

**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 Antônio Simplício 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, um mercado com liquidez um tanto reduzida, 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 no blockchain, e podem, por exemplo, representar diferentes porcentagens do direito da posse do ativo real. Dessa forma, cria-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

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, um mercado com liquidez um tanto reduzida, 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 gerados podem, por exemplo, representar diferentes porcentagens do direito da posse do ativo real. Dessa forma, cria-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.

Keywords: Ativos Imobiliários. Tokenização. Blockchain. Plataforma Digital. Smart Contract.

SUMÁRIO

1 INTRODUÇÃO	8
1.1 MOTIVAÇÃO	8
1.2 OBJETIVO	9
1.3 JUSTIFICATIVA	9
1.4 SOLUÇÕES DE MERCADO	10
1.5 ORGANIZAÇÃO DO TRABALHO	11
2 ASPECTOS CONCEITUAIS	11
2.1 BLOCKCHAIN	11
2.2 SMART CONTRACTS	12
2.3 TOKEN	12
2.3.1 Token Fungível (FT)	13
2.3.2 Token Não Fungível (NFT)	13
3 MÉTODO DO TRABALHO	14
4 ESPECIFICAÇÃO DO SISTEMA	15
4.1 PAPÉIS, ENTIDADES e NOMENCLATURAS DO SISTEMA	15
4.2 REQUISITOS FUNCIONAIS	15
4.3 REQUISITOS NÃO-FUNCIONAIS	16
4.3.1 Escalabilidade	16
4.3.2 Segurança	16
4.4 ARQUITETURA PROPOSTA	18
4.4.1 Interface da plataforma (Front-End)	18
4.4.2 Servidor da Plataforma (Back-End)	19
4.4.3 Banco de dados relacional (Back-end)	19
4.4.4 Blockchain e Smart Contracts	22
4.5 CASOS DE USO PRINCIPAIS	22
4.5.1 Validação de Documentos e Tokenização de um Imóvel	22
4.5.2 Compra e Venda de Porcentagem de Posse de Ativo Imobiliário	24
4.5.3 Registro de garantia de empréstimo	26
4.5.4 Transferência de Posse de Ativo Imobiliário	27
4.5.5 Quitação de Empréstimo	28
4.5.6 Tomada de Garantia de Empréstimo	29
4.5.7 Distribuição de Lucro Proveniente de Aluguel	30
4.5.8 Registro de Usuário na Plataforma	32
4.5.9 Login de Usuário na Plataforma	33
4.5.10 Cadastro de Carteira do Usuário na Plataforma	34
5 DESENVOLVIMENTO DO TRABALHO	35
5.1 TECNOLOGIAS UTILIZADAS	36

5.1.1 Node.js	37
5.1.2 React	37
5.1.3 PostgreSQL	37
5.1.4 Solidity	37
5.1.5 Remix IDE	38
5.1.6 Web3.js	38
5.1.7 Ethereum Goerli Testnet	38
5.2 PROJETO E IMPLEMENTAÇÃO	38
5.2.1 Desenvolvimento dos Smart Contracts	39
5.2.2 Desenvolvimento do Front-End	39
5.2.3 Desenvolvimento do Back-End	39
5.3 TESTES E AVALIAÇÃO	40
5.3.1 Tokenização de Imóvel	40
5.3.2 Transferência de Posse	40
5.3.3 Oferta e Compra de Percentual de Imóvel	40
5.3.4 Garantias de Empréstimos	41
5.3.5 Distribuição de Pagamento de Lucros Provenientes de Aluguel	41
6 CONSIDERAÇÕES FINAIS	41
REFERÊNCIAS	42

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 troca de dono dos ativos por envolver uma troca física no processo, que muitas vezes não é simples de se realizar.

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, a utilização de ativos reais como garantia de empréstimos atualmente 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 cidadão, e que poderia estar sendo utilizado, por exemplo, para garantir outro empréstimo.

1.1 MOTIVAÇÃO

A motivação do presente trabalho vem da observação da utilização de ativos imobiliários para empréstimos e suas limitações atuais com relação à utilização destes ativos em múltiplos empréstimos; a comercialização de frações destes ativos e a distribuição de retornos financeiros envolvendo-os; e surge uma proposta de solução destes cenários através do presente trabalho. 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 do presente trabalho também é bastante relevante no cenário atual devido à sua implementação utilizando a tecnologia do *blockchain*, que permite tornar os processos de tokenização, empréstimo e distribuição de lucros de aluguéis mais transparente, auditável, menos burocrático, seguro e descentralizado. 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], *Tokenized* [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 1.5 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 PAPÉIS, ENTIDADES e NOMENCLATURAS 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; e
- *Administrador*, cuja função é controlar fluxos de validação e controle da plataforma.
- *Bank*, mutuário 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 a disponibilização de ativos *tokenizados* para transações;
- Listar todas as ofertas de ativos da plataforma;

- Transferir porcentagens de ativos entre usuários;
- Associar porcentagem de ativos a garantias de empréstimos;
- Registrar a distribuição de retornos financeiros provenientes dos ativos imobiliários para os donos de porcentagens dos respectivos ativos;
- 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

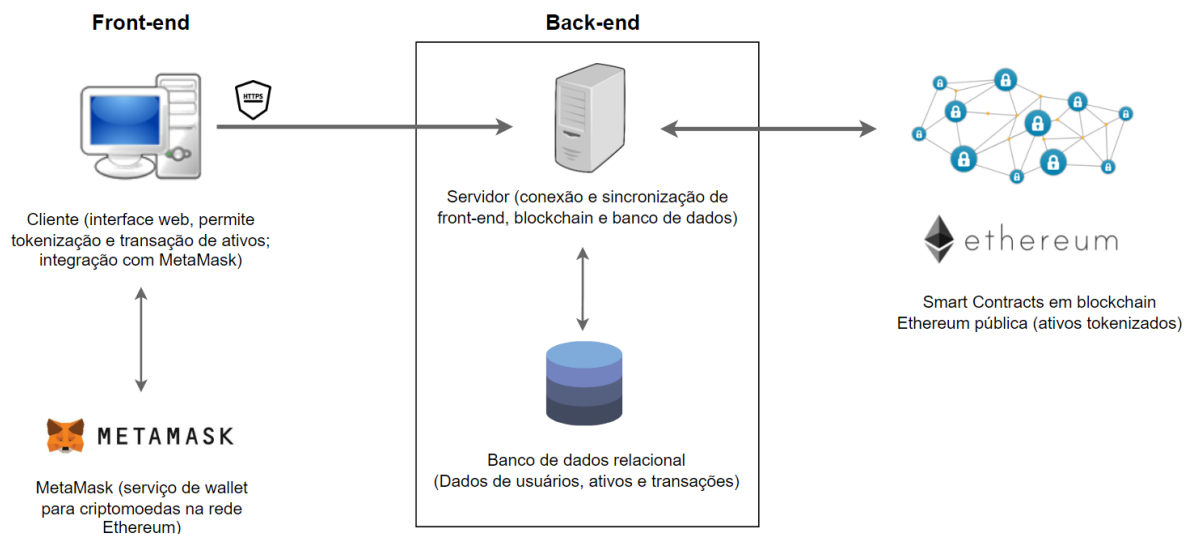


Figura X - Arquitetura Proposta

A arquitetura presente na Figura X dispõe de *Frontend*, *Backend*, *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 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 dono do imóvel irá interagir com uma tela no *front-end* e irá inserir os documentos e dados relativos à 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*, na qual serão associados a um token e as informações necessárias para a plataforma serão registradas em seu banco de dados.

