

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

**PLATAFORMA DIGITAL PARA TOKENIZAÇÃO DE ATIVOS IMOBILIÁRIOS**

De acordo:

A handwritten signature in black ink, reading "Marcos Antonio Simplicio Junior". The signature is written in a cursive style with a long horizontal stroke extending to the right. There are some small marks above the signature, possibly initials or a date.

---

Marcos Antonio Simplicio Junior

**São Paulo**

**2022**

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

**PLATAFORMA DIGITAL PARA TOKENIZAÇÃO DE ATIVOS IMOBILIÁRIOS**

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**

**PLATAFORMA DIGITAL PARA TOKENIZAÇÃO DE ATIVOS IMOBILIÁRIOS**

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 no cotidiano de qualquer pessoa. Eles podem ser ativos reais, como máquinas, veículos, e 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. O mesmo não é muito comum com ativos reais, em particular aqueles de alto valor, como é o caso de imóveis. Com isso em mente, tem ganhado força a ideia de trazer o conceito de tokenização para mercados que lidam com imóveis e ativos de baixa liquidez. A tokenização consiste essencialmente na representação de ativos reais na forma de tokens digitais, comumente usando uma rede *blockchain* como infraestrutura subjacente para garantir a integridade e transparência das transações realizadas. Os tokens podem ser gerenciados por contratos inteligentes (*smart contracts*) capazes de definir regras e funções a serem executadas na *blockchain*. Eles podem, por exemplo, permitir a decomposição do direito da posse do ativo real em parcelas de tamanho arbitrário, bem como ações diversas (e.g., venda ou compartilhamento de lucros) sobre essas parcelas. No setor imobiliário, essa abordagem traz alguns benefícios. Um exemplo é dar maior liquidez a esse tipo de ativo, uma vez que é possível comercializar frações dele por valores proporcionalmente menores. Além disso, dá-se maior flexibilidade ao uso do imóvel tokenizado ao permitir, por exemplo, que ele seja usado como garantia para mais de um empréstimo a baixo risco: afinal, ao usar apenas uma porcentagem do ativo como garantia, evita-se 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 anteriormente descritos.

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

## ABSTRACT

Nowadays, there are several types of assets in the regular life of many people. They can be real assets, such as machines, vehicles and 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. However, this doesn't happen very often with real assets, particularly those with high value, such as properties. With this in mind, the idea of using tokenization in markets that deal with illiquid assets have been growing. 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 can be managed by smart contracts capable of defining rules and functions to be executed in the blockchain. They can allow, for instance, the decomposition of the real asset's right of ownership in portions of arbitrary size, as well as many other actions (e.g., sale or profit sharing) over these portions. In the real estate market, this approach brings some benefits. An example is to grant a greater liquidity to this kind of asset once it enables the commercialization of its fractions for proportionally smaller amounts. Furthermore, the usage of tokenized real estate assets becomes more flexible by allowing, for example, its use as guarantee of multiple loans with low risk: after all, by using just a percentage of the asset as a guarantee we can avoid the scenario 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**

1	Arquitetura Proposta .....	25
2	Diagrama de sequência para registro de novo usuário na plataforma .....	28
3	Diagrama de sequência para autenticação de usuário na plataforma .....	29
4	Diagrama de sequência para cadastro de endereço de carteira de usuário na plataforma .....	30
5	Diagrama de sequência para a tokenização de um imóvel .....	31
6	Diagrama de sequência para a validação dos documentos de um imóvel para o processo de tokenização .....	31
7	Diagrama de sequência para a criação de oferta de porcentagem de ativo tokenizado .....	33
8	Diagrama de sequência para aceite de oferta de ativo imobiliário tokenizado .....	33
9	Diagrama de sequência para a confirmação de compra de ativo imobiliário	34
10	Diagrama de sequência para criação de empréstimo .....	35
11	Diagrama de sequência para confirmação de empréstimo .....	35
12	Diagrama de sequência para registro de quitação de empréstimo .....	36
13	Diagrama de sequência para confirmação de quitação de empréstimo .....	37
14	Diagrama de sequência para tomada de garantia de empréstimo .....	37
15	Diagrama de sequência para registro de pagamento de aluguéis .....	38
16	Diagrama de sequência para visualização dos pagamentos de aluguéis recebidos por um proprietário .....	39
17	Diagrama de sequência para auditoria dos dados do imóvel na <i>blockchain</i>	40
18	Modelagem do banco de dados do projeto .....	50

19	Estrutura de Controladores e serviços do servidor .....	53
20	Documentação do Swagger para as rotas de <i>Smart Contracts</i> , <i>Users</i> , <i>Tokenized Assets</i> , <i>Proposals</i> e <i>Ownerships</i> .....	55
21	Documentação do Swagger para as rotas de <i>Collaterals</i> , <i>Offers</i> , <i>Rent</i> <i>Payments</i> e <i>Authentication</i> .....	56
22	Tela de login e cadastro de usuário .....	57
23	Aba de dados gerais da conta .....	58
24	Aba de imóveis tokenizados do usuário .....	59
25	Registro dos aluguéis recebidos enquanto proprietário do imóvel .....	59
26	Aba de garantias do usuário .....	60
27	Aba de ofertas do usuário abertas no Marketplace .....	60
28	Aba de compras realizadas no Marketplace pelo usuário .....	61
29	Formulário de envio dos dados do imóvel para tokenização .....	62
30	Tela de listagem dos imóveis disponíveis para criação de oferta .....	63
31	Formulário de criação de oferta no Marketplace .....	63
32	Marketplace de imóveis da plataforma .....	64
33	Tela de detalhes da oferta do imóvel .....	64
34	Tela de listagem dos imóveis disponíveis para criação de garantia .....	65
35	Formulário de criação de garantia de empréstimo .....	66
36	Aba listando todas as propostas de tokenização da plataforma .....	67
37	Aba listando todos os pagamentos de ofertas pendentes de confirmação ..	68
38	Aba listando todos os cadastros de garantia de empréstimos pendentes na plataforma .....	69

39	Aba listando todas as garantias de empréstimo registradas na plataforma .	70
40	Aba listando todos os imóveis tokenizados da plataforma .....	70
41	Modal de auditoria dos dados do imóvel tokenizado .....	71
42	Formulário de registro de pagamento de aluguel .....	72

**LISTA DE ABREVIATURAS E SIGLAS**

<b>API</b>	<i>Application Programming Interface</i>
<b>CPF</b>	Cadastro de Pessoa Física
<b>ERC</b>	<i>Ethereum Request for Comments</i>
<b>FT</b>	<i>Fungible token</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>HTTPS</b>	<i>Hypertext Transfer Protocol Secure</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>JWT</b>	<i>JSON Web Token</i>
<b>LARC</b>	Laboratório de Arquitetura e Redes de Computadores
<b>NFT</b>	<i>Non-fungible token</i>
<b>ORM</b>	<i>Object–Relational Mapping</i>
<b>PoS</b>	<i>Proof-of-Stake</i>
<b>PoW</b>	<i>Proof-of-Work</i>
<b>REST</b>	<i>Representational state transfer</i>
<b>SQL</b>	<i>Structured Query Language</i>
<b>USP</b>	Universidade de São Paulo

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 MOTIVAÇÃO .....	13
1.2 OBJETIVO .....	14
1.3 JUSTIFICATIVA .....	15
1.4 SOLUÇÕES DE MERCADO .....	15
1.5 ORGANIZAÇÃO DO TRABALHO .....	16
<b>2 ASPECTOS CONCEITUAIS</b> .....	<b>17</b>
2.1 BLOCKCHAIN .....	17
2.2 SMART CONTRACTS .....	18
2.3 TOKEN .....	19
<b>2.3.1 Token Fungível (FT)</b> .....	<b>19</b>
<b>2.3.2 Token Não Fungível (NFT)</b> .....	<b>20</b>
2.4 CRIPTOGRAFIA ASSIMÉTRICA .....	21
<b>3 MÉTODO DO TRABALHO</b> .....	<b>22</b>
<b>4 ESPECIFICAÇÃO DO SISTEMA</b> .....	<b>23</b>
4.1 AGENTES DO SISTEMA .....	23
4.2 REQUISITOS FUNCIONAIS .....	23
4.3 REQUISITOS NÃO-FUNCIONAIS .....	24
<b>4.3.1 Escalabilidade</b> .....	<b>24</b>
<b>4.3.2 Segurança</b> .....	<b>24</b>
4.4 ARQUITETURA PROPOSTA .....	25
<b>4.4.1 Interface da plataforma (Front-End)</b> .....	<b>26</b>
<b>4.4.2 Servidor da plataforma (Back-End)</b> .....	<b>26</b>
<b>4.4.3 Banco de dados relacional (Back-End)</b> .....	<b>27</b>
<b>4.4.4 Blockchain e Smart Contracts</b> .....	<b>27</b>

4.5 CASOS DE USO PRINCIPAIS .....	27
4.5.1 Registro e login de usuários na plataforma .....	28
4.5.2 Cadastro de carteira do usuário na plataforma .....	29
4.5.3 Validação de documentos e tokenização de um imóvel .....	30
4.5.4 Compra e venda de porcentagem de posse de ativo imobiliário .....	32
4.5.5 Registro de garantia de empréstimo .....	34
4.5.6 Quitação de empréstimo e tomada de garantia .....	36
4.5.7 Distribuição de retornos financeiros provenientes de aluguel .....	38
4.5.8 Auditoria de dados registrados na blockchain .....	39
<b>5 DESENVOLVIMENTO DO TRABALHO .....</b>	<b>41</b>
5.1 PREMISSAS E RESTRIÇÕES .....	41
5.2 TECNOLOGIAS UTILIZADAS .....	43
5.2.1 Node.js .....	43
5.2.2 React .....	43
5.2.3 PostgreSQL .....	43
5.2.4 Solidity .....	44
5.2.5 Remix IDE .....	44
5.2.6 Web3.js .....	44
5.2.7 Ethereum Goerli Testnet .....	44
5.2.8 TypeORM .....	45
5.3 PROJETO E IMPLEMENTAÇÃO .....	45
5.3.1 Desenvolvimento do Smart Contract .....	45
5.3.1.1 Atributos armazenados no smart contract desenvolvido .....	46
5.3.1.2 Métodos desenvolvidos no smart contract .....	47
5.3.1.3 Publicação e implantação do smart contract na blockchain Goerli .....	49
5.3.2 Estrutura do banco de dados relacional .....	49
5.3.3 Desenvolvimento do servidor .....	52

<b>5.3.4 Desenvolvimento do Front-End</b> .....	<b>56</b>
5.3.4.1 Login e cadastro .....	57
5.3.4.2 Perfil de usuário .....	58
5.3.4.3 Tokenização de imóvel .....	61
5.3.4.4 Criar oferta no Marketplace .....	62
5.3.4.5 Marketplace .....	63
5.3.4.6 Criar garantia de empréstimo .....	65
5.3.4.7 Portal do administrador .....	66
<b>5.4 TESTES E AVALIAÇÃO</b> .....	<b>72</b>
<b>5.4.1 Considerações sobre requisitos não-funcionais</b> .....	<b>72</b>
<b>5.4.2 Tokenização de imóvel</b> .....	<b>73</b>
<b>5.4.3 Oferta e compra de percentual de imóvel</b> .....	<b>74</b>
<b>5.4.4 Garantias de empréstimos</b> .....	<b>75</b>
<b>5.4.5 Distribuição de retornos financeiros provenientes de aluguel</b> .....	<b>76</b>
<b>6 CONSIDERAÇÕES FINAIS</b> .....	<b>77</b>
6.1 PERSPECTIVAS DE CONTINUIDADE .....	78
<b>REFERÊNCIAS</b> .....	<b>80</b>
<b>APÊNDICE A - Teste do Caso de Uso de Tokenização de Imóvel</b> .....	<b>86</b>
<b>APÊNDICE B - Teste do Caso de Uso de Oferta e Compra de Percentual de Imóvel</b> .....	<b>88</b>
<b>APÊNDICE C - Teste do Caso de Uso de Garantia de Empréstimo</b> .....	<b>92</b>
<b>APÊNDICE D - Teste do Caso de Distribuição de Retornos Financeiros Provenientes de Aluguel</b> .....	<b>96</b>

## 1 INTRODUÇÃO

Existem diversos tipos de ativos no mercado atual, os quais podem ser classificados em duas categorias: os ativos reais (ou “tangíveis”), e os ativos financeiros (ou “intangíveis”) [1]. Os ativos reais englobam bens materiais que podem ser tocados, como máquinas, veículos, imóveis, moedas antigas, vinhos, entre outros. Já os ativos financeiros são bens intangíveis, como ações, títulos públicos, títulos privados, criptomoedas, 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 incompatível com o valor do próprio imóvel utilizado na garantia, paralisando um valor patrimonial muito maior do que o necessário para garantir o 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 e resoluções 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, apesar de se encontrar em estágio inicial, tem evoluído significativamente com o intuito de que os registros de imóveis como propriedades digitais 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 projetar e 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* deve se conectar a uma rede *blockchain*, na qual estão contidos contratos inteligentes responsáveis pela geração e gestão dos tokens. Uma vez obtidos, os ativos imobiliários tokenizados podem ser utilizados dentro da plataforma como garantia para múltiplos empréstimos, e podem 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 nas seções 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 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]. 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 adição às 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. Uma vez feito isso, a tokenização acontece e a propriedade digital na *blockchain* passa 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: *Vanhall* [11], cuja principal função é fornecer toda a infraestrutura necessária para tokenização de ativos que vão além dos imobiliários; e *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:

A seção 2 apresenta breves introduções aos principais conceitos abordados no trabalho.

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

A seção 4 versa sobre as especificações e os requisitos que o sistema desenvolvido deve cumprir.

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

A seção 6 traz as considerações finais sobre o trabalho desenvolvido, desafios encontrados, 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 em um 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 algoritmos de consenso presentes nas *blockchains* [19]. São esses mecanismos que garantem a existência de uma versão única da sequência de dados de forma descentralizada e resistente a modificações pelos diferentes nós responsáveis por manter a rede. Comumente, todos os dados sobre transações são expostos de forma transparente, garantindo auditabilidade e tornando esta tecnologia extremamente útil em cenários que demandam alta confiabilidade.

As *blockchains* podem ser públicas, permitindo que qualquer nó participe da rede em processos de registro de transações, validação e criação de novos blocos com base em algoritmos de consenso, além de manter a auditoria dos dados acessível a qualquer pessoa. Alternativamente, podem ser construídas *blockchains* privadas, as quais requerem autorização para ingresso e, eventualmente, diferentes níveis de permissão para leitura e registro dos dados dispostos em cadeia.

Dentre as categorias de algoritmos de consenso mais conhecidas, é possível citar o Proof-of-Work (PoW), utilizado pelo Bitcoin, cujo princípio está baseado na resolução de problemas computacionais caros, que exigem alta alocação de recursos para serem solucionados no menor tempo possível. No caso do Bitcoin,

partindo do hash do último bloco da cadeia e o incrementando de um *nonce* (número aleatório), o problema envolve encontrar um número que, quando calculado o seu hash, seja retornado um valor que inicie com uma certa quantidade de bits zero, de tal forma que o esforço computacional é exponencial à quantidade de bits zero requerida. Resolvido o problema por um nó da rede, a verificação da resposta exige poder computacional muito menor que a resolução e, caso validada, é feito um *broadcast* para a inclusão de um novo bloco na rede e o nó é recompensado com incentivos financeiros por meio de Bitcoins [18].

Como alternativa ao Proof-of-Work, outra categoria de algoritmos de consenso amplamente conhecida é o Proof-of-Stake (PoS), que passou a ser utilizada pela rede de *blockchain* Ethereum em 2022, na qual nós validadores devem possuir uma determinada quantia de moedas da rede para que sejam sorteados para validar um bloco, tendo chances proporcionais à quantidade de moedas que possuem. Parte das moedas desses nós são tomadas como garantia da adequação às regras de criação de novos blocos e, caso a validação ocorra, estes nós são recompensados com taxas sobre as transações realizadas pelos usuários da rede [20]. Um ponto a se destacar em relação a este tipo de algoritmo é o fato de, por não envolver resolução de problemas com alto custo computacional, tem-se um consumo energético menor com nível de segurança equiparável ao do PoW e menor barreira de entrada para validadores, uma vez que não é necessário o alto investimento em hardwares para mineração [21].

## 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 na *blockchain*, determinado automaticamente durante a sua implantação inicial e que através dele é possível que usuários da *blockchain* invoquem as diversas funções implementadas no *smart contract*, que poderá fazer cálculos, verificar condições, realizar transações automaticamente para a rede dependendo de seu

estado interno e das condições verificadas [22] e até mesmo invocar funções de outros *smart contracts* presentes na rede.

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 [23].

## 2.3 TOKEN

No contexto de *blockchains*, um conceito bastante importante é o de token. Em linhas gerais, 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 às *blockchains*, temos que um token é um objeto digital criado através de um *smart contract* existente na rede e que representa um ativo físico ou digital qualquer, seja ele dinheiro, um bem, uma propriedade ou uma arte.

Um token na *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.

No contexto de *blockchain Ethereum*, as diferentes categorias de tokens são desenvolvidas com base em protocolos, chamados *ERC (Ethereum Request for Comments)* que definem regras e padrões a serem cumpridos e que servem como interfaces a serem implementadas pelos *smart contracts* dos tokens [24].

Tradicionalmente, os tokens são classificados em duas categorias, dependendo de sua natureza fungível ou não fungível. Essas categorias serão descritas nas subseções a seguir [25] [26].

### 2.3.1 Token Fungível (FT)

A primeira categoria é a de tokens fungíveis. Em sua maioria, são implementados pelo padrão ERC-20, proposto por Fabian Vogelsteller em 2015 [27], que tem como principais características:

- implementa tokens passíveis de substituição,

- define um valor inicial fixo de emissão de tokens na *blockchain* e permite a leitura deste valor,
- permite transações de tokens entre contas na *blockchain*,
- permite buscar o balanço de tokens atual de uma conta,
- permite aprovação do uso de determinada quantidade de tokens de uma conta por terceiros

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 outra nota de um real, 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 balanço na conta de ambas.

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)**

A segunda categoria mencionada é a de tokens não fungíveis. Eles são implementados com base no padrão ERC-721, proposto por Dieter Shirley et al. em 2017 [28], cuja principal característica é permitir a criação de tokens completamente únicos, de tal forma que um *smart contract* que segue este padrão, quando enviado para a *blockchain*, gera apenas uma unidade do token.

Ao contrário dos tokens fungíveis, os tokens não fungíveis (ou NFTs) são comumente gerados para representar algum ativo específico e colecionável, como por exemplo uma arte digital, que pode ou não ter alto valor agregado por conta de seu caráter único.

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 são intercambiáveis.

Esta categoria de tokens tem ganhado 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 [29].

## 2.4 CRIPTOGRAFIA ASSIMÉTRICA

Um conceito importante da área de segurança da informação, e que possui grande relevância no contexto da blockchain é o de criptografia assimétrica, também conhecida como criptografia de chave pública. Essa forma de criptografia utiliza um par de chaves, sendo uma delas pública e outra privada, podendo a chave pública ser conhecida por qualquer pessoa e a chave privada não devendo ser exposta a terceiros, sendo conhecida apenas por seu dono.

Este modelo de criptografia é utilizado para a realização de diversos serviços no mundo digital como, por exemplo, a cifração e decifração de dados num contexto geral e, no contexto da blockchain, utilizado para a manutenção de carteiras na rede blockchain, sendo a chave pública utilizada como o endereço da carteira de um usuário na rede e a chave privada a informação que o usuário utiliza para assinar transações na rede blockchain (e.g. para transferir o saldo de sua carteira para a carteira de outro usuário na rede).

### 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* escolhida. Em seguida, foi implementado o *web app*, composto pela interface (*Front-End*) na qual usuários poderão interagir e enviar solicitações de serviços ao servidor (*Back-End*) integrado ao banco de dados do projeto e à *blockchain* escolhida.

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

## 4 ESPECIFICAÇÃO DO SISTEMA

Nesta seção, 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 utilização de imóveis como garantia de mais de um empréstimo.

### 4.1 AGENTES DO SISTEMA

Na plataforma, haverá quatro 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 [30].

