UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO POLITÉCNICO DA UFSM
CURSO DE SISTEMAS PARA INTERNET

Gabriel Andres Silva Gimenez

CRIPTOMOEDA: SMART CONTRACT NA REDE ETHEREUM

Santa Maria, RS

2023

Gabriel Andres Silva Gimenez

CRIPTOMOEDA: SMART CONTRACT NA REDE ETHEREUM

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas para Internet da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Tecnólogo em Sistemas para Internet.**

ORIENTADOR: Prof. Dr. Marcos Alexandre Rose Silva

Santa Maria, RS

2023

Gabriel Andres Silva Gimenez

CRIPTOMOEDA: SMART CONTRACT NA REDE ETHEREUM

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas para Internet da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Tecnólogo em Sistemas para Internet.**

Aprovado em 12 de dezembro de 2023.

Marcos Alexandre Rose Silva, Dr. (UFSM)
(Presidente/Orientador)

Giani Petri, Dr. (UFSM)

Andressa Falcade, Dr. (UFSM)

Santa Maria, RS
2023

RESUMO

CRIPTOMOEDA: SMART CONTRACT NA REDE ETHEREUM

AUTOR: Gabriel Andres Silva Gimenez
ORIENTADOR: Marcos Alexandre Rose Silva

A criação do Bitcoin fez com que as criptomoedas se tornassem populares, e surgiu a necessidade de investigar formas de transacionar um ativo digital onde a criptografia se torna o intermediário com o intuito de prover segurança, etc. O surgimento da criptomoeda Ethereum e de sua rede abriram uma nova porta para a utilização da tecnologia da Blockchain, os contratos inteligentes (*Smart Contracts*) permitindo aplicações descentralizadas, inovando assim a utilidade da Blockchain. Este novo nicho de tecnologia originou a criação de frameworks que disponibilizam um gerenciamento, automatização e facilitação do desenvolvimento dos *Smart Contracts*. Neste contexto, este TCC tem por intuito investigar os conceitos e as tecnologias relacionados à Blockchain, incluindo os *Smart Contracts*, bem como investigar e exemplificar a utilização do framework HardHat para o desenvolvimento de uma *Smart Contract* na rede Ethereum.

Palavras-chave: Criptomoeda. Smart contract. Ethereum. Blockchain.

ABSTRACT

CRYPTOCURRENCY: SMART CONTRACT ON THE ETHEREUM NETWORK

AUTHOR: Gabriel Andres Silva Gimenez
ADVISOR: Marcos Alexandre Rose Silva

The creation of Bitcoin made cryptocurrencies become popular, and the need arose to investigate ways of transacting a digital asset where cryptography becomes the intermediary in order to provide security, etc. opened a new door for the use of Blockchain technology, *Smart Contracts* allowing decentralized applications, thus innovating the usefulness of Blockchain. This new technology niche led to the creation of frameworks that provide management, automation and facilitation of the development of *Smart Contracts*. In this context, this TCC aims to investigate concepts and technologies related to Blockchain, including *Smart Contracts*, as well as to investigate and exemplify the use of the HardHat framework for the development of a *Smart Contracts* on the Ethereum network.

Keywords: Cryptocurrency. Smart contract. Ethereum. Blockchain.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

As criptomoedas permitem a população mundial uma alternativa aos métodos convencionais de transacionar dinheiro, tirando assim uma parte do poder dos bancos e governos em cima da população, somado a isso a segurança da blockchain, a anonimidade da transação, a irreversibilidade da transação depois de ter sido adicionada a blockchain e a disponibilidade que a rede tem 24 horas nos 7 dias da semana, tudo isso a um custo baixíssimo comparado as transações normais, tornam a blockchain uma alternativa para pessoas que querem proteger seu dinheiro. Contudo, essa não é a única vertente, o fato da tecnologia ter criado uma disrupção tecnológica que possibilitou a transação de ativos de maneira digital,

transformou a blockchain em uma referência nesse assunto. Sendo considerada como fonte de inspiração por governos para servir como base da implementação de dinheiro virtual oficial de um país (NAKAMOTO, S., 2008).

Os *Smart Contracts* são contratos inteligentes que podem ser implementados de maneira parcial ou completamente digital. Eles fornecem a possibilidade de criar um código que seguirá passos específicos e de maneira imutável, que podem ser usados como caráter formal, uma negociação regida por lei, ou de caráter informal.

Como, por exemplo, um contrato digital que garante que no dia de aniversário de uma pessoa, será enviada uma quantia de criptomoedas fornecida por quem quer enviar o valor como presente. O contrato pode automaticamente tentar enviar o valor da carteira do presenteador.

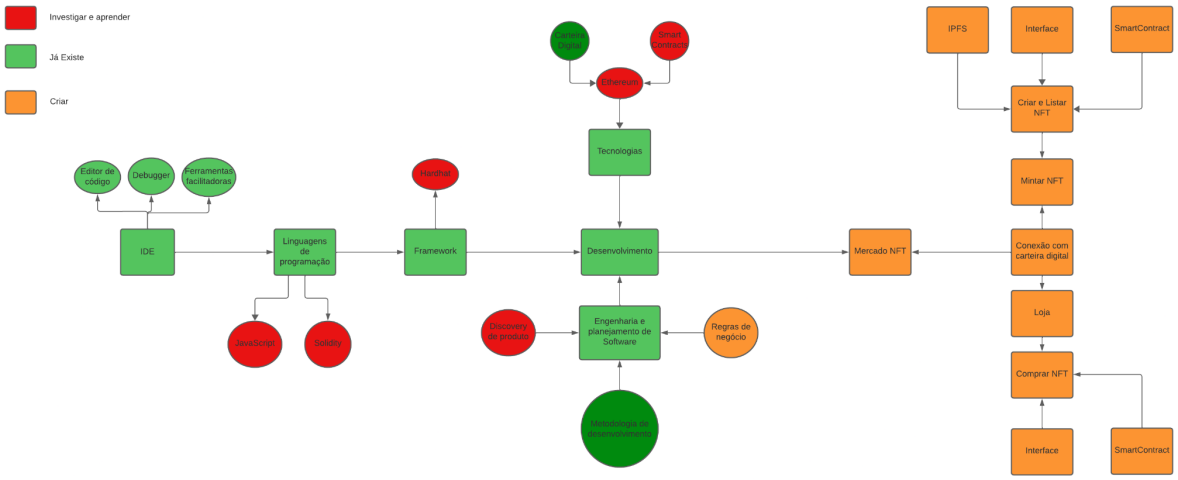
O trabalho de conclusão de curso tem o foco de investigar o funcionamento da blockchain e implementar um *Smart Contract* utilizando o framework HardHat, onde o projeto é voltado a criar uma carteira mercado de NFT (Tokens Não Fungíveis) que são ativos digitais únicos e indivisíveis registrados em uma blockchain, qualquer outra forma de mídia digital pode se tornar uma NFT. A singularidade e a propriedade exclusiva de cada NFT são o que os torna valiosos e distintos de outros tipos de ativos digitais.