Seguem abaixo 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.

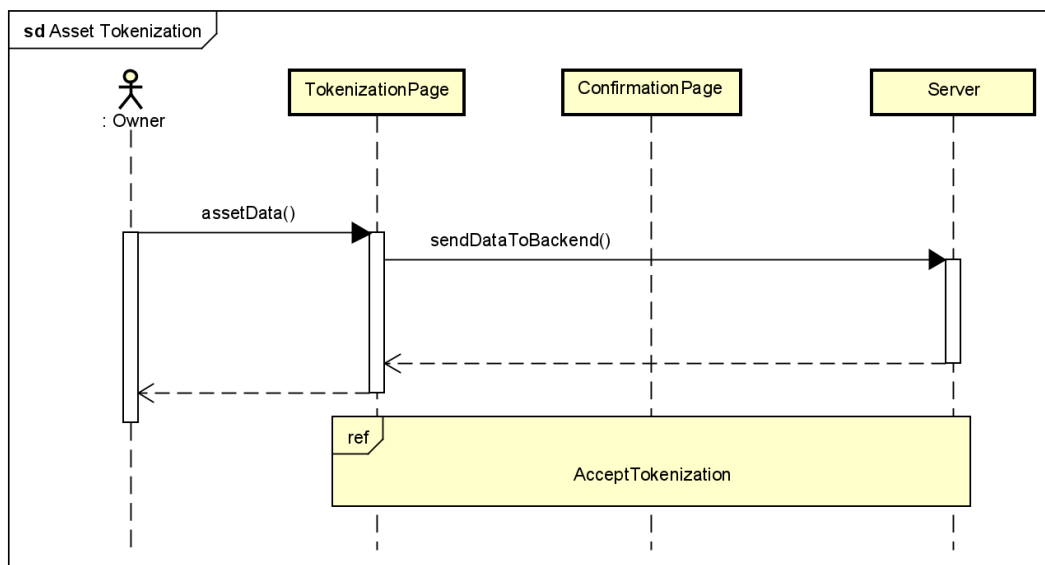


Figura X - Diagrama de sequência para a tokenização de um imóvel

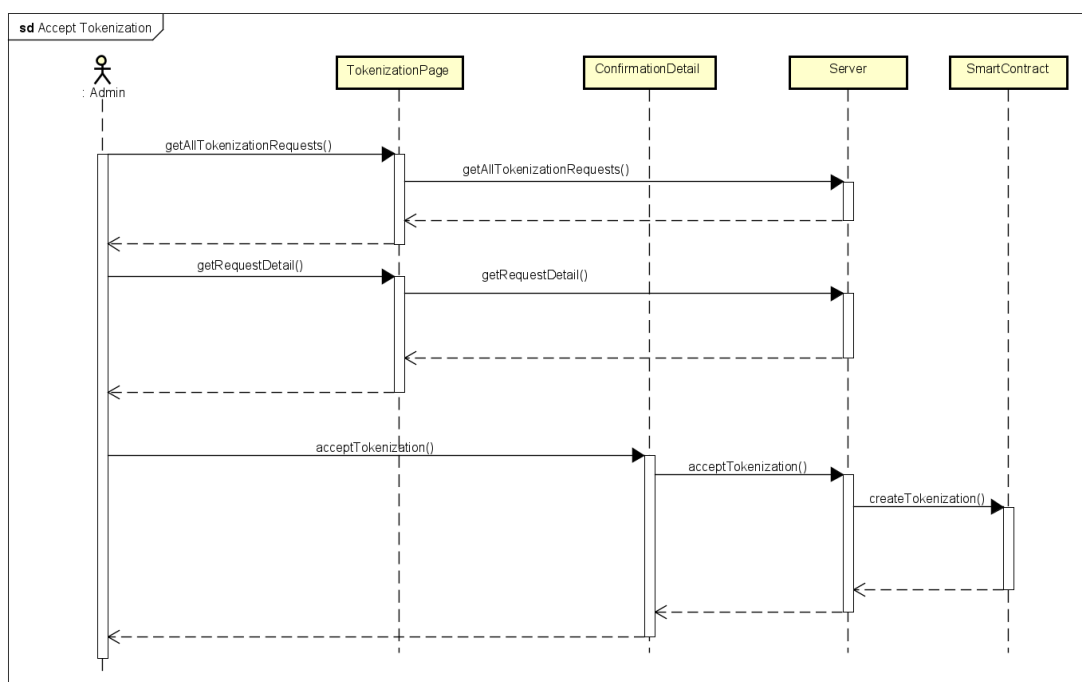


Figura X - Diagrama de sequência para a validação dos documentos de um imóvel para o processo de tokenização

4.5.2 Compra e Venda de Porcentagem de Posse de Ativo Imobiliário

Para viabilizar que diversas pessoas possuam porcentagens de um ativo tokenizado, é necessário que seja possível a compra de uma porcentagem de um ativo.

Para isso, o comprador deverá acessar uma tela no *front-end* com uma lista de todas as ofertas disponíveis (que deverão ser colocadas à venda pelos donos dos imóveis tokenizados) 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 também poderá realizar a proposta de compra de porcentagem do ativo.

Após o registro da proposta de compra do ativo, o comprador deverá então realizar o pagamento para a plataforma que irá então repassar o valor pago ao dono do imóvel através de um usuário administrador. Então, após o processo de repasse do valor do ativo do comprador para o dono do ativo, um administrador da plataforma deverá acessar esta para então registrar a confirmação do pagamento e assim a plataforma irá registrar a transferência de posse na blockchain, bem como atualizar as informações de posse em seu banco de dados.

Seguem abaixo 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.

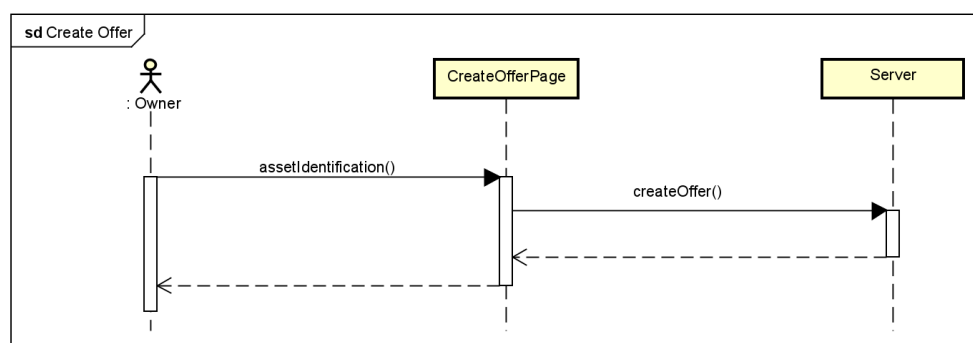


Figura X - Diagrama de sequência para a criação de oferta de venda de porcentagem de posse de ativo *tokenizado*

