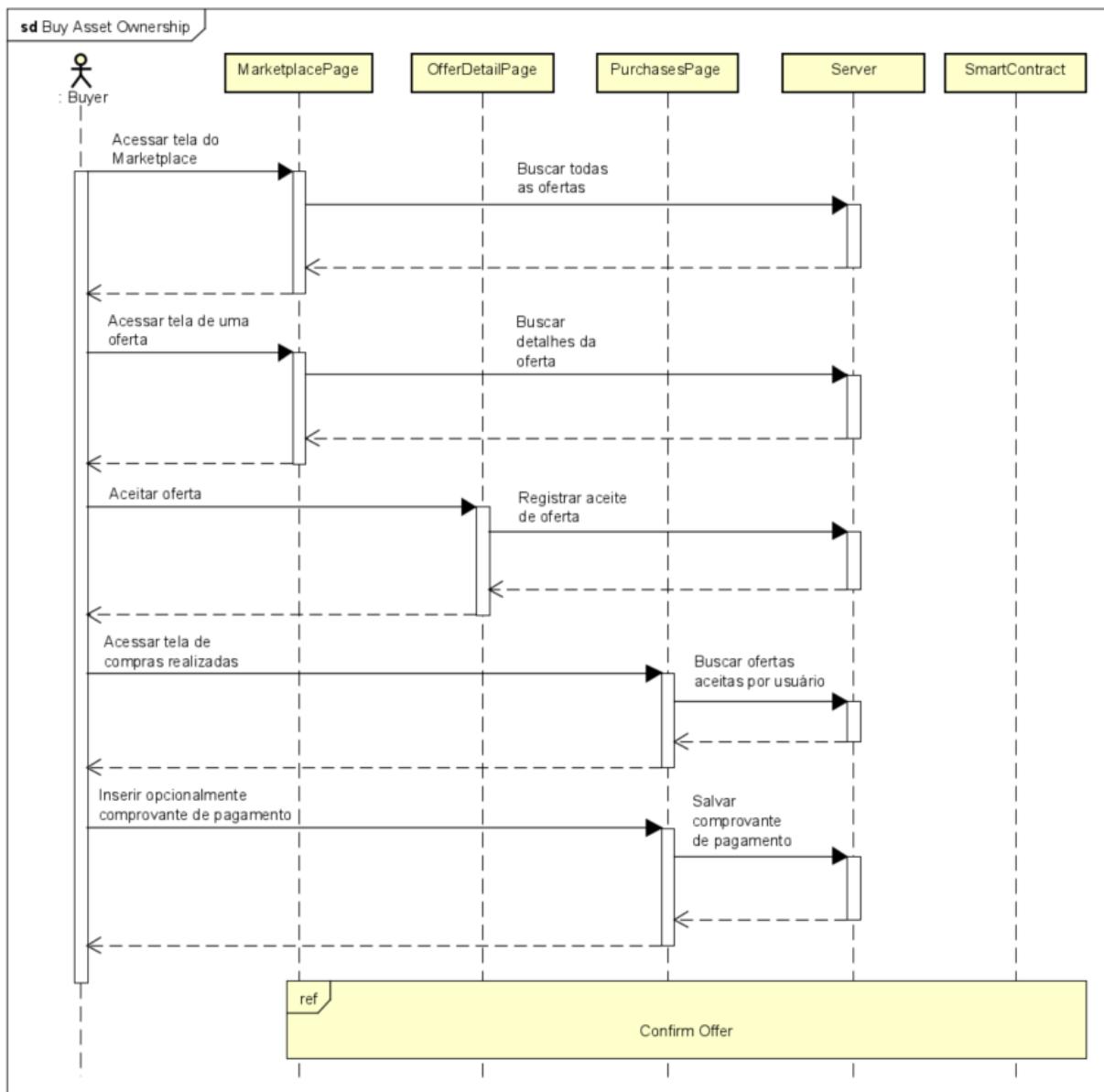


Figura X - Diagrama de sequência para aceite de oferta de ativo imobiliário tokenizado

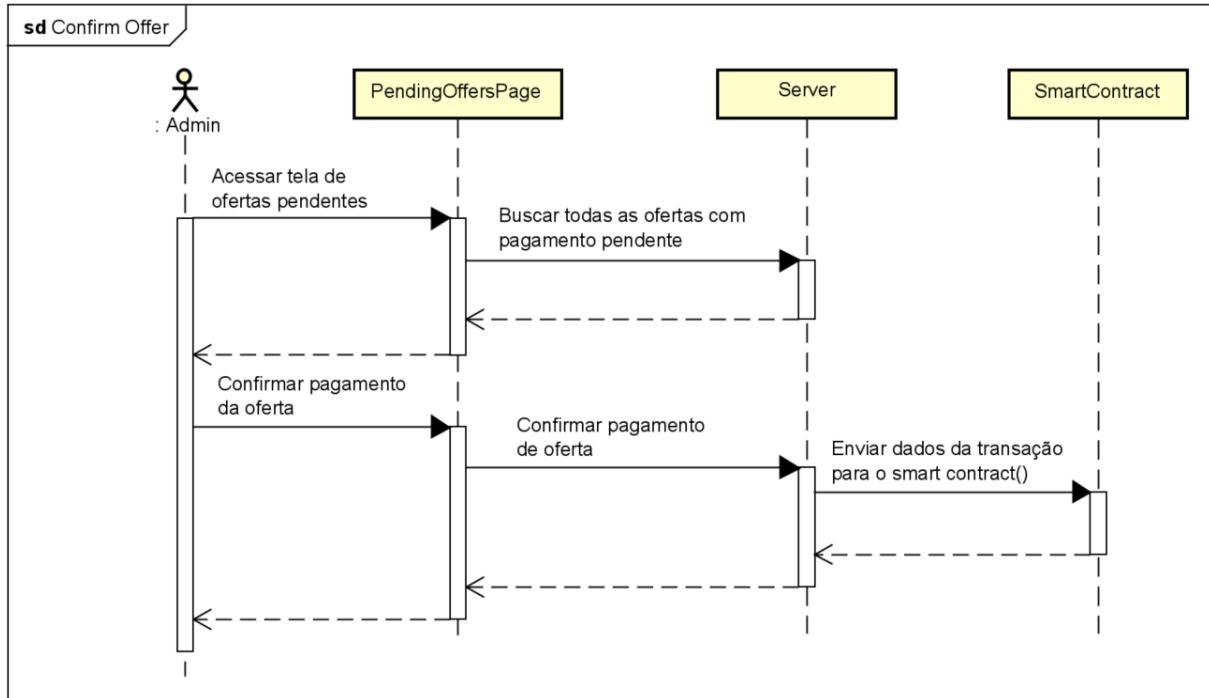


Figura X - Diagrama de sequência para a confirmação de compra de ativo imobiliário

4.5.5 Registro de garantia de empréstimo

Para o cenário de garantia de empréstimo, o dono do ativo imobiliário *tokenizado* será o devedor e irá acordar os termos do empréstimo com o credor, seja ele um banco ou uma pessoa física.

De posse das informações de empréstimo acordadas entre ambas as partes, neste projeto representadas por data de vencimento do empréstimo e porcentagem do imóvel tokenizado dado em garantia, o dono do ativo irá inserir estas condições na tela de criação de garantias e estas serão enviadas para o servidor, que irá armazená-las no banco de dados e as manterá no estado de aguardo de confirmação por parte de um administrador.

O administrador, por sua vez, irá verificar as informações da garantia do empréstimo e confirmar o valor emprestado pelo credor ao devedor (acordado por fora da plataforma), e então irá acessar a página de garantias pendentes da plataforma, onde deverá confirmar o registro da garantia para que os dados sejam enviados para o *smart contract* do imóvel na *blockchain* e também atualizados no banco de dados da plataforma.

As Figuras X e X, os diagramas de sequência deste caso de uso envolvendo o dono do ativo, o administrador, as telas utilizadas, o servidor e o *smart contract*,

além das devidas mensagens trocadas entre os elementos para que seja possível o empréstimo.

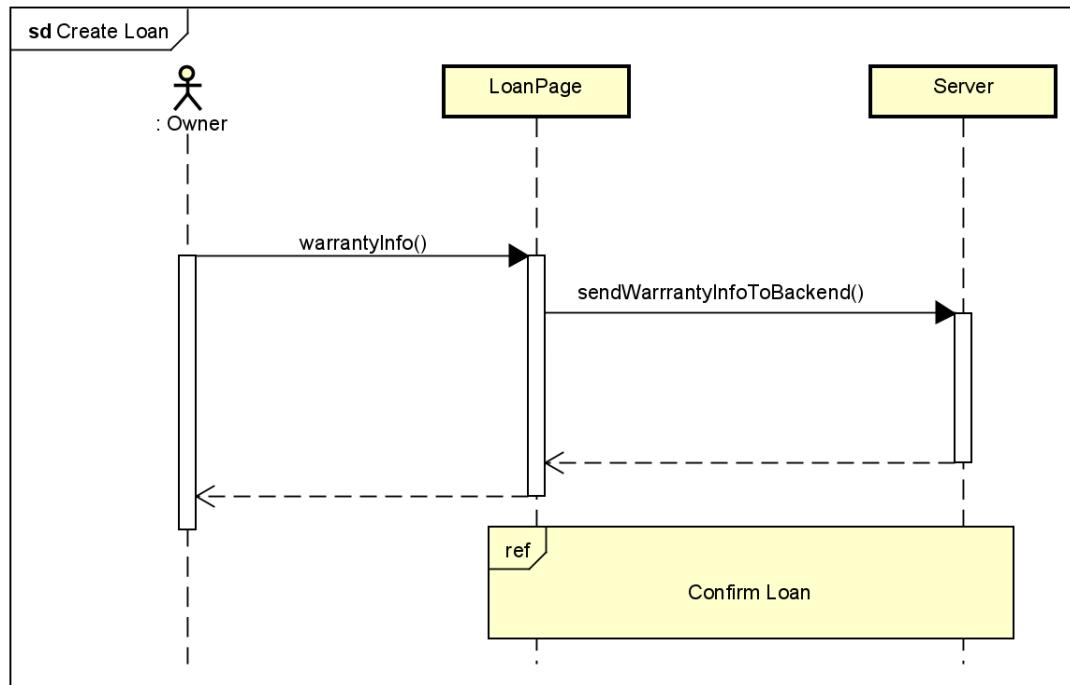


Figura X - Diagrama de sequência para criação de empréstimo

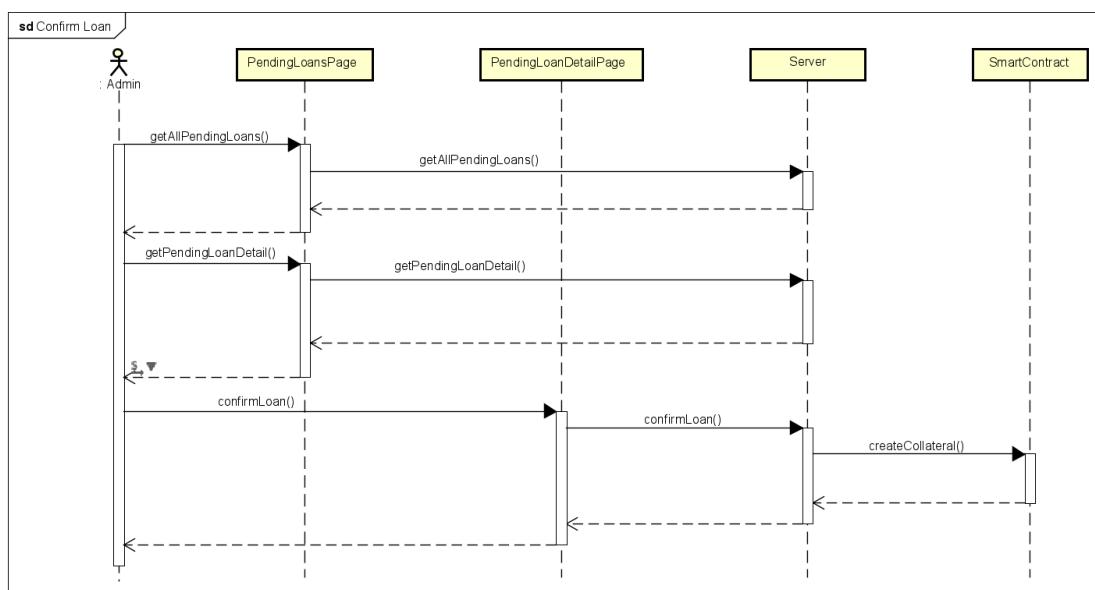


Figura X - Diagrama de sequência para confirmação de empréstimo

4.5.6 Quitação de empréstimo e tomada de garantia

Após associar uma certa porcentagem de um ativo imobiliário como garantia de um empréstimo, o dono do imóvel deverá quitá-lo antes que a data de vencimento do empréstimo seja alcançada.

Para isso, é necessário um caso de uso na plataforma em que o devedor consiga interagir com a interface da plataforma e registrar que quitou o empréstimo associado ao ativo imobiliário.