1.1 Objetivo

Investigar o funcionamento da blockchain e suas tecnologias, com o objetivo de desenvolver um *Smart Contract* na rede Ethereum para um mercado NFT.

1.2 Metodologia

Figura 1 – Metodologia



Fonte: Autoria própria.

A Figura 1 apresenta a metodologia desta pesquisa, concentrando-se na exploração e criação de elementos essenciais. A abordagem abrange a análise de conceitos, a identificação de implementações anteriores e a subsequente geração de novos componentes. Utilizando uma paleta de cores, o tom laranja destaca elementos destinados à criação a partir do zero, representando a esfera da inovação. A cor verde, por sua vez, representa elementos já conhecidos e dominados, beneficiando-se de implementações consolidadas, integrados como recursos prévios. A tonalidade vermelha na figura representa elementos existentes que demandam investigação e aprendizado para serem compreendidos e implementados. Essa etapa é crucial para integrar efetivamente os elementos criados (laranja) e os já conhecidos e dominados (verde), assegurando uma abordagem holística e fundamentada no desenvolvimento do trabalho.

1.3 ORGANIZAÇÃO DO TRABALHO

O Capítulo 2 explica o funcionamento da blockchain do Bitcoin, compara a blockchain do Bitcoin com a do Ethereum, explica a blockchain do Ethereum com seus *Smart contracts* e nos introduz ao conceito de IPFS (Interplanetary File System). O Capítulo 3 traz uma explicação sobre o que é framework e uma introdução ao framework HardHat e reúne as informações e demonstra como é a estrutura mínima necessária para o desenvolvimento fluido de um *Smart contract*. O Capítulo 4 aborda o desenvolvimento de um mercado de NFTs, destacando a criação de um aplicativo descentralizado para a mintagem (criação da NFT), apresentação dessas NFTs e venda. Capítulo 5 tem as considerações finais e trabalhos futuros.

2 CRIPTOMOEDA

Criptomoeda é um termo que surgiu para denominar uma moeda digital que utiliza criptografia como intermediário e garantidor de segurança. Sua principal diferença do mercado monetário tradicional é o fato de que as criptomoedas na sua grande maioria são descentralizadas. Dessa forma, seu valor e emissão não dependem de governos ou instituições financeiras para funcionar (NAKAMOTO, S., 2008).

A primeira criptomoeda descentralizada foi o Bitcoin, e seu criador é conhecido apenas pelo pseudônimo Satoshi Nakamoto. O fato da identidade real ser desconhecida fortalece a ideia de descentralização (NAKAMOTO, S., 2008).

2.1 Bitcoin

Satoshi Nakamoto percebeu que, entre um dos problemas das transações bancárias no mundo, encontrava-se a necessidade de um intermediário que garantisse a confiabilidade e garantia dos valores e transações. Sendo assim, as transações não são totalmente imutáveis, pois ficam à mercê da confiança de um terceiro e dos demais poderes (Governo, banco central e etc.) (NAKAMOTO, S., 2008). Tentando sugerir uma solução para esse problema, ele desenvolveu o Bitcoin. Contudo, para fazer a teoria de uma moeda digital descentralizada funcionar foi necessário pensar em uma arquitetura de software que fizesse o mesmo papel que um intermediário de pagamentos tradicional.

Satoshi Nakamoto então criou a *Blockchain,* que consiste em um livro-razão digital publicamente compartilhado via internet onde suas transações e seu histórico são inalteráveis. Para chegar nessa solução, ele combinou uma série de tecnologias e algoritmos como: uma rede peer-to-peer, com assinaturas digitais, servidores timestamp, prova de trabalho e árvore Merkle, criando assim uma alternativa às transações como eram conhecidas (NAKAMOTO, S., 2008), A seguir, exploraremos detalhadamente esses componentes e seu papel na integridade da *Blockchain.*

a) **Peer-to-peer:** Refere-se a um tipo de sistema que trabalha de maneira descentralizada. No caso das criptomoedas o conceito é aplicado a rede, ou

seja, os envolvidos em uma transação são ao mesmo tempo, clientes e servidores mútuos de maneira que não haja necessidade de um terceiro para mediar a comunicação (transação) (FANNING, S., & PARKER, B.,2000).

b) Assinatura Digital: Tecnologia que permite a verificação da integridade e autenticidade de uma mensagem. O remetente escreve a mensagem e a criptografa usando um algoritmo chave privado que só ele possui. Para o destinatário conseguir ler o remetente envia uma chave criptográfica pública que transforma a mensagem em texto plano novamente. Apenas a chave pública correta desvenda a mensagem e apenas uma chave privada consegue criar aquela chave pública. Dessa forma, o remetente tem a garantia que apenas os possuidores da chave pública irão ler a mensagem, e o destinatário tem a garantia que aquela mensagem foi realmente escrita pelo remetente (RIVEST, R. L., SHAMIR, A., & ADLEMAN, L. M,1978).

c) Servidores Timestamp: São servidores que garantem um registro de tempo de maneira altamente precisa, no caso das criptomoedas são utilizados para garantir que uma transação ocorreu precisamente em um tempo X. No caso das criptomoedas eles contribuem para impossibilitar a adulteração da ordem das transações (LAMPORT, L.,1978).