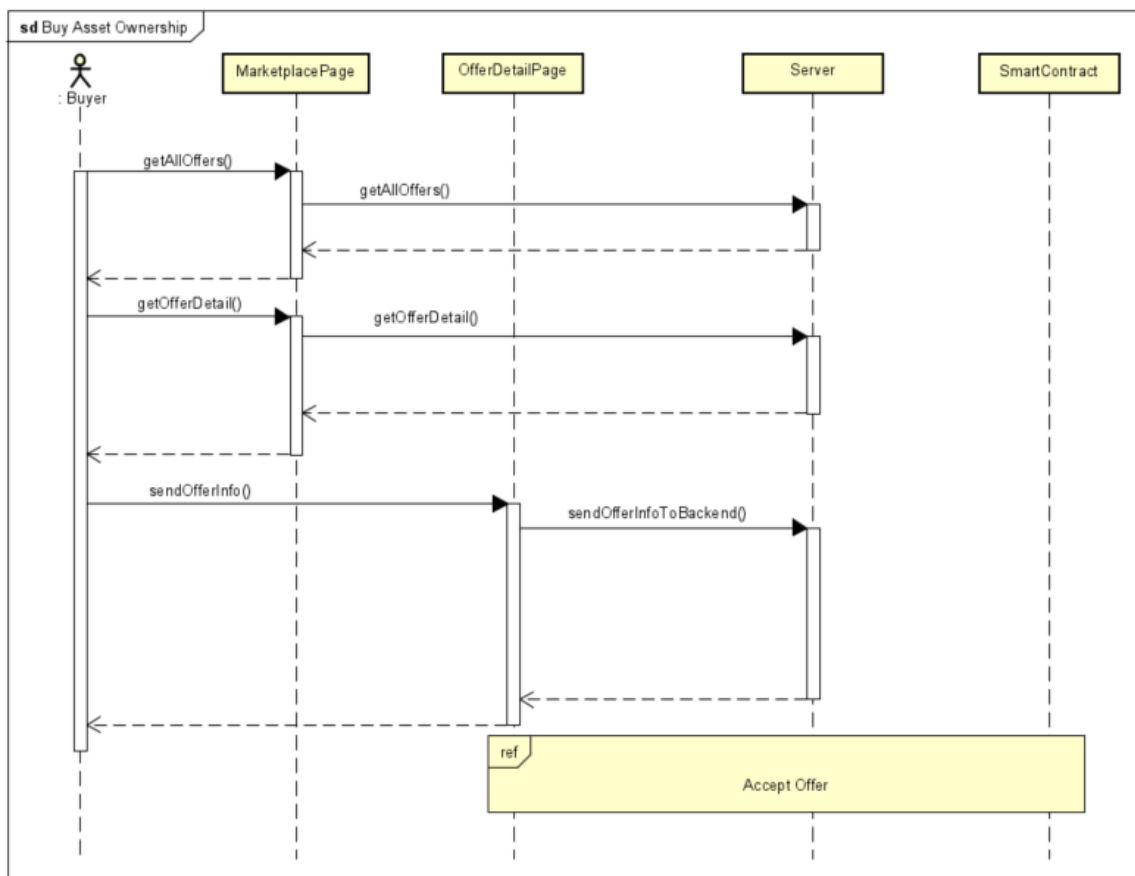


Figura X - Diagrama de sequência para a criação de proposta de compra de porcentagem de posse de ativo imobiliário

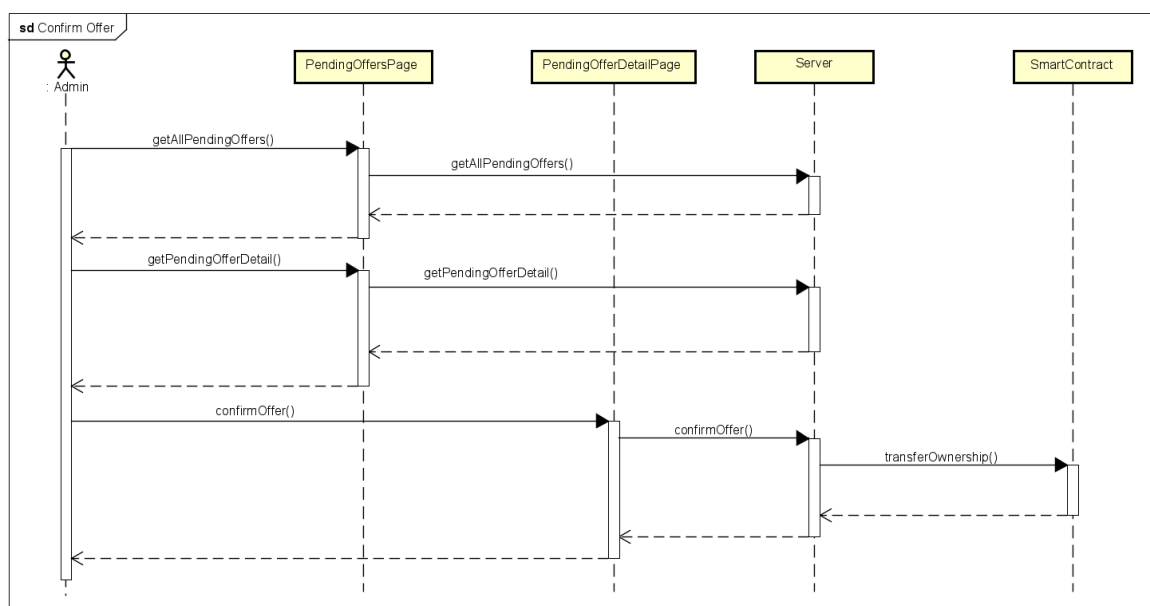


Figura X - Diagrama de sequência para a confirmação de compra de ativo imobiliário

4.5.3 Registro de garantia de empréstimo

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

De posse das informações de empréstimo acordadas entre ambas as partes, o dono do ativo irá inserir estas condições na tela de criação de empréstimo e estas serão enviadas para o servidor, que irá armazená-las no banco de dados e ficará no estado de aguardo de confirmação por parte de um administrador.

Após a criação do registro de empréstimo por parte do dono do ativo, um administrador irá verificar as informações do empréstimo bem como irá confirmar o valor do empréstimo entregue pelo mutuário ao dono do ativo, e então irá acessar a página de empréstimos pendentes da plataforma, onde deverá confirmar que o empréstimo foi criado corretamente, fazendo com que a plataforma registre os dados do empréstimo na blockchain através do *smart contract* e armazene os dados necessários em seu banco de dados.

Seguem abaixo 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.