Após este registro, um usuário administrador irá realizar a verificação do pagamento e então deverá interagir com o Front-End de modo a confirmar a quitação, para que plataforma registre na *blockchain* a exclusão da garantia e libere para o proprietário a utilização da porcentagem do ativo que estava garantida. Por fim, a plataforma também deve atualizar o banco de dados próprio com as novas informações de posse para se manter atualizada em relação à *blockchain*.

As Figuras X e X mostram os diagramas de sequência deste caso de uso envolvendo o dono do ativo, o administrador, as telas utilizadas, o servidor e o *smart contract*, além das devidas mensagens trocadas entre os elementos para que seja possível a quitação do empréstimo de um ativo imobiliário tokenizado.

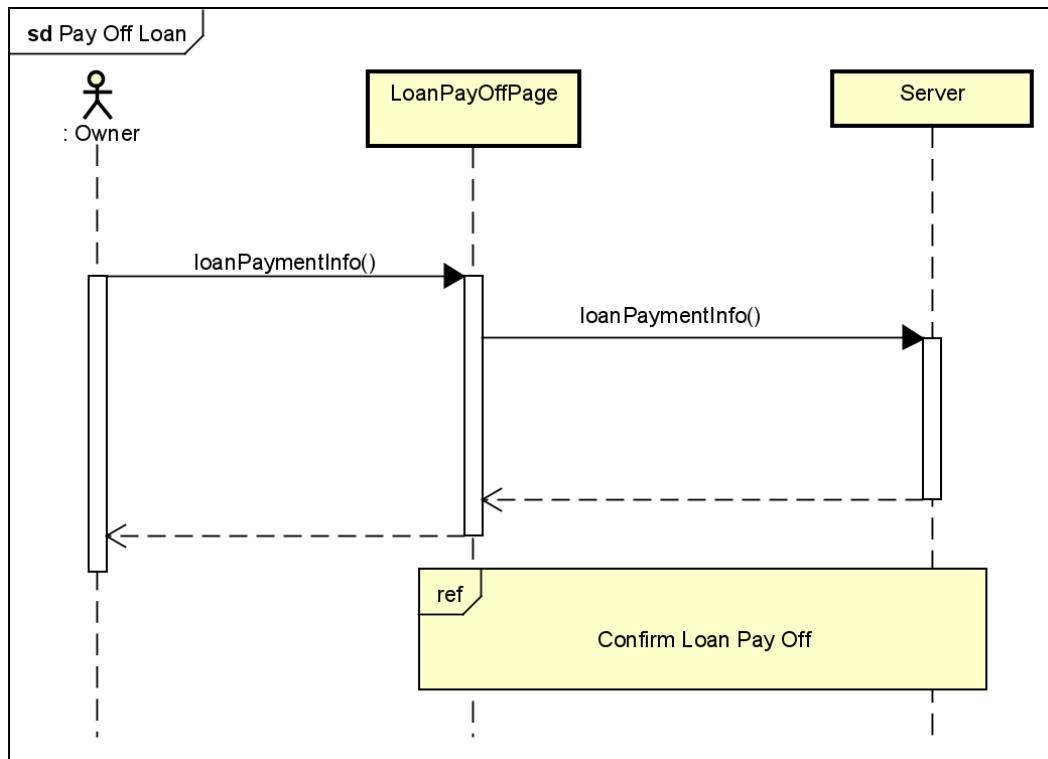


Figura X - Diagrama de sequência para registro de quitação de empréstimo

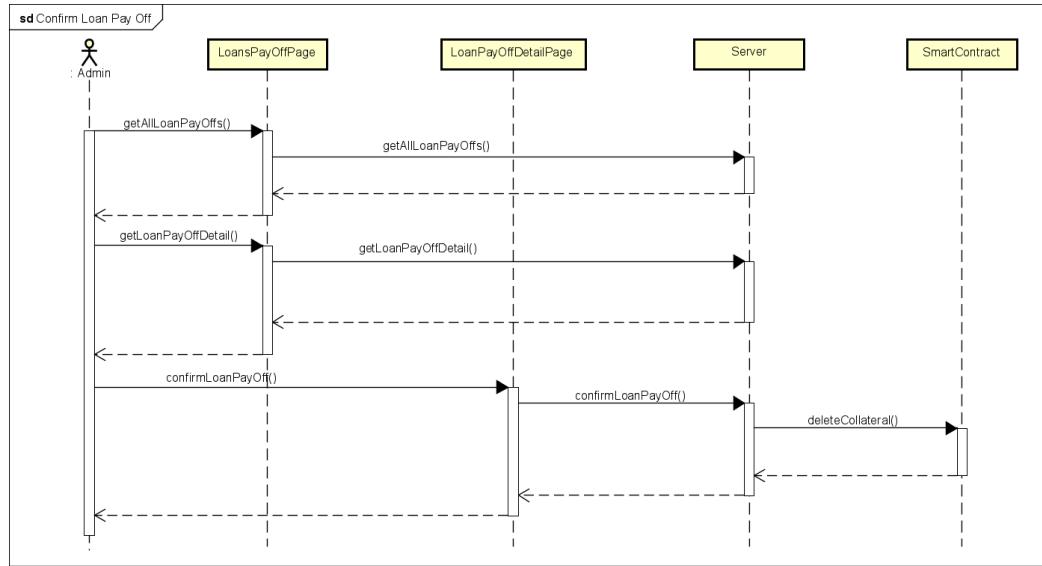


Figura X - Diagrama de sequência para confirmação de quitação de empréstimo

Caso o dono do imóvel não realize a quitação do empréstimo até a data de vencimento, o credor terá o direito de tomar posse da garantia utilizada no empréstimo. Para isso, o administrador é responsável por interagir com uma tela no *Front-End* para registrar a tomada da garantia do empréstimo.

Ao interagir com a tela, é feita a validação de se a data de vencimento já foi alcançada ou não, e, caso tenha sido, a plataforma irá registrar na blockchain a transferência de porcentagem do ativo imobiliário do devedor para o credor e também irá armazenar as informações necessárias em seu banco de dados. Este fluxo é ilustrado pelo diagrama de sequência na Figura X.

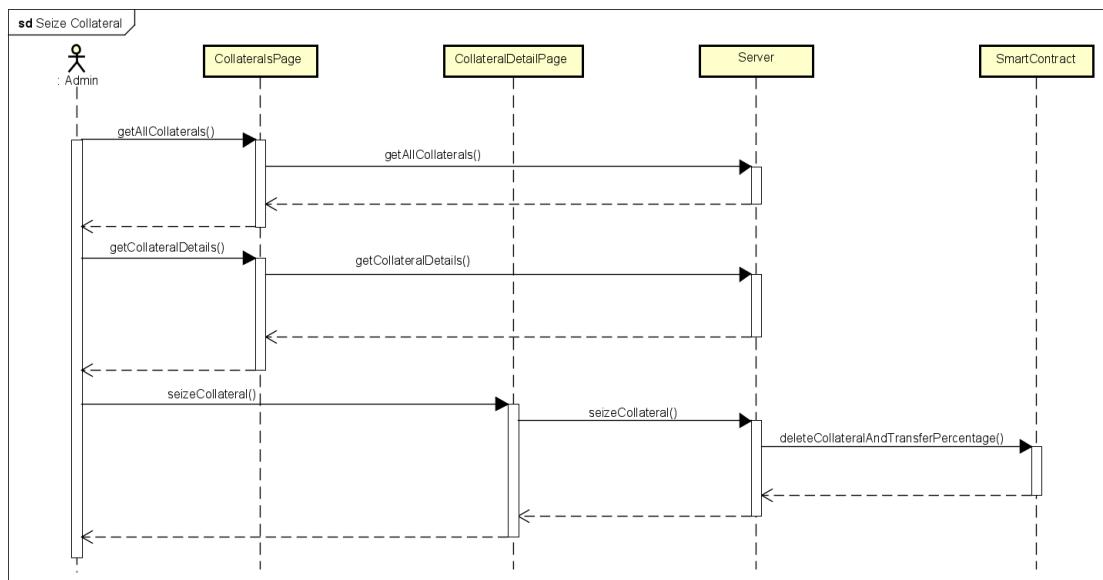


Figura X - Diagrama de sequência para tomada de garantia de empréstimo

4.5.7 Distribuição de retornos financeiros provenientes de aluguel

Para o cenário de distribuição de retornos financeiros provenientes de aluguel, simulando o pagamento do inquilino para a plataforma, o administrador poderá interagir com uma tela no *Front-End* para cadastrar o valor recebido. Feito o cadastro, o servidor envia o valor e a data do dia vigente para o *smart contract* do imóvel na *blockchain*, onde é feito o registro do valor total repartido entre todos os proprietários do imóvel de forma proporcional à porcentagem de posse de cada um. Além disso, estes dados também são salvos no banco de dados da plataforma, de forma sincronizada com a *blockchain*.

Ademais, cada um dos proprietários do ativo tokenizado pode acessar uma tela no *Front-End* para visualizar o registro de todos os pagamentos de aluguéis recebidos proporcionais à porcentagem que detinha no momento do pagamento.

As Figuras X e X mostram os diagramas de sequência deste caso de uso envolvendo o dono do ativo, o administrador, as telas utilizadas, o servidor e o *smart contract*, além das devidas mensagens trocadas entre os elementos para que seja possível a distribuição dos pagamentos de aluguéis entre todos os proprietários do ativo imobiliário tokenizado.