d) Prova de trabalho: Prova de trabalho ou mais conhecido como proof of work(PoW) é o método que a blockchain do Bitcoin utiliza para validar suas transações e proteger sua rede. se baseia em um sistema em que os ‘’Mineradores’’ (indivíduos ou empresas) que fornecem poder computacional e competem entre si realizando cálculos matemáticos complexos para validação dos blocos na rede, onde o primeiro participante a chegar na conclusão é recompensado em criptomoedas, e o bloco é adicionado à rede. Devido ao gigantesco poder computacional requerido para ser realizado os cálculos matemáticos para validações das transações e adição dos blocos acabam tornando a rede mais segura. Ressalta-se que na rede do Bitcoin a recompensa é em Bitcoin (NAKAMOTO, S., 2008).

e) Merkle Tree: Merkle tree é um tipo de estrutura de dados que permite a otimização da verificação e validação da totalidade dos dados sem precisar deles completos. A árvore é gerada através de agrupamentos de hashes das transações em duplas, e geram um hash novamente, é assim sucessivamente até que reste apenas um hash, o hash da raiz da árvore (cada bloco possui sua própria árvore), para validar as transações o hash raiz é validado, evitando ser necessária a validação de todas as transações do bloco, permitindo assim que a rede continue rápida e com validações constantes (DHUMWAD, S., SUKHADEVE, M., NAIK, C., MANJUNATH, K.N. AND PRABHU, S., 2017).

Existem inúmeras possibilidades de utilização do sistema de blockchain, contudo seu uso mais difundido encontra-se na comercialização de ativos digitais (NAKAMOTO, S., 2008), Nesse cenário, torna-se crucial compreender as diferenças com outra rede.

2.2 BITCOIN x ETHEREUM

Enquanto o Bitcoin tem como objetivo servir de rede e de moeda para transações de valores, o Ethereum vem com outros ideais. Vitalik Buterin, criador do Ethereum, viu que a blockchain do Bitcoin possuía uma limitação de poder, pois surgiu com a ideia de apenas transacionar Bitcoin, devido ao fato de que sua estrutura de parte programável era muito rudimentar. Assim teve a ideia de criar a blockchain do Ethereum (BUTERIN, V.,2014).

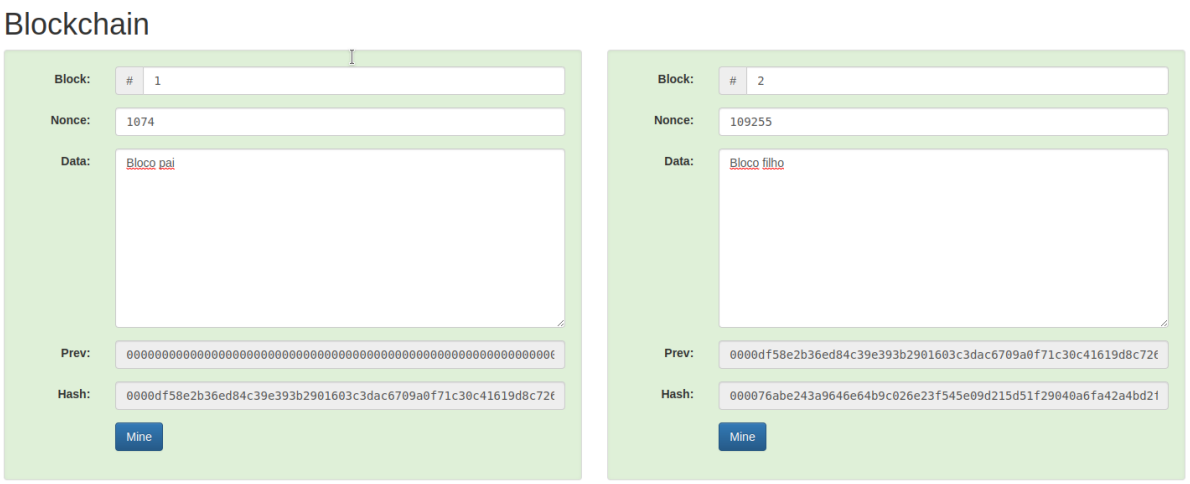
Uma *blockchain* que possuía uma linguagem de scripts melhor e que permite customizações de utilização de maneira programável. Aumentando assim a capacidade de utilização da tecnologia. Se na ideia inicial apenas Bitcoin poderia ser transacionado, agora com a Blockchain da Ethereum, não somente a criptomoeda (Ether) pode ser transacionada, como a mesma pode ser utilizada como valor de troca para contratos e negociações de outro ativos, um exemplo comum atualmente é a venda de arte digital.

2.2.1 Funcionamento da Blockchain da Ethereum

Funciona como um livro razão onde haverá o histórico de transações entre os usuários da rede. Para que essas transações sejam validadas elas primeiramente precisam ser agrupadas em um local que guarde seus dados e seu estado. Dessa premissa chega-se ao bloco (“block”).

Contudo, agrupar as transações em blocos é somente uma parte do processo, ainda é necessário que se garanta a confiabilidade do histórico. Para isso, o Ethereum espelhando-se na ideia de Blockchain utilizada no Bitcoin estipulou um sistema de herança onde um bloco sempre irá referenciar o bloco anterior (bloco pai). Este agrupamento síncrono de blocos interligados um ao outro é semelhante a uma corrente, deste motivo vem a segunda palavra formadora do termo (“chain”) (BUTERIN, V.,2014). A Figura 2 contém uma transação normal de dois blocos sem terem sidos alterados desde a criação. No exemplo cada bloco contém 5 informações: Block (número do bloco), Nonce (token que auxilia na autenticação), Data (conteúdo do bloco), Prev (Bloco anterior) e o Hash (Criptografia do bloco. É gerada uma criptografia de acordo com o conteúdo do bloco).