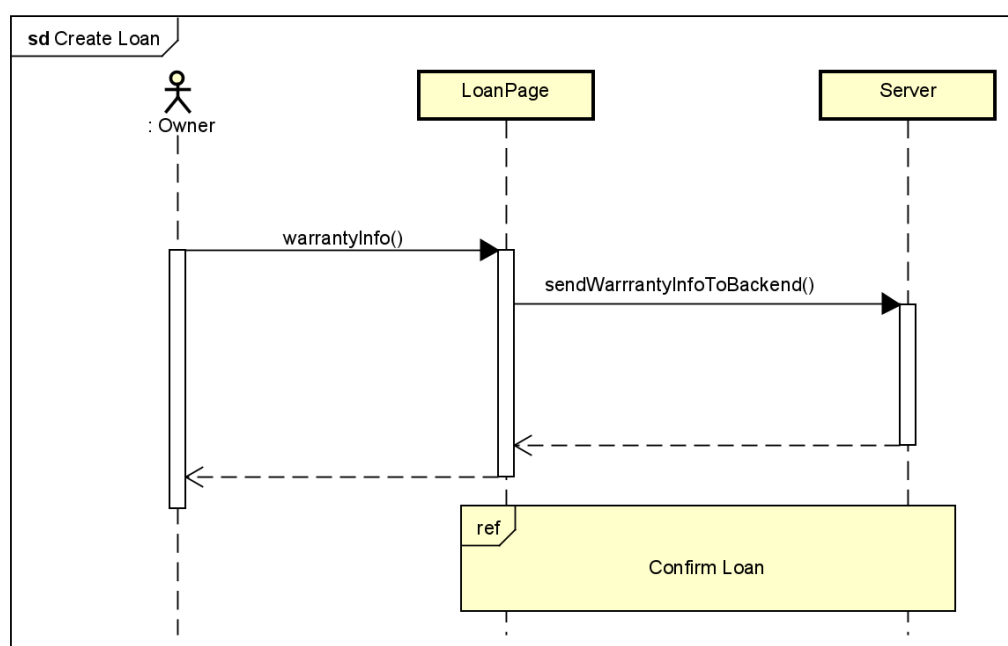


Figura X - Diagrama de sequência para criação de empréstimo

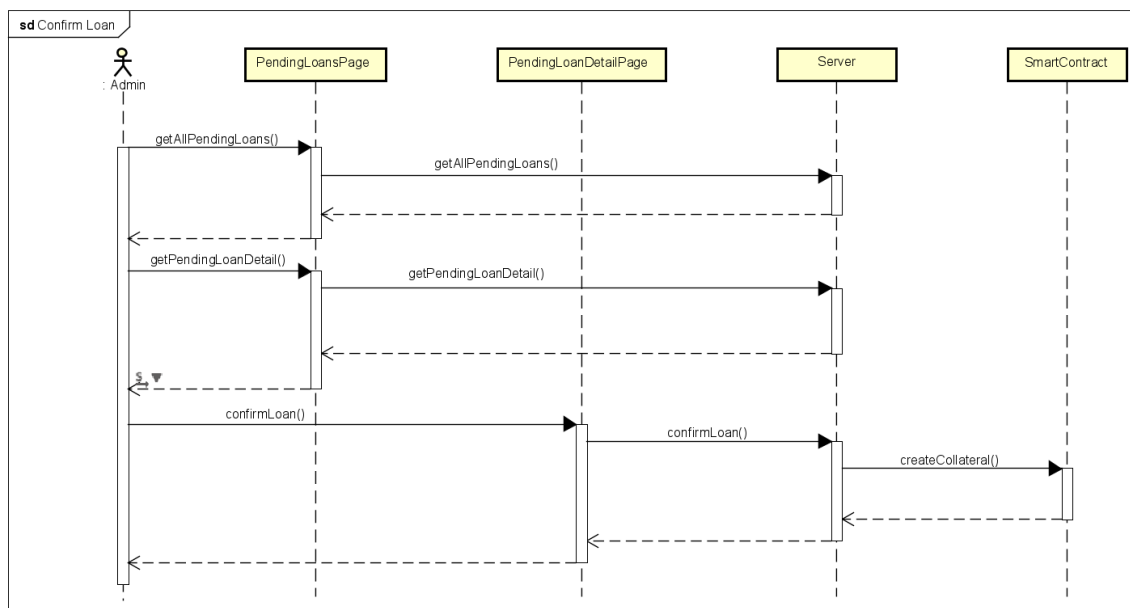


Figura X - Diagrama de sequência para confirmação de empréstimo

4.5.4 Transferência de Posse de Ativo Imobiliário

O dono de um ativo imobiliário *tokenizado* também poderá transferir a posse do ativo através da plataforma. Para isso, o dono deverá interagir com a tela de transferência de posse no *front-end* e inserir as informações necessárias para a transferência, como por exemplo, informações do novo dono e eventuais porcentagens da posse a serem transferidas. Com essas informações a interface enviará estes dados para o servidor que irá interagir com o *smart contract* e com o banco de dados da plataforma enviando e recebendo as informações necessárias.

Segue abaixo o diagrama de sequência deste caso de uso envolvendo o dono do ativo, a tela utilizada, o servidor e o *smart contract*, além das devidas mensagens trocadas entre os elementos para que seja possível a transferência de posse do ativo imobiliário tokenizado.

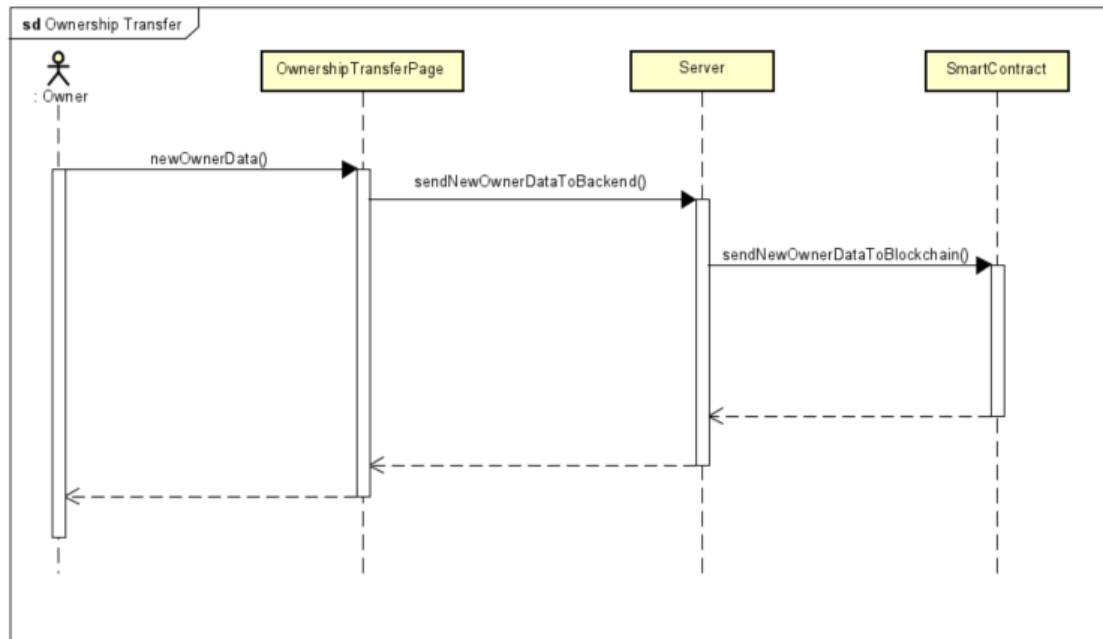


Figura X - Diagrama de sequência para o cenário de transferência de posse de ativo tokenizado

4.5.5 Quitação de Empréstimo

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 expiração do empréstimo seja alcançada.

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

Após o registro do pagamento do empréstimo pelo dono do ativo, um usuário administrador irá realizar a verificação do pagamento e então deverá interagir com o front-end de modo a confirmar a quitação, e então a plataforma irá registrar na blockchain a exclusão do empréstimo, liberar para o dono a utilização da porcentagem do ativo que estava associada ao empréstimo e então atualizar o banco de dados próprio com as novas informações de posse para se manter atualizada com relação ao blockchain.

Seguem abaixo 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.