#### 4.3.2 Segurança

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

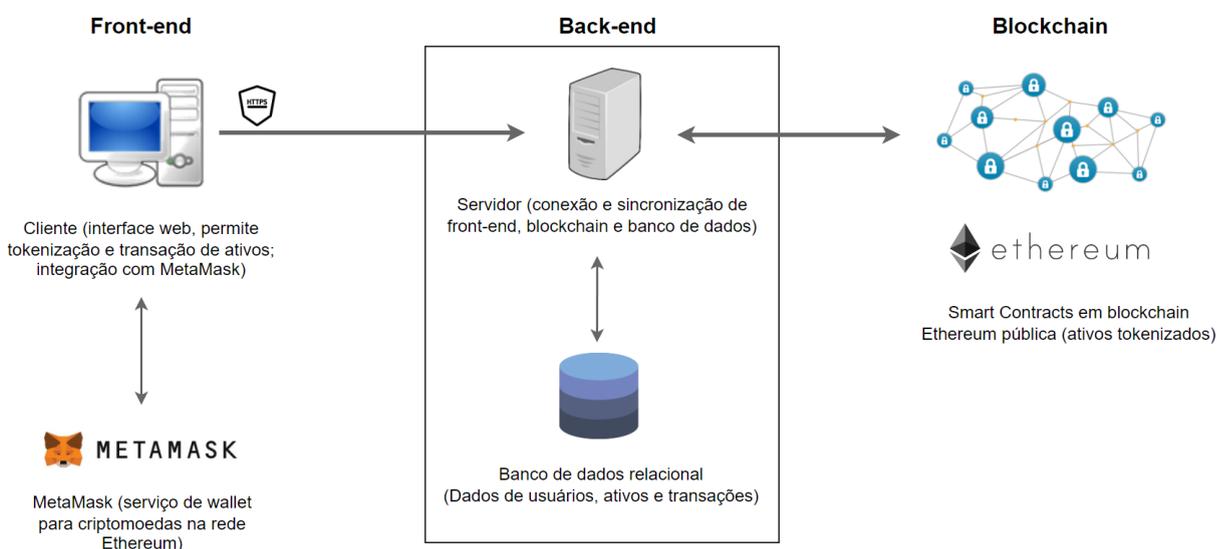
Para isso, um dos requisitos não-funcionais do projeto diz respeito ao processo de *login* dos usuários da plataforma. Neste processo, será necessário

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 1 - Arquitetura Proposta**

A arquitetura presente na Figura 1 dispõe de *Front-End* (que se integra ao serviço da MetaMask), *Servidor*, *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 4.2, 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 então o *Front-End* enviará 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 [31], 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. O *Front-End* e o *Back-End* foram desenvolvidos em aplicações distintas, 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 2 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.

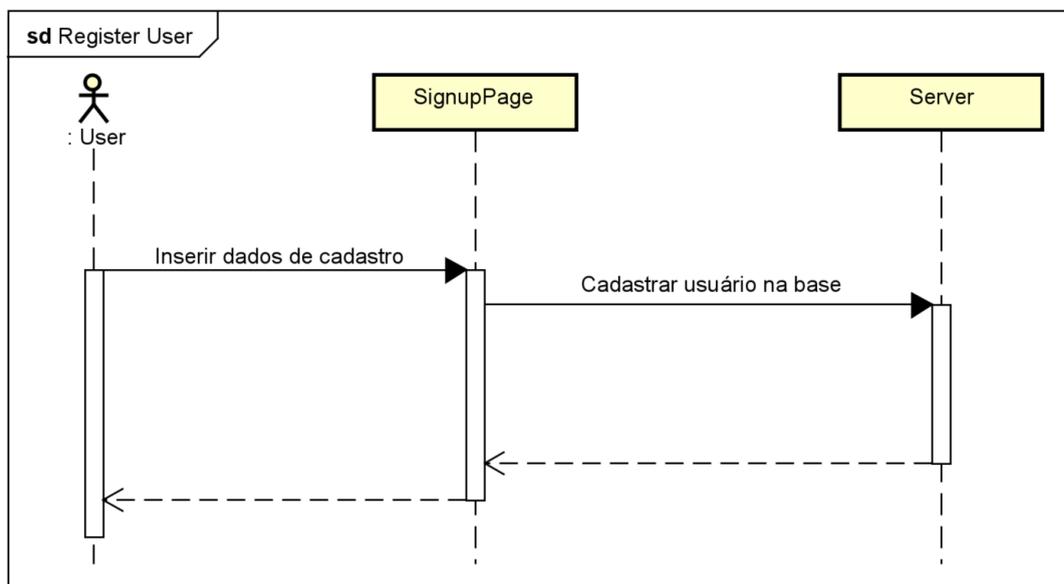


Figura 2 - Diagrama de sequência para registro de novo usuário na plataforma

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 3.

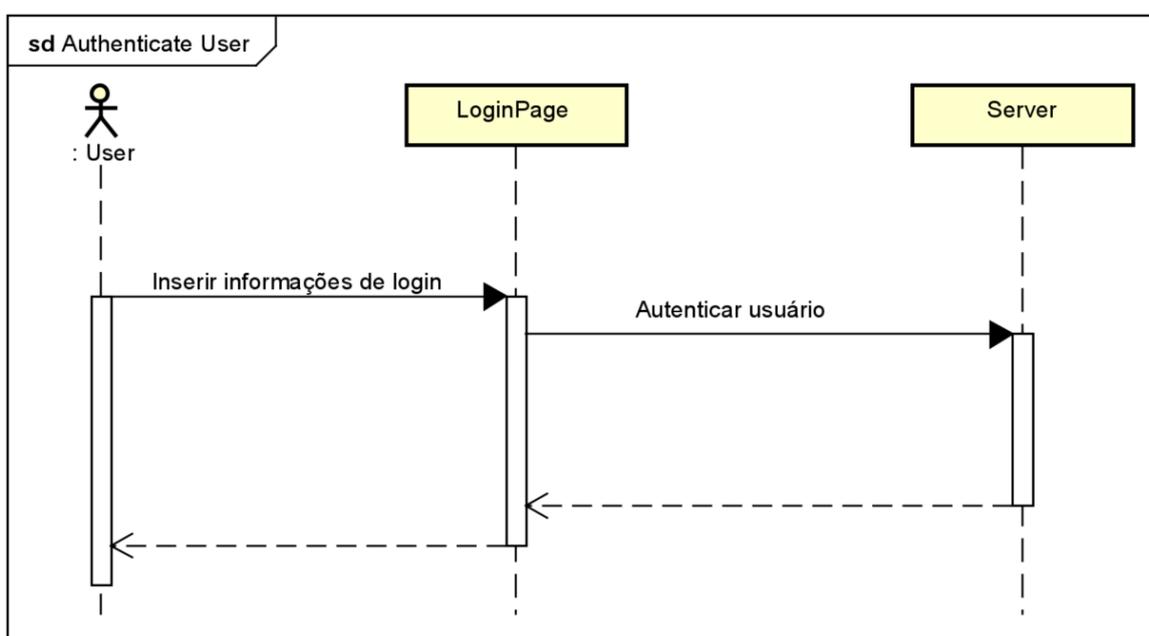
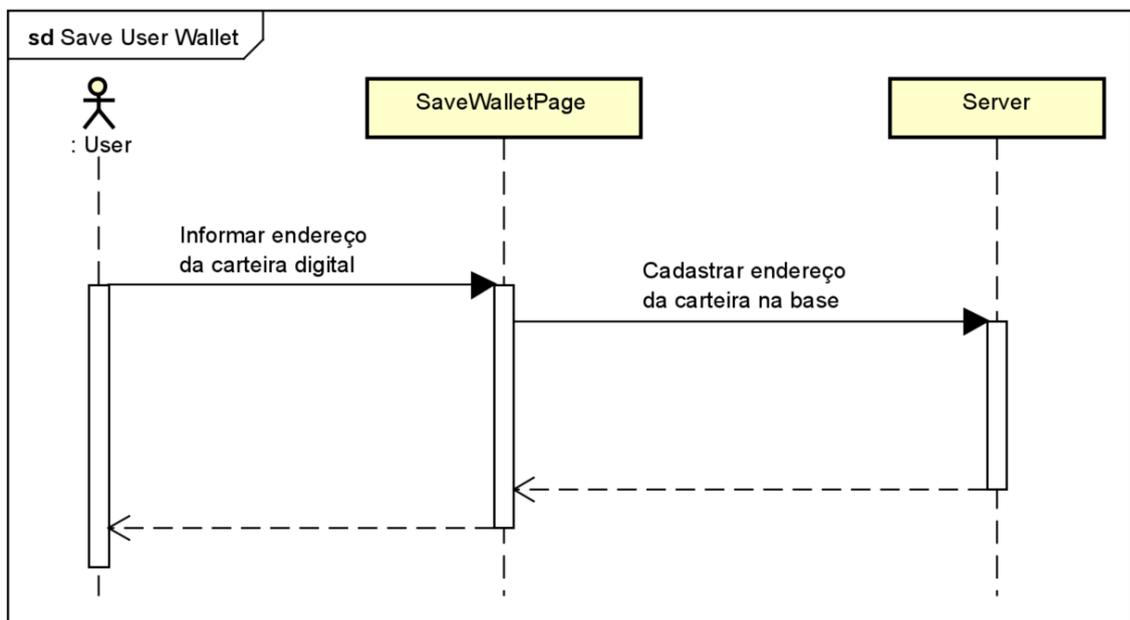


Figura 3 - Diagrama de sequência para autenticação de usuário na plataforma

#### 4.5.2 Cadastro de carteira do usuário na plataforma

Após o usuário se cadastrar e se 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 4 mostra o diagrama de sequência com todas as etapas descritas.



**Figura 4 - Diagrama de sequência para cadastro de endereço de carteira de usuário na plataforma**

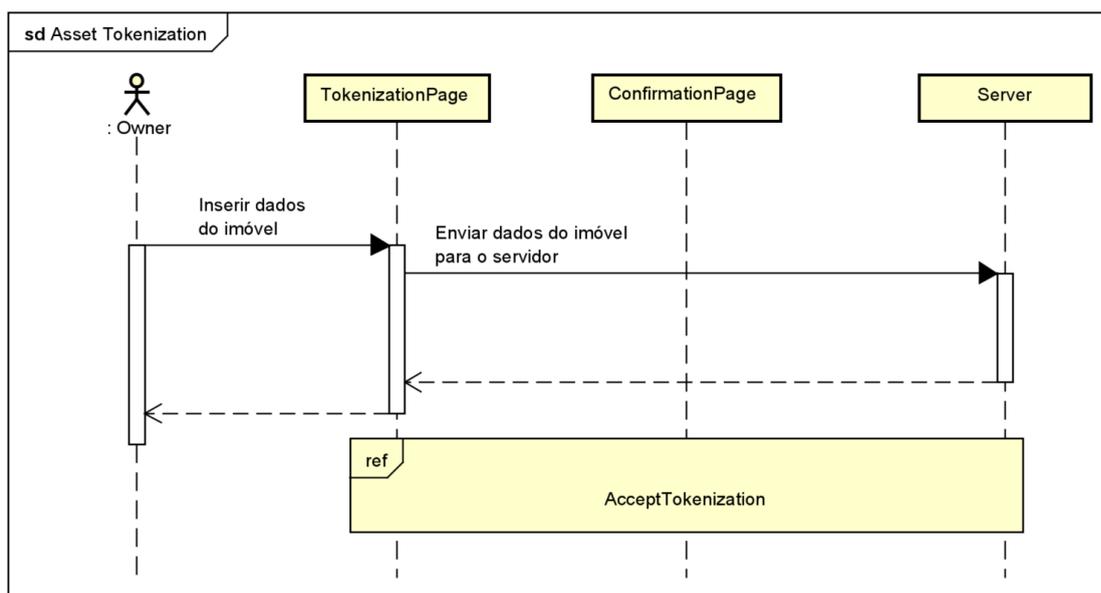
#### 4.5.3 Validação de documentos e tokenização de um imóvel

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

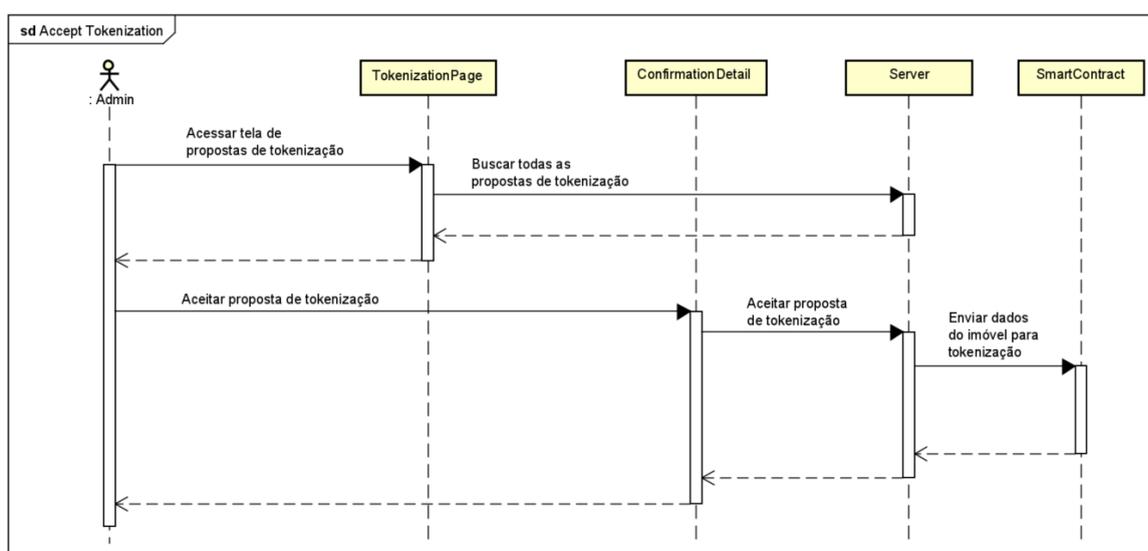
Neste caso de uso, o proprietário do imóvel irá interagir com uma tela no *Front-End* e irá inserir os dados de endereço, área útil, número de registro em cartório e um documento identificando a posse do ativo imobiliário físico. Estas informações serão enviadas para o servidor e ficarão disponíveis 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 5 e 6 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 5 - Diagrama de sequência para a tokenização de um imóvel**



**Figura 6 - 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 realizado por fora da plataforma desenvolvida, com apenas seu registro realizado na plataforma). 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 7, 8 e 9, 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.

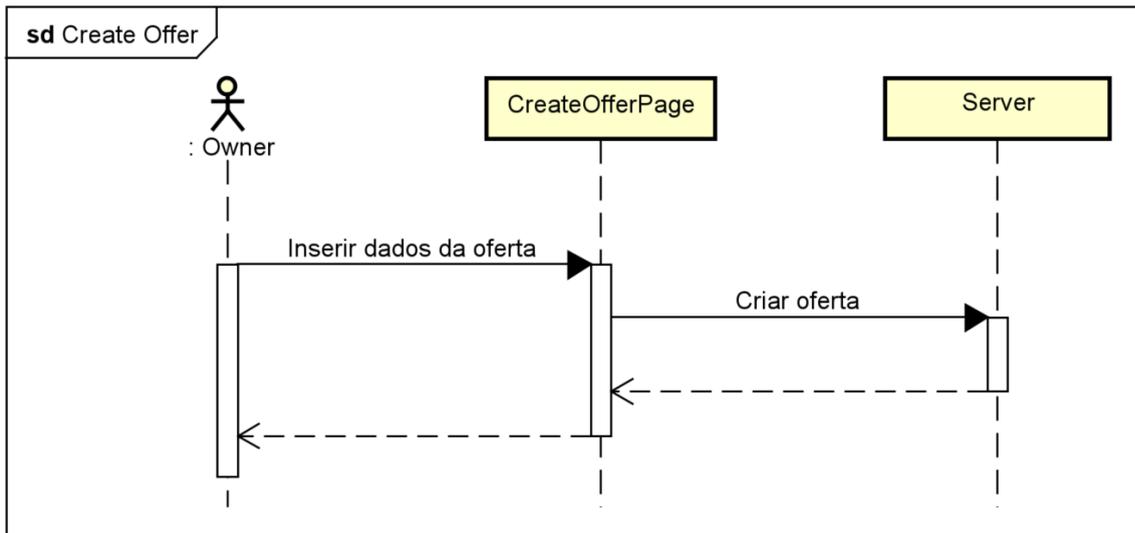


Figura 7 - Diagrama de seqüência para a criação de oferta de porcentagem de ativo tokenizado

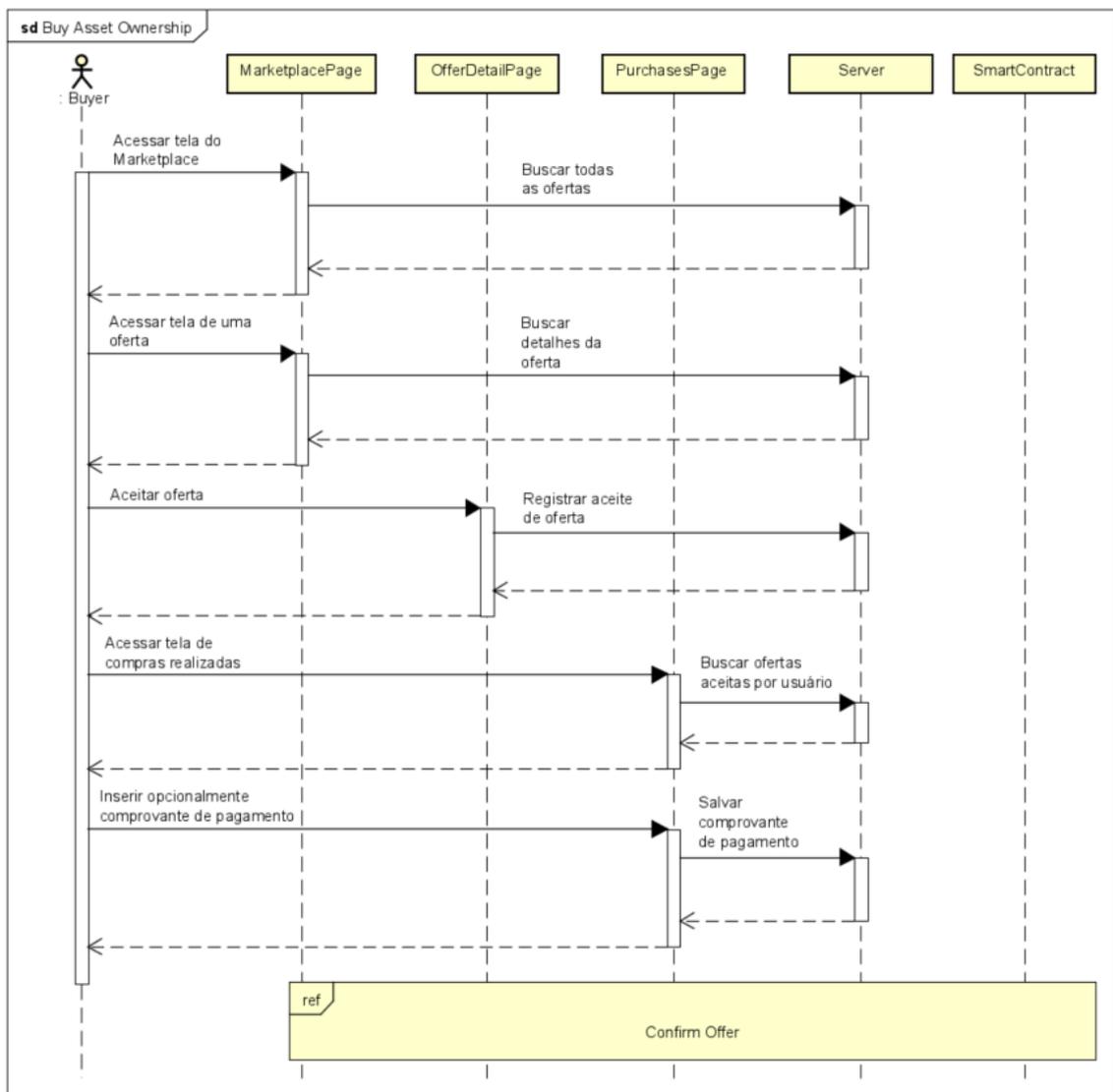


Figura 8 - Diagrama de seqüência para aceite de oferta de ativo imobiliário tokenizado