Figura 2 – Bloco pai e bloco filho validados



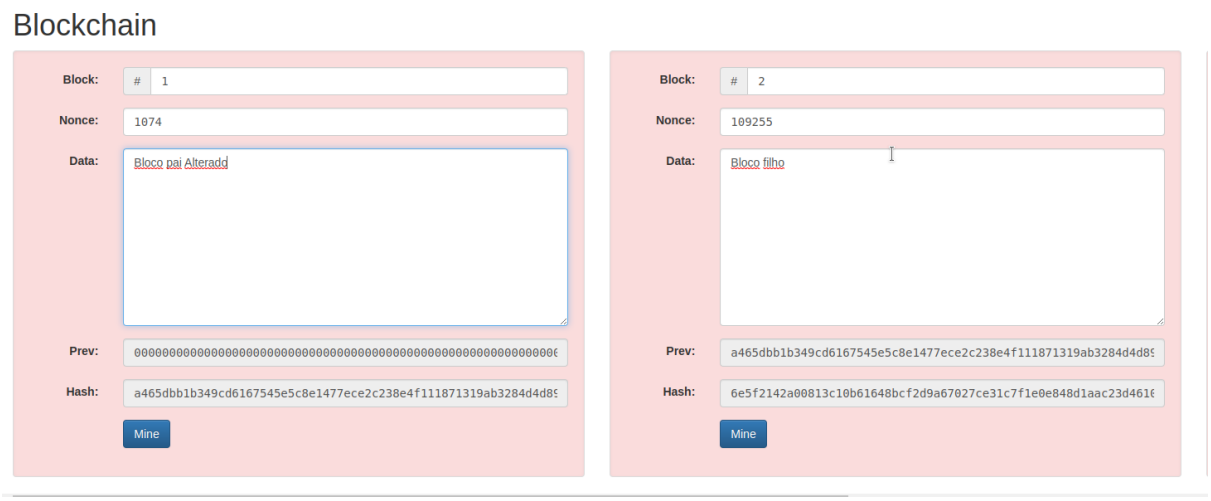
Fonte: Autoria própria.

Sintetizando, para as transações serem validadas são agrupadas em blocos, estes blocos referenciam o anterior (pai), formando uma corrente de informação que é compartilhada entre uma rede. Dessa maneira, qualquer alteração de dados feita

em um bloco irá também alterar os subsequentes, porém essa alteração somente será validada se houver o consenso da maioria da rede.

Há um exemplo do bloco pai tendo o conteúdo alterado, na Figura 3 o conteúdo do bloco pai é “Bloco pai”, na Figura 3 o conteúdo do bloco pai é alterado para “Bloco pai Alterado”, dessa forma quebrando a hash de herança com o bloco filho, pois o hash do bloco pai passa a ser diferente do que continha no filho, consequentemente quebrando toda a cadeia após a alteração do bloco pai.

Figura 3 – Bloco pai e bloco filho invalidados por alteração



Fonte: Autoria própria.

a) Tempo de resolução do bloco: As unidades de medida estipuladas pela rede Ethereum são slots e epochs, sendo slot o total de doze segundos e epoch um total de doze slots. A cada slot um validador será escolhido de maneira aleatória para ser o iniciador de um bloco, enviando o bloco para outros nós. Dentro deste tempo, outros validadores também escolhidos de maneira aleatória, irão ser escolhidos para dar o voto se o bloco é autêntico ou não. Por fim, o resultado da votação é o que a maioria estipula.

b) A importância de nós no funcionamento da rede: Todo computador que interage com a rede Ethereum deve concordar com a adição de novos blocos, assim como o estado atual da corrente de informações (consenso).

Estes computadores recebem o nome de nós (nodes). Sendo os nós os garantidores da redundância de dados entre todos os usuários da rede.

c) O conceito de consenso: Quando se utiliza o termo consenso em relação à rede Ethereum, está se firmando um acordo feito entre partes onde fica estipulado que a decisão de uma maioria formada por uma porcentagem específica da rede (66% de nós) irá determinar o que é válido. Dessa forma, toda rede deve tomar como verdade o que a maioria afirmou.

Para que seja estabelecido esse consenso de maioria na rede, um mecanismo de consenso é aplicado. Os mais comuns são prova de trabalho (proof of work), e prova de participação (proof of stake, tradução livre). Particularmente a rede Ethereum já utilizou ambos, sendo implementada inicialmente com a prova de trabalho, e atualmente após o dia 15/09/2022 trocando a implementação para prova de participação, no evento conhecido como **The Merge.** Tendo como uma das finalidades principais reduzir o consumo energético para o funcionamento da rede (DE VRIES, A., 2022).

d) Proof of Stake: O funcionamento da prova por participação, consiste do seguinte desenvolvimento: Usuários que quiserem tornar-se validadores devem aportar uma quantidade X de Eth (criptomoeda oficial da rede Ethereum) em um contrato inteligente (*Smart Contract*). Após o usuário tornar-se validador, o mesmo receberá a responsabilidade de contribuir com a confiabilidade da rede, validando novos blocos que forem enviados para ela, e também criando alguns blocos novos.

Para que haja incentivo em participar da rede, esses validadores são premiados com o Eth ao longo do tempo caso trabalhem de maneira honesta. Contudo havendo comportamento desonesto, os mesmos serão penalizados (KAPENGUT, E., & MIZRACH, B.,2022).

Assim, garante-se que os usuários ajam com honestidade, devido à grande penalidade financeira ao não seguir a mesma.

e) Processo detalhado para tornar-se um validador: Após o aporte de 32 Eth em um contrato inteligente, o novo usuário validador deve instalar um

software separado em três partes: Cliente executor, executor de consenso, e validador. Estando pronto para ser um validador, o usuário fica registrado em um sistema de fila que limita a taxa de entrada de novos validadores. Assim, a rede mantém-se organizada e estável.

Depois de inserido como um novo validador, o usuário começará a receber blocos de transações aos quais irá re-executá-los, verificar sua autenticidade e por fim emitir um voto se o bloco está ou não válido para a rede.

Diferente do sistema antigo de prova de trabalho, onde a dificuldade da resolução do hash determinava o tempo de validação do bloco. Na prova de participação, o tempo para a resolução de um bloco é fixo.

Lidando com validadores mal intencionados: Para evitar com que validadores mal intencionados interfiram maleficamente na rede de propósito, punições como perda do valor aportado foram implementadas (KAPENGUT, E., & MIZRACH, B.,2022).