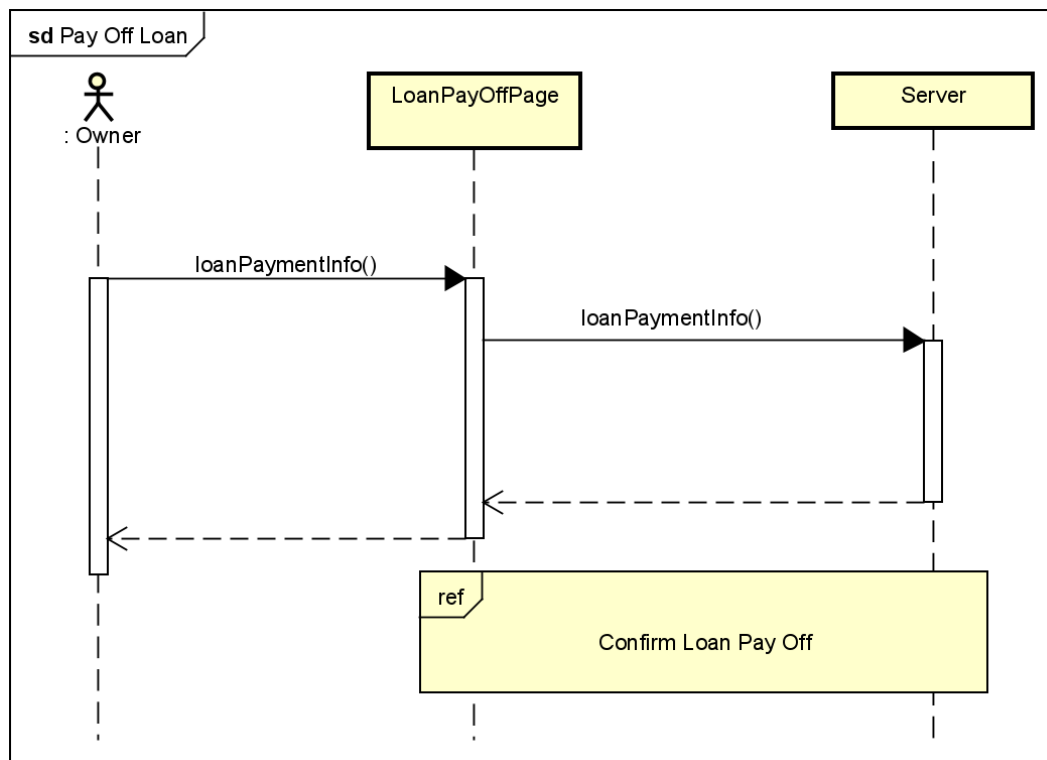


Figura X - Diagrama de sequência para registro de quitação de empréstimo

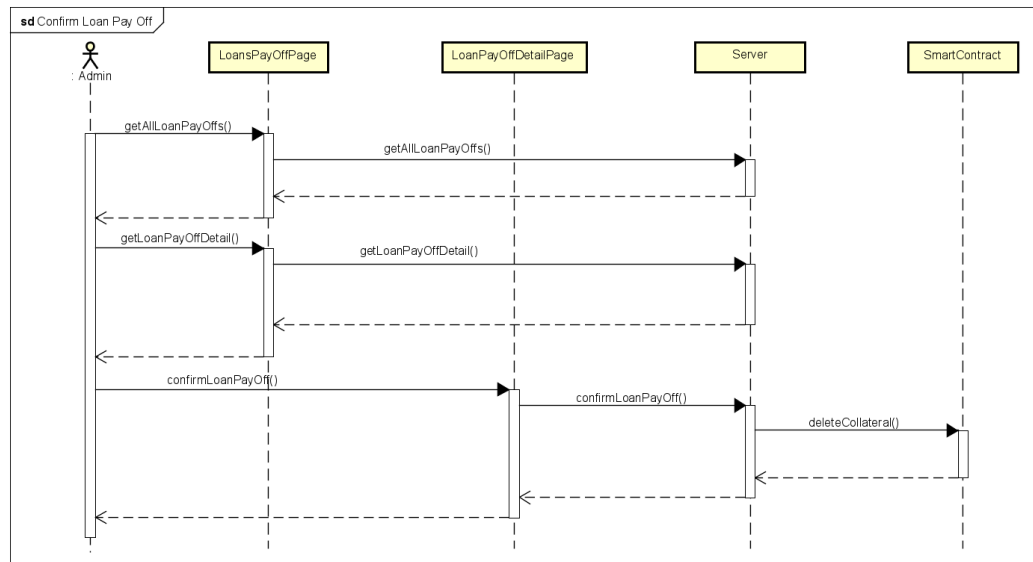


Figura X - Diagrama de sequência para confirmação de quitação de empréstimo

4.5.6 Tomada de Garantia de Empréstimo

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 expiração do empréstimo seja alcançada.

Caso o dono do imóvel não realize a quitação do empréstimo de uma porcentagem de um ativo imobiliário, o mutuário terá o direito de tomar posse da garantia utilizada no empréstimo. Para isso, é necessário que o mutuário possa interagir com uma tela no *front-end* e possa registrar a tomada da garantia do empréstimo.

Ao interagir com a tela, a plataforma irá verificar se a data de expiração já foi alcançada ou não, e caso tenha sido, a plataforma irá registrar no blockchain a transferência de porcentagem do ativo imobiliário do mutuante para o mutuário e também irá armazenar as informações necessárias em seu banco de dados.

Segue abaixo o diagrama de sequência deste caso de uso envolvendo o administrador, as telas utilizadas, o servidor e o *smart contract*, além das devidas mensagens trocadas entre os elementos para que seja possível a tomada de porcentagem de ativo imobiliário tokenizado utilizado como garantia de empréstimo.

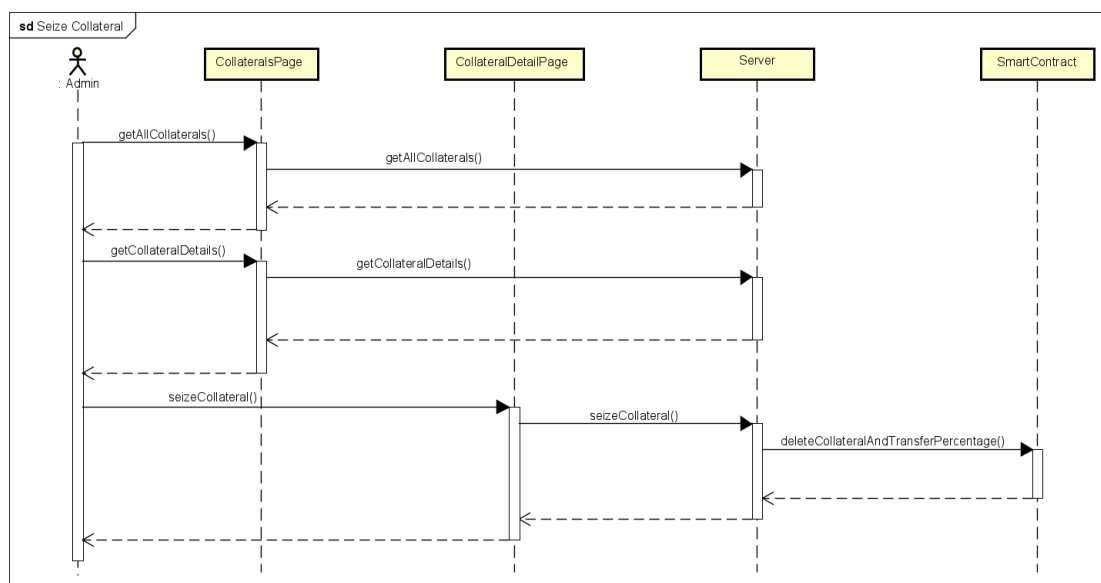


Figura X - Diagrama de sequência para tomada de garantia de empréstimo

4.5.7 Distribuição de retornos financeiros provenientes de Aluguel

Para o cenário de distribuição de lucros provenientes de aluguel, o proprietário do ativo imobiliário poderá cadastrar os lucros derivados do ativo na plataforma, que será encarregada de registrar esta distribuição do valor cadastrado entre os diversos proprietários de porcentagem do ativo em questão de acordo com a porcentagem possuída por cada um deles.

Para isto, o proprietário irá interagir com uma tela no *front-end* da plataforma e nela poderá selecionar o ativo do qual quer distribuir os lucros provenientes de aluguel da sua lista de ativos que possui e então poderá registrar o valor a ser distribuído para os proprietários, que será armazenado no banco de dados.