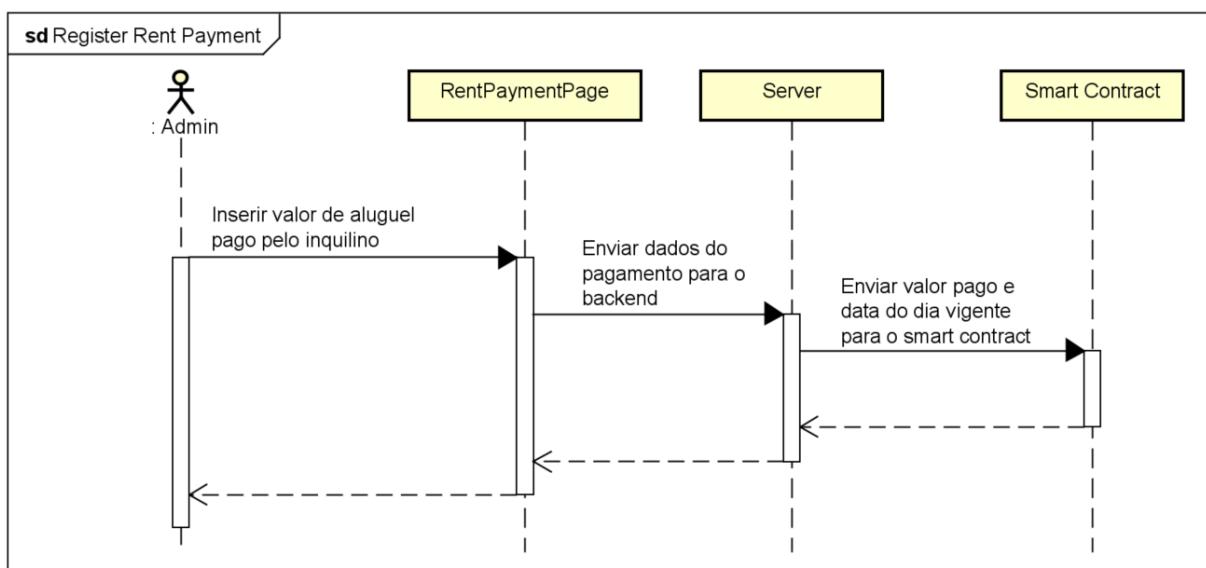


Figura X - Diagrama de sequência para registro de pagamento de aluguéis

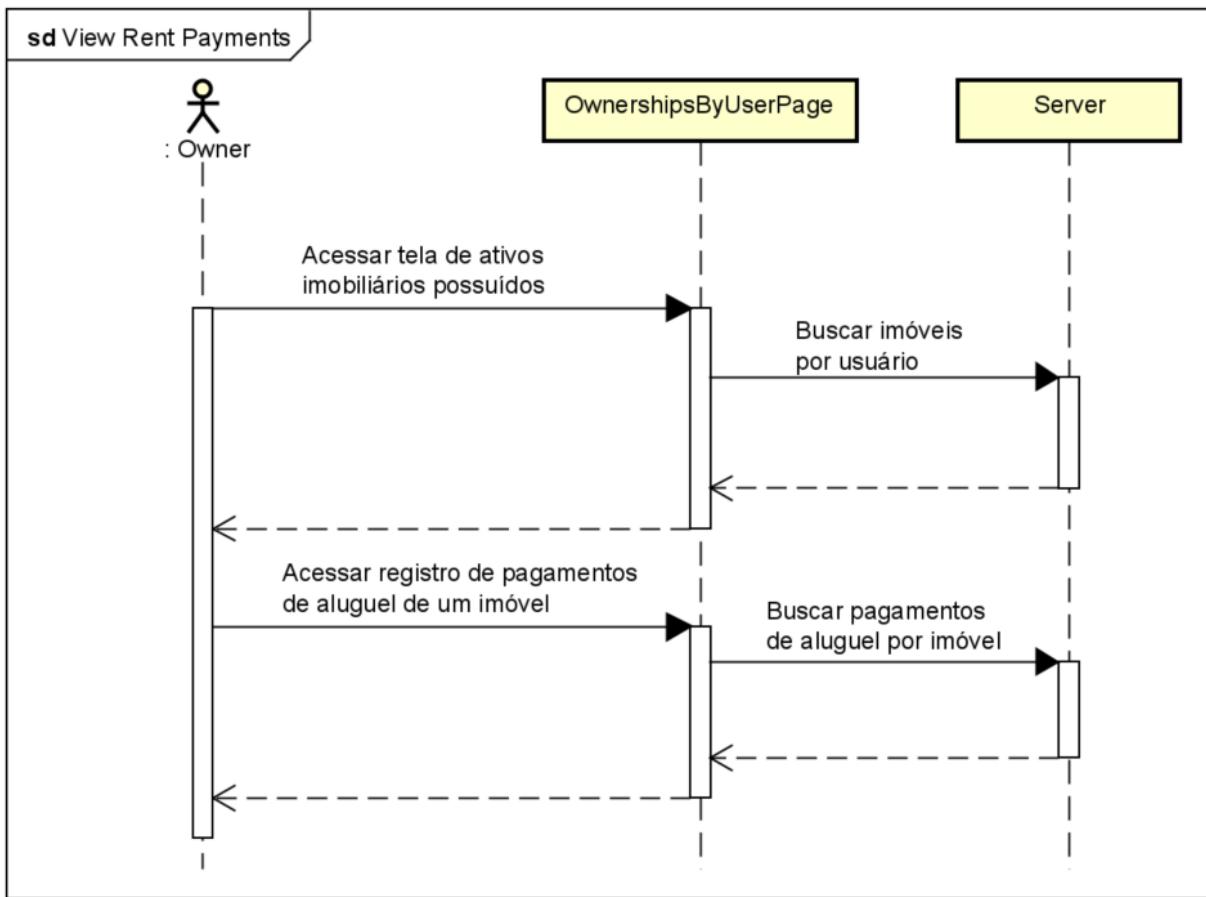


Figura X - Diagrama de sequência para visualização dos pagamentos de aluguéis recebidos por um proprietário

4.5.8 Auditoria de dados registrados na blockchain

O último caso de uso mapeado no presente projeto é o de auditoria dos dados do imóvel registrados em blockchain. Para isso, o administrador terá acesso a uma tela no *Front-End* listando todos os ativos tokenizados da plataforma. Para cada um deles, será possível interagir de modo que a plataforma busque e exiba os dados registrados de forma imutável na *blockchain*, relativos aos proprietários, seus respectivos pagamentos de aluguéis recebidos e suas garantias criadas, além dos dados gerais do imóvel tokenizado em questão.

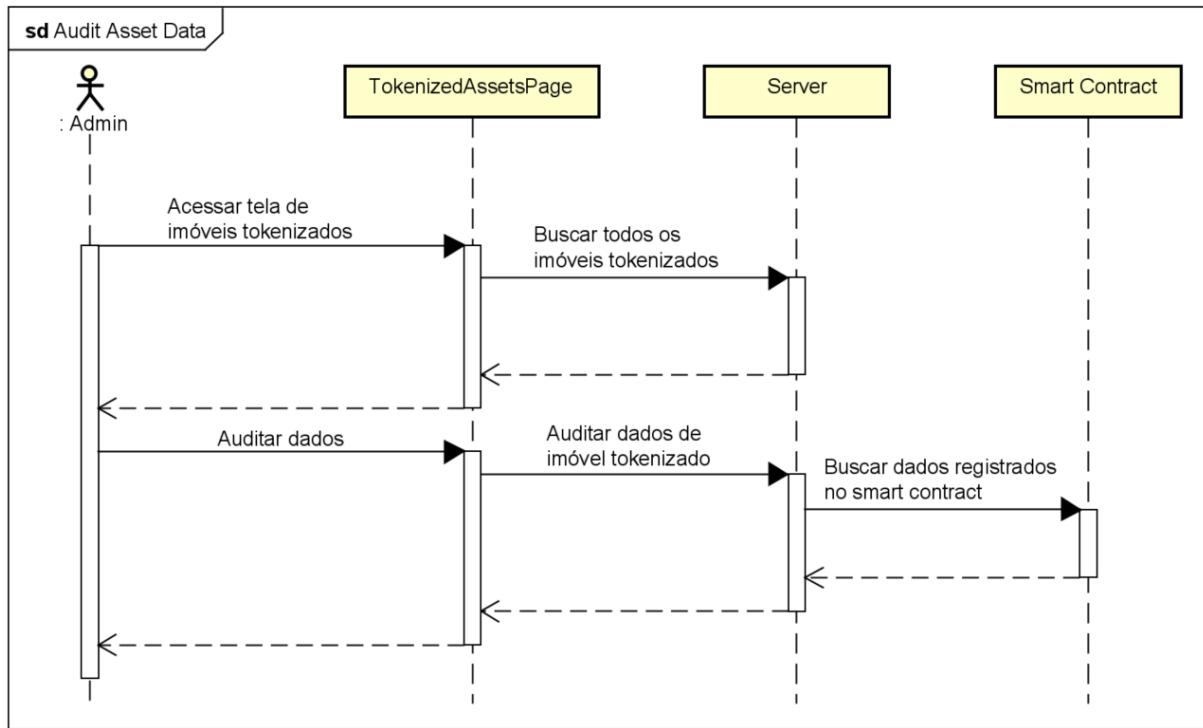


Figura X - Diagrama de sequência para auditoria dos dados do imóvel na blockchain

5 DESENVOLVIMENTO DO TRABALHO

Para a implementação do projeto, é necessário que sejam feitas diversas assunções com relação ao funcionamento e o escopo do projeto, e este capítulo tem o intuito de listar e explicá-las.

Primeiramente, como descrito nos capítulos anteriores, atualmente não é permitido que um mesmo imóvel seja utilizado como garantia de múltiplos empréstimos. A primeira premissa do projeto é que esta restrição seja alterada e que passe a ser permitido por lei que um imóvel sirva de garantia para mais de um empréstimo.

Outra premissa do projeto é a utilização de porcentagens dos ativos imobiliários durante o processo de tokenização, empréstimo e distribuição de lucros provenientes de aluguéis na plataforma, ou seja, ao tokenizar seu ativo imobiliário, o dono irá possuir 100% de seu ativo tokenizado e a partir daí, todas as transações serão realizadas com base em porcentagens do ativo, preservando a confidencialidade do valor absoluto do imóvel. Assim, os interessados em comprar uma porcentagem do ativo ou então um banco com o qual certa porcentagem do

ativo será utilizada como garantia terão que realizar uma avaliação própria do valor em reais do imóvel para avaliar a viabilidade do investimento.

Ainda sobre a posse de porcentagens do imóvel, o presente trabalho irá separar a posse de porcentagens da posse efetiva do imóvel, ou seja, serão utilizados dois atributos diferentes de posse, um atributo representando os donos de porcentagens e suas respectivas porcentagens possuídas e outro atributo para a posse efetiva do ativo imobiliário, que será atribuído apenas a um dono efetivo. Assim, será possível, por exemplo, que haja um dono efetivo de um ativo imobiliário que possui 0% do imóvel, enquanto os outros usuários que são donos de porcentagens possuem 100% do imóvel no total, mas mesmo assim preservando a posse do ativo imobiliário ao dono efetivo, que irá estar atrelado ao atributo que representa o dono efetivo do imóvel.

Com relação à implementação da plataforma em si, será realizada uma simplificação da representação do ativo imobiliário, utilizando-se apenas algumas informações tomadas como relevantes pelo grupo, sendo elas endereço, área útil total, número de registro em cartório e algum documento de identificação da posse do imóvel, pois sabe-se que o registro de um ativo imobiliário num cartório necessita de mais inúmeras informações [28], que foram consideradas externas ao escopo do projeto, visto que o foco principal do presente trabalho são as funcionalidades propostas, bastando apenas que sejam utilizadas informações suficientes para que seja possível distinguir-se unicamente cada um dos ativos.

Ademais, com relação à transferências de porcentagens dos ativos, bem como o empréstimo e a distribuição de aluguéis, foi assumido que todas as transações monetárias serão realizadas por fora da plataforma, através do método que as partes envolvidas combinarem, seja pagamento em cédulas, transferências bancárias ou outros meios, com a plataforma servindo apenas para o registro dessas transações e das informações necessárias e suficientes para que seja possível comprovar que ocorreram ou irão ocorrer as transações monetárias envolvendo os ativos imobiliários.

5.1 TECNOLOGIAS UTILIZADAS

Nesta seção, estão listadas as principais tecnologias utilizadas na implementação do projeto, fundamentais para o desenvolvimento de forma mais ágil e padronizada dos objetivos deste.

5.1.1 Node.js

Node.js é um ambiente de execução *open-source* em JavaScript, idealizado para o desenvolvimento de aplicações escaláveis do lado do servidor [29]. No projeto, será usado na implementação do Back-End, por meio do framework NestJS [30], que define uma estrutura de desenvolvimento do código a ser desenvolvido de maneira a agilizar e padronizar a implementação dos requisitos de projeto necessários ao fornecer uma série de ferramentas facilitadoras durante o desenvolvimento.

5.1.2 React

React é uma biblioteca *open-source* de JavaScript usada para desenvolvimento de interfaces para usuários baseada em componentes *UI* (*User Interface*, ou Interface de Usuário) [31]. Será utilizado no desenvolvimento do Front-End do projeto em conjunto com outras bibliotecas com as quais possui integração, de modo a viabilizar um desenvolvimento mais padronizado e ágil da interface de usuário que permitirá que o usuário execute as funções existentes na plataforma.

5.1.3 PostgreSQL

PostgreSQL é um banco de dados relacional *open-source* que utiliza e estende a linguagem SQL [32], e será utilizado para armazenar os dados de ativos, usuários e transações na plataforma.

5.1.4 Solidity

Solidity é uma linguagem de programação de alto nível e orientada a objetos utilizada para a implementação de *smart contracts* que podem ser executados por

EVMs (*Ethereum Virtual Machine*) numa rede de blockchain Ethereum [33]. É através desta linguagem de programação que serão implementadas as funcionalidades de tokenização, empréstimo e distribuição de aluguéis na plataforma, armazenando as informações necessárias num blockchain Ethereum, bem como em estados internos contidos no *smart contract* a ser desenvolvido.

5.1.5 Remix IDE

Remix IDE é um ambiente de desenvolvimento integrado (*IDE*) que permite o desenvolvimento, implantação e teste de *smart contracts* para aplicações de redes *blockchain* através de uma aplicação *web* [34]. É através deste ambiente de desenvolvimento que serão desenvolvidas as principais funcionalidades do projeto envolvendo *blockchain* e *smart contracts*.

5.1.6 Web3.js

Web3.js é uma coleção de bibliotecas JavaScript que implementam diversas funcionalidades de interação com nós de uma rede *blockchain* Ethereum [35]. Esta coleção de bibliotecas será utilizada tanto no desenvolvimento do Front-End como no desenvolvimento do Back-End da plataforma, sendo responsável pela comunicação da aplicação desenvolvida com os *smart contracts* desenvolvidos.

5.1.7 Ethereum Goerli Testnet

Devido à utilização da tecnologia do blockchain Ethereum no projeto, foi necessária a escolha de uma rede Ethereum a ser utilizada durante o projeto.

Dentre as alternativas de redes Ethereum a serem escolhidas, decidiu-se utilizar a rede de testes Goerli, uma rede blockchain pública utilizada para testes, em que não é necessário que os usuários utilizem Ethereum real para realizar as transações de criação de *smart contracts* bem como a utilização de seus métodos. Ademais, também é utilizado no projeto o *Etherscan*, um explorador de blocos para redes Ethereum, no qual é possível pesquisar dados públicos de transações, blocos, endereços de carteiras e *smart contracts* registrados na rede.

5.1.8 TypeORM

TypeORM é um ORM compátivel com aplicações desenvolvida em Node.js, responsável por realizar o mapeamento de classes declaradas no código para entidades do banco de dados, e vice-versa, além de permitir consultas e atualizações no banco de dados sem a necessidade da escrita de scripts em SQL. [36]

5.2 PROJETO E IMPLEMENTAÇÃO

Definidos os casos de uso a serem implementados, bem como a estrutura e arquitetura do projeto e as tecnologias a serem utilizadas, foi desenvolvido o *smart contract* a ser utilizado para a criação e transação dos imóveis de forma tokenizada, o servidor da plataforma que lida com as regras de negócio dos casos de uso especificados e o *Front-End* que constitui a interface da plataforma com o usuário.