Os nós validadores têm a obrigação de estar em execução. É entendido que um usuário quando se dispõe a se tornar um nó mantenha uma conexão estável e o computador disponível para participar das validações ou criações de blocos quando for solicitado, para recompensar a disponibilidade o validador é pago em Eth. Os que não participam quando solicitados são punidos com perdas das recompensas, podendo até terem suas participações excluídas (32 Eth minimo).

Por serem nós validadores, abre-se uma porta que torna possível o nó ter atitudes desonestas, ora seja para ganho pessoal ou sabotar a própria rede, são consideradas atitudes desonestas, por exemplo:

- propor vários blocos em um único slot para enviar atestados contraditórios aos outros validadores.

Caso o nó que está agindo desonestamente seja punido, a punição pode ser de 1% do valor apostado (32 Eth) para um único validador, até 100% para eventos de redução em massa (quando muitos nós validadores estão agindo desonestamente ao mesmo tempo).

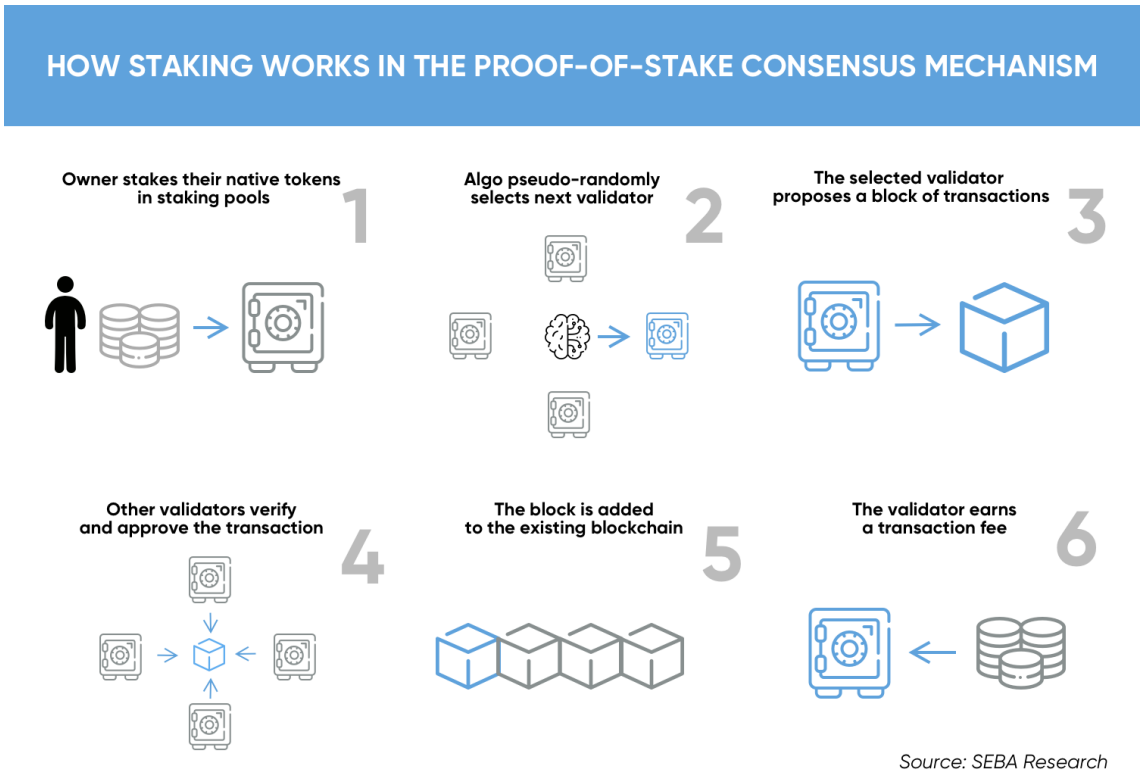
g) Comprovação da transação na rede: Uma transação só tem o status alterado para finalizada quando ela pertencer a um bloco que está no estágio

onde para sua mudança uma quantidade significativa de Eth será queimada (eliminada da carteira do validador). Para que este status receba a atualização, um método de checagem (checkpoint) é implementado:

- Cada primeiro bloco de uma epoch é escolhido como checkpoint,
- Os checkpoints são validados por pares, epoch (penúltima epoch) -> epoch (mais recente)
- Quando o número de votos válidos equivaler a mais de 66% do valor aportado na rede, o checkpoint mais recente atualiza seu estado para verificado e o checkpoint anterior atualiza seu estado para finalizado.
- Para que uma transação não seja finalizada, é preciso que o voto discordante obtenha pelo menos um terço da votação.

A Figura 4 exemplifica o funcionamento abstraído do sistema de proof of stake, onde um indivíduo coloca seus Eth em uma piscina de staking para se tornar validador, em seguida o algoritmo pseudo randômico seleciona um validador entre todos para gerar o bloco, depois seleciona outros validadores para validarem o bloco que o primeiro validador gerou, o bloco é validado e adicionado a blockchain, por fim os validadores que participaram do processo da validação ganham recompensas em Eth.

Figura 4 – Exemplificação do funcionamento do proof of stake



Fonte:(COINCODEX, 2023)

h) Gas-Fee: Toda transação feita na rede Ethereum possui uma taxa aplicada, devido ao fato da transação usar esforço computacional de terceiros para funcionar. O nome utilizado para calcular o esforço computacional na rede Ethereum é chamado de GAS. O GAS é calculado com uma unidade de medida chamada GWEI. O GWEI equivale a 0.000000001 Eth (10-9 Eth). Dessa forma, ao realizar uma transação na rede Ethereum o pagamento é feito em Etér (Eth), para calcular a quantidade de Éter a ser pago utiliza-se o GWEI, dessa forma fazendo uma regra de três, chegasse ao valor da taxa. Quanto maior o poder computacional aplicado na transação, maior o GAS-Fee a ser cobrado.

O valor de relação do GWEI com o Eth é fixo. Porém a variação do Eth com outras medidas monetárias é variável, exemplo o dólar americano. Dessa forma o valor da transação convertido para outras unidades monetárias é intermitente.

Aplicando todas as partes deste complexo sistema, a Blockchain do Ethereum consolida-se como um computador de uso geral ou conhecido como EVM (Ethereum Virtual Machine), o qual permite que qualquer usuário crie um Smart contract para adicionar funcionalidades a blockchain (Buterin, V.,2014).