Após o registro do valor a ser pago pelo proprietário do ativo imobiliário, um administrador poderá interagir com uma tela do *front-end* da plataforma para obter a lista de lucros a serem distribuídos que estão registrados na plataforma e então poderá acessar o valor a ser distribuído do imóvel em específico para que possa verificar a distribuição dos lucros. De posse da informação do valor que o proprietário registrou que distribuiu para os donos de porcentagens do ativo, o administrador irá verificar se a distribuição foi feita corretamente pelo proprietário e então poderá confirmar a distribuição pela interface da plataforma.

Com a confirmação do administrador pela interface da aplicação, o *back-end* irá se comunicar com o smart contract e enviará os dados do pagamento realizado para registrar no blockchain a distribuição de lucros.

Seguem abaixo 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 de lucros provenientes de aluguel relacionados ao ativo imobiliário tokenizado.

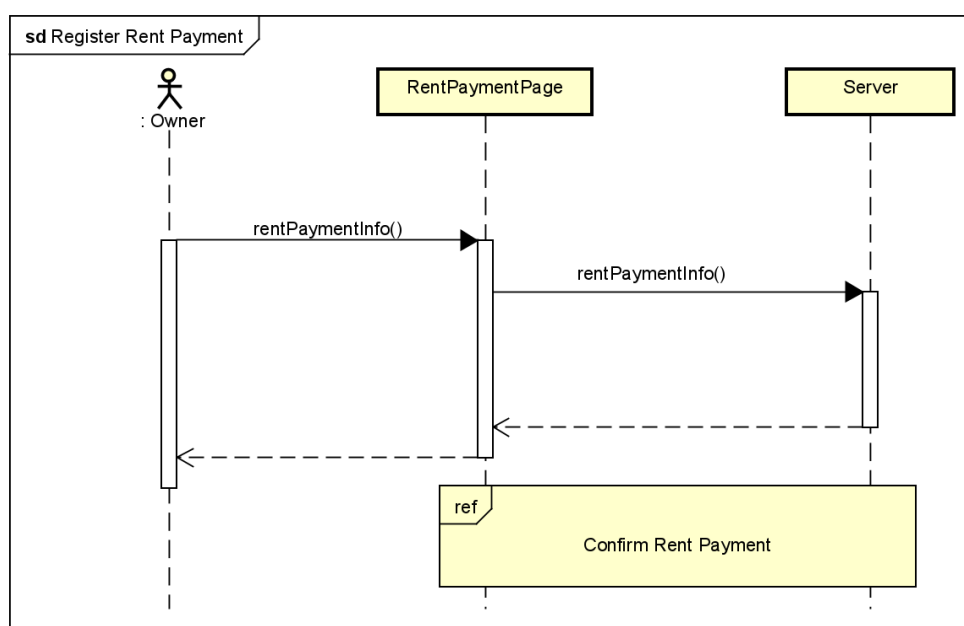


Figura X - Diagrama de sequência para registro de pagamento de lucros provenientes de aluguel

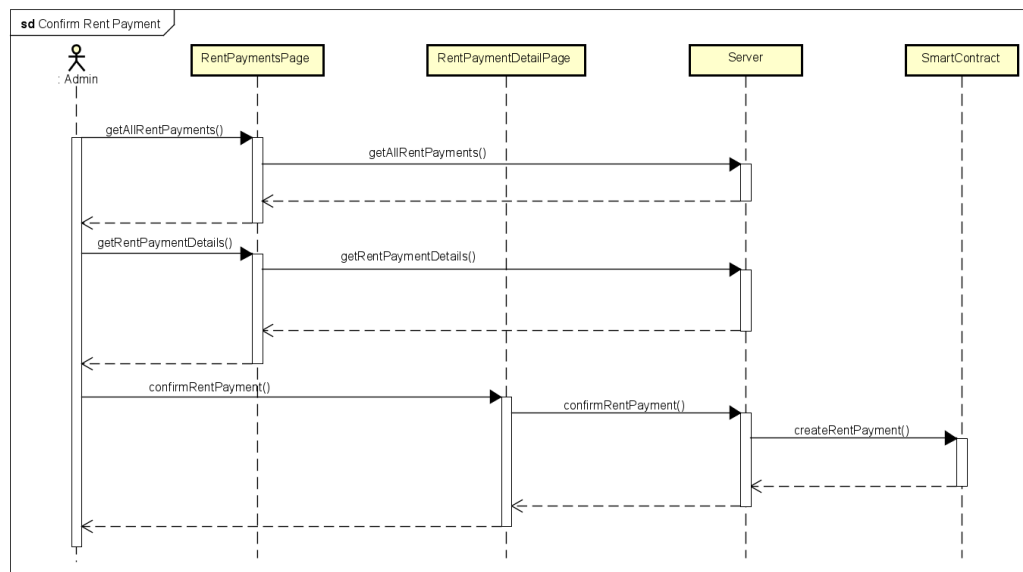


Figura X - Diagrama de sequência para confirmação do pagamento de lucros provenientes de aluguel

4.5.8 Registro de Usuário 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 e senha.

Para realizar este cadastro, 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 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.

Segue abaixo 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.

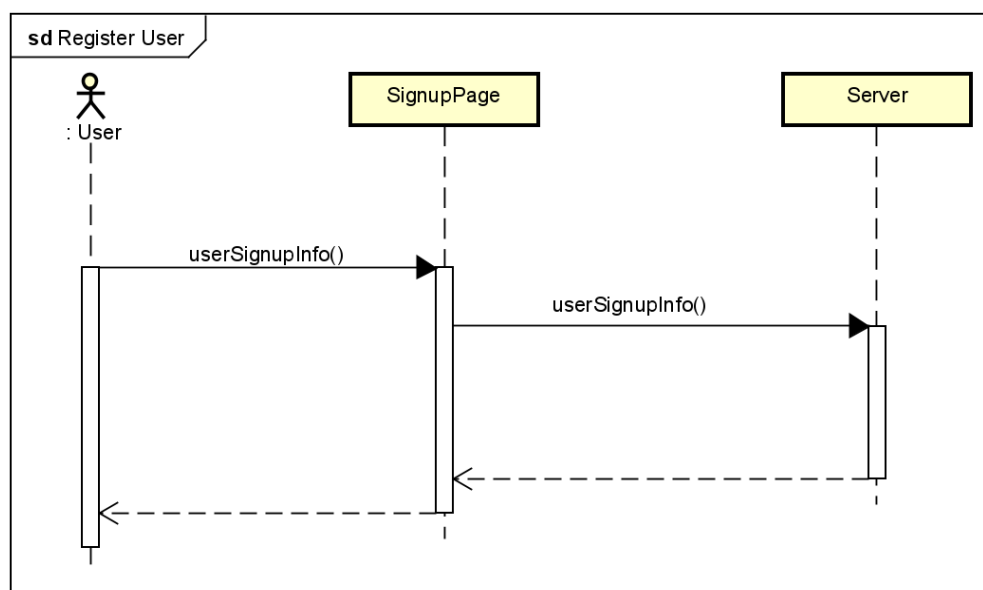


Figura X - Diagrama de sequência para registro de novo usuário na plataforma

4.5.9 Login de 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* na 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 verificar a autenticidade do usuário através da validação do nome de usuário e da senha enviados pelo *front-end*, verificando o usuário e senha armazenados no banco de dados da plataforma durante a etapa de registro do usuário na plataforma, e então, 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.

Segue abaixo 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 a autenticação de um usuário na plataforma.