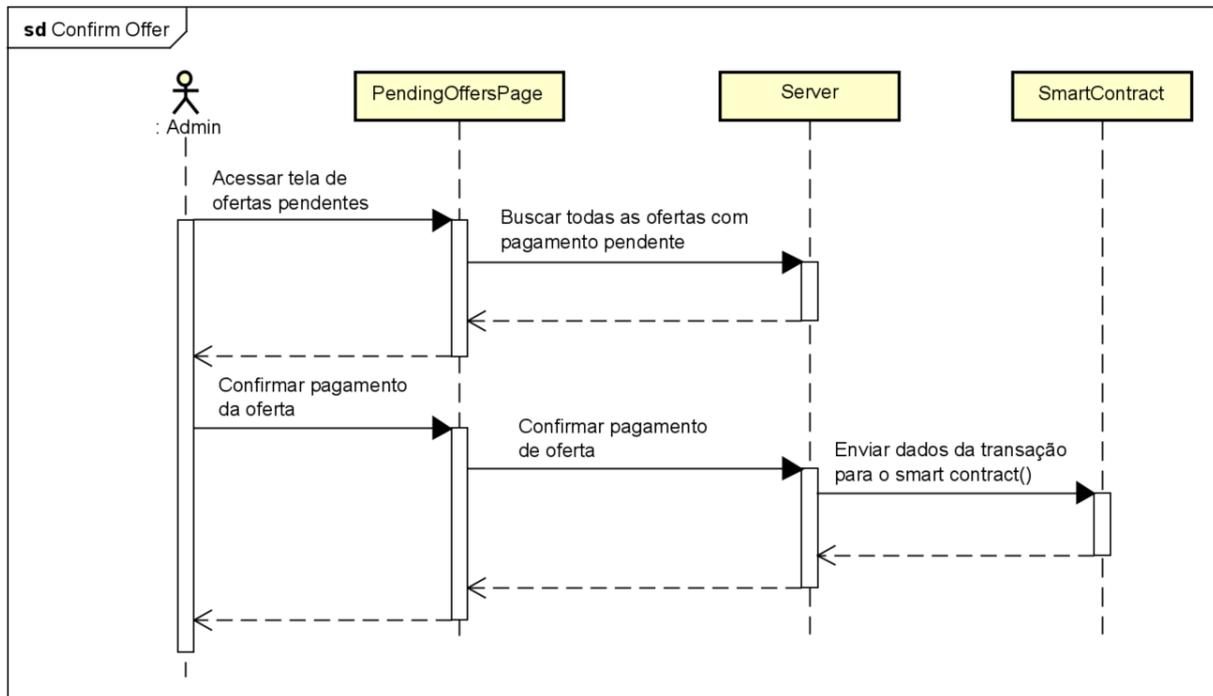


Figura 9 - Diagrama de sequência para a confirmação de compra de ativo imobiliário

#### 4.5.5 Registro de garantia de empréstimo

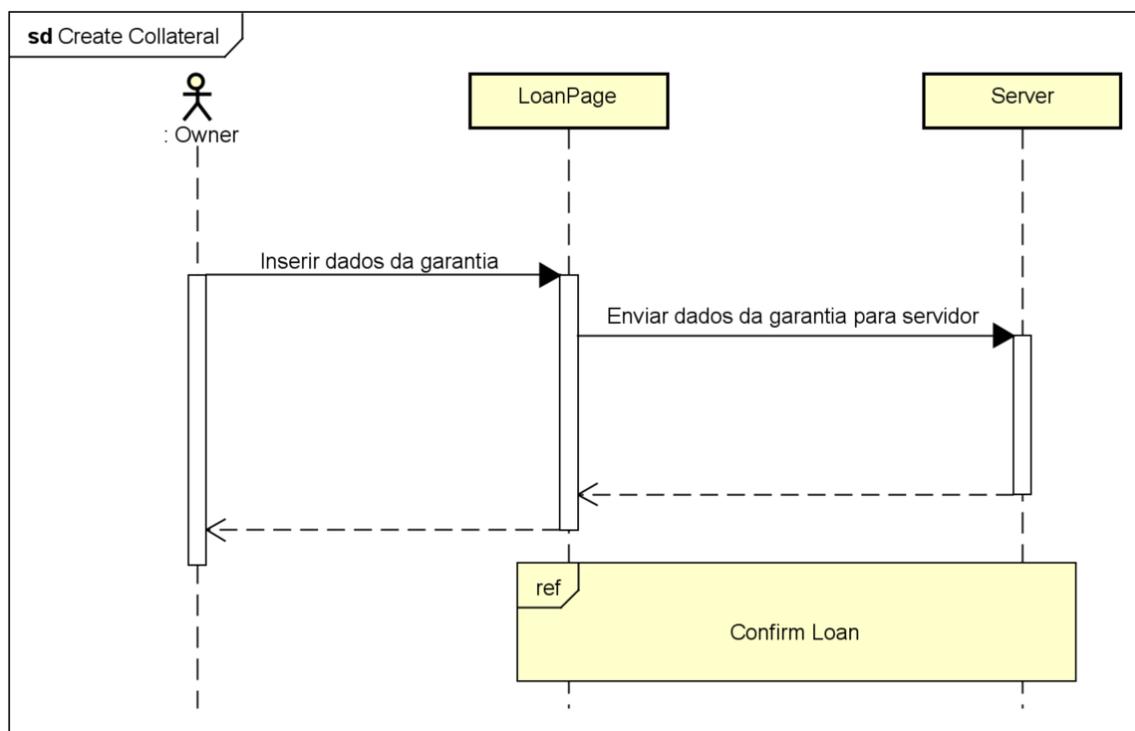
Para o cenário de garantia de empréstimo, o dono do ativo imobiliário tokenizado será o devedor e irá acordar os termos do empréstimo com o credor, seja ele um banco ou uma pessoa física.

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.

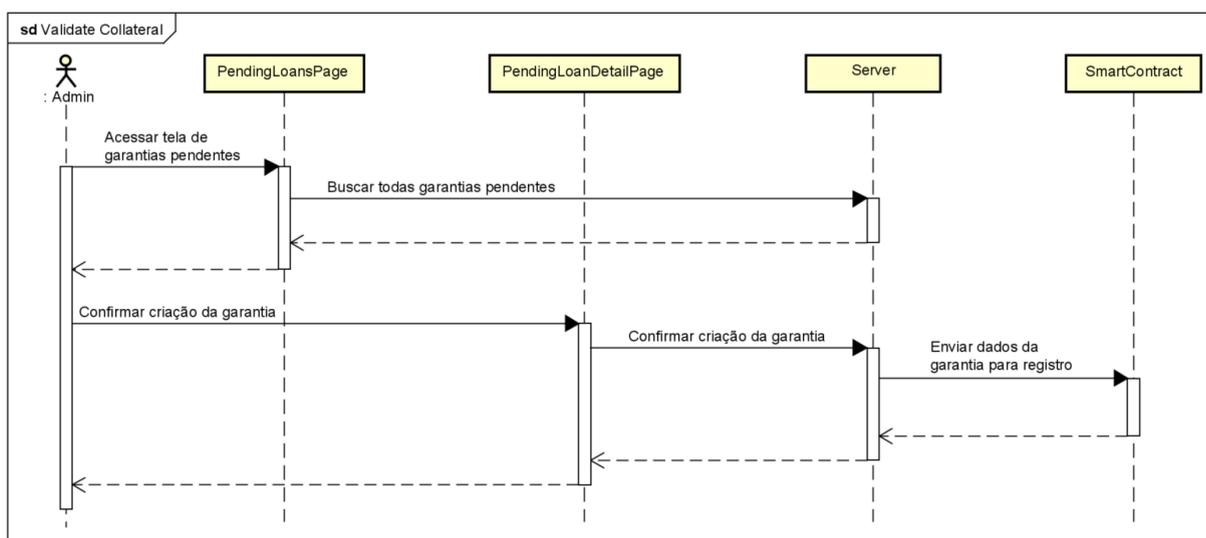
O administrador, por sua vez, irá verificar as informações da garantia do empréstimo e confirmar o valor emprestado pelo credor ao devedor (acordado por fora da plataforma), e então irá acessar a página de garantias pendentes da plataforma, onde deverá confirmar o registro da garantia para que os dados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

As Figuras 10 e 11 apresentam 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 10 - Diagrama de sequência para criação de empréstimo**



**Figura 11 - 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 12 e 13 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.

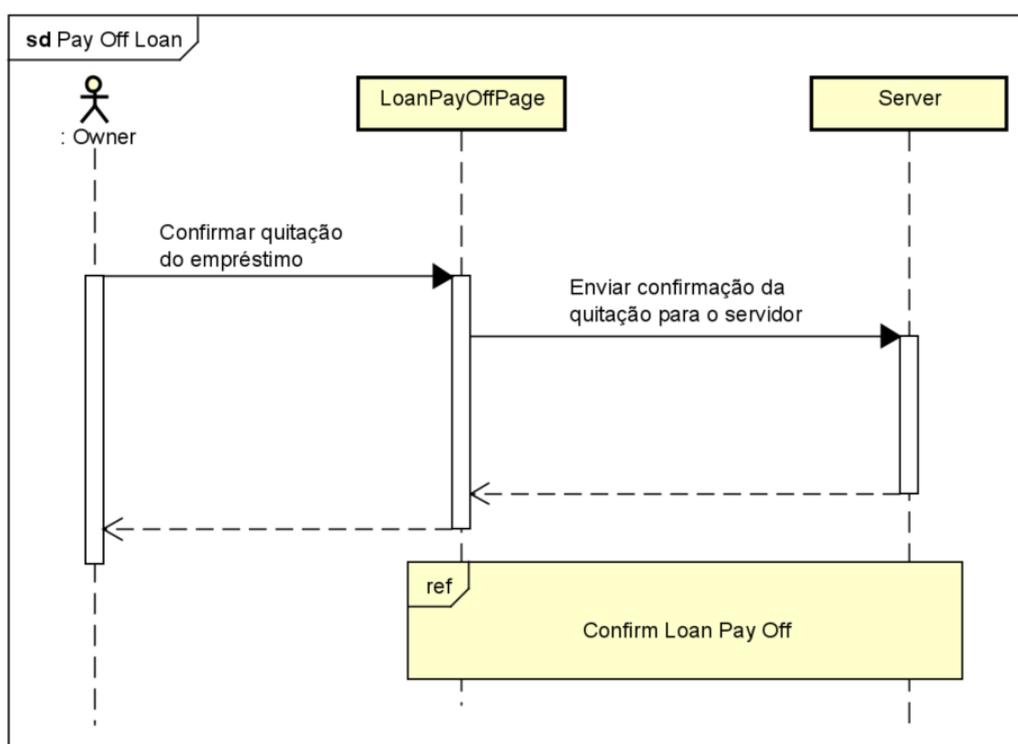
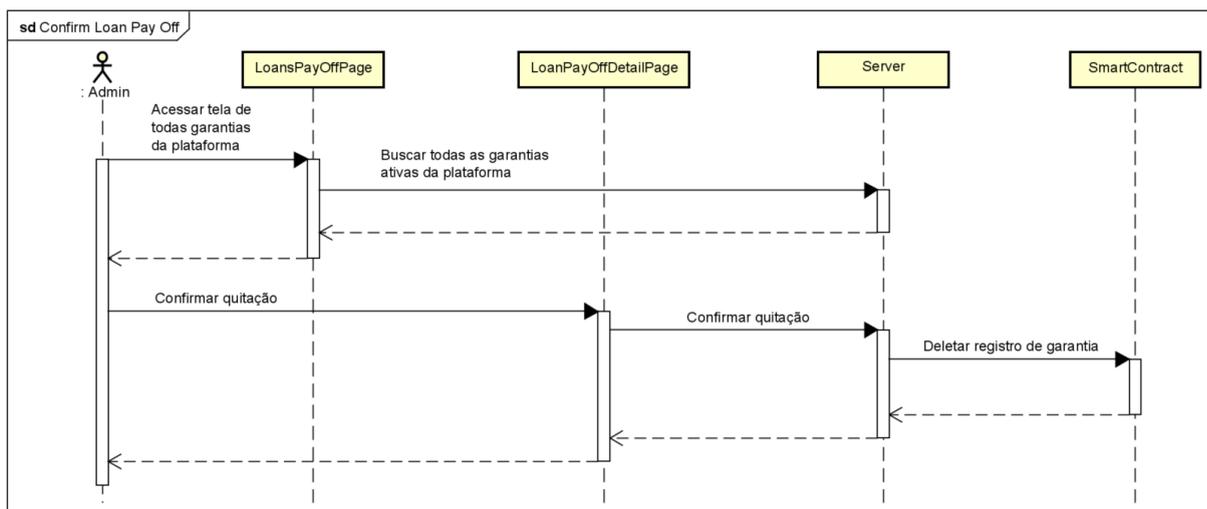


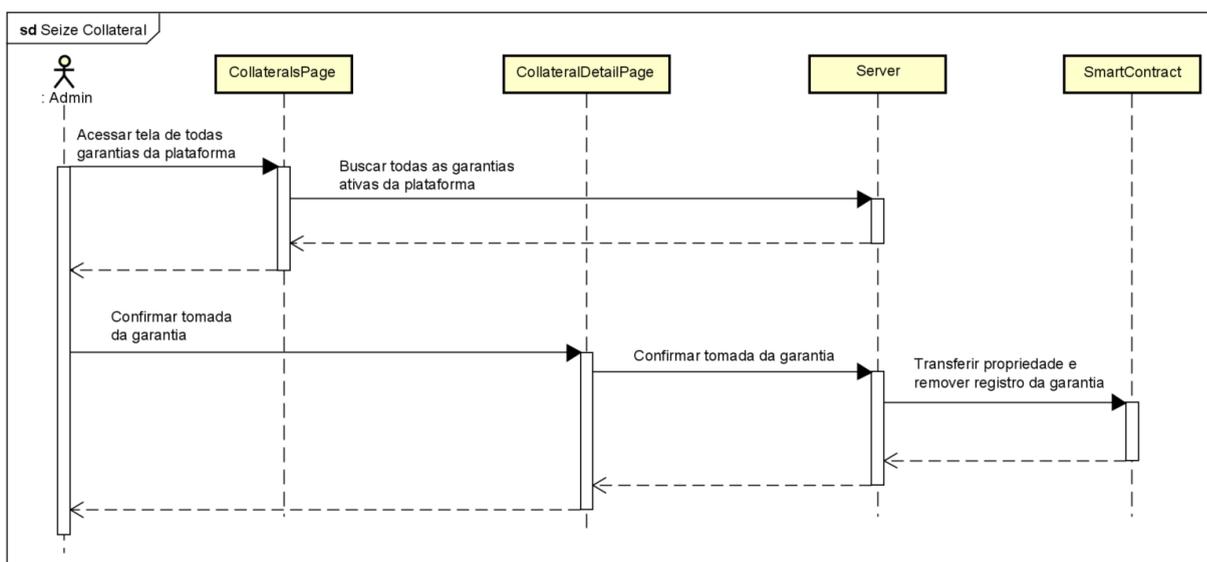
Figura 12 - Diagrama de sequência para registro de quitação de empréstimo



**Figura 13 - 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 confirmar a tomada da garantia do empréstimo pelo credor.

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 14.



**Figura 14 - 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 15 e 16 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 do pagamento de aluguel entre todos os proprietários do ativo imobiliário tokenizado.

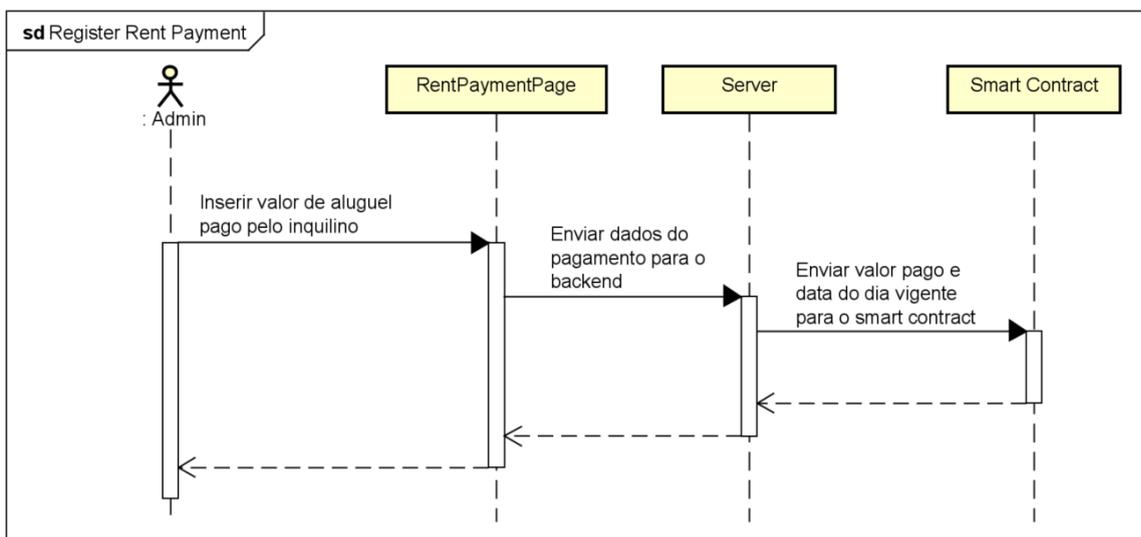


Figura 15 - Diagrama de sequência para registro de pagamento de aluguéis

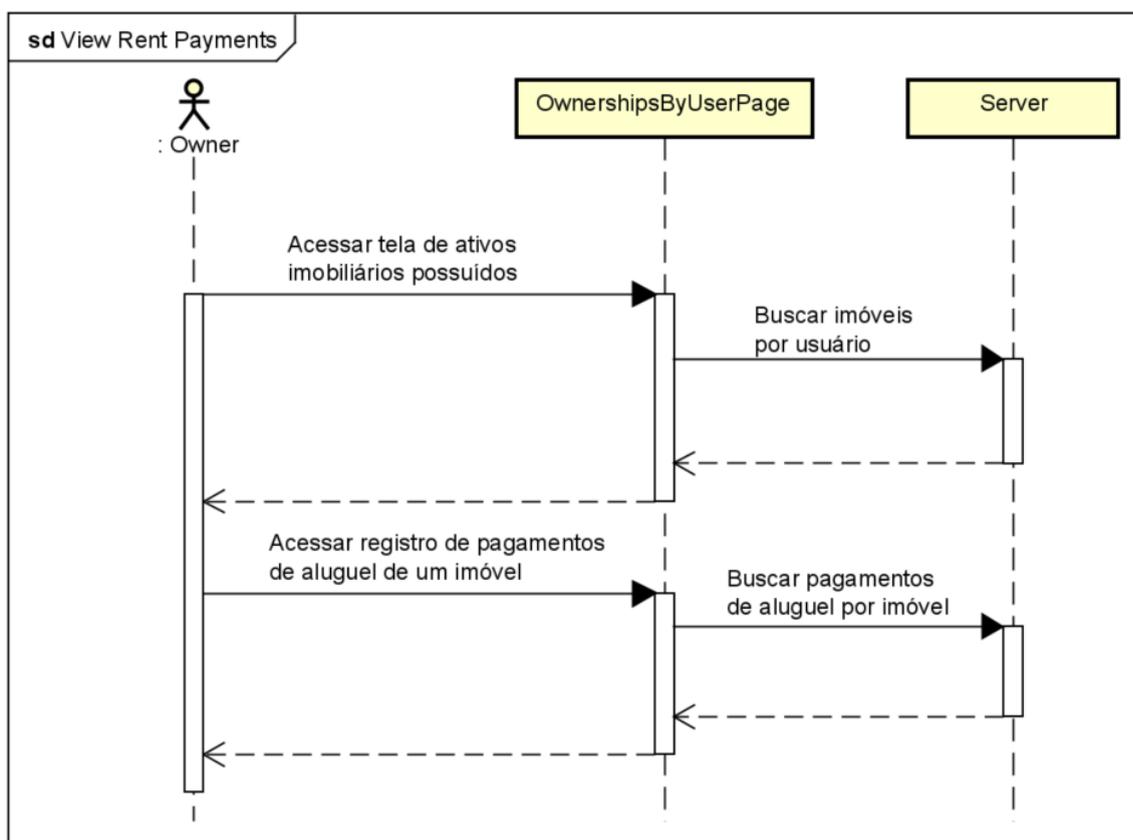
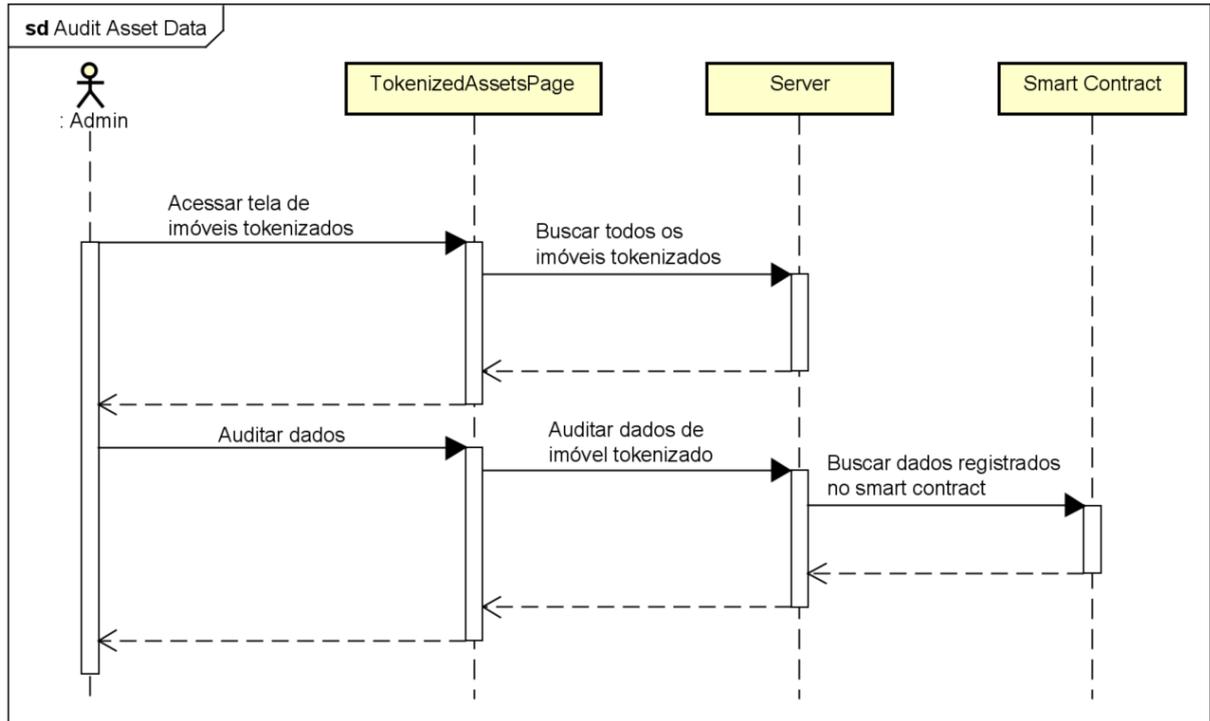