2.2.2 *Smart contract*

O *Smart contract* é um conceito tecnológico que visa dar mais funcionalidade a rede Ethereum. Fornecendo uma solução que possibilita negociações que vão além de criptomoedas em uma rede distribuída. Dessa forma a blockchain se torna uma tecnologia ainda mais útil. O conteúdo dos contratos fica gravado no sistema de máquina virtual da Ethereum EVM (Ethereum Virtual Machine). Sendo os mesmos abertos à visualização do público, porém inalteráveis (ZOU, W., LO, D., KOCHHAR, P. S., LE, X. B. D., XIA, X., FENG, Y., ... & XU, B.,2019).

Os Smarts contracts são considerados Apis (Interfaces de programação que retornam dados) abertas, ou seja, podem ser utilizados por qualquer usuário, ser referenciados por outros, ou até mesmo um *Smart contract* criar outros. Eles possuem o nome *smart* pois são auto-executáveis a partir que a condição requerida por ele seja atendida, permitindo assim a criação de aplicativos descentralizados, possibilitando aos desenvolvedores criarem e implementarem serviços de complexidade baixa a elevada. Os aplicativos são executados em plataformas descentralizadas.

a) DApps: Com o surgimento da blockchain do Ethereum nasceu o termo DApps (Aplicativos Descentralizados), que são aplicativos (web, desktop ou móbiles) que utilizam Contratos inteligentes dentro da blockchain e uma interface front-end para facilitar a utilização de seus usuários, estendendo assim as vantagens que a blockchain fornece aos seus usuários.

b) Limitações dos Smart Contracts: Os contratos inteligentes possuem algumas limitações, como a do tamanho máximo que um contrato pode possuir 24 Kilobyte, isso acontece pois se um contrato tiver mais que 24

Kilobyte de tamanho ele ficará sem gás para ser completo, mas pode ser contornado com o Padrão Diamond.

Existe também limitações por causa da infraestrutura que a blockchain fornece ao ser fechada para o mundo externo, as informações de fora da blockchain não podem ser solicitadas, pois isso pode comprometer a prova do mecanismo de consenso, o que é ponto chave na blockchain. Para contornar essa limitação os *Smart Contracts* utilizam “oráculos” quando precisam de informações de fora da blockchain.

c) Oráculos: Os oráculos são *Smart Contracts* dentro da blockchain que possuem ligação com blockchains privadas, o contrato dentro da cadeia solicita os dados que quer ao contrato de fora da cadeia, o qual consulta as informações em Apis, assim criando o *Smart Contract* Híbrido, que consiste um Smart contract que tem conexão com a blockchain e com a blockchain externa (Blockchain do oráculo). Os oráculos podem servir como enviadores de informação para fora da blockchain pública, como para trazer informações das blockchains privadas. Dessa forma conectando o mundo externo ao mundo interno (BUTERIN, V.,2014).

A Figura 5 possui um exemplo de *Smart Contract* que realiza a venda de bitcoin como se fosse um caixeiro automático, clientes podem comprar bitcoin do proprietário do contrato pagando 100 Eth por bitcoin, porém apenas o dono do contrato consegue fazer a recarga do saldo de bitcoin no *Smart Contract*.

Figura 5 – Exemplificação de um *Smart Contract* que realiza a venda de bitcoin

```
// SPDX-License-Identifier: MIT
 pragma solidity 0.8 .8 ;

contract VendaDeBitcoin {

    // Declara as variáveis de estado do contrato
    address public owner;
```

```
    mapping (address => uint) public bitcoin;

    // Quando o contrato 'Venda de Bitcoin' é implantado:
       // 1. defina o endereço de implantação como proprietário do
contrato
     // 2. defina o saldo de bitcoin do contrato inteligente implantado
para 100
    constructor() {
        owner = msg.sender;
        bitcoin[address(this)] = 100;
    }

     // Permitir que o proprietário aumente o saldo de bitcoin do
contrato inteligente
    function refill(uint amount) public {
            require(msg.sender == owner, "Somente o proprietario pode
recarregar.");
        bitcoin[address(this)] += amount;
    }

    // Permitir que qualquer pessoa compre bitcoin
        function purchase(uint amount) public payable {
         require(msg.value >= amount * 100 ether, "Você deve pagar pelo
menos 100 ETH por bitcoin");
            require(bitcoin[address(this)] >= amount, "Não há bitcoin
suficientes em estoque para concluir esta compra");
        bitcoin[address(this)] -= amount;
        bitcoin[msg.sender] += amount;
    }
}
```

Fonte: Autoria própria.