Neste capítulo serão detalhadas as implementações realizadas destes diferentes componentes do projeto, ilustrando-se de formas adequadas os resultados obtidos para cada parte constituinte da plataforma.

5.2.1 Desenvolvimento do Smart Contract

Para a representação dos imóveis na forma de *tokens* no blockchain foi desenvolvido um *smart contract* com base nos padrões ERC-20 e ERC-721, padrões bastante utilizados para *tokens* fungíveis e não fungíveis na rede de blockchain Ethereum, respectivamente.

Este *smart contract* possui diversos atributos que são armazenados através de seu estado e que são referentes às informações utilizadas para identificar o imóvel, bem como as informações utilizadas para identificar os proprietários de porcentagem de posse do imóvel tokenizado, informação do dono efetivo do imóvel, o registro de garantias e o registro de pagamentos de aluguéis referentes ao imóvel. Além disso, o *smart contract* possui diversos métodos que implementam as regras de negócio de transferência de posse, criação e deleção de garantias associadas a empréstimos e auditoria das informações do imóvel, donos de porcentagens,

garantias atribuídas ao imóvel e pagamentos de lucros provenientes da utilização do imóvel.

Neste sentido, é possível notar que cada imóvel tokenizado irá ser associado a um *smart contract*, que será implantado no blockchain no momento de sua tokenização e será associado a um endereço público no blockchain que será armazenado pela plataforma em seu banco de dados.

Nos próximos itens serão melhor detalhados os atributos contidos no smart contract desenvolvido, bem como as funções desenvolvidas pelo grupo para garantir a execução das regras de negócio necessárias para o presente trabalho e o processo de publicação e implantação do contrato na rede blockchain Goerli escolhida para a utilização no projeto.

5.2.1.1 Atributos armazenados no *smart contract* desenvolvido

Nesta seção, serão detalhados os atributos presentes no *smart contract* desenvolvido, explicitando-se seus nomes e funções. São eles:

- ***assetAddress***: Atributo do tipo *string* que registra o endereço físico do imóvel tokenizado.
- ***assetUsableArea***: Atributo do tipo *uint* (número inteiro positivo) onde é registrado o total de área útil em metros quadrados (m^2) do imóvel tokenizado.
- ***assetId***: Atributo do tipo *uint* que armazena o número do registro em cartório do imóvel.
- ***creator***: Atributo do tipo *address* (endereço de carteira no blockchain) que é associado à chave pública da carteira da plataforma.
- ***effectiveOwner***: Atributo do tipo *address* que é associado à chave pública da dono efetivo atual do imóvel tokenizado.
- ***owners***: Lista de atributos do tipo *address* referentes aos endereços das carteiras dos atuais donos do imóvel tokenizado.
- ***percentageOwners***: Lista de endereços de carteiras que associa cada carteira à uma estrutura denominada de *Owner* que armazena todos os dados relativos à posse de cada dono do imóvel, desenvolvida pelo grupo e descrita a seguir:

- A estrutura *Owner* é composta por 5 campos, sendo eles ***shares*** (do tipo *uint*, representando a porcentagem de posse do ativo que o dono possui), ***walletAddress*** (*address*, representando o endereço da carteira do dono em questão), ***isEffectiveOwner*** (booleano, verdadeiro caso o dono em questão seja dono efetivo do imóvel, e falso caso contrário), ***collaterals*** (lista de *Collateral*, estrutura definida a seguir) e ***rentPayments*** (lista de *RentPayment*, estrutura definida a seguir).
 - ***Collateral***: Estrutura contendo as informações referentes à uma garantia associada a um dono. Composta pelos atributos ***bankId*** (*address*, referente à carteira do usuário credor), ***collateralShares*** (*uint*, representando a porcentagem de posse do ativo que foi utilizada pelo dono como garantia do empréstimo) e ***expirationDate*** (*uint*, data de expiração referente ao prazo da quitação do empréstimo realizado).
 - ***RentPayment***: Estrutura contendo as informações de pagamento de lucro proveniente do imóvel tokenizado e recebido pelo dono em questão. Constituído por três atributos do tipo *uint*, sendo eles ***amount*** (valor recebido pelo dono da porcentagem do imóvel em questão), ***shares*** (porcentagem possuída pelo dono em questão no momento do pagamento do lucro) e ***paymentDate*** (data do pagamento do lucro referente ao imóvel tokenizado).
- ***totalSupply***: Atributo do tipo *uint* representando o número total de tokens em oferta do imóvel tokenizado. Foi utilizado o valor de 1000 *tokens* no presente trabalho, com transferência mínima de 1 *token*, proporcionando uma granularidade de 0.1% ao ativo tokenizado.

5.2.1.2 Métodos desenvolvidos no *smart contract*

Nesta seção, serão detalhados os métodos presentes no *smart contract* desenvolvido para execução das regras de negócio dos casos de uso mapeados, explicitando-se seus nomes e funções. São eles:

- ***constructor***: Método utilizado para a criação de uma nova tokenização através dos dados do imóvel e do usuário que está tokenizando seu imóvel. Recebe como parâmetros o endereço da carteira do usuário que está criando a tokenização do seu imóvel, o endereço físico do imóvel, o total de área útil do imóvel expresso em metros quadrados (m^2) e o número do registro em cartório do imóvel. Atribui 100% da porcentagem de posse (1000 *tokens*) ao usuário e lhe garante a posse efetiva do imóvel tokenizado.
- ***transferOwnership***: Método utilizado para a execução das regras de negócio do caso de uso de compra e venda de porcentagem e da posse efetiva do imóvel tokenizado. Recebe como parâmetros a porcentagem a ser transferida, o endereço da carteira do vendedor e do comprador, bem como um parâmetro booleano que é verdadeiro quando a transferência de porcentagem envolve também a transferência da posse efetiva e falso caso contrário. Registra a transferência de porcentagem de posse do vendedor para o comprador
- ***createCollateral***: Método utilizado para a execução do caso de uso de cadastro de garantia utilizando o imóvel como garantia do empréstimo. Recebe como parâmetros o endereço da carteira do credor e do devedor, a porcentagem a ser utilizada como garantia do empréstimo e a data de expiração deste. Associa a porcentagem e a data de expiração indicadas nos parâmetros do método ao endereço da carteira do credor, bloqueando a movimentação da devida porcentagem do ativo da conta do devedor.
- ***deleteCollateral***: Método utilizado para registrar a quitação de um empréstimo e também utilizado durante o fluxo de tomada de posse de garantia. Recebe como parâmetros de execução o endereço da carteira do devedor e do credor, a porcentagem de posse do ativo utilizada como garantia e a data de expiração do empréstimo. Elimina o registro do empréstimo cujos atributos foram recebidos como parâmetros da execução do método.
- ***registerRentPayment***: Método para realizar o registro de pagamento de lucros provenientes de aluguel do ativo imobiliário tokenizado. Recebe como parâmetros o valor a ser distribuído entre os donos de porcentagem de posse do imóvel tokenizado e a data da realização do pagamento. Efetua o registro do pagamento de cada um dos donos do ativo tokenizado de forma

proporcional à porcentagem que cada um possui, registrando também a data do pagamento.

5.2.1.3 Publicação e implantação do *smart contract* no blockchain Goerli

Através do *smart contract* desenvolvido e com seus atributos e métodos descritos nas seções anteriores, a criação e a publicação dos smart contracts referentes aos imóveis tokenizados podem ser efetuados.

Na solução desenvolvida no presente trabalho, a plataforma efetua todas as comunicações com o blockchain (ou seja, o usuário da plataforma não precisa realizar nenhuma transação, apenas necessita cadastrar sua chave pública na plataforma, que irá utilizar este endereço de carteira para associar às transações envolvendo ativos tokenizados), sejam elas a criação de um *smart contract* associado a um determinado imóvel ou a execução de algum dos métodos de tal *smart contract* no blockchain e, portanto, a plataforma possui sua própria carteira no blockchain e utiliza seus próprios recursos (no caso da rede de teste Ethereum, a moeda Ether, de sigla ETH) para enviar tais transações para o blockchain.

Esta comunicação com o blockchain é realizada pelo servidor da plataforma, que se conecta em um nó da blockchain da rede de testes Ethereum Goerli disponibilizado pela plataforma Infura [X], e através dele assina e envia todas as transações necessárias nos fluxos dos casos de uso implementados para o blockchain utilizado.

5.2.2 Estrutura do banco de dados relacional

A Figura X mostra a modelagem do banco de dados, na qual é possível visualizar as tabelas de entidades e seus relacionamentos com as demais.

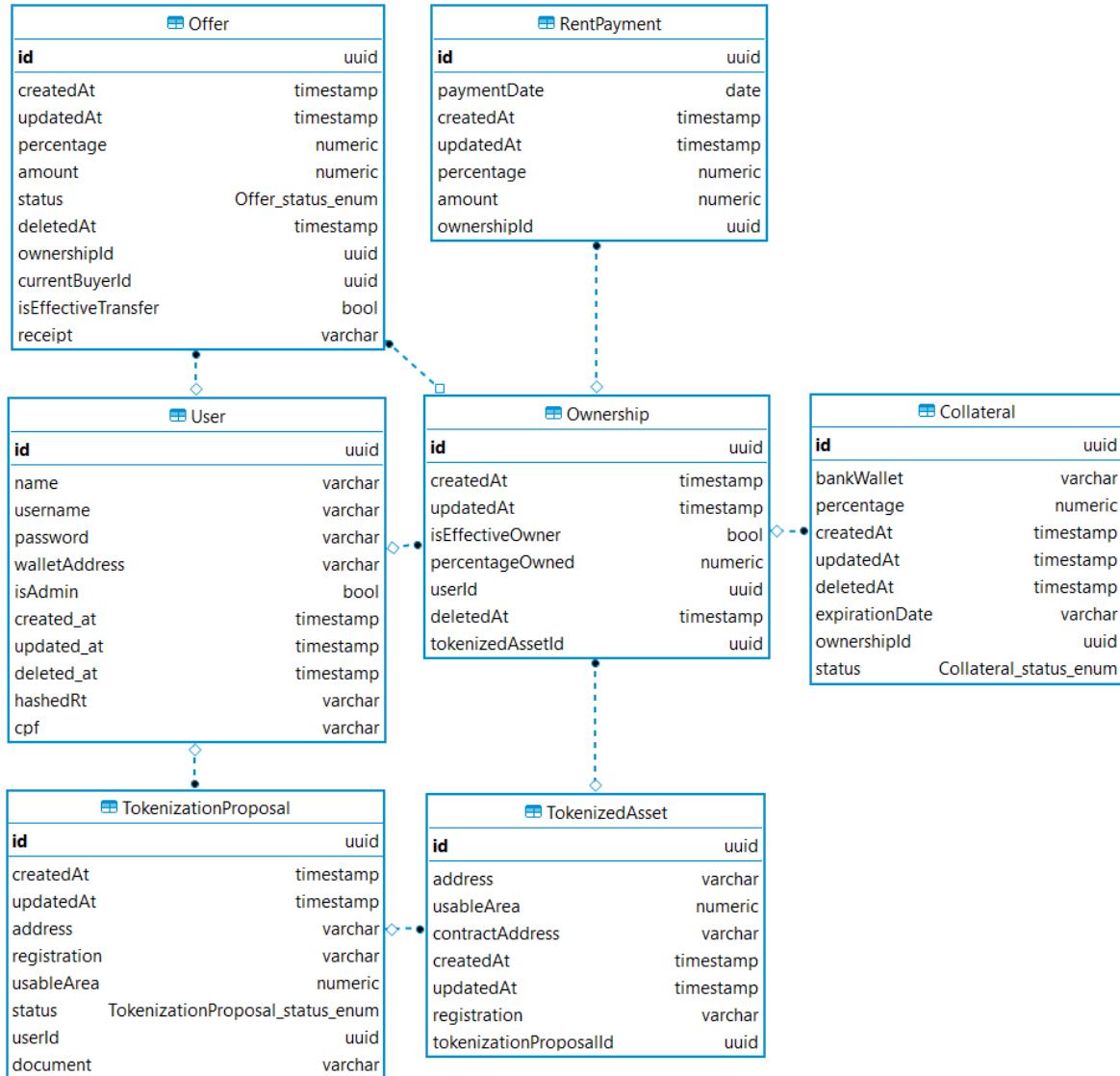


Figura X - Modelagem do banco de dados do projeto

Detalhando cada uma das entidades idealizadas:

- **User**: Tabela de usuários registrados na plataforma. Junto dos dados básicos de cadastro, também são armazenados o endereço da carteira digital do usuário e um campo identificando se o usuário é administrador da plataforma ou não.
- **TokenizationProposal**: Armazena as propostas de tokenização feitas por usuários. Para propor uma tokenização, são necessários os dados de endereço, número do registro em cartório, área útil e documento do imóvel. As propostas podem ter os seguintes status: *PENDING* - proposta

aguardando validação da plataforma; *APPROVED* - proposta aprovada pela plataforma; e *REFUSED* - proposta recusada pela plataforma.