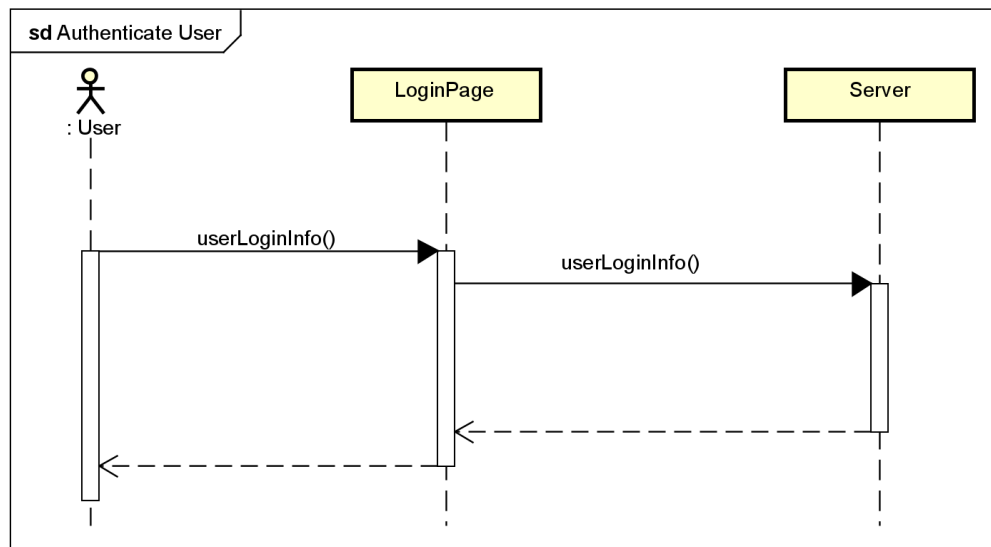


Figura X - Diagrama de sequência para autenticação de usuário na plataforma

4.5.10 Cadastro de Carteira do Usuário na Plataforma

Após o usuário registrar-se autenticar-se na plataforma, para que seja possível que o usuário utilize as funcionalidades da plataforma que envolvem transações no utilizando *smart-contracts* e *blockchain*, ele deverá conectar sua carteira de Ethereum à plataforma, e com isso poderá ser realizar transações e ser associado à atributos e métodos utilizados no *smart-contract* para que sejam implementadas as funcionalidades descritas nos casos de uso envolvendo tokenização, compra, venda e transferência de posse de ativos imobiliários, além da utilização de ativos tokenizados em empréstimos e a distribuição de retornos financeiros associados à ativos tokenizados na plataforma.

Para isso, o usuário autenticado irá acessar a página de cadastro de carteira na plataforma na interface do *front-end* e irá inserir a chave pública da sua carteira (endereço da carteira do usuário na rede Ethereum) num formulário e então irá enviar esta informação para o *back-end*, que atualizará o cadastro do usuário com o endereço de sua carteira e então o usuário estará apto a utilizar as funcionalidades descritas acima da plataforma.

Segue abaixo 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 a conexão da carteira Ethereum

do usuário à plataforma, de modo que viabilize este usuário a utilizar as funcionalidades implementadas pela plataforma.

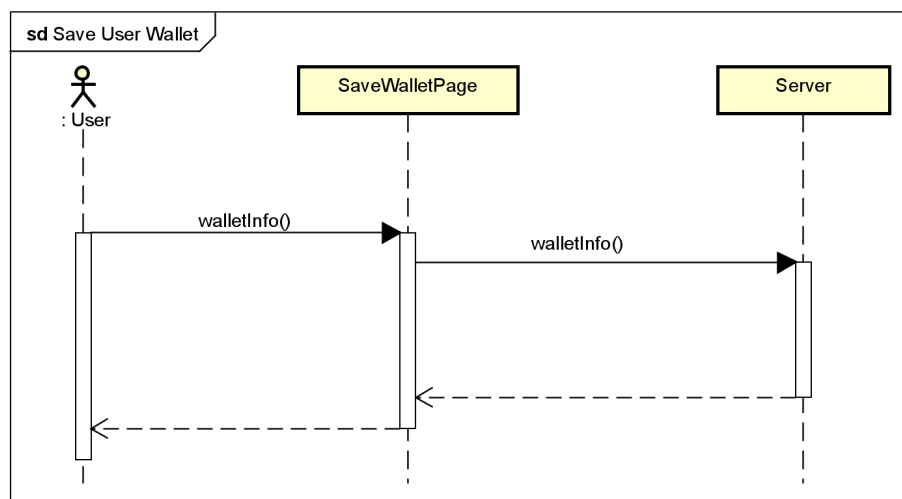


Figura X - Diagrama de sequência para cadastro de endereço de carteira de usuário na plataforma

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, como endereço do imóvel, área útil total e número de identificação 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 é nas funcionalidades propostas, e não na representação de um ativo imobiliário, bastando apenas que sejam utilizadas informações suficientes para que seja possível distinguir-se unicamente cada um dos ativos imobiliários.

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 uma 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.

Devido ao teor de

5.2 PROJETO E IMPLEMENTAÇÃO

Capítulo a ser desenvolvido.

Este capítulo irá descrever a implementação do smart-contract, modelagem do banco de dados, as funcionalidades desenvolvidas no *back-end* e as interfaces desenvolvidas no *front-end*.

5.2.1 Desenvolvimento dos Smart Contracts

Capítulo a ser desenvolvido.

Este capítulo irá descrever a estrutura do *smart contract* desenvolvido, detalhando seus atributos, métodos, regras de negócio, implementação dos diferentes papéis de usuários da plataforma e também como o processo de publicação e implantação do contrato na rede blockchain Goerli escolhida para a utilização no projeto.

5.2.2 Desenvolvimento do Front-End

Capítulo a ser desenvolvido.

Este capítulo irá descrever as diferentes telas desenvolvidas para suprir os diferentes casos de uso mapeados e descritos nas seções anteriores, bem como detalhar a integração da interface com a carteira do usuário para que ele possa utilizá-la na plataforma.

5.2.3 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.