2.3 IPFS (Sistema de Arquivos Interplanetário)

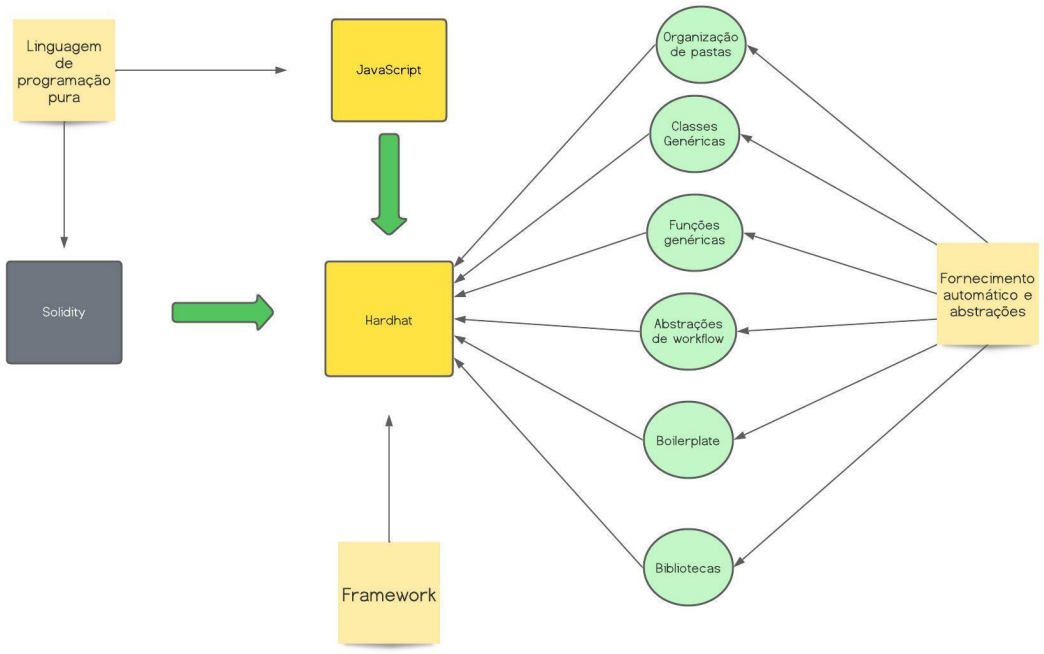
Uma rede peer-to-peer possibilita o acesso a arquivos identificados por hashes exclusivos via IPFS (Interplanetary File System). Esta tecnologia visa aumentar a durabilidade e eficiência da Internet através da criação de um sistema descentralizado de compartilhamento de arquivos. Substituir o método tradicional de servidores centralizados por uma rede de computadores distribuídos (nós) é o conceito central. A rede IPFS opera por cada nó contendo uma cópia dos dados, e quando um arquivo é solicitado, ele é localizado e entregue a partir dos nós mais próximos ou acessíveis, resultando em elevada eficiência e velocidade na acessibilidade. Endereçamento baseado em conteúdo, criptografia, prova de trabalho e árvore Merkle são utilizados pelo IPFS para garantir a segurança e integridade dos dados. O sistema descentralizado do IPFS permite a distribuição eficaz de conteúdo na Internet, ao mesmo tempo que é resistente à censura e falhas (BENET, J., ZHAO, Q., & ET AL.,2014).

3 FRAMEWORK

Para garantir uma melhora na produtividade do desenvolvimento de software, um projeto pode utilizar um “framework”. Ele é a base da construção do projeto, fornecendo um conjunto de classes que podem ser estendidas e implementadas de maneira a fornecer um padrão e uma estrutura ao software. Abstraindo assim algumas preocupações que o desenvolvedor teria no desenvolvimento (CAVICHIOLI, J. B., 2002).

A Figura 6 exemplifica a facilitação que um framework fornece ao desenvolvedor no contexto de reunir todas as ferramentas necessárias para facilitar o desenvolvimento, assim demonstrando as vantagens de utilizá-lo ao invés de apenas a linguagem de programação.

Figura 6 - Exemplo das facilitações de um framework



Fonte: Autoria própria.

3.1 IMPLEMENTANDO UM SMART CONTRACT NA REDE ETHEREUM

a) Artefatos obrigatórios a um projeto de *Smart contract*:

- Linguagem de Programação
- Conexão - Integração com a rede Ethereum
- Arquivo de código que conterá a lógica do *Smart contract*
- Arquivo de código que fará a implementação do *Smart contract* na rede
- Valor monetário (Criptomoeda da rede) que será utilizado para o pagamento da taxa de implementação

b) Artefatos Opcionais que um projeto de *Smart contract* podem possuir:

- IDE (ambiente de implementação integrado) para facilitação de visualização e escrita do código
- Framework para optimização da produtividade
- Documentação das tecnologias para facilitação da resolução de problemas técnicos

Através de todos os tipos de artefatos há a possibilidade de exemplificar o que é necessário para a implementação de um *Smart Contract*, que contém um fluxo do conjuntos dessas partes, onde inicia com o fluxo da parte esquerda com o desenvolvedor, que irá ter as ideias para desenvolver, uma IDE para facilitar o desenvolvimento, uma ou mais linguagens que supram seu desenvolvimento, seguindo um Framework para facilitar a preparação do ambiente de desenvolvimento, auxiliar com suas bibliotecas e ferramentas, conectando assim todas as metodologias e tecnologias que podem ser utilizadas para o desenvolvimento, por exemplo, para o mercado de NFT, como apresentado no próximo capítulo.

O framework HardHat foi escolhido para implementar o contrato na rede Ethereum devido ao fato de ser um dos mais populares pela heurística comum, que é a sua garantia de estabilidade no desenvolvimento e sua priorização de um desenvolvimento baseado em testes (HARDHAT, 2023). Dessa forma, o HardHat oferece um sistema completo no desenvolvimento para realizar a edição, os testes e a implementação de um *Smart Contract*.

4 MERCADO NFT

Combinando as diversas tecnologias inovadoras, concebe-se um aplicativo descentralizado com a capacidade de criar (mintar) NFTs (tokens não fungíveis) e apresentá-los na própria aplicação. Essa abordagem visa otimizar o processo de transações NFT, proporcionando uma experiência mais eficiente e direta (ETHEREUM FOUNDATION, 2023).

A plataforma oferece aos usuários uma interface, permitindo a exploração, compra e venda de ativos digitais exclusivos. Nesse ambiente inovador, os usuários podem criar NFTs de artes colecionáveis. Esses NFTs são então listados dentro do aplicativo, criando um mercado digital. Para impulsionar a participação e garantir a sustentabilidade da plataforma, é implementada uma estrutura de taxas sobre todas as transações realizadas no aplicativo.

Essa taxa não só suporta a operação contínua da aplicação, mas também estabelece um modelo de negócios viável para os desenvolvedores e mantenedores da plataforma. Essa sinergia de tecnologias não apenas simplifica a criação e transação de NFTs, mas também torna o acesso a ativos digitais exclusivos. Ao oferecer aos criadores uma nova forma de monetização e aos usuários uma experiência enriquecedora na economia digital descentralizada, a plataforma busca transformar a interação com ativos digitais de maneira inovadora (FORBES ADVISOR, 2023).