Figura 16 - Diagrama de sequência para visualização dos pagamentos de aluguéis recebidos por um proprietário

#### 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 na *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 17 - Diagrama de sequência para auditoria dos dados do imóvel na blockchain**

## 5 DESENVOLVIMENTO DO TRABALHO

Nesta seção, serão detalhadas as premissas e restrições adotadas para o desenvolvimento do projeto, bem como serão descritas as tecnologias utilizadas durante o desenvolvimento da plataforma. Em seguida, serão abordados de forma mais aprofundada os detalhes da implementação de cada um dos componentes da plataforma desenvolvida e, por fim, serão listados os diversos testes realizados para a validação das funcionalidades e dos casos de uso mapeados nas seções anteriores.

### 5.1 PREMISSAS E RESTRIÇÕES

Para a implementação do projeto, é necessário que sejam feitas suposições com relação ao funcionamento e o escopo do projeto, e esta seção tem o intuito de listar e explicá-las.

Primeiramente, como descrito nas seções 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. 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 propriedade efetiva do imóvel (também referida como posse efetiva), 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 ao proprietário legal do imóvel, que está indicado no registro do

imóvel no cartório de imóveis da região. 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. No caso da transferência da posse efetiva entre indivíduos, entende-se que são necessários procedimentos cartoriais para efetivar a operação além da transferência pela plataforma, porém estes não foram englobados no escopo do projeto.

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 [32], 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. Dentro deste contexto, destaca-se a premissa de que as informações dos imóveis não poderão ser tokenizadas por fora da plataforma desenvolvida. Dessa forma, caberá ao administrador verificar a validade das informações do ativo e garantir que o imóvel já não tenha sido tokenizado em alguma blockchain externa à plataforma.

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.

## 5.2 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.

### 5.2.1 Node.js

Node.js [33] é um ambiente de execução *open-source* em JavaScript, idealizado para o desenvolvimento de aplicações escaláveis do lado do servidor. No projeto, foi usado na implementação do *Back-End*, por meio do framework NestJS [34], 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.

### 5.2.2 React

*React* [35] é uma biblioteca *open-source* de JavaScript usada para desenvolvimento de interfaces para usuários baseada em componentes *UI* (*User Interface*, ou Interface de Usuário). Foi 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.

### 5.2.3 PostgreSQL

*PostgreSQL* [36] é um banco de dados relacional *open-source* que utiliza e estende a linguagem SQL, e foi utilizado para armazenar os dados de ativos, usuários e transações na plataforma.

#### 5.2.4 Solidity

*Solidity* [37] 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. É através desta linguagem de programação que foram 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* desenvolvido.

#### 5.2.5 Remix IDE

*Remix IDE* [38] 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*. É através deste ambiente de desenvolvimento que foram desenvolvidas as principais funcionalidades do projeto envolvendo *blockchain* e *smart contracts*.

#### 5.2.6 Web3.js

Web3.js [39] é uma coleção de bibliotecas JavaScript que implementam diversas funcionalidades de interação com nós de uma rede *blockchain* Ethereum. Esta coleção de bibliotecas foi 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.

#### 5.2.7 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.

Além disso, também foi utilizado no projeto o *Etherscan* [40], 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.

### 5.2.8 TypeORM

TypeORM [41] é um ORM (*Object Relational Mapping*, ou mapeador objeto-relacional) compatível com aplicações desenvolvidas 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.

## 5.3 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.

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

### 5.3.1 Desenvolvimento do *Smart Contract*

Para a representação dos imóveis na forma de tokens na *blockchain*, foi desenvolvido um *smart contract* adaptando os padrões ERC-20 [27] e ERC-721 [28], que são amplamente 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.

#### 5.3.1.1 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<sup>2</sup>) 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 do 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.

#### 5.3.1.2 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<sup>2</sup>) 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.

#### 5.3.1.3 Publicação e implantação do *smart contract* na *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 a *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* na *blockchain* e, portanto, a plataforma possui sua própria carteira na *blockchain* e utiliza seus próprios recursos (no caso da rede de teste Ethereum, a moeda Ether, de sigla ETH) para enviar tais transações para a *blockchain*.

Esta comunicação com a *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 [42], e através dele assina e envia todas as transações necessárias nos fluxos dos casos de uso implementados para a *blockchain* utilizado.

#### 5.3.2 Estrutura do banco de dados relacional

A Figura 18 mostra a modelagem do banco de dados, na qual é possível visualizar as tabelas de entidades e seus relacionamentos com as demais.

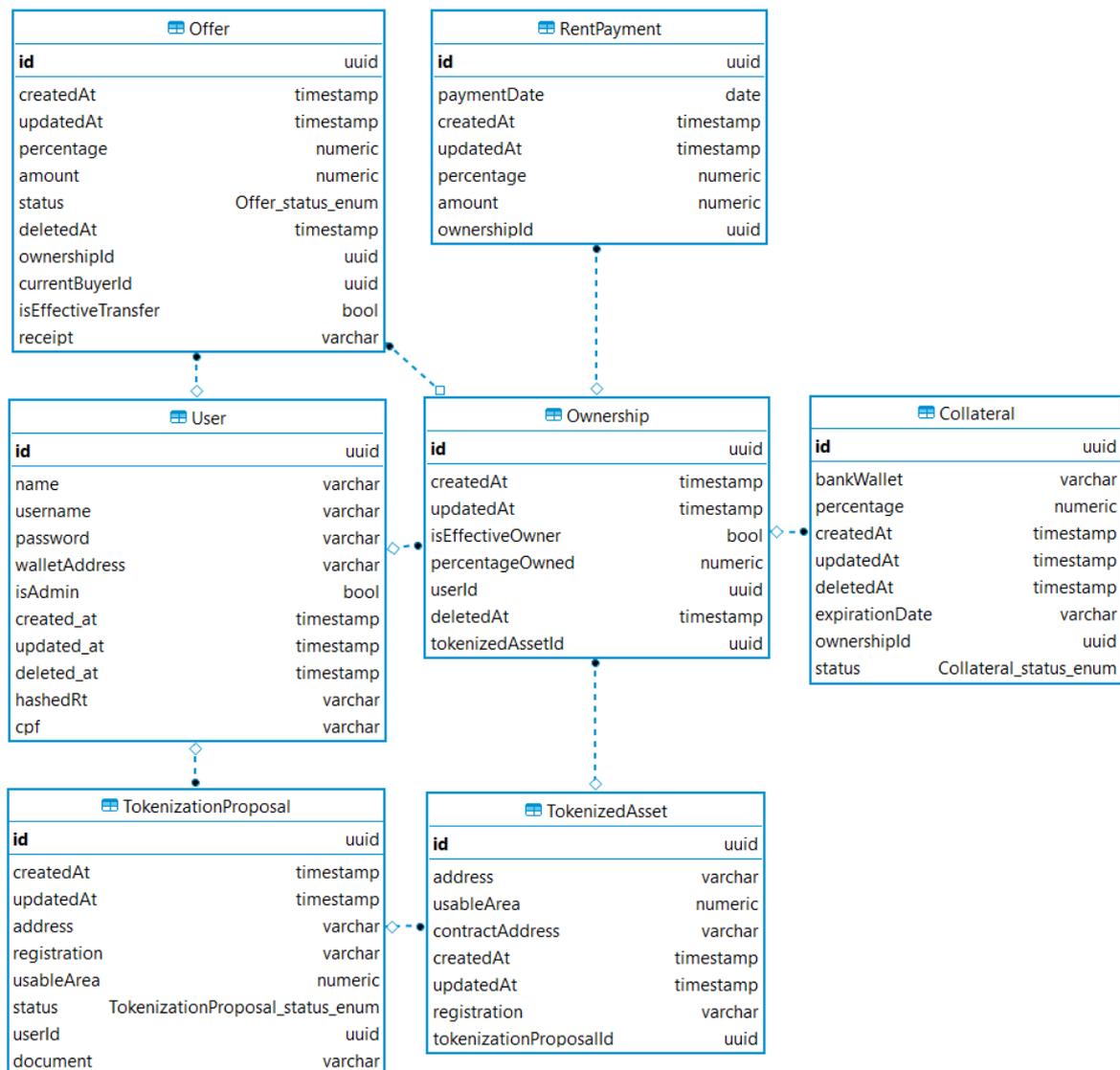


Figura 18 - Modelagem do banco de dados do projeto

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 (em moeda fiduciária), 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

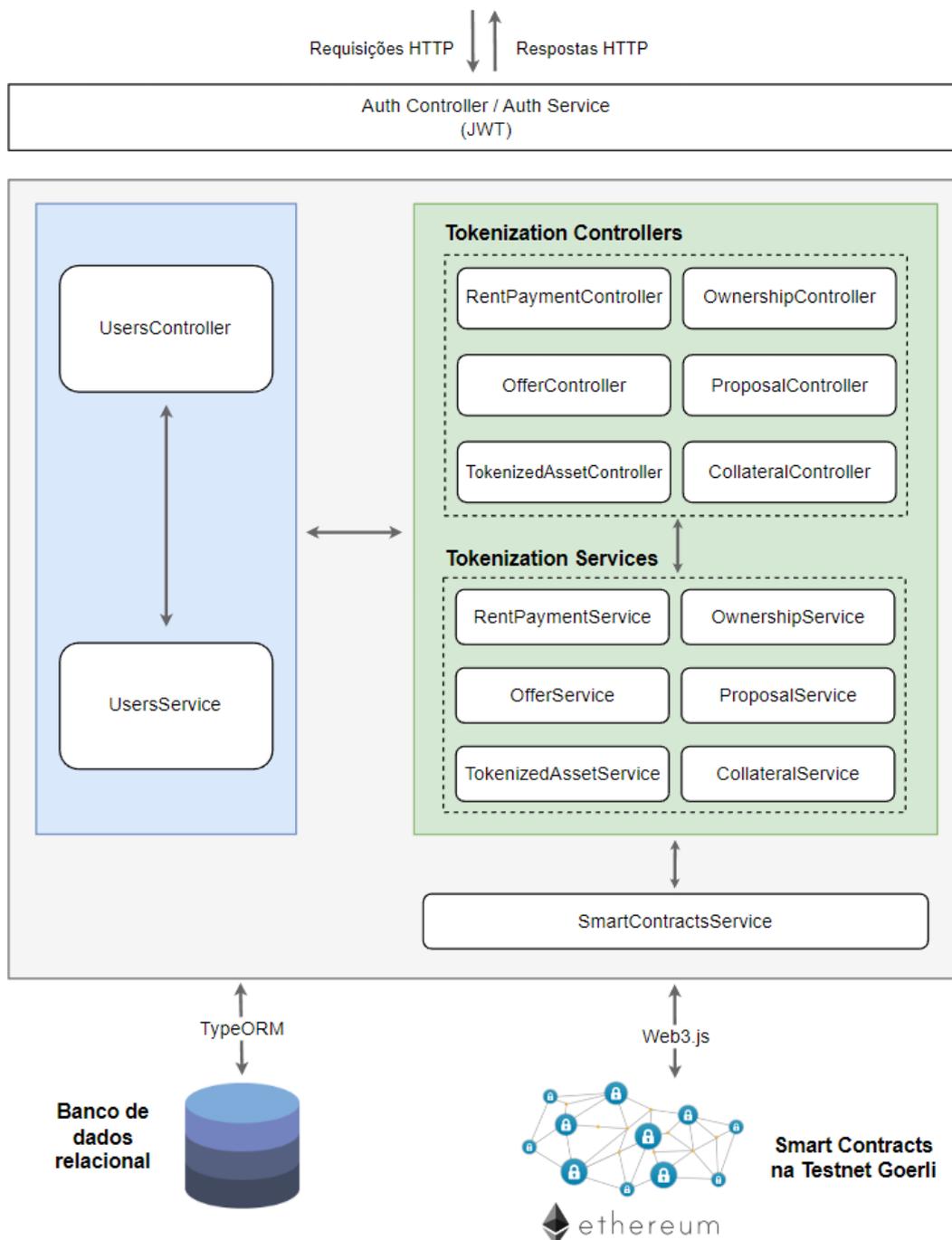
imóvel será salvo um registro com o valor pago proporcional a sua participação, o percentual do valor total do aluguel e a data do pagamento.

### 5.3.3 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 GIT (<https://github.com/lsantos0142/asset-tokenization-back>), no qual os integrantes do grupo puderam trabalhar simultaneamente. Ao fim da implementação do servidor da plataforma, o código foi implantado através do serviço de hospedagem de sistemas *Back-End* da empresa Railway [43] utilizando-se diretamente do código presente no repositório GIT.

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; 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 19 - Estrutura de Controladores e serviços do servidor**

A Figura 19 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 e autorização baseadas 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 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.

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 framework utilizado para o desenvolvimento do servidor permite gerar a documentação das rotas da *API* de forma automática, utilizando o módulo do *Swagger* [44]. 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.

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 20 - 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 21 - Documentação do Swagger para as rotas de Collaterals, Offers, Rent Payments e Authentication**

### 5.3.4 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 foi estruturado de modo a separar as telas referentes às funcionalidades em pastas contendo os códigos-fonte específicos de cada uma das funcionalidades.

A aplicação foi criada em um repositório GIT (<https://github.com/lsantos0142/asset-tokenization-front>), no qual os integrantes do grupo puderam trabalhar simultaneamente.

Nas próximas seções serão detalhadas as diversas telas desenvolvidas para a plataforma e que, ao final do desenvolvimento, foram implantadas publicamente através do serviço de hospedagem de sistemas *Front-End* da empresa Vercel [45] utilizando-se diretamente do código presente no repositório do GIT.

#### 5.3.4.1 Login e cadastro

A tela inicial da plataforma é a de *login* e cadastro de usuário (Figura 22), 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.

Tokenização de Ativos Imobiliários

**Fazer Login**

Usuário \*  
Digite seu usuário

Senha \*  
Digite sua senha

Fazer Login

OU

**Registrar**

Nome \*  
Digite seu nome

Usuário \*  
Digite seu usuário

CPF \*  
Digite seu CPF

Senha \*  
Digite sua senha

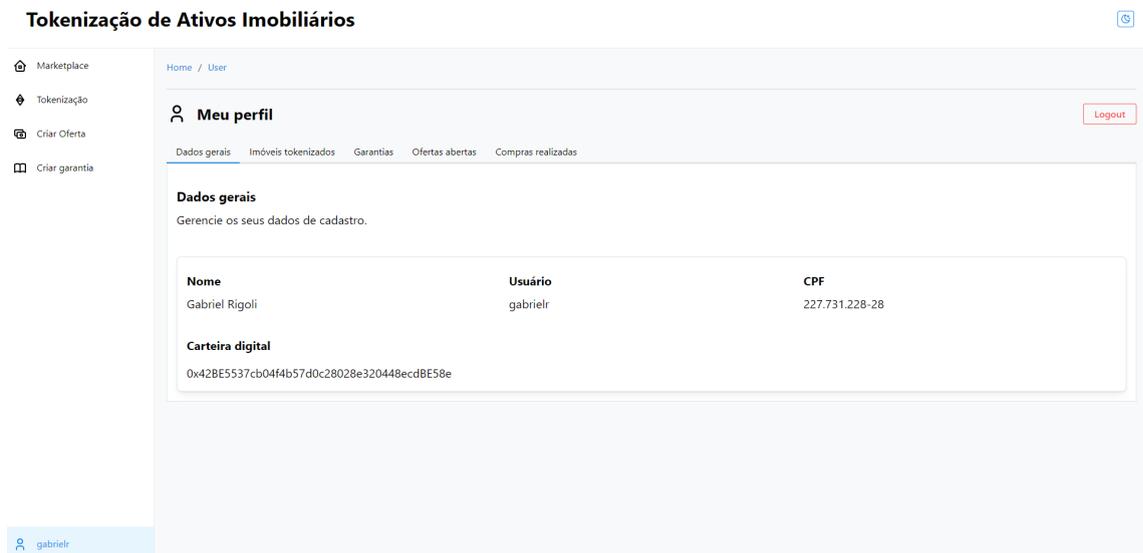
Registrar

Figura 22 - Tela de login e cadastro de usuário

### 5.3.4.2 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 23) 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 23 - Aba de dados gerais da conta**

A segunda aba (Figura 24) 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 exibidos em sua parte inferior. Nesta tela, o botão apresentado tem a função

de abrir um *modal* (Figura 25) que mostra um registro de todos os pagamentos de alugueis já recebidos enquanto proprietário do imóvel.

The screenshot shows the 'Tokenização de Ativos Imobiliários' user profile page. The user is logged in as 'joaosilva'. The page displays three tokenized properties:

Propriedade	Área Útil	Número do Registro	Porcentagem de Posse	Porcentagem disponível para operações
Rua Embuaçu, 499 - Vila Mariana	250 m <sup>2</sup>	23146	100 %	40 %
R. Santa Cruz, 776 - Vila Mariana	130 m <sup>2</sup>	33785	100 %	100 %
Avenida Pompeia, 227 - Pompeia	110 m <sup>2</sup>	26734	100 %	25,9 %

Each property card includes a 'DONO' status, a 'Visualizar no Etherscan' link, and a 'Aluguéis recebidos' button.

Figura 24 - Aba de imóveis tokenizados do usuário

The screenshot shows the same user profile page as Figure 24, but with a modal open over the 'Aluguéis recebidos' button for the property 'Rua Embuaçu, 499 - Vila Mariana'. The modal displays the following data:

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

Figura 25 - Registro dos alugueis recebidos enquanto proprietário do imóvel

A terceira aba da página de perfil do usuário (Figura 26) mostra todas as garantias de empréstimos que o usuário possui registrada 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.