- **TokenizedAsset:** Quando uma proposta de tokenização é aceita pela plataforma, os dados do imóvel são enviados para a blockchain e associados a um token, cujo endereço é salvo nesta tabela, juntamente com os dados do imóvel.
- **Ownership:** Nesta tabela, é armazenada uma estrutura de dados que representa a participação de um usuário na posse de um imóvel *tokenizado*. O atributo *isEffectiveOwner* indica se o usuário é o proprietário efetivo do imóvel (registrado em cartório) e *percentageOwned* indica o percentual que o usuário detém do imóvel, podendo variar de 0 a 1.
- **Offer:** A posse e os percentuais dos imóveis poderão ser transacionados na plataforma por meio de ofertas criadas pelos donos. Para isso, são armazenados os dados de valor, porcentagem oferecida e status da oferta, que pode assumir os seguintes valores: *AVAILABLE* - oferta disponível para compras; *WAITING_PAYMENT* - oferta aguardando confirmação de pagamento; *ACCEPTED* - oferta aceita e paga; e *CANCELED* - oferta cancelada. Caso esteja no status *WAITING_PAYMENT*, necessariamente terá um usuário comprador associado por meio da chave *currentBuyerId*. Ademais, o campo opcional *receipt* armazena o comprovante de pagamento da oferta.
- **Collateral:** Esta tabela armazena os dados da funcionalidade de garantia de empréstimos, em que *bankWallet* é o endereço da carteira digital do credor, *expirationDate* é a data de vencimento do empréstimo, *percentage* é o percentual do imóvel dado em garantia e status podendo assumir os seguintes valores: *PENDING_CONFIRMATION* - garantia pendente de validação pela plataforma; *ACTIVE* - garantia ativa na plataforma; *AWAITING_LOAN_PAYMENT_VALIDATION* - aguardando confirmação da quitação do empréstimo; *SEIZED* - garantia tomada pelo credor; e *CANCELED* - garantia cancelada na plataforma.
- **RentPayment:** Os pagamentos de aluguéis de imóveis tokenizados com mais de um dono são registrados nesta tabela. Sempre que o administrador registrar um pagamento de um aluguel, para cada dono de um percentual do

imóvel será salvo um registro com o valor pago proporcional a sua participação, o percentual do valor total do aluguel e a data do pagamento.

5.2.3 Desenvolvimento do servidor

O servidor desenvolvido para o back-end do projeto trata-se de uma *API REST*, e é responsável por implementar as regras de negócio da plataforma, enviar e interagir com os *smart contracts* na *blockchain* e armazenar dados relevantes num banco de dados próprio. A aplicação foi criada em um repositório no *github* chamado *asset-tokenization-back*, no qual os integrantes do grupo puderam trabalhar simultaneamente. O padrão de desenvolvimento adotado é composto por controladores, encarregados de tratar requisições *HTTP* recebidas nas rotas da *API*, validar os dados contidos nestas requisições, direcioná-los corretamente para os serviços que contenham a lógica necessária para interpretar estes dados e, por fim, gerar uma resposta *HTTP* para a origem da requisição. Os serviços, no escopo deste projeto, têm as seguintes atribuições: Criar, ler, atualizar ou remover entidades e relacionamentos do banco de dados, por meio do *TypeORM*; enviar *smart contracts* com dados de imóveis para a *blockchain*; e persistir ou atualizar dados dos *smart contracts* já enviados anteriormente para a *blockchain*.

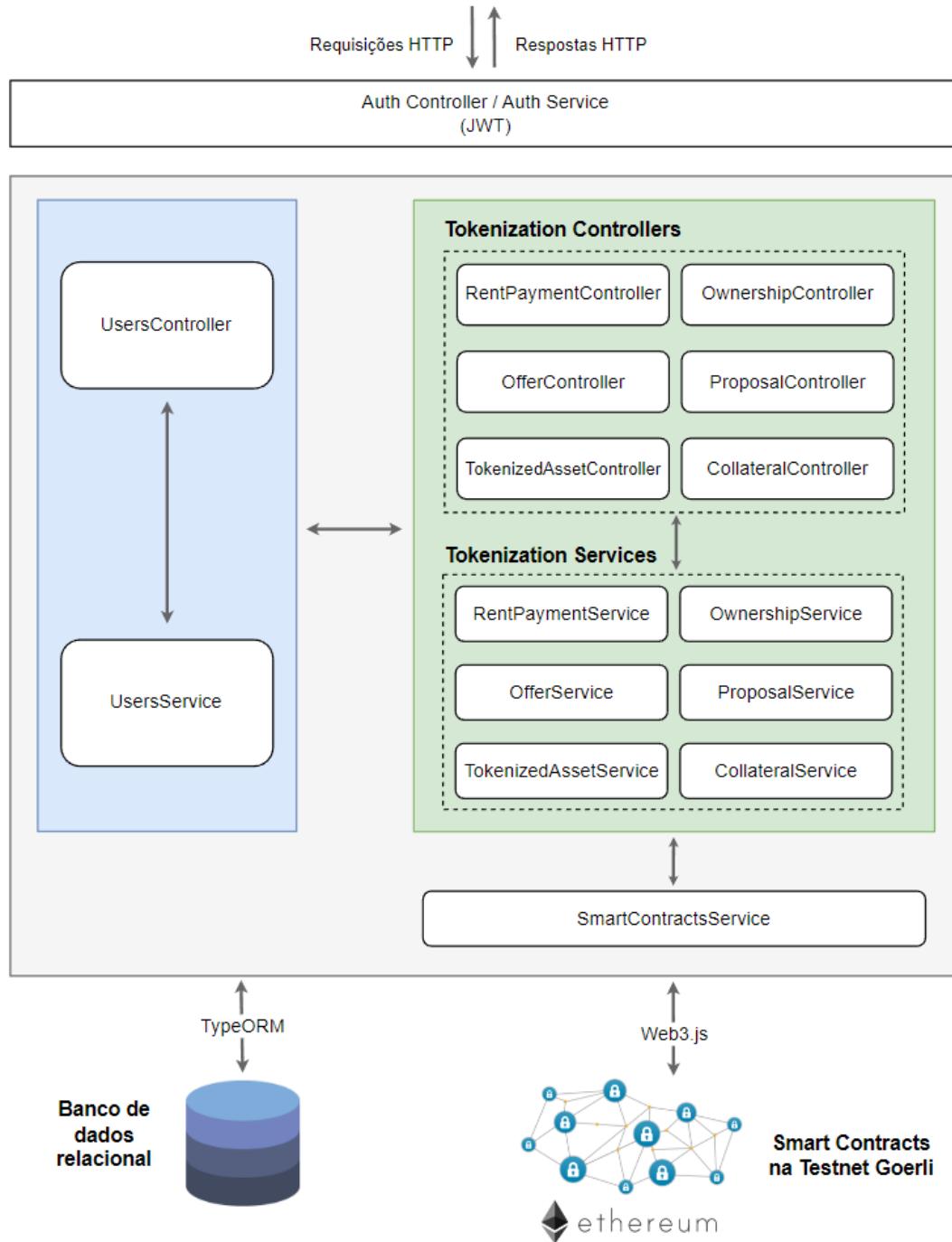


Figura X - Estrutura de Controladores e serviços do servidor

A Figura X retrata a estrutura do projeto seguindo o padrão de desenvolvimento detalhado. Com exceção da rota de *login*, todas as requisições que chegam ao servidor passam pelo *AuthService*, serviço responsável por realizar a autenticação baseada em *JWT* e garantir que apenas requisições feitas por usuários registrados no servidor sejam autorizadas. O controlador *UserController* é responsável por receber todas as requisições dentro do escopo de criação, leitura e

atualização de usuários, enquanto os demais controladores, todos associados a rotas derivadas de *tokenized-asset/*, são responsáveis por receber requisições dos fluxos relacionados a tokenização de ativos, compra e venda de ativos, garantias de empréstimos, registro de pagamentos de aluguel e auditoria dos dados de imóveis na *blockchain*.

Em relação aos serviços implementados, o *SmartContractsService* é, em geral, utilizado apenas por outros serviços do projeto, e concentra todos os métodos necessários para enviar *smart contracts* para a *blockchain* e acessar as suas respectivas funções, por meio da utilização das bibliotecas do *Web3* para *Javascript*.

Os demais serviços, por sua vez, são encarregados de coordenar a leitura e atualização das suas respectivas entidades no banco de dados (identificadas no nome dos serviços). Dentre eles, utilizam os métodos do serviço de *smart contracts*, quando necessário, o *ProposalService*, no qual as propostas de tokenização aceitas são enviadas para *blockchain* para tokenizar o imóvel; o *OwnershipService*, responsável por orquestrar as posses fracionadas de imóveis dos proprietários e persisti-las tanto no banco de dados, quanto na *blockchain*; o *CollateralService*, que contém toda a lógica de criação, tomada ou desalienação de garantias; o *RentPaymentService*, responsável pelo fluxo de registro dos pagamentos de aluguel na *blockchain*; e o *OfferService*, que utiliza o método de transferência de posse do *smart contract* quando uma venda de posse fracionada é realizada na plataforma.

O NestJs, framework utilizado para o desenvolvimento do servidor, permite gerar a documentação das rotas da *API* de forma automática, utilizando o módulo do *Swagger [X]*. Por meio desta ferramenta, todas as rotas da aplicação são detalhadas em uma interface que agrupa de acordo com os controladores de que fazem parte. Para cada rota, é mostrado o tipo de requisição HTTP a ser feito, a mensagem da requisição e a mensagem de resposta esperada, o que viabiliza testes rápidos e práticos e facilita consideravelmente o desenvolvimento do *Front-End*, que deve enviar as mensagens para o *back-end* e tratar os retornos de acordo com as especificações detalhadas nesta documentação.

Users

GET	/users
POST	/users
GET	/users/{id}
PUT	/users/{id}
DELETE	/users/{id}
PUT	/users/{id}/{wallet}

Tokenized Assets

GET	/tokenized-asset/get-all
GET	/tokenized-asset/audit/{contractAddress}

Tokenized Assets / Proposal

POST	/tokenized-asset/proposal/create
GET	/tokenized-asset/proposal/get-pending
PUT	/tokenized-asset/proposal/refuse/{id}
PUT	/tokenized-asset/proposal/accept/{id}
GET	/tokenized-asset/proposal/get-document/{id}

Tokenized Assets / Ownership

GET	/tokenized-asset/ownership/get-by-user/{id}
POST	/tokenized-asset/ownership/transfer

Figura X - Documentação do Swagger para as rotas de Smart Contracts, Users, Tokenized Assets, Proposals e Ownerships

Tokenized Assets / Collateral

- GET** /tokenized-asset/collateral/get-by-user/{id}
- GET** /tokenized-asset/collateral/get-by-bank/{id}
- POST** /tokenized-asset/collateral/create
- PUT** /tokenized-asset/collateral/reject/{id}
- PUT** /tokenized-asset/collateral/validate/{id}
- PUT** /tokenized-asset/collateral/register-loan-payment/{id}
- DELETE** /tokenized-asset/collateral/delete
- PUT** /tokenized-asset/collateral/seize/{id}
- GET** /tokenized-asset/collateral/get-all

Tokenized Assets / Offers

- GET** /tokenized-asset/offer/get-by-ownership/{id}
- GET** /tokenized-asset/offer/get-by-user/{id}
- GET** /tokenized-asset/offer/get-by-buyer/{id}
- GET** /tokenized-asset/offer/get-by-id/{id}
- POST** /tokenized-asset/offer/add-receipt
- GET** /tokenized-asset/offer/get-all
- PUT** /tokenized-asset/offer/accept
- PUT** /tokenized-asset/offer/reject-payment/{id}
- PUT** /tokenized-asset/offer/validate-payment/{id}
- POST** /tokenized-asset/offer/create

Tokenized Assets / Rent Payments

- GET** /tokenized-asset/rent-payment/get-by-ownership/{id}
- POST** /tokenized-asset/rent-payment/create

Authentication

- POST** /auth/login
- POST** /auth/signup
- POST** /auth/logout
- POST** /auth/refresh

Figura X - Documentação do Swagger para as rotas de Collaterals, Offers, Rent Payments e Authentication

5.2.4 Desenvolvimento do Front-End

Durante o levantamento dos diversos casos de uso a serem desenvolvidos no presente trabalho, foram identificadas diversas telas a serem desenvolvidas para a interface do usuário, cada uma dedicada a uma funcionalidade especificada em seu devido caso de uso.

Com base nessas telas mapeadas, o Front-End da plataforma, desenvolvido utilizando a biblioteca React, foi estruturado de modo a separar as telas referentes às funcionalidades em pastas contendo os códigos-fonte associados a tais funcionalidades.

5.2.4.1 Login e cadastro

A tela inicial da plataforma é a de *login* e cadastro de usuário (Figura X), que implementa o fluxo descrito na seção 4.5.1. Nela, o usuário tem a opção de realizar o *login* com *username* e senha para se autenticar, ou de cadastrar uma nova conta, informando nome completo, *username*, CPF e senha, sendo *username* e CPF dados únicos na plataforma.