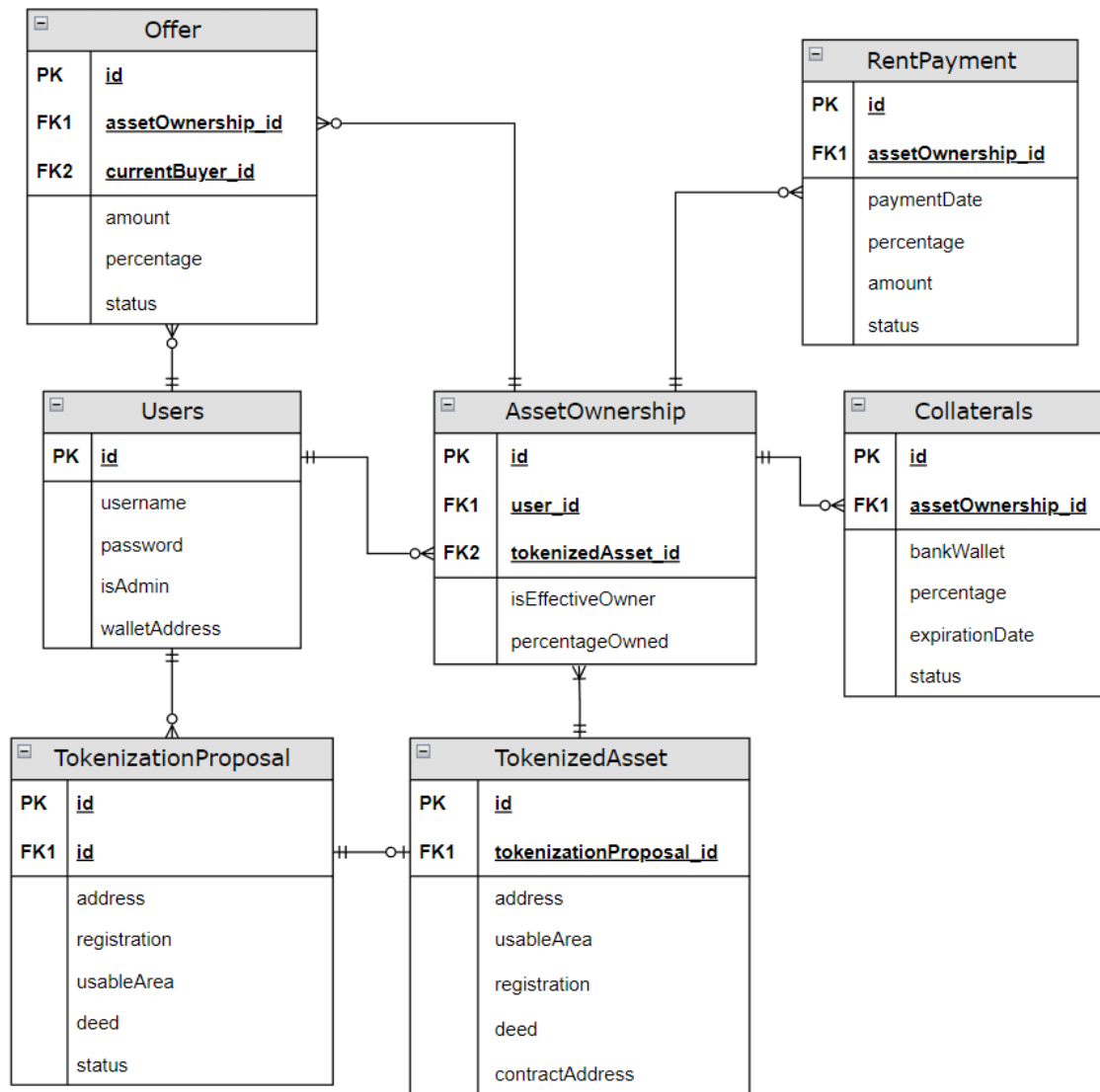


Figura X - Modelagem do banco de dados do projeto

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 escrituras 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.
- **AssetOwnership:** 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 dono 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.
- **Collateral:** Esta tabela armazena os dados da funcionalidade de garantia de empréstimos, em que *bankWallet* é o endereço da carteira digital do banco, *expirationDate* é a data de expiração do empréstimo, *percentage* é o percentual do imóvel dado em garantia e status podendo assumir os seguintes valores: *PENDING* - garantia pendente de validação pela plataforma; *ACTIVE* - garantia ativa na plataforma; *AWAITING_LOAN_PAYMENT_VALIDATION* - aguardando confirmação do pagamento do empréstimo; 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 proprietário efetivo do imóvel 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.

5.2.4 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*.

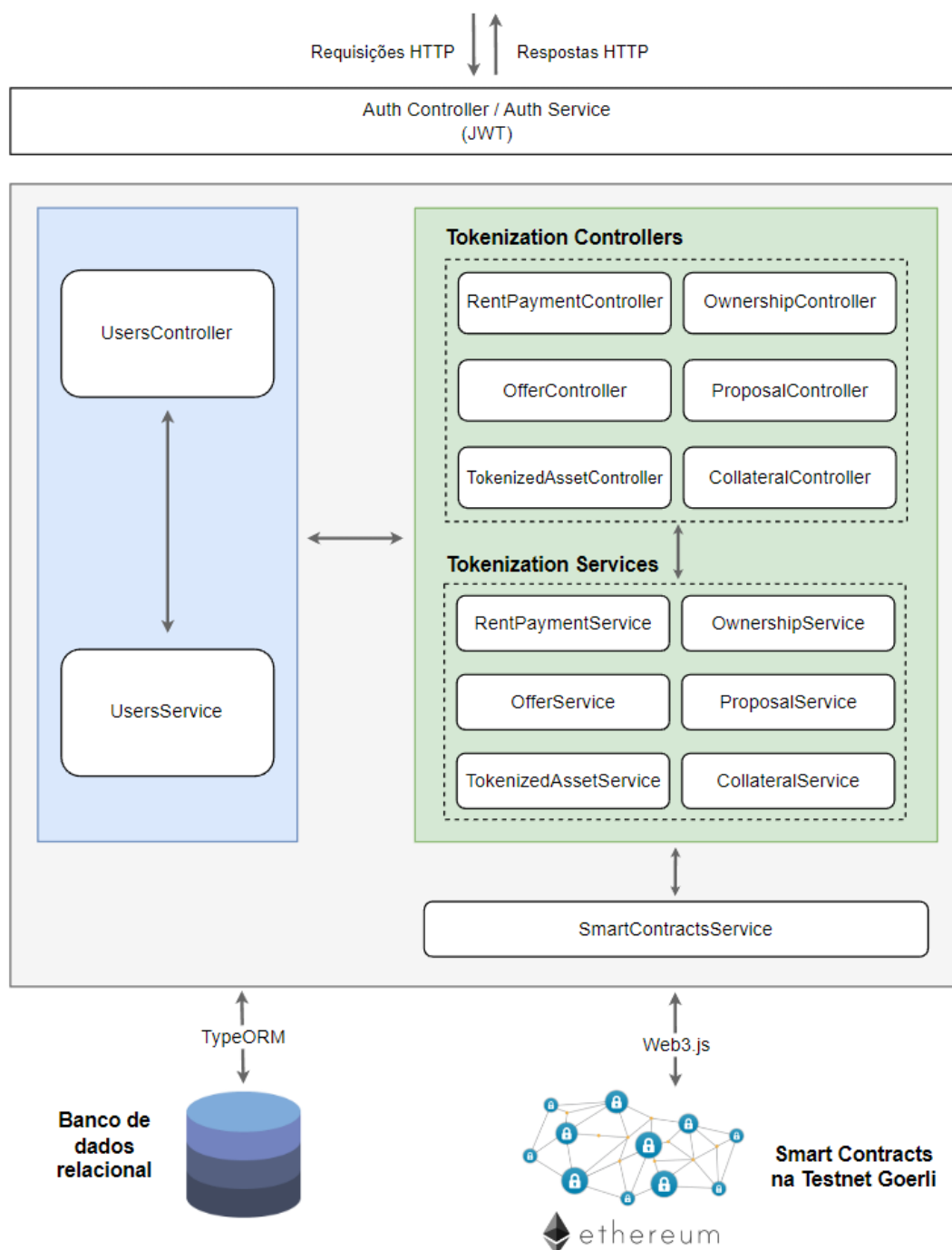


Figura X - Estrutura de Controladores e serviços do servidor

A Figura X retrata a estrutura do projeto seguindo o padrão de desenvolvimento detalhado. O controlador *UserController* é responsável por receber todas as requisições dentro do escopo de criação, autenticação, 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 as 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.

PRINTS SWAGGER

5.3 TESTES E AVALIAÇÃO

Capítulo a ser desenvolvido.

Os testes a serem realizados correspondem à validação das funcionalidades mapeadas nos casos de uso, e através dos resultados obtidos será possível avaliar-se 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 Transferência de posse

Capítulo a ser desenvolvido.

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

5.3.3 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.

5.3.4 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.

5.3.5 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://tokenyzed.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-nonfungible-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.