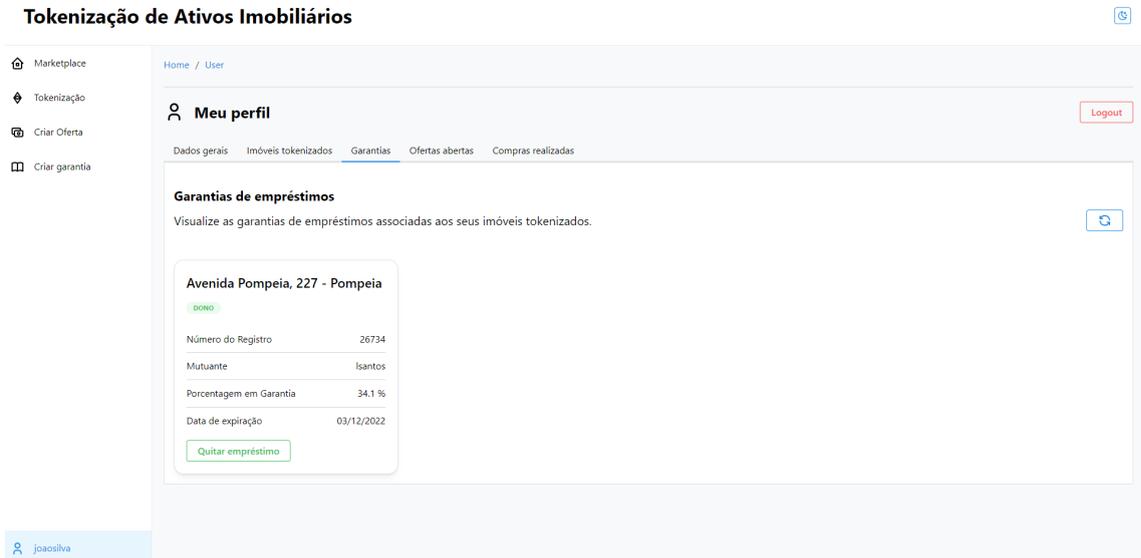


Figura 26 - Aba de garantias do usuário

A quarta aba da página de perfil do usuário (Figura 27) 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.

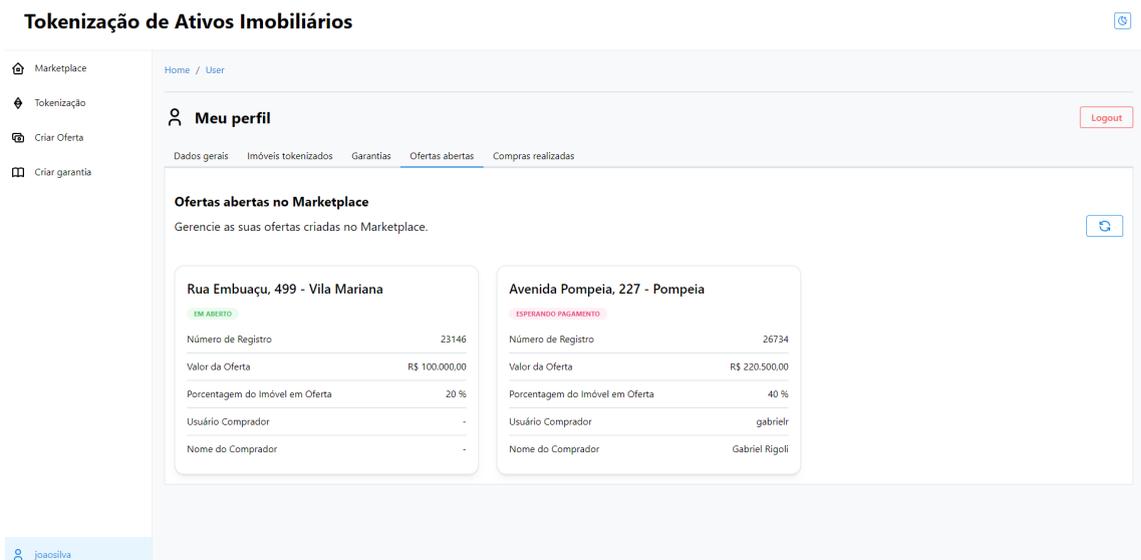
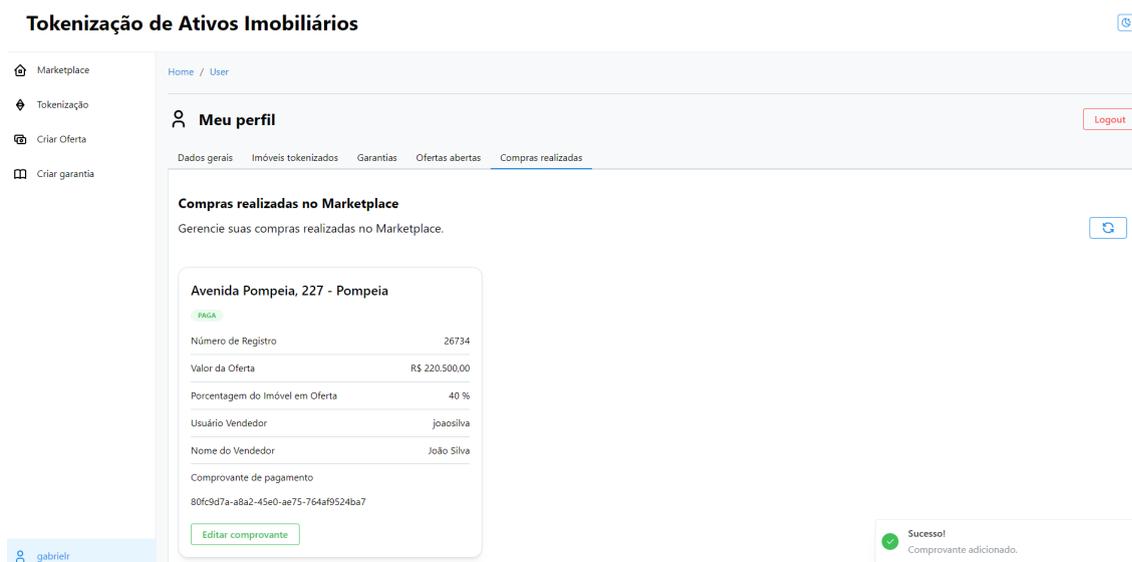


Figura 27 - Aba de ofertas do usuário abertas no Marketplace

Por fim, a quinta aba da tela do perfil do usuário (Figura 28) 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 28 - Aba de compras realizadas no Marketplace pelo usuário**

#### 5.3.4.3 Tokenização de imóvel

Ao selecionar a opção “Tokenização” no menu lateral, o usuário é redirecionado para uma tela (Figura 29) 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 a chave pública da carteira digital cadastrada em sua conta.

**Tokenização de Ativos Imobiliários**

Marketplace / Home / Criar Proposta de Tokenização

**Tokenize o seu imóvel**

Insira os dados do seu imóvel para análise da plataforma. Caso os dados sejam aprovados, você receberá em sua carteira digital o imóvel tokenizado.

Endereço do Imóvel \*  
Ex: Av. Prof. Luciano Gualberto, 390

Área Útil do Imóvel (m²) \*  
Ex: 100

Número do Registro em Cartório \*  
Ex: 27

Endereço da Carteira do Dono Efetivo \*  
0x428e5537cd04f657d0c28028e32044becd8e58e

Documentos do imóvel (Ex.: Escrituras, comprovante de posse, etc) \*  
Selecionar arquivo

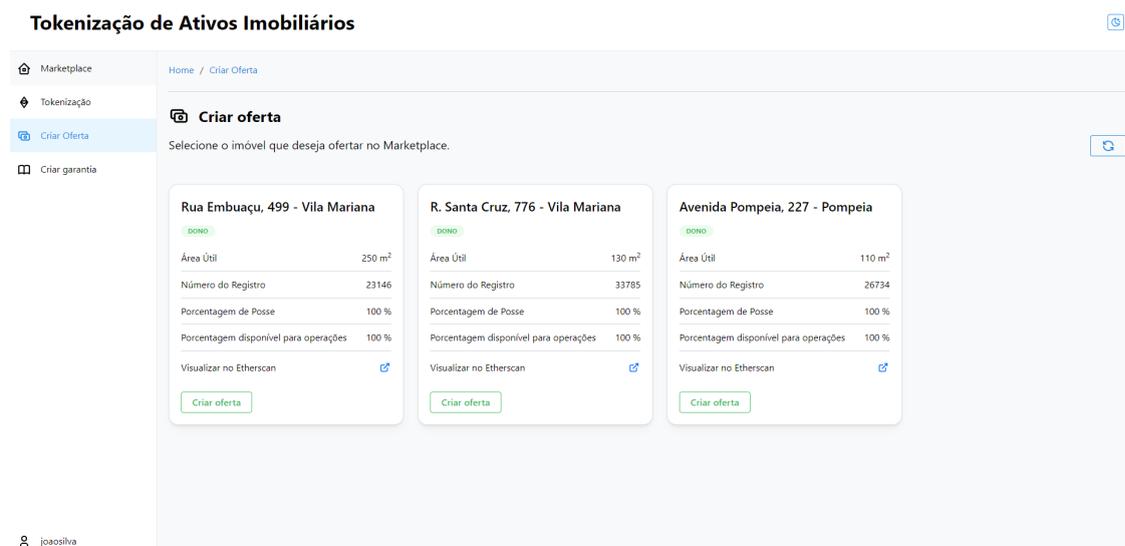
Enviar dados do imóvel

gabriel

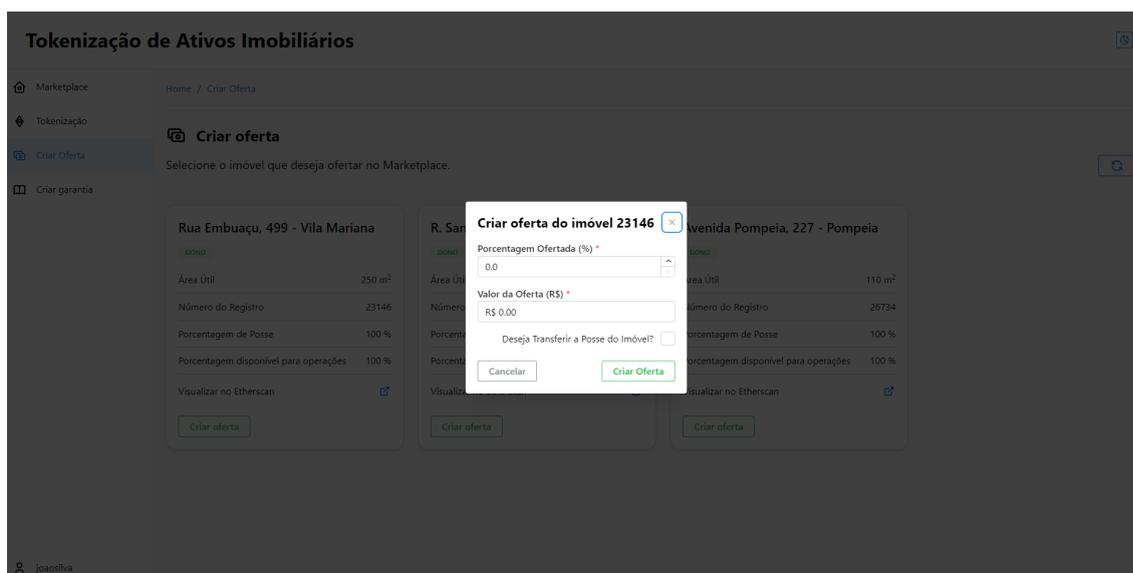
**Figura 29 - Formulário de envio dos dados do imóvel para tokenização**

#### 5.3.4.4 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 30), serão listados em *cards* todos os imóveis do usuário, cada um possuindo um botão que abre um modal (Figura 31) 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 30 - Tela de listagem dos imóveis disponíveis para criação de oferta**



**Figura 31 - Formulário de criação de oferta no Marketplace**

### 5.3.4.5 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 32). 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.

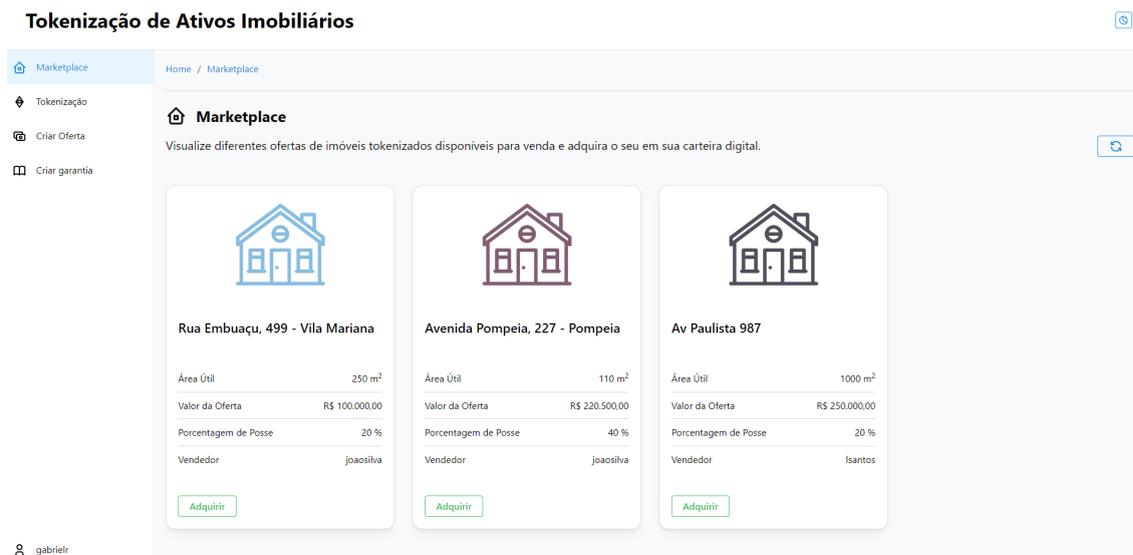


Figura 32 - Marketplace de imóveis da plataforma

Clicando no botão de “Adquirir”, o usuário é redirecionado para a tela de detalhes da oferta (Figura 33), na qual são mostradas mais informações do 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.

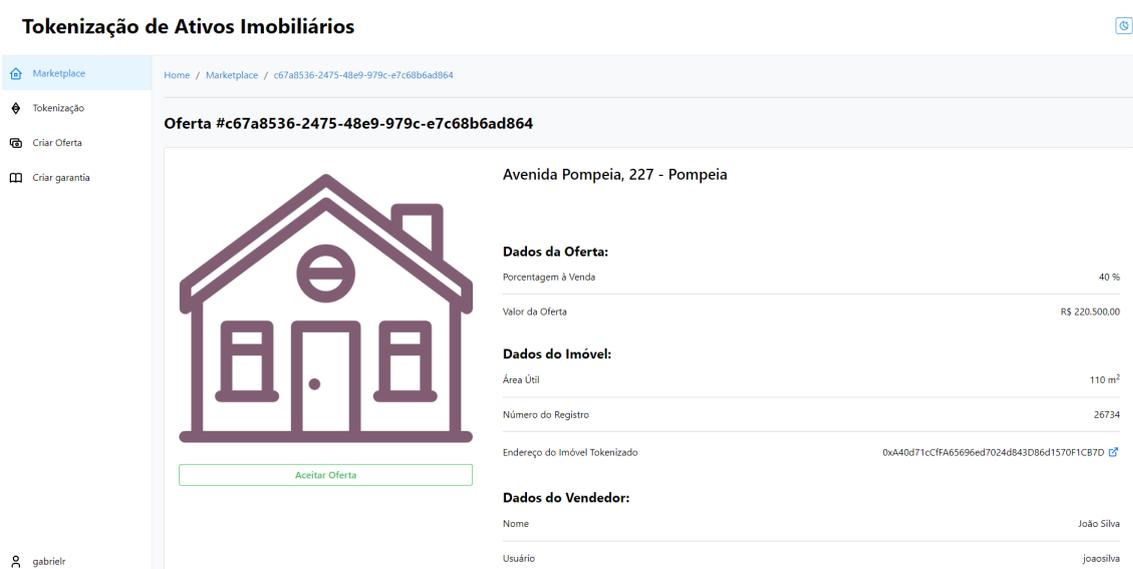


Figura 33 - Tela de detalhes da oferta do imóvel

### 5.3.4.6 Criar garantia de empréstimo

De forma semelhante à tela de criação de oferta, a tela de criação de garantia (Figura 34), 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 35) 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.

Imóvel	Área Útil	Número do Registro	Porcentagem de Posse	Porcentagem disponível para operações
Rua Embuaçu, 499 - Vila Mariana	250 m <sup>2</sup>	23146	100 %	100 %
R. Santa Cruz, 776 - Vila Mariana	130 m <sup>2</sup>	33785	100 %	100 %
Avenida Pompeia, 227 - Pompeia	110 m <sup>2</sup>	26734	100 %	100 %

Visualizar no Etherscan

Cadastrar garantia

**Figura 34 - Tela de listagem dos imóveis disponíveis para criação de garantia**

The image shows a web application interface for 'Tokenização de Ativos Imobiliários'. A modal window is open, titled 'Cadastrar imóvel 33785 como garantia de empréstimo'. The modal contains the following fields:

- Endereço do imóvel tokenizado \***: 0x268032581b76e0b881c333e6e88b0d42cAD971F0
- Data de expiração do empréstimo \***: Dezembro 3, 2022
- Nome de usuário do banco \***: Escolha um banco
- Porcentagem utilizada como garantia (%) \***: 0.0

Buttons for 'Cancelar' and 'Cadastrar Garantia' are visible at the bottom of the modal. The background shows a sidebar with 'Criar garantia' selected and a list of properties, including 'Rua Embuaçu, 499 - Vila Mariana'.

Figura 35 - Formulário de criação de garantia de empréstimo

#### 5.3.4.7 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 36) 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 é enviado 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.

**Tokenização de Ativos Imobiliários**

Marketplace  
Tokenização  
Criar Oferta  
Criar garantia

Home / Portal Admin

**Portal Admin**

Propostas de tokenização | Pagamentos de ofertas | Novas garantias | Quitações de empréstimos | Imóveis tokenizados

**Propostas de Tokenização Ativas**  
Gerencie as propostas de tokenização ativas, aceite ou rejeite de acordo com os dados fornecidos.

Rua Embuaçu, 499 - Vila Mariana	R. Santa Cruz, 776 - Vila Mariana	Avenida Pompeia, 227 - Pompeia
EM ABERTO	EM ABERTO	EM ABERTO
Nome do Usuário: joasilva	Nome do Usuário: joasilva	Nome do Usuário: joasilva
Nome: João Silva	Nome: João Silva	Nome: João Silva
CPF do Usuário: 337.731.228-28	CPF do Usuário: 337.731.228-28	CPF do Usuário: 337.731.228-28
Área útil: 250 m <sup>2</sup>	Área útil: 130 m <sup>2</sup>	Área útil: 110 m <sup>2</sup>
Número de Registro: 23146	Número de Registro: 33785	Número de Registro: 26734
Visualizar documentos	Visualizar documentos	Visualizar documentos
Rejeitar   Aceitar	Rejeitar   Aceitar	Rejeitar   Aceitar

userAdmin  
Portal Admin

**Figura 36 - Aba listando todas as propostas de tokenização da plataforma**

A segunda aba presente no portal Admin (Figura 37), por sua vez, é responsável por listar todas as ofertas que foram aceitas no Marketplace e que estão com o status pendente de validação do pagamento. Cada *card* contém as informações da oferta em andamento, podendo ou não exibir o comprovante de pagamento, caso o comprador tenha registrado-o 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.

**Tokenização de Ativos Imobiliários**

Marketplace  
Tokenização  
Criar Oferta  
Criar garantia

Home / Portal Admin

**Portal Admin**

Propostas de tokenização Pagamentos de ofertas Novas garantias Quitações de empréstimos Imóveis tokenizados

**Ofertas Esperando Confirmação de Pagamento**

Gerencie as ofertas com compradores já definidos. Caso a oferta tenha sido paga, valide o pagamento no respectivo card abaixo.

**Av Paulista 987**

ESPERANDO PAGAMENTO

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	-

Rejeitar
Confirmar

**Avenida Pompeia, 227 - Pompeia**

ESPERANDO PAGAMENTO

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	-

Rejeitar
Confirmar

userAdmin  
Portal Admin

**Figura 37 - Aba listando todos os pagamentos de ofertas pendentes de confirmação**

A terceira aba presente no portal Admin (Figura 38) 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.