The screenshot shows a web page with two main sections: 'Fazer Login' (Login) and 'Registrar' (Register). The 'Fazer Login' section contains fields for 'Usuário' (Username) and 'Senha' (Password), both marked with a red asterisk indicating they are required. Below these fields is a blue 'Fazer Login' button. A horizontal line with the word 'OU' (Or) separates the login section from the registration section. The 'Registrar' section contains fields for 'Nome' (Name), 'Usuário', 'CPF', and 'Senha', all marked with red asterisks. Below these fields is a blue 'Registrar' button. The entire page has a light gray background and is titled 'Tokenização de Ativos Imobiliários' at the top left.

Figura X - Tela de login e cadastro de usuário

5.2.4.2 Perfil de usuário

Uma vez autenticado na plataforma, o usuário tem acesso a todas as funcionalidades propostas no projeto. Ao clicar em seu *username* na parte inferior do menu lateral, ele é redirecionado para a tela de dados pessoais, na qual diferentes informações são expostas por meio de abas.

A primeira aba (Figura X) mostra dados gerais da conta e possibilita o cadastro da chave pública do usuário, caso ainda não a tenha cadastrado, de acordo com o fluxo descrito na seção 4.5.2.

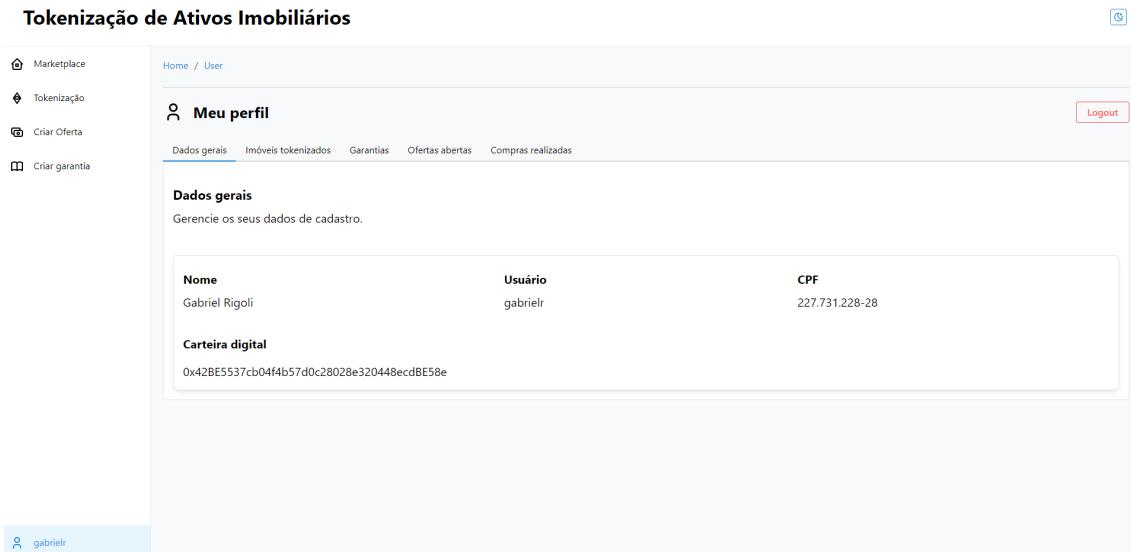


Figura X - Aba de dados gerais da conta

A segunda aba (Figura X) exibe uma lista de *cards* (cartões) com os dados de cada imóvel tokenizado que o usuário detém em sua carteira. Dentre os dados, estão o indicador de posse efetiva do imóvel (onde lê-se “Dono”), a área útil do imóvel, o número de registro em cartório, a porcentagem do imóvel sob posse do usuário, a porcentagem do imóvel disponível para operações (dar em garantia ou vender) e um link do *Etherscan* para validação do *smart contract* do imóvel na *blockchain*.

Destaca-se na utilização destes *cards* o fato de eles terem sido desenvolvidos como um componente genérico, o qual recebe como parâmetro obrigatório a estrutura de dados da propriedade do imóvel e parâmetro opcional os botões que serão renderizados em sua parte inferior. Nesta tela, o botão renderizado tem a função de abrir um *modal* (Figura X) que mostra um registro de todos os pagamentos de aluguéis já recebidos enquanto proprietário do imóvel.

Tokenização de Ativos Imobiliários

Home / User

Meu perfil

Dados gerais Imóveis tokenizados Garantias Ofertas abertas Compras realizadas Logout

Imóveis Tokenizados

Visualize os imóveis tokenizados que você possui em sua carteira digital.

Rua Embuaçu, 499 - Vila Mariana	R. Santa Cruz, 776 - Vila Mariana	Avenida Pompeia, 227 - Pompeia
DONO	DONO	DONO
Área Útil	250 m ²	130 m ²
Número do Registro	23146	33785
Porcentagem de Posse	100 %	100 %
Porcentagem disponível para operações	40 %	100 %
Visualizar no Etherscan	Visualizar no Etherscan	Visualizar no Etherscan
Aluguéis recebidos	Aluguéis recebidos	Aluguéis recebidos

joaosilva

Figura X - Aba de imóveis tokenizados do usuário

Tokenização de Ativos Imobiliários

Home / User

Meu perfil

Dados gerais Imóveis tokenizados Garantias Ofertas abertas Compras realizadas Logout

Imóveis Tokenizados

Visualize os imóveis tokenizados que você possui em sua carteira digital.

Rua Embuaçu, 499 - Vila Mariana	Avenida Pompeia, 227 - Pompeia
DONO	DONO
Área Útil	250 m ²
Número do Registro	23146
Porcentagem de Posse	100 %
Porcentagem disponível para operações	40 %
Visualizar no Etherscan	Visualizar no Etherscan
Aluguéis recebidos	Aluguéis recebidos

Pagamentos de aluguel do imóvel 26734

Data do pagamento	% do total	Valor recebido
2022-12-03	60 %	R\$ 6.000,00
2023-01-03	60 %	R\$ 6.000,00
2023-02-03	60 %	R\$ 6.000,00
2023-03-03	60 %	R\$ 6.000,00

Visualizar no Etherscan Aluguéis recebidos

joaosilva

Figura X - Registro dos aluguéis recebidos enquanto proprietário do imóvel

A terceira aba da página de perfil do usuário (Figura X) mostra todas as garantias de empréstimos que o usuário possui registrado na plataforma por meio de uma lista de cards, que listam o endereço e o número de registro do imóvel, o usuário credor, a porcentagem do imóvel dada em garantia e a data de vencimento do empréstimo. Na parte inferior dos cards, há um botão para o usuário sinalizar que quitou o empréstimo, para que o administrador valide e aceite ou não a desalienação do imóvel, como explicado na seção 4.5.6.

Tokenização de Ativos Imobiliários

Home / User

Meu perfil

Dados gerais Imóveis tokenizados Garantias Ofertas abertas Compras realizadas Logout

Garantias de empréstimos

Visualize as garantias de empréstimos associadas aos seus imóveis tokenizados.

Avenida Pompeia, 227 - Pompeia
DONO

Número do Registro: 26734
Mutuante: Isantos
Porcentagem em Garantia: 34.1 %
Data de expiração: 03/12/2022

Quitar empréstimo

joaosilva

Figura X - Aba de garantias do usuário

A quarta aba da página de perfil do usuário (Figura X) mostra todas as ofertas de imóveis que o usuário já realizou no *Marketplace* por meio de *cards*, cada qual listando os dados da oferta, do imóvel e, caso exista, do comprador do imóvel.

Tokenização de Ativos Imobiliários

Home / User

Meu perfil

Dados gerais Imóveis tokenizados Garantias Ofertas abertas Compras realizadas Logout

Ofertas abertas no Marketplace

Gerencie as suas ofertas criadas no Marketplace.

Rua Embuaçu, 499 - Vila Mariana
EM ABERTO

Número de Registro: 23146
Valor da Oferta: R\$ 100.000,00
Porcentagem do Imóvel em Oferta: 20 %
Usuário Comprador: -
Nome do Comprador: -

Avenida Pompeia, 227 - Pompeia
ESPERANDO PAGAMENTO

Número de Registro: 26734
Valor da Oferta: R\$ 220.500,00
Porcentagem do Imóvel em Oferta: 40 %
Usuário Comprador: gabrielr
Nome do Comprador: Gabriel Rigoli

joaosilva

Figura X - Aba de ofertas do usuário abertas no Marketplace

Por fim, a quinta aba da tela do perfil do usuário (Figura X) lista por meio de *cards* todas as compras realizadas no *Marketplace* pelo usuário, mostrando o status de cada operação, os dados da oferta, do imóvel e do vendedor. Além disso, cada

card também possui um botão para o usuário adicionar ou editar o comprovante da compra, funcionalidade que compõe o fluxo descrito na seção 4.5.4.

Figura X - Aba de compras realizadas no Marketplace pelo usuário

5.2.4.3 Tokenização de imóvel

Ao selecionar a opção “Tokenização” no menu lateral, o usuário é redirecionado para uma tela (Figura X) com um formulário no qual ele pode informar os dados do imóvel que deseja tokenizar na plataforma. São pedidos o endereço, a área útil, o número de registro em cartório e a documentação do imóvel, dados estes que são enviados para análise do administrador para a realização da tokenização, de acordo com o fluxo descrito na seção 4.5.3. O botão de envio dos dados só é habilitado caso o usuário possua chave pública da carteira digital cadastrada em sua conta.

Figura X - Formulário de envio dos dados do imóvel para tokenização

5.2.4.4 Criar oferta no Marketplace

Tendo posse de ativos imobiliários tokenizados na plataforma, o usuário pode criar uma oferta no Marketplace selecionando a opção “Criar oferta” no menu lateral. Ao ser redirecionado para a tela (Figura X), serão listados em cards todos os imóveis do usuário, cada um possuindo um botão que abre um modal (Figura X) solicitando as informações necessárias para criação da oferta na plataforma. Uma vez que os dados são submetidos, a oferta já se torna visível no Marketplace para outros usuários, conforme explicado na seção 4.5.4.

Figura X - Tela de listagem dos imóveis disponíveis para criação de oferta

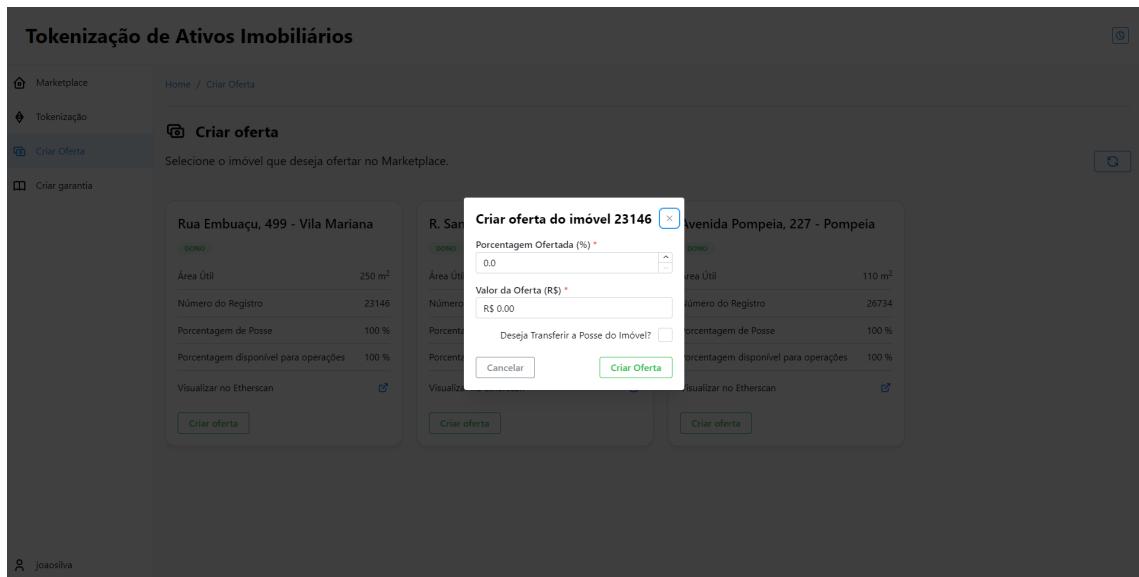


Figura X - Formulário de criação de oferta no Marketplace

5.2.4.5 Marketplace

O Marketplace é a primeira tela com a qual o usuário tem contato ao se autenticar na plataforma, podendo também acessá-la por meio do menu lateral selecionando a opção “Marketplace” (Figura X). Nela, são listadas em cards todas as ofertas de imóveis disponíveis na plataforma, cada um contendo os dados do imóvel, o vendedor, o preço e a porcentagem de posse à venda.