**Tokenização de Ativos Imobiliários**

Marketplace  
Tokenização  
Criar Oferta  
Criar garantia

Home / Portal Admin

**Portal Admin**

Propostas de tokenização Pagamentos de ofertas **Novas garantias** Quitações de empréstimos Imóveis tokenizados

**Validação de novas garantias de empréstimo**  
Gerencie as novas garantias cadastradas pelos usuários no portal. Caso uma garantia tenha sido de fato acordada entre as partes envolvidas e esteja com os dados corretos, valide a criação em seu respectivo card.

Rua Embuaçu, 499 - Vila Mariana		Avenida Pompeia, 227 - Pompeia	
ESPERANDO CONFIRMAÇÃO			
Usuário Mutuário	joaosilva	Usuário Mutuário	joaosilva
Usuário Mutuante	cleiton123	Usuário Mutuante	lsantos
Porcentagem em Garantia	40 %	Porcentagem em Garantia	34.1 %
Data de Expiração do Empréstimo	03/12/2022	Data de Expiração do Empréstimo	03/12/2022
Número de Registro do Imóvel	23146	Número de Registro do Imóvel	26734
<input type="button" value="Rejeitar"/> <input type="button" value="Confirmar"/>		<input type="button" value="Rejeitar"/> <input type="button" value="Confirmar"/>	

userAdmin  
Portal Admin

**Figura 38 - Aba listando todos os cadastros de garantia de empréstimos pendentes na plataforma**

A quarta aba do portal do administrador (Figura 39) exibe 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 fluxos de quitação e tomada de garantia descrito na seção 4.5.6.

### Tokenização de Ativos Imobiliários

Home / Portal Admin

**Portal Admin**

Propostas de tokenização Pagamentos de ofertas Novas garantias Quitações de empréstimos Imóveis tokenizados

**Empréstimos Esperando Confirmação de Pagamento**

Valide os pagamentos de empréstimos com imóveis tokenizados como garantia. Caso o pagamento tenha sido feito, confirme para desalienar o imóvel da operação, caso contrário, se a data de expiração for atingida, confirme a tomada da garantia pelo banco.

Avenida Pompeia, 227 - Pompeia		Rua Mauro 32974	
Usuário Mutuário	joaosilva	Usuário Mutuário	lsantos
Usuário Mutuante	lsantos	Usuário Mutuante	gabrielr
Porcentagem em Garantia	34.1 %	Porcentagem em Garantia	15 %
Data de Expiração do Empréstimo	03/12/2022	Data de Expiração do Empréstimo	08/12/2022
Número de Registro do Imóvel	26734	Número de Registro do Imóvel	34

Tomada de posse Confirmar Pagamento

**Figura 39 - Aba listando todas as garantias de empréstimo registradas na plataforma**

A última aba presente no portal Admin (Figura 40) apresenta uma tabela contendo os dados de todos os imóveis tokenizados da plataforma. Para cada um deles, são disponibilizadas as ações de “Auditar” e “Registrar aluguel”.

### Tokenização de Ativos Imobiliários

Home / Portal Admin

**Portal Admin**

Propostas de tokenização Pagamentos de ofertas Novas garantias Quitações de empréstimos Imóveis tokenizados

**Gestor de imóveis tokenizados**

Gerencie todos os imóveis tokenizados na plataforma.

Id	Endereço	Nº registro	Área útil	Endereço na blockchain	Ações
f0973eca-85cf-435e-b5c4-18b478681e8a	Av Jabaquara 12345	2167	2183791 m <sup>2</sup>	0x983dC5d11b182b3b826275286f490151f862584a	<a href="#">Auditar</a> <a href="#">Registrar aluguel</a>
f00723f8-493c-41d2-b70c-e6ece5291c11	Av Paulista 987	123	1000 m <sup>2</sup>	0x1ACFF485be7e62427177a19ca106981e1C40D7d	<a href="#">Auditar</a> <a href="#">Registrar aluguel</a>
5ec4d177-0101-46ef-b035-0eaccc81a08	Rua Mauro 32974	34	1243 m <sup>2</sup>	0x657887931bd7949f01e45219471D6ED8259201d8	<a href="#">Auditar</a> <a href="#">Registrar aluguel</a>
98c7fc9f-3c86-4698-a6b7-2572ad5e126c	Rua Embuaçu, 499 - Vila Mariana	23146	250 m <sup>2</sup>	0x01592a70Ace9Aa16D991955b15140bFa2AD0E7c8	<a href="#">Auditar</a> <a href="#">Registrar aluguel</a>
80149341-d046-43f1-820d-7531cf7bd7a7	R. Santa Cruz, 776 - Vila Mariana	33785	130 m <sup>2</sup>	0x28B032581b76E0b881C333e8E88B042cAD971f0	<a href="#">Auditar</a> <a href="#">Registrar aluguel</a>
3ba63a26-4ba5-4d94-a2cc-71d8a00bca6a	Avenida Pompeia, 227 - Pompeia	26734	110 m <sup>2</sup>	0xA40d71ccfA65696ed7024d843D86d1570f1CB7D	<a href="#">Auditar</a> <a href="#">Registrar aluguel</a>
c91dd675-348f-4f99-8a94-d8a25c5202e	Rua Guaçuá, 117 - Saúde	45881	220 m <sup>2</sup>	0x7f921E48710d9A60Fdc8a6a684136D1682DFE29e	<a href="#">Auditar</a> <a href="#">Registrar aluguel</a>

**Figura 40 - Aba listando todos os imóveis tokenizados da plataforma**

A primeira ação é responsável por abrir um modal (Figura 41) 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.

**Dados do imóvel registrados em Blockchain**  
Endereço do Smart Contract: 0xA40d71cCFA65696e7024d843D86a1570f1CB7D

**Nome do proprietário: João Silva** PROPRIETÁRIO DETIVO  
CPF: 337.731.229-28  
Percentual do imóvel: 60%  
Endereço da carteira: 0x7E2C0E4Ac30885253118d79c998124E83a0556

**Pagamentos de alugueis**

Data do pagamento	% do total	Valor recebido
2022-12-03T23:25:59.000Z	60 %	R\$ 6.000,00

**Garantias de empréstimo realizadas**

Data de expiração	% do imóvel	Carteira do banco
2022-12-03T23:03:28.000Z	34,1 %	0xa4c56783b296Ea29E8Ee154bb59c2907E6774371

**Nome do proprietário: Gabriel Rigoli**  
CPF: 227.731.229-28  
Percentual do imóvel: 40%  
Endereço da carteira: 0xa28E5537cb044b57d0c28028a320448ec8E558

**Pagamentos de alugueis**

Data do pagamento	% do total	Valor recebido
2022-12-03T23:25:59.000Z	40 %	R\$ 4.000,00

**Garantias de empréstimo realizadas**  
Proprietário não possui nenhuma garantia associada a este imóvel.

**Figura 41 - Modal de auditoria dos dados do imóvel tokenizado**

A segunda ação, por sua vez, implementa o fluxo descrito na seção 4.5.7, por meio de um modal (Figura 42) no qual o administrador cadastra o valor recebido no mês como 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.

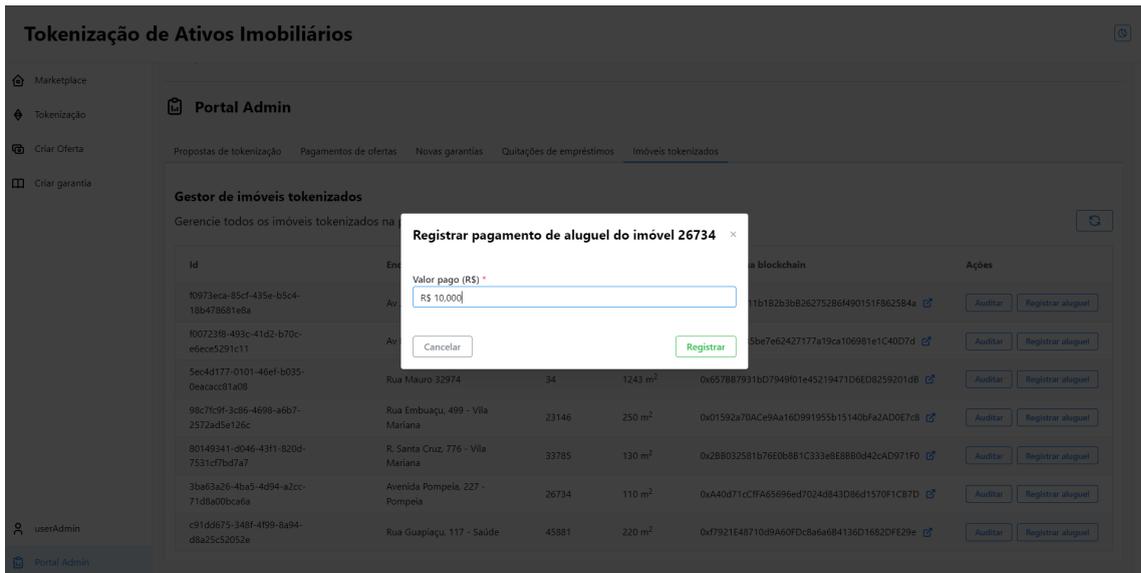


Figura 42 - Formulário de registro de pagamento de aluguel

## 5.4 TESTES E AVALIAÇÃO

Os testes realizados correspondem à validação de todas as funcionalidades mapeadas nos casos de uso e nos requisitos funcionais sob diferentes cenários. Por meio dos resultados obtidos, foi possível avaliar o sucesso da implementação de cada uma das funcionalidades propostas no início do projeto.

Nesta seção serão descritos apenas os testes realizados para os principais casos de uso (que englobam as principais funcionalidades), relatando-se os resultados obtidos.

### 5.4.1 Considerações sobre requisitos não-funcionais

Com relação aos requisitos de escalabilidade, a arquitetura desenvolvida foi pensada de modo a ser facilmente escalável, com o *Front-End* e o *Back-End* hospedados em plataformas diferentes e dissociando as regras de negócio da plataforma da interface do usuário.

Nesse aspecto, durante os testes realizados, a plataforma não apresentou nenhuma queda de desempenho devido a problemas de performance e escalabilidade, apenas durante as interações que exigem comunicação com a *blockchain* ocorre um pequeno atraso na resposta intrínseco à execução de funções dos *smart contracts*, mas que não prejudica a usabilidade da plataforma.

Ademais, os serviços de hospedagem utilizados para ambos os componentes da plataforma possuem um mecanismo de aumento do poder computacional automático conforme o aumento de demanda, garantindo escalabilidade até os limites configurados durante a hospedagem.

Com relação à segurança, a plataforma foi desenvolvida utilizando-se autenticação e autorização através de tokens JWT, um padrão da indústria quando trata-se de comunicação segura entre duas partes de uma aplicação distribuída.

Além disso, a comunicação entre todas as partes da plataforma entre si e com o usuário é realizada utilizando-se o protocolo HTTPS, que confere confidencialidade e integridade a todos os dados trafegados durante a utilização da plataforma por parte do usuário.

Por fim, a utilização de *smart contracts* na *blockchain* mostrou, durante os testes, garantir imutabilidade, auditabilidade e transparência aos dados e operações envolvendo os imóveis tokenizados, de tal forma que, dispondo do endereço do *smart contract* do imóvel (disponibilizado para os proprietários, para administradores e durante o aceite de uma oferta no Marketplace), é possível verificar todos os seus dados registrados na *blockchain* (e.g. proprietários, garantias, pagamentos de aluguéis, etc) por meio de ferramentas como o *Etherscan* ou o próprio *Remix IDE*.

#### 5.4.2 Tokenização de imóvel

Para a avaliação do funcionamento do caso de uso de tokenização de um imóvel, foi realizado o fluxo do cadastro das informações do imóvel por parte do dono do imóvel e em seguida o fluxo da validação dos documentos por parte de um usuário administrador.

Para o caso de uso de tokenização de um imóvel, foram testadas diversas situações. São elas:

Cenário testado	Resultado esperado
Formulário preenchido com informações corretas sobre o imóvel (cenário ideal)	Imóvel tokenizado com suas informações associadas ao <i>smart contract</i>
Usuário tenta tokenizar imóvel com número de registro já cadastrado	Envio de formulário rejeitado

Informações preenchidas incorretamente pelo dono	Usuário administrador rejeita a tokenização e o imóvel não é tokenizado
--	---

**Quadro 1 - Cenários de teste para o caso de uso de tokenização de imóvel**

Após a realização dos cenários descritos acima, foi possível confirmar o sucesso da implementação de todos os possíveis cenários referentes a este caso de uso de tokenização de um imóvel.

Maiores registros da realização do teste de funcionamento referente a este caso de uso se encontram no **APÊNDICE A - Teste do Caso de Uso de Tokenização de Imóvel**.

#### 5.4.3 Oferta e compra de percentual de imóvel

Para a realização da validação do caso de uso de compra e venda de uma porcentagem de um imóvel, foram realizados os fluxos de criação de uma oferta por parte de um usuário dono de um imóvel, o fluxo de aceite de oferta por parte do comprador e em seguida a realização do fluxo de confirmação da compra do percentual do ativo imobiliário através de um usuário administrador.

Para o caso de uso de compra e venda de porcentagem de imóvel, foram testadas diversas situações. São elas:

<b>Cenário testado</b>	<b>Resultado esperado</b>
Cenário ideal sem transferência de propriedade efetiva	Registrada com sucesso a transferência do percentual do imóvel do dono original para o usuário comprador, sem transferência da posse efetiva do imóvel
Cenário ideal com transferência de propriedade efetiva entre o usuário dono e o comprador	Registrada com sucesso a transferência do percentual do imóvel e da propriedade efetiva do dono original para o usuário comprador
Dono tenta criar oferta com mais porcentagem ofertada do que disponível	Envio do formulário de criação de oferta rejeitado
Dono tenta criar oferta com transferência de posse sem ser proprietário efetivo do imóvel	Envio do formulário de criação de oferta rejeitado
Comprador não confirma pagamento da oferta	Usuário administrador rejeita a compra e a

do imóvel tokenizado	oferta do imóvel volta a ser exibida no marketplace
Dono possui propriedade efetiva e vende toda a porcentagem do imóvel que possui	Dono mantém a propriedade efetiva do imóvel e ao final irá possuir 0% de porcentagem de posse do ativo
Dono possui propriedade efetiva e vende toda a porcentagem do imóvel que possui juntamente com a propriedade efetiva	Dono passa a possuir 0% de porcentagem do imóvel, sendo o imóvel removido de sua carteira e o comprador passa a ser o proprietário efetivo do imóvel
Dono não possui propriedade efetiva e vende toda a porcentagem do imóvel que possui	Dono passa a possuir 0% de porcentagem do imóvel, sendo o imóvel removido de sua carteira

**Quadro 2 - Cenários de teste para o caso de uso oferta e compra de porcentagem de imóvel**

Após a realização dos cenários descritos acima, foi possível confirmar o sucesso da implementação de todos os possíveis cenários referentes a este caso de uso de oferta e compra de porcentagem de imóvel.

Maiores registros da realização do teste de funcionamento referente a este caso de uso se encontram no **APÊNDICE B - Teste do Caso de Uso de Oferta e Compra de Percentual de Imóvel**.

#### **5.4.4 Garantias de empréstimos**

Para a realização da validação dos testes envolvendo o caso de uso de utilização de um imóvel tokenizado como garantia de um empréstimo, foi necessária a execução dos fluxos de criação de empréstimo através do usuário dono do ativo tokenizado, confirmação do empréstimo por parte de um usuário administrador, registro de quitação de empréstimo por parte do dono com confirmação de um usuário administrador, incluindo também o cenário de tomada da garantia executado por um usuário administrador da plataforma para o caso onde o dono não registra a quitação do empréstimo até a data de expiração estipulada na criação do empréstimo.

Para o caso de uso de utilização de um ativo imobiliário tokenizado como garantia de empréstimos, foram testadas diversas situações. São elas:

Cenário testado	Resultado esperado
Cenário ideal de registro de utilização de imóvel como garantia de empréstimo	Porcentagem do imóvel registrada como garantia do empréstimo
Cenário ideal de registro de dois ou mais empréstimos utilizando o mesmo ativo tokenizado	Diferentes porcentagens do imóvel registradas como garantias nos diferentes empréstimos realizados
Cenário ideal de quitação de empréstimo	Empréstimo quitado e porcentagem do ativo associada ao empréstimo é liberada para utilização pelo dono
Cenário ideal de tomada de garantia	Porcentagem do imóvel referente à garantia do empréstimo é transferida da carteira do devedor para a carteira do credor
Dono do imóvel insere dados incorretos no registro do empréstimo	Usuário administrador rejeita o registro da utilização do ativo imobiliário como garantia do empréstimo

**Quadro 3 - Cenários de teste para o caso de uso de garantias de empréstimos**

Após a realização dos cenários descritos acima, foi possível confirmar o sucesso da implementação de todos os possíveis cenários referentes a este caso de utilização de porcentagem de um ativo imobiliário tokenizado como garantia de empréstimos.

Maiores registros da realização do teste de funcionamento referente a este caso de uso se encontram no **APÊNDICE C - Teste do Caso de Uso de Garantia de Empréstimo**.

#### **5.4.5 Distribuição de retornos financeiros provenientes de aluguel**

Por fim, o último conjunto de testes realizado teve como foco validar o caso de uso de distribuição de aluguel entre os proprietários de um imóvel, de modo proporcional à porcentagem de posse que cada um deles possui. Para isso, foi executado o fluxo de registro de pagamento do aluguel de um imóvel pelo usuário administrador em diferentes configurações de proprietários do imóvel e valor de aluguel pago. O **Quadro 4** lista os cenários de testes levantados para validar este caso de uso.

Cenário testado	Resultado esperado
Registro do valor do aluguel pelo administrador, imóvel com 2 proprietários com 50% de posse	Valor do aluguel é distribuído igualmente entre os 2 proprietários
Registro de valor do aluguel com 2 casas decimais	Valor distribuído corretamente entre os proprietários, com 2 casas decimais de precisão
Registro do pagamento do aluguel de um imóvel com dono efetivo possuindo 0% de porcentagem de posse	Proprietário com 0% de posse não recebe aluguel, sendo o valor distribuído apenas entre os demais
Proprietário vende ou compra mais participação do imóvel após já ter recebido aluguéis em algum momento	Porcentagem que detinha no momento do recebimento do aluguel permanece registrada, garantindo auditorias futuras

**Quadro 4 - Cenários de teste para o caso de uso de registro de pagamento de aluguel**

Com a execução correta dos cenários listados acima, foi possível confirmar o sucesso da implementação do caso de uso de distribuição de pagamentos de aluguéis.

Maiores registros da realização do teste de funcionamento referente a este caso de uso se encontram no **APÊNDICE D - Teste do Caso de Uso de Registro de pagamento de aluguel**.

## 6 CONSIDERAÇÕES FINAIS

O projeto aqui descrito tem como principal objetivo trazer uma solução que viabilize o ganho de liquidez de ativos imobiliários, de modo a facilitar transações e ampliar as possibilidades de negócio envolvendo esta categoria de ativos, como em cenários de aluguéis ou de garantias de empréstimo.

A plataforma proposta é capaz de enviar *smart contracts* para uma *blockchain* e associar imóveis a tokens que representam a sua posse, permitindo assim a realização de transações e registros com frações destes imóveis.

A versão final do projeto englobou todas as funcionalidades inicialmente propostas, e isso foi possível muito por conta da pesquisa por soluções análogas e pela escolha adequada das tecnologias utilizadas, balanceando a familiaridade dos membros da equipe com a viabilidade técnica para implementação do que foi proposto. Apesar de todas as funcionalidades propostas terem sido atingidas,

entende-se que a solução desenvolvida ainda apresenta uma principal limitação, que é a dificuldade de se averiguar se um ativo já foi tokenizado em outras blockchains (para evitar o caso da tokenização de um mesmo imóvel em múltiplas plataformas), ainda não existindo meios bem explorados para que isso de fato seja possível.

Além disso, como citado anteriormente, as discussões jurídicas sobre o uso da tokenização para registro de imóveis como propriedades digitais ainda estão em discussão e, portanto, ainda existem muitos entraves legais para que a solução desenvolvida possa eventualmente ser utilizada em escala por diferentes entidades do mercado. Apesar disso, o foco deste estudo se manteve em provar a viabilidade da tecnologia inserida no mercado imobiliário, e isto foi alcançado com êxito por meio da implementação da plataforma, que agrega e adapta funcionalidades presentes nas soluções de mercado levantadas, como Netspaces [9] e RealT [13], que fazem parte do estado da arte no contexto de tokenização de ativos imobiliários.