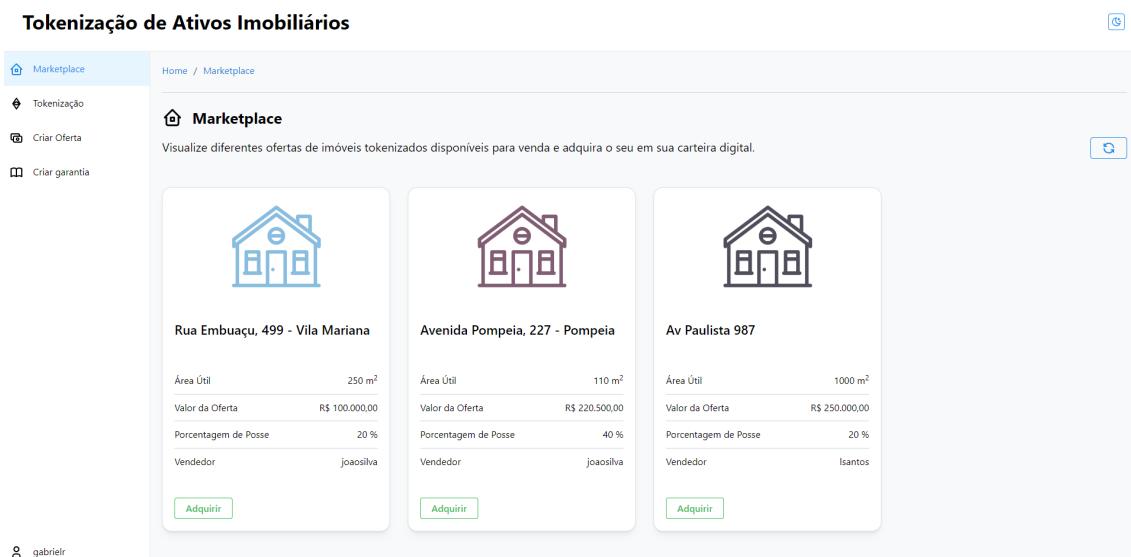


Figura X - Marketplace de imóveis da plataforma

Clicando no botão de “Adquirir”, o usuário é redirecionado para a tela de detalhes da oferta (Figura X), na qual são mostradas mais informações da imóvel, incluindo o endereço do *smart contract* na *blockchain* e o link do *Etherscan* para auditoria do contrato. Ao clicar em “Aceitar oferta”, o usuário é notificado que deverá realizar o pagamento e, assim que houver a confirmação pelo administrador, receberá a porcentagem do imóvel tokenizado em sua carteira digital, conforme anteriormente descrito na seção 4.5.4.

The screenshot shows a web application for real estate tokenization. The left sidebar has a 'Marketplace' tab selected, with options for 'Tokenização', 'Criar Oferta', and 'Criar garantia'. The main content area shows a property listing for 'Oferta #c67a8536-2475-48e9-979c-e7c68b6ad864'. It features a stylized house icon and the address 'Avenida Pompeia, 227 - Pompeia'. Below the address are sections for 'Dados da Oferta' (Offer Data) and 'Dados do Imóvel' (Property Data). 'Dados da Oferta' includes 'Porcentagem à Venda' (40 %) and 'Valor da Oferta' (R\$ 220.500,00). 'Dados do Imóvel' includes 'Área Útil' (110 m²) and 'Número do Registro' (26734). An 'Enderereço do Imóvel Tokenizado' field contains the value '0xA40d71cCfFA65696ed7024d843D86d1570F1CB7D' with a copy icon. Below these are sections for 'Dados do Vendedor' (Seller Data), which lists 'Nome' (João Silva) and 'Usuário' (joaosilva). A green button at the bottom left says 'Aceitar Oferta' (Accept Offer).

Figura X - Tela de detalhes da oferta do imóvel

5.2.4.6 Criar garantia de empréstimo

De forma semelhante à tela de criação de oferta, a tela de criação de garantia (Figura X), acessível por meio da opção “Criar garantia” no menu lateral, lista todos os imóveis tokenizados do usuário disponíveis para a operação, cada qual em seu respectivo card que possui o botão de cadastrar garantia, que, ao ser clicado, abre um modal (Figura X) solicitando a data de vencimento do empréstimo, o usuário credor e o percentual do imóvel a ser dado em garantia. Após a submissão destes dados, é necessária a ação do administrador para validar e aceitar o registro da garantia, em conformidade com o fluxo descrito na seção 4.5.5.

Tokenização de Ativos Imobiliários

Home / Empréstimo

Criar garantia de empréstimo

Selecione o imóvel que deseja associar a uma garantia de empréstimo.

Rua Embuaçu, 499 - Vila Mariana	R. Santa Cruz, 776 - Vila Mariana	Avenida Pompeia, 227 - Pompeia
DONO	DONO	DONO
Área Útil 250 m ²	Área Útil 130 m ²	Área Útil 110 m ²
Número do Registro 23146	Número do Registro 33785	Número do Registro 26734
Porcentagem de Posse 100 %	Porcentagem de Posse 100 %	Porcentagem de Posse 100 %
Porcentagem disponível para operações 100 %	Porcentagem disponível para operações 100 %	Porcentagem disponível para operações 100 %
Visualizar no Etherscan	Visualizar no Etherscan	Visualizar no Etherscan
Cadastrar garantia	Cadastrar garantia	Cadastrar garantia

joaosilva

Figura X - Tela de listagem dos imóveis disponíveis para criação de garantia

Cadastrar imóvel 33785 como garantia de empréstimo

Endereço do imóvel tokenizado * 0x2B032581b76f0b8B1C33e8E8BB0d42cAD971F0

Data de expiração do empréstimo * Dezembro 3, 2022

Nome de usuário do banco * Escolha um banco

Porcentagem utilizada como garantia (%) * 0.0

Cadastrar Garantia

Figura X - Formulário de criação de garantia de empréstimo

5.2.4.7 Portal do administrador

Caso o usuário tenha papel de administrador da plataforma, será liberado a ele o acesso ao portal Admin, cujas funcionalidades presentes têm como foco validar fluxos de cadastro e pagamento da plataforma, como já citado brevemente nas seções anteriores. Este portal é organizado em formato de abas, assim como a página de perfil do usuário, e possui cinco delas:

A primeira aba (Figura X) lista por meio de *cards* todas as propostas de tokenização de imóveis enviadas por usuários da plataforma na tela de Tokenização, tendo o administrador as opções de aceitar ou rejeitar a proposta após analisar os dados e documentos do imóvel. No momento em que a proposta é aceita, o *smart contract* contendo os dados do imóvel são enviados para a *blockchain* e o proprietário já é capaz de visualizar seu imóvel na sua página de perfil, concluindo o fluxo de tokenização de imóvel descrito na seção 4.5.3.

Proposta de Tokenização Ativa	Nome do Usuário	CPF do Usuário	Área útil	Número de Registro	Visualizar documentos	Ação
Rua Embuaçu, 499 - Vila Mariana	joaosilva	337.731.228-28	250 m ²	23146	Visualizar documentos	Rejeitar Aceitar
R. Santa Cruz, 776 - Vila Mariana	joaosilva	337.731.228-28	130 m ²	33785	Visualizar documentos	Rejeitar Aceitar
Avenida Pompeia, 227 - Pompeia	joaosilva	337.731.228-28	110 m ²	26734	Visualizar documentos	Rejeitar Aceitar

Figura X - Aba listando todas as propostas de tokenização da plataforma

A segunda aba presente no portal Admin (Figura X), por sua vez, é responsável por listar todas as ofertas que foram aceitas no Marketplace e que estão com o status pendente validação de pagamento. Cada *card* contém as informações da oferta em andamento, podendo ou não exibir o comprovante de pagamento, caso o comprador tenha registrado na tela de Compras realizadas. Tendo a confirmação do pagamento, o administrador pode clicar no botão Aceitar e, com isso, será feita a transferência da posse do imóvel entre os usuários envolvidos na operação, concluindo o fluxo de compra e venda de imóvel descrito na seção 4.5.4.

The screenshot shows the 'Portal Admin' section of the platform. On the left sidebar, under 'userAdmin', there's a 'Portal Admin' button. The main content area is titled 'Ofertas Esperando Confirmação de Pagamento'. It lists two offers:

- Av Paulista 987** (Offer ID 123):

Número de registro	123
Valor da oferta	R\$ 100.000,00
Porcentagem do imóvel ofertada	10 %
CPF do comprador	445.961.338-75
Nome do comprador	Lucas Santos
Usuário do comprador	lsantos
CPF do vendedor	351.086.060-87
Nome do vendedor	Cleiton Jorge
Usuário do vendedor	cleiton123
Comprovante de pagamento	-

 Buttons: Rejeitar (Red), Confirmar (Green)
- Avenida Pompeia, 227 - Pompeia** (Offer ID 26734):

Número de registro	26734
Valor da oferta	R\$ 220.500,00
Porcentagem do imóvel ofertada	40 %
CPF do comprador	227.731.228-28
Nome do comprador	Gabriel Rigoli
Usuário do comprador	gabrielr
CPF do vendedor	337.731.228-28
Nome do vendedor	João Silva
Usuário do vendedor	joaosilva
Comprovante de pagamento	-

 Buttons: Rejeitar (Red), Confirmar (Green)

Figura X - Aba listando todos os pagamentos de ofertas pendentes de confirmação

A terceira aba presente no portal Admin (Figura X) é responsável por listar todas os cadastros de garantia aguardando validação para se tornarem vigentes na plataforma. O administrador, de posse dos dados presentes em cada *card*, pode rejeitar ou aceitar o registro da garantia, tornando a garantia ativa na plataforma e registrando-a no *smart contract* do imóvel na *blockchain*, concluindo assim o fluxo de registro de garantia descrito na seção 4.5.5.

The screenshot shows the 'Portal Admin' section of the platform. On the left sidebar, under 'userAdmin', there's a 'Portal Admin' button. The main content area is titled 'Validação de novas garantias de empréstimo'. It lists two new guarantees:

- Rua Embuá, 499 - Vila Mariana** (Guarantor ID joaosilva):

Usuário Mutuário	joaosilva
Usuário Mutuante	cleiton123
Porcentagem em Garantia	40 %
Data de Expiração do Empréstimo	03/12/2022
Número de Registro do Imóvel	23146

 Buttons: Rejeitar (Red), Confirmar (Green)
- Avenida Pompeia, 227 - Pompeia** (Guarantor ID joaosilva):

Usuário Mutuário	joaosilva
Usuário Mutuante	lsantos
Porcentagem em Garantia	34,1 %
Data de Expiração do Empréstimo	03/12/2022
Número de Registro do Imóvel	26734

 Buttons: Rejeitar (Red), Confirmar (Green)

Figura X - Aba listando todos os cadastros de garantia de empréstimos pendentes na plataforma

A quarta aba do portal administrador (Figura X) possui uma lista de todas as garantias de empréstimos vigentes na plataforma. Caso o devedor tenha registrado a quitação do empréstimo, o botão “Confirmar pagamento” ficará habilitado e será responsável por desalienar a porcentagem do imóvel e torná-la novamente utilizável pelo usuário na plataforma. Caso a data de vencimento do empréstimo tenha sido atingida e o devedor não o tenha quitado, o botão “Tomada de posse” ficará habilitado e será responsável por transferir a posse da porcentagem do imóvel da carteira do devedor para a carteira do credor. As funcionalidades presentes nessa aba concluem os fluxo de quitação e tomada de garantia descrito na seção 4.5.6

Figura X - Aba listando todas as garantias de empréstimo registradas na plataforma

A última aba presente no portal Admin (Figura X) renderiza uma tabela contendo os dados de todos os imóveis tokenizados da plataforma. Para cada um deles, são disponibilizadas as ações de “Registrar aluguel” e “Auditar”.

ID	Endereço	Nº registro	Área útil	Endereço na blockchain	Ações
f0973eca-85cf-435e-b5c4-18b478681e8a	Av Jabaquara 12345	2167	2183791 m ²	0x983dC5d11b1B2j3bB26275286f490151f8625b4a	Auditar Registrar aluguel
f00723f8-493c-41d2-b70c-e6ece5291c11	Av Paulista 987	123	1000 m ²	0x1ACFF14b5b7e62427177a19ca106981e1c40D7d	Auditar Registrar aluguel
Sec4d177-0101-46ef-b035-0eaac81a08	Rua Mauro 32974	34	1243 m ²	0x6578B7931bD7949f01e45219471D6ED8259201dB	Auditar Registrar aluguel
98c7fc9f-3c86-4698-a6b7-2572ad5e126c	Rua Embuáçu, 499 - Vila Mariana	23146	250 m ²	0x01592a70ACe9a16D99155b15140bFa2AD0E7c8	Auditar Registrar aluguel
80149341-d046-43f1-820d-7531cf7bd7a7	R. Santa Cruz, 776 - Vila Mariana	33785	130 m ²	0x2BB032581b76E0b8B1C333e8EBB0d42cAD971f0	Auditar Registrar aluguel
3ba63a26-4b45-4d94-a2cc-71d8a0bca6a	Avenida Pompeia, 227 - Pompeia	26734	110 m ²	0xA40d71cCfA65696ed7024d843D86d1570F1CB7D	Auditar Registrar aluguel
c91dd675-348f-4f99-8a94-d8a25c2052e	Rua Guapáçu, 117 - Saúde	45881	220 m ²	0xf7921E48710d9A60FDc8a6a84136D1682DFE29e	Auditar Registrar aluguel

Figura X - Aba listando todas os imóveis tokenizados da plataforma

A primeira ação implementa o fluxo descrito na seção 4.5.7, por meio de um modal (Figura X) no qual o administrador cadastra o valor recebido no mês do aluguel daquele imóvel. Este valor é dividido tanto no servidor, quanto no *smart contract*, entre os proprietários do imóvel de forma proporcional à porcentagem da posse que cada um detém.