## 6.1 PERSPECTIVAS DE CONTINUIDADE

Apesar da plataforma desenvolvida cumprir o objetivo proposto inicialmente, foram notados diversos pontos de melhorias para torná-la mais próxima da realidade do mercado em termos de funcionalidades e cumprimento de requisitos funcionais e não-funcionais.

Por ser uma prova de conceito, foram tomadas algumas simplificações no desenvolvimento do projeto, e a principal delas foi a utilização de apenas endereço, área útil, número de registro e algum documento para identificar o imóvel. Como um próximo passo claro, está o levantamento de todos os dados do imóvel necessários para viabilizar legalmente a utilização desta tecnologia no mercado.

Além disso, outra decisão de projeto foi a de assumir que os pagamentos seriam realizados *off-chain*, ou seja, por fora da *blockchain*. Neste contexto, um ponto de melhoria seria também implementar transações *on-chain* (via *blockchain*), o que não só possibilitaria que os usuários assinassem transações com suas próprias carteiras digitais, como também excluiria a necessidade da atuação do administrador como intermediador das transações ou da integração com serviços de pagamentos nos fluxos de registro de aluguéis e de compra e venda de porcentagens de ativos no marketplace.

Outro ponto de melhoria seria migrar a utilização da rede Ethereum Goerli para a rede pública do Ethereum onde, combinada com a realização de pagamentos *on-chain*, os tokens transacionados realmente possuem valor atrelado, diferentemente da rede de testes, apenas utilizada para o teste de funcionamento. A utilização da rede pública Ethereum também traria maior confiabilidade à plataforma devido ao seu caráter público e ao maior número de nós participantes da rede que iriam executar seu algoritmo de consenso.

Também seria interessante, no futuro, a utilização de um sistema de custódia de chaves privadas por parte da própria da plataforma, eliminando a necessidade dos usuários possuírem e gerenciarem suas próprias carteiras na rede Ethereum através da MetaMask. Este sistema de custódia poderia implementar mecanismos de segurança como o *secret-sharing* (sistema que permite a distribuição de uma chave privada em um grupo de atores), o *multi-signature* (sistema que requer múltiplas assinaturas de diversos atores para a realização de uma transação) ou o *multi-party computation* (sistema onde cada um dos diversos atores irá gerar sua parte de uma chave e estas partes serão utilizadas para a geração da chave completa quando necessário). [46] [47]

Por fim, entende-se que a tokenização tem diversas aplicações no mercado imobiliário, que vão além das funcionalidades aqui desenvolvidas. Assim, outro foco para uma possível continuação deste projeto seria a de idealizar e implementar novos casos de uso, como por exemplo, de financiamento de imóveis, fundos de investimentos ou crowdfunding, de modo a tornar a plataforma mais completa e versátil frente ao mercado imobiliário.

## 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] SATOSHI, N. Bitcoin: A Peer-to-Peer Electronic Cash System. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>. 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] CASTOR, A. A (Short) Guide to Blockchain Consensus Protocols. Disponível em: <<https://www.coindesk.com/markets/2017/03/04/a-short-guide-to-blockchain-consensus-protocols>>. Acesso em: 13 dez. 2022.

[21] SMITH, C. PROOF-OF-STAKE (POS). Disponível em: <<https://ethereum.org/en/developers/docs/consensus-mechanisms/pos>>. Acesso em: 13 dez. 2022.

[22] 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.

[23] 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.

[24] CHAINANI, V. What are Ethereum request for comments (ERC) Standards. Disponível em: <[https://dev.to/envoy\\_/ks-what-are-ethereum-request-for-comments-erc-standards-5f80](https://dev.to/envoy_/ks-what-are-ethereum-request-for-comments-erc-standards-5f80)>. Acesso em: 13 dez. 2022.

[25] 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.

[26] 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.

[27] WACKEROW, P. ERC-20 TOKEN STANDARD. Disponível em: <<https://ethereum.org/en/developers/docs/standards/tokens/erc-20>>. Acesso em: 13 dez. 2022.

[28] SHIRLEY, D. EIP-721: Non-Fungible Token Standard. Disponível em: <<https://eips.ethereum.org/EIPS/eip-721>>. Acesso em: 13 dez. 2022.

[29] 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.

[30] 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.

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

[32] 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.

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

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

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

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

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

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

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

[40] ETHERSCAN. Página inicial. Disponível em: <<https://etherscan.io/>>. Acesso em: 13 dez. 2022.

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

[42] INFURA. Página inicial. Disponível em: <<https://www.infura.io/>>. Acesso em: 13 dez. 2022.

[43] RAILWAY. Página inicial. Disponível em: <<https://railway.app/>>. Acesso em: 13 dez. 2022.

[44] SWAGGER. Página inicial. Disponível em: <<https://swagger.io/>>. Acesso em: 13 dez. 2022.

[45] VERCEL. Página inicial. Disponível em: <<https://vercel.com/>>. Acesso em: 13 dez. 2022.

[46] ZHOU, L. Secret Sharing. Disponível em: <<https://www.cs.cornell.edu/courses/cs513/2000SP/SecretSharing.html>>. Acesso em: 13 dez. 2022.

[47] SCHWEIGER, L. Crypto Custody: The gateway to institutional adoption. Disponível em: <<https://www.blockdata.tech/blog/general/crypto-custody-the-gateway-to-institutional-adoption>>. Acesso em: 13 dez. 2022.

## APÊNDICE A - Teste do Caso de Uso de Tokenização de Imóvel

Seguem abaixo capturas de tela da plataforma referentes ao teste do fluxo do cenário de tokenização de um ativo imobiliário:

A interface de usuário para a tokenização de ativos imobiliários. O título principal é "Tokenização de Ativos Imobiliários". O menu lateral contém: Marketplace, Tokenização (selecionado), Criar Oferta e Criar garantia. O caminho de navegação é "Home / Criar Proposta de Tokenização".

### Tokenize o seu imóvel

Insira os dados do seu imóvel para análise da plataforma. Caso os dados sejam aprovados, você receberá em sua carteira digital o imóvel tokenizado.

Endereço do Imóvel \*  
R. Pedro Zaccaria, 1300

Área Útil do Imóvel (m²) \*  
2135

Número do Registro em Cartório \*  
3829

Endereço da Carteira do Dono Efetivo \*  
0x949DE174294BD2752a9A7172EF344eEc6dEc7F8e

Documentos do imóvel (Ex: Escrituras, comprovante de posse, etc) \*  
Escritura.pdf

Enviar dados do imóvel

cleiton123

Figura A.1 - Preenchimento das informações de tokenização de um ativo imobiliário

O Portal Admin para a tokenização de ativos imobiliários. O menu lateral contém: Marketplace, Tokenização, Criar Oferta e Criar garantia. O caminho de navegação é "Portal Admin".

### Portal Admin

Propostas de tokenização | Pagamentos de ofertas | Novas garantias | Quitações de empréstimos | Imóveis tokenizados

#### Propostas de Tokenização Ativas

Gerencie as propostas de tokenização ativas, aceite ou rejeite de acordo com os dados fornecidos.

R. Pedro Zaccaria, 1300

Nome do Usuário: cleiton123

Nome: Cleiton Jorge

CPF do Usuário: 351.086.060-87

Área Útil: 2135 m²

Número de Registro: 3829

Visualizar documentos

Deseja Aceitar Esta Proposta?

Cancelar Aceitar

Rejeitar Aceitar

Figura A.2 - Aceite da proposta de tokenização por parte de usuário administrador

## Tokenização de Ativos Imobiliários

Marketplace

Tokenização

Criar Oferta

Criar garantia

### Av Jabaquara 12345

Área Útil 2183791 m<sup>2</sup>

Número do Registro 2167

Porcentagem de Posse 10 %

Porcentagem disponível para operações 10 %

Visualizar no Etherscan [↗](#)

Aluguéis recebidos

### Rua Mauro 32974

Área Útil 1243 m<sup>2</sup>

Número do Registro 34

Porcentagem de Posse 10 %

Porcentagem disponível para operações 10 %

Visualizar no Etherscan [↗](#)

Aluguéis recebidos

### Av Paulista 987 DONO

Área Útil 1000 m<sup>2</sup>

Número do Registro 123

Porcentagem de Posse 35 %

Porcentagem disponível para operações 25 %

Visualizar no Etherscan [↗](#)

Aluguéis recebidos

### R. Pedro Zaccaria, 1300 DONO

Área Útil 2135 m<sup>2</sup>

Número do Registro 3829

Porcentagem de Posse 100 %

Porcentagem disponível para operações 100 %

Visualizar no Etherscan [↗](#)

Aluguéis recebidos

delton123

Figura A.3 - Imóvel tokenizado apresentado na página de perfil do usuário

## APÊNDICE B - Teste do Caso de Uso de Oferta e Compra de Percentual de Imóvel

Seguem abaixo capturas de tela da plataforma referentes ao teste do fluxo do cenário ideal de compra e venda de um ativo imobiliário tokenizado:

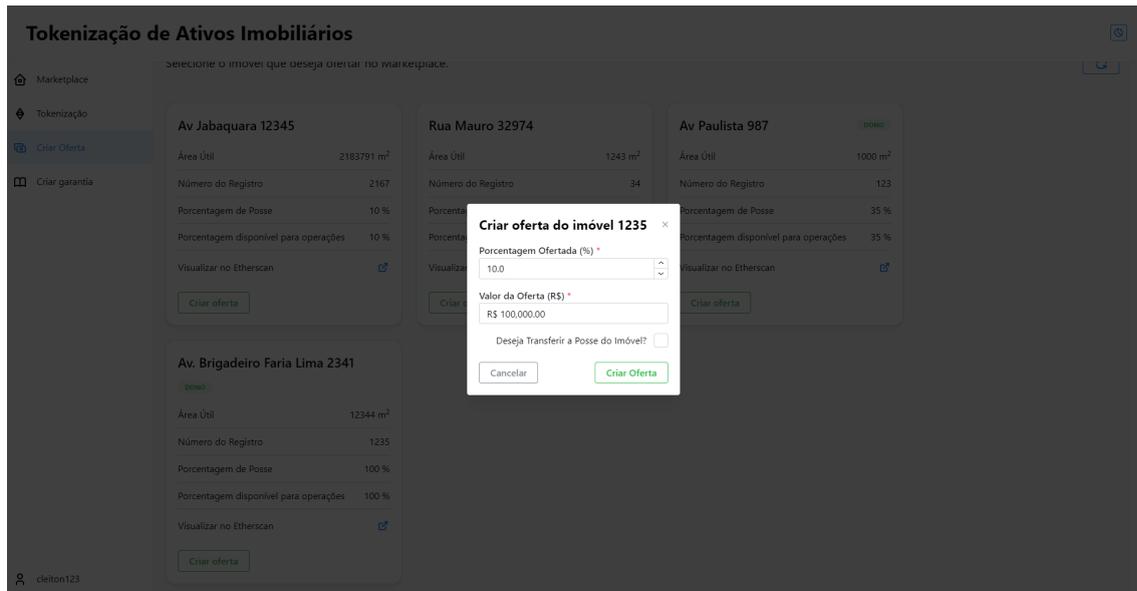


Figura B.1 - Preenchimento das informações de criação de oferta

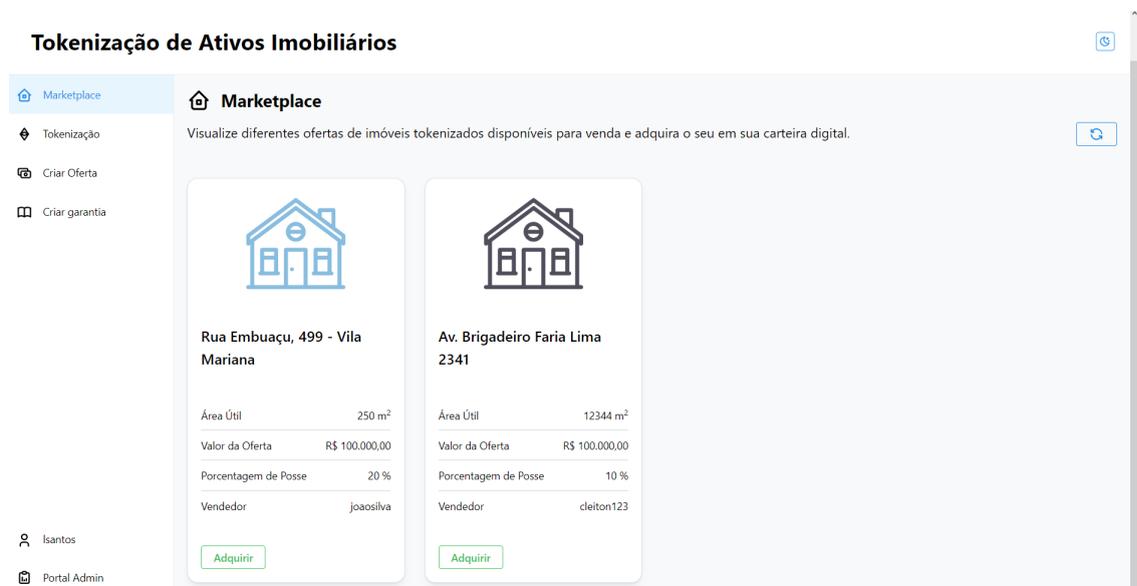


Figura B.2 - Oferta disponível no marketplace após sua criação

**Tokenização de Ativos Imobiliários**

Home / Marketplace / be6a0a89-5e68-4adb-9790-41e604a28cbf

**Oferta #be6a0a89-5e68-4adb-9790-41e604a28cbf**

Av. Brigadeiro Faria Lima 2341

**Dados da Oferta:**

Porcentagem à Venda	10 %
Valor da Oferta	R\$ 100.000,00

**Dados do Imóvel:**

Área Útil	12344 m <sup>2</sup>
Número do Registro	1235
Endereço do Imóvel Tokenizado	0x1171c74ac159E558e48468b95ED51E6a2692dAfb

**Dados do Vendedor:**

Nome	Cleiton Jorge
Usuário	cleiton123

[Aceitar Oferta](#)

Isantos Portal Admin

**Figura B.3 - Detalhes da oferta logo antes de seu aceite pelo comprador**

**Tokenização de Ativos Imobiliários**

Home / User

**Meu perfil** Logout

Dados gerais Imóvel tokenizados Garantias Ofertas abertas Compras realizadas

**Compras realizadas no Marketplace**

Gerencie suas compras realizadas

**Av. Paulista 987** PAGA

Número de Registro	1235
Valor da Oferta	R\$ 100.000,00
Porcentagem do Imóvel em Oferta	50 %
Usuário Vendedor	cleiton123
Nome do Vendedor	Cleiton Jorge
Comprovante de pagamento	Transação XYZ

[Editar comprovante](#)

**Av. Brigadeiro Faria Lima 2341** ESPERANDO PAGAMENTO

Número de Registro	1235
Valor da Oferta	R\$ 100.000,00
Porcentagem do Imóvel em Oferta	10 %
Usuário Vendedor	cleiton123
Nome do Vendedor	Cleiton Jorge
Comprovante de pagamento	-

[Adicionar comprovante](#)

**Adicionar comprovante de pagamento à compra do imóvel 1235**

Comprovante de pagamento \*

Transação: 0x5e871

[Cancelar](#) [Adicionar](#)

Isantos Portal Admin

**Figura B.4 - Registro do comprovante de pagamento de oferta por parte do comprador**

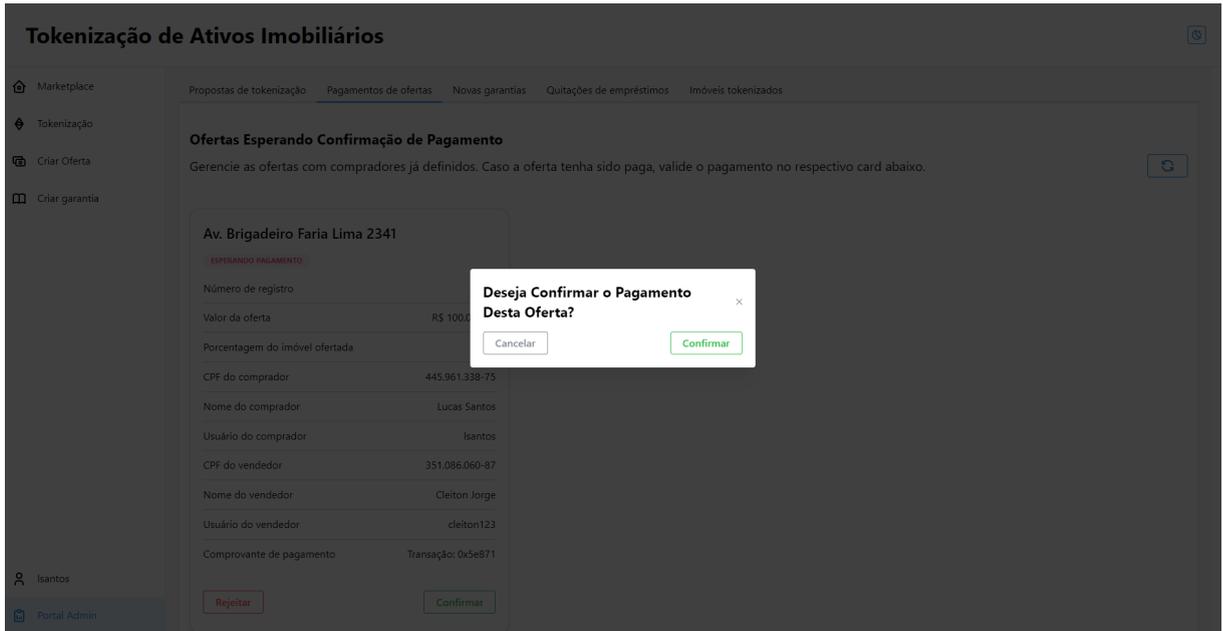


Figura B.5 - Confirmação do pagamento de oferta por parte de usuário administrador

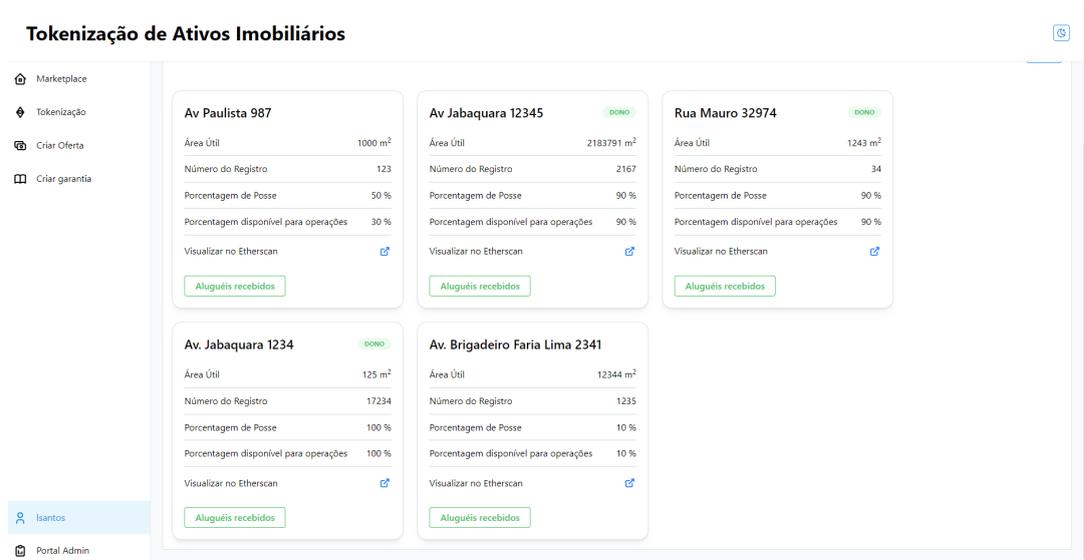


Figura B.6 - Porcentagem do ativo transferido para a conta do comprador

### Tokenização de Ativos Imobiliários

Marketplace  
Tokenização  
Criar Oferta  
Criar garantia

<b>Av Jabaquara 12345</b>	<b>Rua Mauro 32974</b>	<b>Av Paulista 987</b> <span>DONO</span>
Área Útil: 2183791 m <sup>2</sup>	Área Útil: 1243 m <sup>2</sup>	Área Útil: 1000 m <sup>2</sup>
Número do Registro: 2167	Número do Registro: 34	Número do Registro: 123
Porcentagem de Posse: 10 %	Porcentagem de Posse: 10 %	Porcentagem de Posse: 35 %
Porcentagem disponível para operações: 10 %	Porcentagem disponível para operações: 10 %	Porcentagem disponível para operações: 25 %
Visualizar no Etherscan <a href="#">↗</a>	Visualizar no Etherscan <a href="#">↗</a>	Visualizar no Etherscan <a href="#">↗</a>
<a href="#">Aluguéis recebidos</a>	<a href="#">Aluguéis recebidos</a>	<a href="#">Aluguéis recebidos</a>

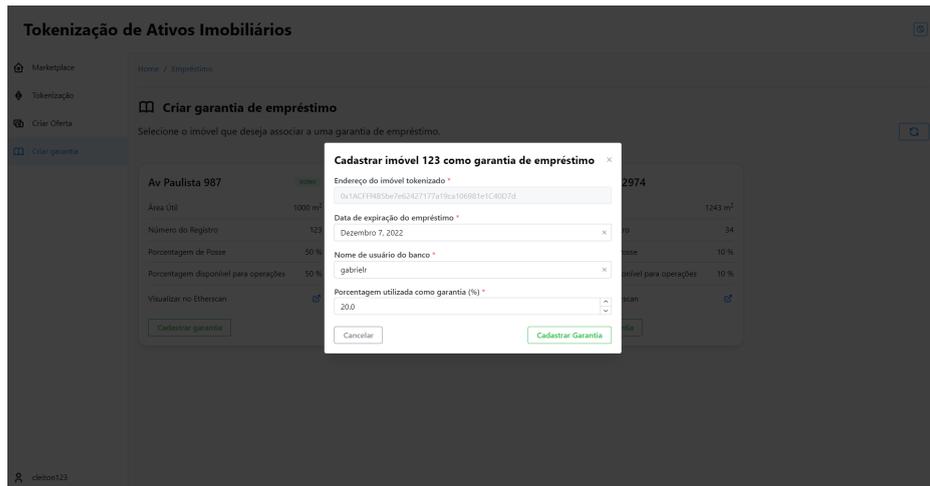
<b>Av. Brigadeiro Faria Lima 2341</b>
<span>DONO</span>
Área Útil: 12344 m <sup>2</sup>
Número do Registro: 1235
Porcentagem de Posse: 90 %
Porcentagem disponível para operações: 90 %
Visualizar no Etherscan <a href="#">↗</a>
<a href="#">Aluguéis recebidos</a>

cleiton123

Figura B.7 - Porcentagem do ativo transferido da conta do vendedor

## APÊNDICE C - Teste do Caso de Uso de Garantia de Empréstimo

Seguem abaixo capturas de tela da plataforma referentes ao teste do fluxo do cenário ideal de criação de empréstimo utilizando porcentagem de ativo tokenizado como garantia:



**Tokenização de Ativos Imobiliários**

Home / Empréstimo

Marketplace

Tokenização

Criar Oferta

Criar garantia

### Criar garantia de empréstimo

Selecione o imóvel que deseja associar a uma garantia de empréstimo

Av Paulista 987	123	1243 m <sup>2</sup>	10 %
Área Útil	1000 m <sup>2</sup>		10 %
Número do Registro	123		10 %
Porcentagem de Posse	50 %		10 %
Porcentagem disponível para operações	50 %		10 %

Visualizar no Etherscan

Cadastrar garantia

#### Cadastrar imóvel 123 como garantia de empréstimo

Endereço do imóvel tokenizado \*

DATACTTPRESB81e62427177a15ba106981e1c48D79d

Data de expiração do empréstimo \*

Dezembro 7, 2022

Nome de usuário do banco \*

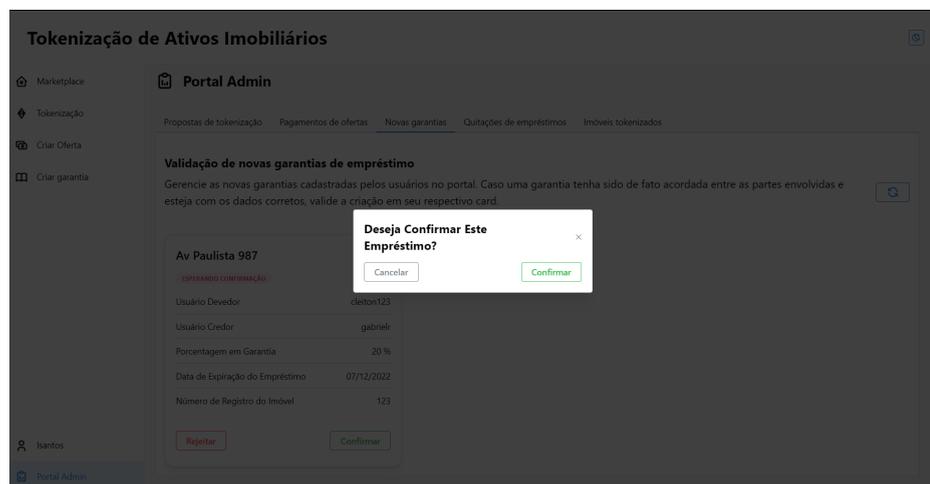
gabriel

Porcentagem utilizada como garantia (%) \*

20,0

Cancelar Cadastrar Garantia

Figura C.1 - Preenchimento do cadastro de utilização de porcentagem de um imóvel tokenizado como garantia de empréstimo realizado pelo devedor



**Tokenização de Ativos Imobiliários**

Portal Admin

Propostas de tokenização Pagamentos de ofertas Novas garantias Quitações de empréstimos Imóveis tokenizados

### Validação de novas garantias de empréstimo

Gerencie as novas garantias cadastradas pelos usuários no portal. Caso uma garantia tenha sido de fato acordada entre as partes envolvidas e esteja com os dados corretos, valide a criação em seu respectivo card.

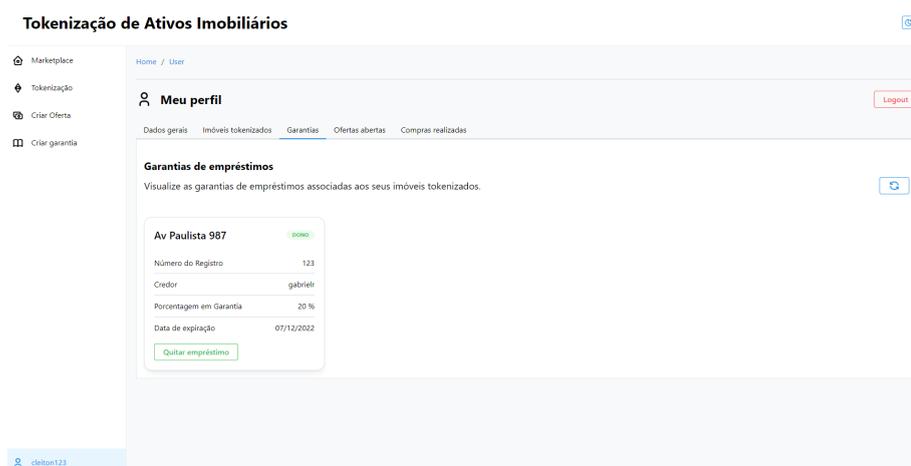
Av Paulista 987	erro de validação
Usuário Devedor	cleiton123
Usuário Credor	gabriel
Porcentagem em Garantia	20 %
Data de Expiração do Empréstimo	07/12/2022
Número de Registro do Imóvel	123

Registrar Confirmar

#### Deseja Confirmar Este Empréstimo?

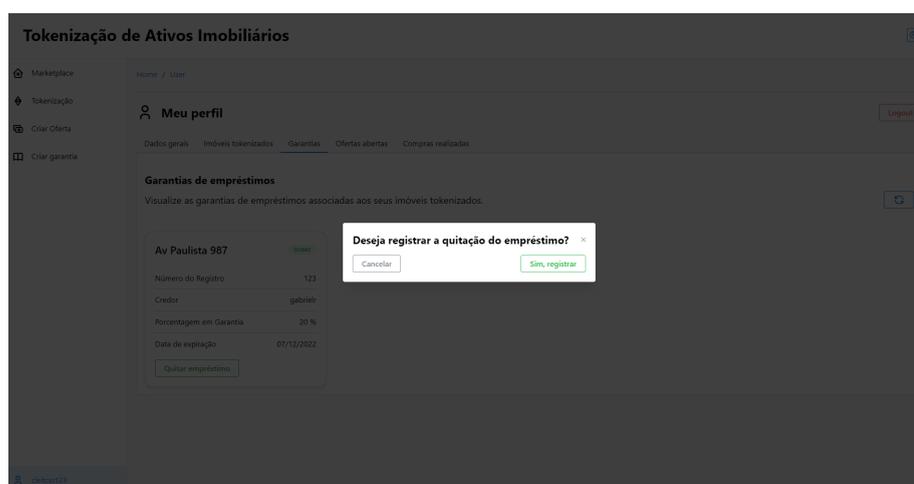
Cancelar Confirmar

Figura C.2 - Usuário administrador confirma cadastro de utilização de porcentagem do imóvel como garantia de empréstimo

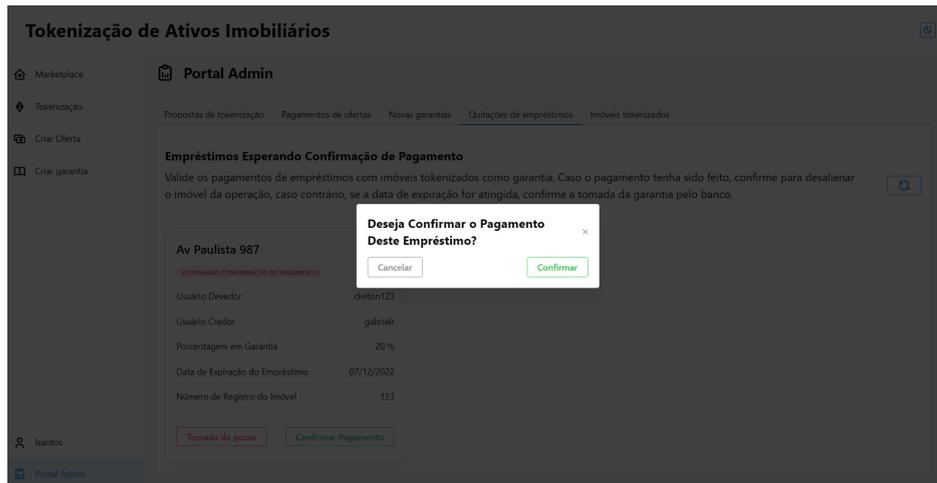


**Figura C.3 - Registro da utilização de porcentagem do imóvel como garantia de empréstimo exibido no perfil do usuário devedor**

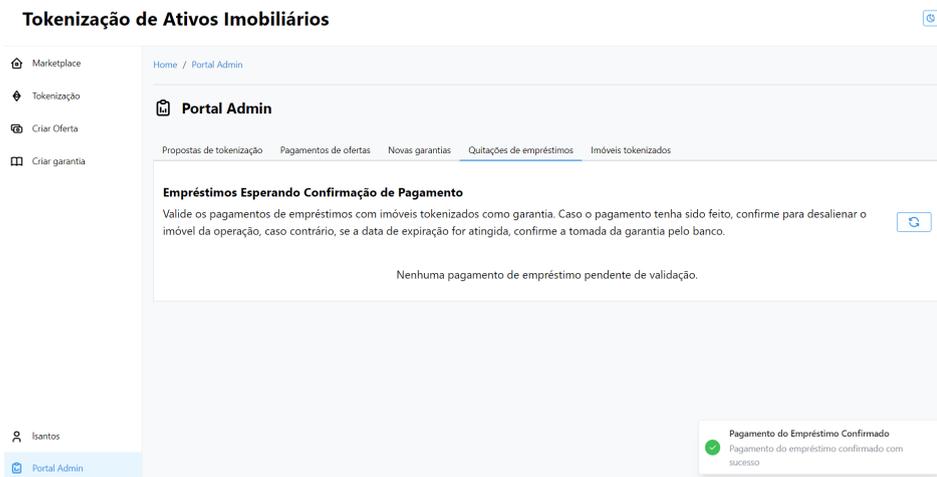
Seguem abaixo capturas de tela da plataforma referentes ao teste do fluxo do cenário de quitação de empréstimo:



**Figura C.4 - Registro da quitação do empréstimo pelo devedor**

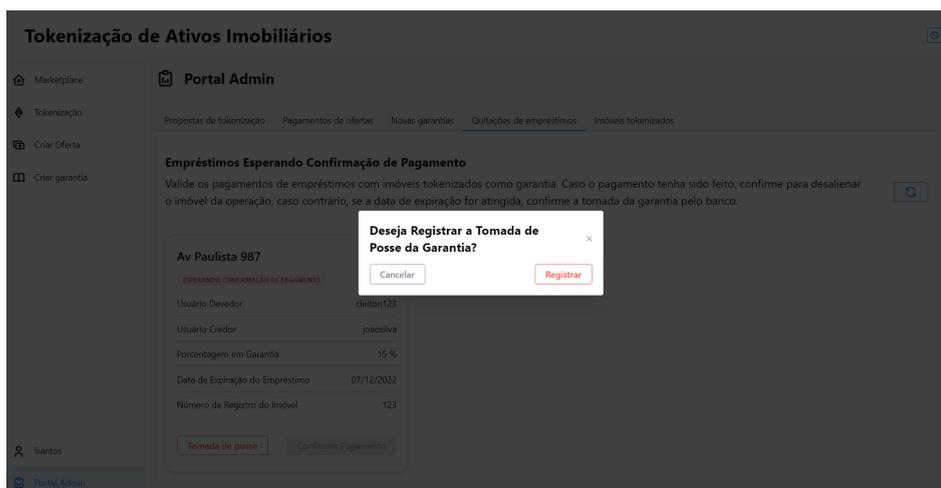


**Figura C.5 - Usuário administrador confirma a quitação do empréstimo**

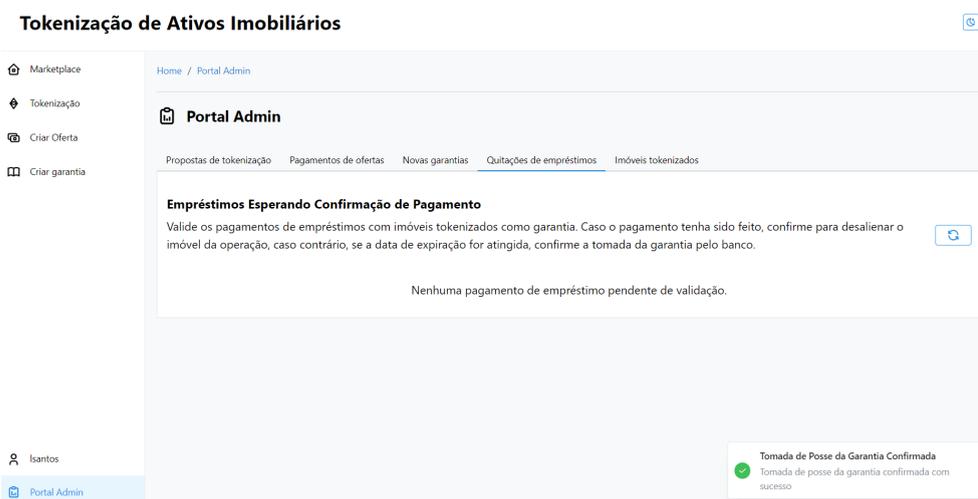


**Figura C.6 - Confirmação de quitação de empréstimo pelo usuário administrador bem sucedida**

Seguem abaixo capturas de tela da plataforma referentes ao teste do fluxo do cenário de tomada de posse:



**Figura C.7 - Usuário administrador registra a tomada de posse da garantia do empréstimo após sua data de expiração**



**Figura C.8 - Confirmação de tomada de posse da porcentagem do imóvel pelo usuário administrador bem sucedida**

## APÊNDICE D - Teste do Caso de Distribuição de Retornos Financeiros Provenientes de Aluguel

Seguem abaixo capturas de tela da plataforma referentes ao teste do fluxo do cenário ideal pagamento de lucros provenientes de aluguel de imóvel tokenizado:

A captura de tela mostra a interface de 'Tokenização de Ativos Imobiliários'. No topo, há um menu lateral com opções como 'Marketplace', 'Tokenização', 'Criar Oferta' e 'Criar garantia'. O conteúdo principal é uma tabela com as seguintes colunas: 'Id', 'Endereço', 'Nº registro', 'Área útil', 'Endereço na blockchain' e 'Ações'. Uma modal de diálogo intitulada 'Registrar pagamento de aluguel do imóvel 3829' está aberta no centro da tela, permitindo o registro de um valor pago de R\$ 2.500,00. A modal contém campos para 'Valor pago (R\$)', um botão 'Cancelar' e um botão 'Registrar'.

Id	Endereço	Nº registro	Área útil	Endereço na blockchain	Ações
f0973eca-85cf-435e-b5c4-18b478681e8a	Av Jabaquara 12345	2167	2183791 m²	0x983dC5d11b182b3b826275286490151f862584a	Audit   Registrar aluguel
f00723f8-493c-41d2-b70c-e6ece5291c11	Av Paulista 987	123	1000 m²	0x1ACFF4B5be7e62427177a19ca106981e1C40D7d	Audit   Registrar aluguel
5ec4d177-0101-46ef-b035-0eacacc81a08	Rua - V			95219471D6ED8259201d8	Audit   Registrar aluguel
98c7fc9f-3c86-4698-a6b7-2572ad5e126c	Rua - V			81955b15140bfa2AD0E7c8	Audit   Registrar aluguel
80149341-d046-43f1-820d-7531cf7bd7a7	R. S			333e8E88B0d42cAD971f0	Audit   Registrar aluguel
3ba63a26-4ba5-4d94-a2cc-71d8a00bca6a	Ave 221			024d843D86d1570f1CB7D	Audit   Registrar aluguel
c91dd675-348f-4f99-8a94-d8a25c52052e	Rua Guapiacu, 117 - Saúde	45881	220 m²	0xf7921E48710d9A60Fdc8a6a6B4136D1682DFE29e	Audit   Registrar aluguel
4cd3b76d-b9ac-44db-9b77-778d6ab182b4	Av. Jabaquara 1234	17234	125 m²	0x78bd6161338273c7b19D3f63f185e75e215e0Ccb	Audit   Registrar aluguel
1a84b9e6-07f6-4a3b-b065-c9c273967484	Av. Brigadeiro Faria Lima 2341	1235	12344 m²	0x1171c74aC159E558e48468b95ED51E6a2692d4fb	Audit   Registrar aluguel
fdc9c2d7-67b5-4f60-9d71-bb43ecfe9a7c	R. Pedro Zaccaria, 1300	3829	2135 m²	0x80a02A7adAD82Add86077034b22e1964740bbAc9	Audit   Registrar aluguel

Figura D.1 - Usuário administrador registra pagamento de lucro proveniente de aluguel do ativo tokenizado

A captura de tela mostra a interface de 'Tokenização de Ativos Imobiliários' com o modal de auditoria de um imóvel tokenizado aberto. O modal contém as seguintes informações:

- Dados do imóvel registrados em Blockchain:** Endereço do Smart Contract: 0x80a02A7adAD82Add86077034b22e1964740bbAc9
- Nome do proprietário: Cleiton Jorge** (PROPRIETÁRIO EFETIVO)
  - CNPJ: 351.086.060-87
  - Percentual do imóvel: 50%
  - Endereço da carteira: 0x9490E1742948D02752a9A7172EF344eC6dEC7F8e
- Pagamentos de alugueis:**

Data do pagamento	% do total	Valor recebido
2022-12-10T01:17:58.000Z	50 %	R\$ 1.250,00
- Garantias de empréstimo realizadas:** Proprietário não possui nenhuma garantia associada a este imóvel.
- Nome do proprietário: Lucas Santos**
  - CNPJ: 445.961.338-75
  - Percentual do imóvel: 50%
  - Endereço da carteira: 0xe4c56783d08da2968EC154b659c2907E6774371
- Pagamentos de alugueis:**

Data do pagamento	% do total	Valor recebido
2022-12-10T01:17:58.000Z	50 %	R\$ 1.250,00
- Garantias de empréstimo realizadas:** Proprietário não possui nenhuma garantia associada a este imóvel.

Figura D.2 - Valor do pagamento de aluguel registrado pelo administrador é dividido entre os diferentes proprietários do imóvel e exibido no modal de auditoria do ativo