Figura X - Formulário de registro de pagamento de aluguel

A segunda ação, por sua vez, é responsável por abrir um modal (Figura X) contendo informações de cada proprietário do imóvel, incluindo dados gerais da

conta, garantias registradas pelo usuário com o imóvel em questão e os pagamentos de aluguéis recebidos. Estes dados são buscados diretamente do *smart contract*, como uma forma de auditar os dados imutáveis do imóvel na *blockchain*. Adicionalmente, este modal também apresenta o endereço do *smart contract* do imóvel e um link para o *Etherscan* como mais uma forma de auditar os dados. Esta ação implementa o último dos casos de uso descritos anteriormente, encontrado na seção 4.5.8.

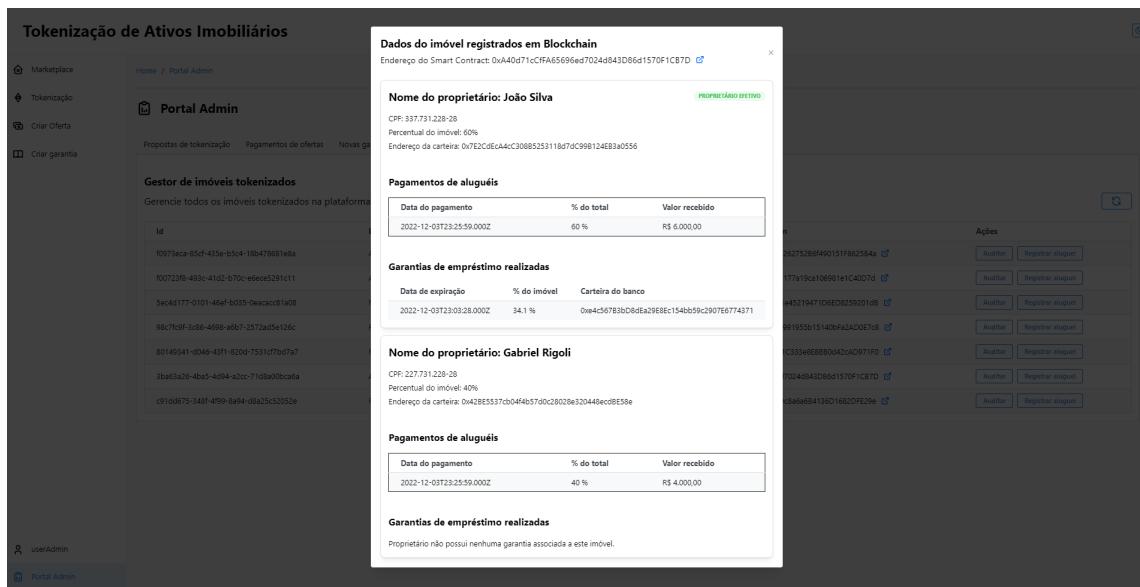


Figura X - Modal de auditoria dos dados do imóvel tokenizado

5.3 TESTES E AVALIAÇÃO

Os testes a serem realizados correspondem à validação das funcionalidades mapeadas nos casos de uso, e por meio dos resultados obtidos será possível avaliar o sucesso da implementação de cada uma das funcionalidades propostas no início do projeto.

5.3.1 Tokenização de imóvel

Capítulo a ser desenvolvido.

Este capítulo irá descrever os diferentes testes a serem realizados para verificar o funcionamento do fluxo de tokenização de imóveis através da plataforma, detalhando também os resultados obtidos.

5.3.2 Oferta e compra de percentual de imóvel

Capítulo a ser desenvolvido.

Este capítulo irá descrever os diferentes testes a serem realizados para verificar o funcionamento do processo de compra e venda de ativos imobiliários tokenizados através da plataforma, detalhando também os resultados obtidos.

5.3.3 Garantias de empréstimos

Capítulo a ser desenvolvido.

Este capítulo irá descrever os diferentes testes a serem realizados para verificar o funcionamento dos casos de uso envolvendo a utilização de porcentagem de ativos imobiliários tokenizados como garantia de empréstimos através da plataforma, detalhando também os resultados obtidos.

5.3.4 Distribuição de retornos financeiros provenientes de aluguel

Capítulo a ser desenvolvido.

Este capítulo irá descrever os diferentes testes a serem realizados para verificar o funcionamento da funcionalidade de distribuição de retornos financeiros obtidos através de ativos imobiliários tokenizados utilizando a plataforma, detalhando também os resultados obtidos.

6 CONSIDERAÇÕES FINAIS

Capítulo a ser desenvolvido.

Este capítulo irá descrever as diferentes observações geradas durante a implementação, bem como analisar os resultados obtidos em comparação com as funcionalidades mapeadas inicialmente e descrever possíveis pontos de melhoria que poderiam ser implementados numa eventual continuação do projeto e suas justificativas.

REFERÊNCIAS

[1] FARACHE, A. ATIVOS REAIS X ATIVOS FINANCEIROS: GUIA COMPLETO. Disponível em: <<https://blog.hurst.capital/blog/ativos-reais-x-ativos-financeiros>>. Acesso em: 10 jul. 2022.

[2] PINTO, L. Direitos reais de garantias. Disponível em: <<https://jus.com.br/artigos/51992/direito-reais-de-garantias>>. Acesso em: 10 jul. 2022.

[3] MOURA, J. Uso de imóvel como garantia de mais de um empréstimo requer mudança na lei. Disponível em: <<https://www1.folha.uol.com.br/mercado/2020/02/uso-de-imovel-como-garantia-de-mais-de-um-emprestimo-requer-mudanca-na-lei.shtml>>. Acesso em: 10 jul. 2022.

[4] AMARAL, L. Câmara aprova permitir que imóvel seja usado como garantia em mais de um empréstimo. Disponível em: <<https://www.cnnbrasil.com.br/business/camara-aprova-permitir-que-imovel-seja-usado-como-garantia-em-mais-de-um-emprestimo/>>. Acesso em: 10 jul. 2022.

[5] AMARO, M. PL pode permitir que um único imóvel seja usado como garantia em vários empréstimos. Disponível em: <<https://www.infomoney.com.br/minhas-financas/pl-pode-permitir-que-um-unico-imovel-seja-usado-como-garantia-em-varios-emprestimos/>>. Acesso em: 10 jul. 2022.

[6] BERTÃO, N. Tokenização das coisas chega ao mercado imobiliário e de florestas. Disponível em: <<https://valorinveste.globo.com/objetivo/hora-de-investir/noticia/2021/10/20/tokenizacao-das-coisas-chega-ao-mercado-imobiliario-e-de-florestas.ghtml>>. Acesso em: 10 jul. 2022.

[7] MILAGRE, J. Blockchain e a tokenização de imóveis no Brasil: Aspectos jurídicos e desafios regulatórios. Disponível em:

<<https://direitodigital.adv.br/artigos/blockchain-e-a-tokenizacao-de-imoveis-no-brasil-aspectos-juridicos-e-desafios-regulatorios/?print=pdf>>. Acesso em: 18 out. 2022.

[8] CARTER, R. The Ultimate List of Blockchain Statistics (2022). Disponível em: <<https://findstack.com/blockchain-statistics/>>. Acesso em: 10 jul. 2022.

[9] NETSPACES. Página inicial. Disponível em: <<https://netspaces.org>>. Acesso em: 18 out. 2022.

[10] NETSPACES. Regulamento da PROPRIEDADE DIGITAL v0.5. Disponível em: <<https://api-landing.netspaces.org/static/netspaces-%20Regulamento%20da%20propriedade%20Digital%20-%20v05.pdf>>. Acesso em: 18 out. 2022.

[11] VANHALL. Tokenização Imobiliária. Disponível em: <<https://vanhall.io/tokenizacao-imobiliaria>>. Acesso em: 18 out. 2022.

[12] SYNC. Página inicial. Disponível em: <<https://synctoken.com.br/>>. Acesso em: 18 out. 2022.

[13] REALT. Página inicial. Disponível em: <<https://realt.co/>>. Acesso em: 18 out. 2022.

[14] TOKENIZED. Página inicial. Disponível em: <<https://tokenized.io/>>. Acesso em: 18 out. 2022.

[15] DIGISHARES. Página inicial. Disponível em: <<https://digishares.io/real-estate-tokenization>>. Acesso em: 18 out. 2022.

[16] REDSWAN CRE. Página inicial. Disponível em: <<https://redswan.io/>>. Acesso em: 18 out. 2022.

[17] TOKENY. Página inicial. Disponível em: <<https://tokeny.com/tokenized-real-estate>>. Acesso em: 18 out. 2022.

[18] BLOCKCHAIN. In: WIKIPÉDIA: a enciclopédia livre. Disponível em: <<https://en.wikipedia.org/wiki/Blockchain>>. Acesso em: 10 jul. 2022.

[19] BASTIANI, A. O que é mecanismo de consenso? Disponível em: <<https://www.criptofacil.com/o-que-e-mecanismo-de-consenso/>>. Acesso em: 10 jul. 2022.

[20] SEN, S. An overview of how smart contracts work on Ethereum. Disponível em: <<https://www.quicknode.com/guides/solidity/an-overview-of-how-smart-contracts-work-on-ethereum>>. Acesso em: 10 jul. 2022.

[21] REAL-WORLD Use Cases for Smart Contracts and dApps. Disponível em: <<https://www.gemini.com/cryptopedia/smart-contract-examples-smart-contract-use-cases>>. Acesso em: 10 jul. 2022.

[22] NON-FUNGIBLE Token (NFT) vs. Fungible Token (FT). Disponível em: <<https://criptomais.com/en/token-non-fungible-nft-vs-token-fungible-ft/>>. Acesso em: 10 jul. 2022.

[23] KHILLAR, S. Difference Between Fungible and Non Fungible Tokens. Disponível em: <<http://www.differencebetween.net/technology/difference-between-fungible-and-non-fungible-tokens/>>. Acesso em: 10 jul. 2022.

[24] FUNGIBLE vs nonfungible tokens: What is the difference?. Disponível em: <<https://cointelegraph.com/nonfungible-tokens-for-beginners/fungible-vs-nongurable-tokens-what-is-the-difference>>. Acesso em: 10 jul. 2022.

[25] REPORTLINKER. The Global Non-fungible Token Market size is expected to reach \$97.6 billion by 2028, rising at a market growth of 31.6% CAGR during the forecast period. Disponível em:

<<https://www.globenewswire.com/news-release/2022/06/29/2470994/0/en/The-Global-Non-fungible-Token-Market-size-is-expected-to-reach-97-6-billion-by-2028-rising-at->>

[a-market-growth-of-31-6-CAGR-during-the-forecast-period.html](#). Acesso em: 10 jul. 2022.

[26] SETOR imobiliário bate recordes de lançamentos e vendas em 2021. Disponível em: <<https://oespecialista.com.br/lancamentos-vendas-imoveis-recorde-2021/>>. Acesso em: 10 jul. 2022.

[27] METAMASK. A crypto wallet & gateway to blockchain apps. Disponível em: <<https://metamask.io/>>. Acesso em: 16 out, 2022.

[28] DOCUMENTOS para registrar imóvel: o que é necessário? Disponível em: <<https://blog.vittaresidencial.com.br/documentos-para-registrar-imovel-o-que-e-necessario/>>. Acesso em: 10 jul. 2022.

[29] NODE JS. About Node.Js. Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 10 jul. 2022.

[30] NEST JS. Página inicial. Disponível em: <<https://nestjs.com/>>. Acesso em: 10 jul. 2022.

[31] REACT. Página inicial. Disponível em: <<https://reactjs.org/>>. Acesso em: 10 jul. 2022.

[32] POSTGRESQL. About PostgreSQL. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 10 jul. 2022.

[33] SOLIDITY. Página inicial. Disponível em: <<https://docs.soliditylang.org/en/v0.8.15/>>. Acesso em: 10 jul. 2022.

[34] REMIX. Página inicial. Disponível em: <<https://remix-project.org/>>. Acesso em: 10 jul. 2022

[35] WEB3.JS. Página inicial. Disponível em: <<https://web3js.readthedocs.io/en/v1.7.4/>>. Acesso em: 10 jul. 2022.

[36] TYPEORM. Página inicial. Disponível em: <<https://typeorm.io/>>. Acesso em: 04 dez. 2022