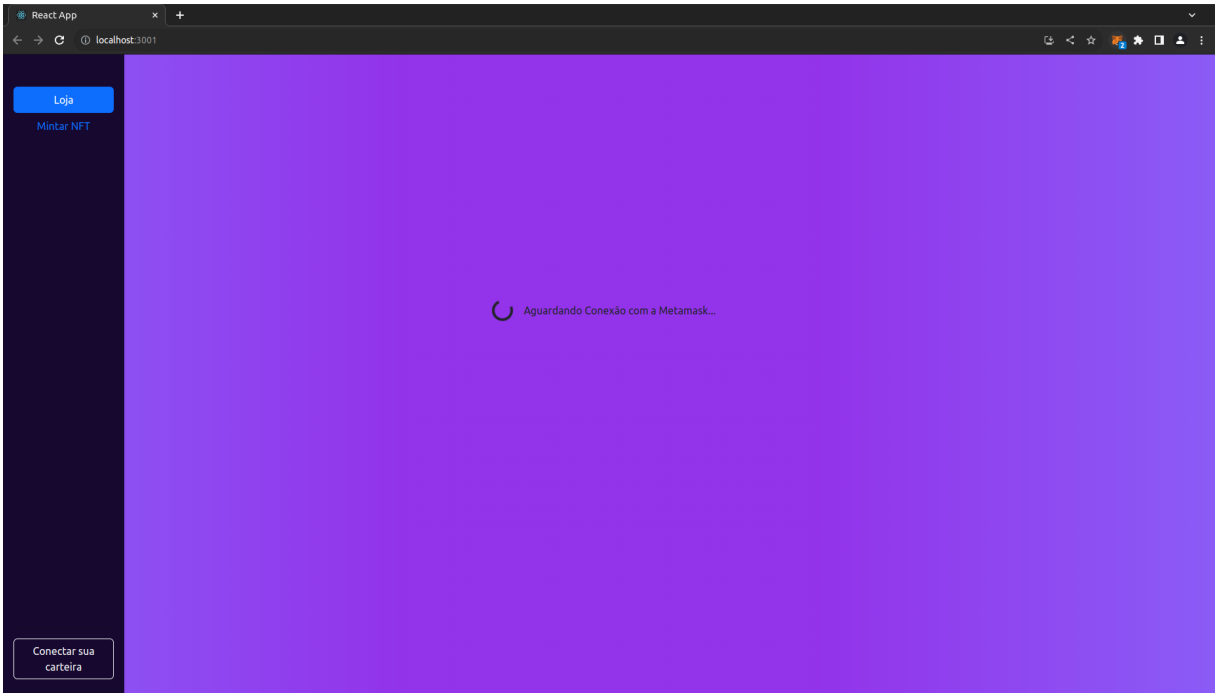
A seguir há a explicação da aplicação desenvolvida pelo autor deste trabalho, em que o objetivo principal é criar uma aplicação web que, por meio de uma carteira de terceiros, facilite ao usuário a criação e listagem de NFTs para comercializá-las com outros usuários da aplicação, tudo de forma anônima, gerando também lucro por disponibilizar a aplicação para os terceiros interessados.

4.1 ACESSANDO A APLICAÇÃO

Metamask, uma carteira digital, essencial para dar sequência à experiência (METAMASK DOCUMENTATION, 2020). Essa conexão é vital, pois permite que a aplicação se ligue à blockchain Ethereum. Assim, os usuários têm a capacidade de percorrer as páginas, explorando a criação e visualização de NFTs em um marketplace descentralizado. Se a carteira não estiver conectada, a aplicação

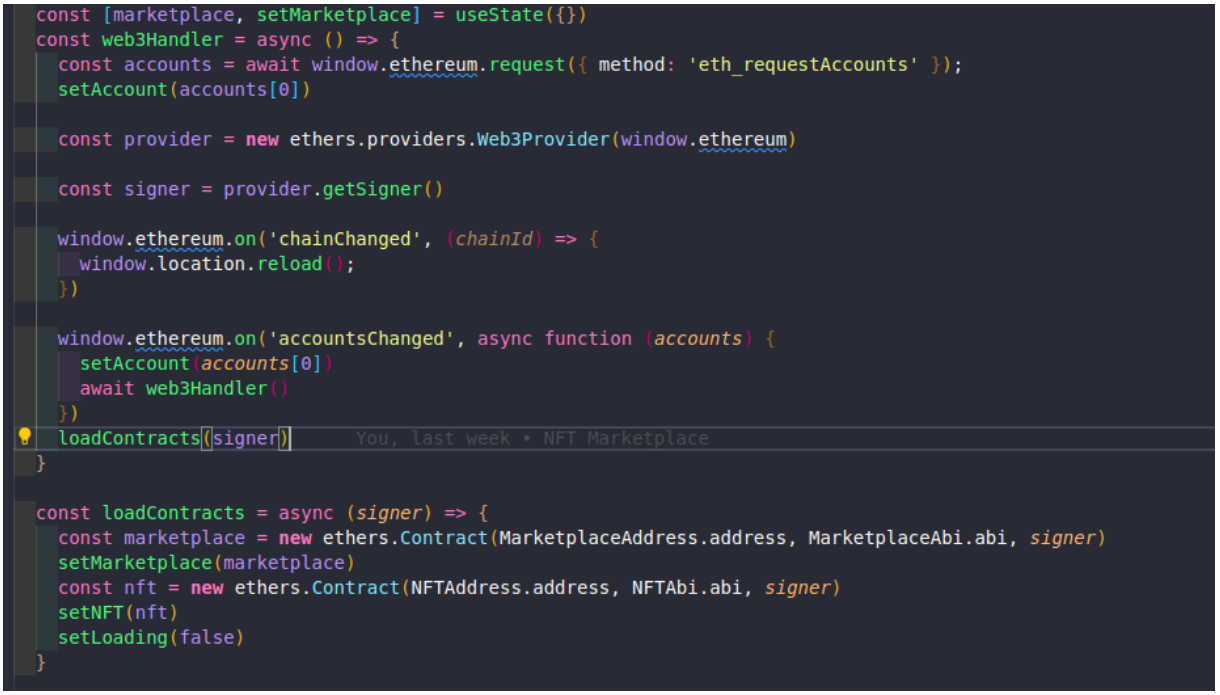
permanece em espera até que a conexão seja estabelecida. Clicando no canto inferior do menu (Conectar sua carteira), Figura 7, a aplicação solicita a conexão com a carteira. Uma vez conectada, os usuários são direcionados para a página da loja, onde podem começar a explorar e interagir com os NFTs disponíveis.

Figura 7 - Página inicial



Fonte: Autoria própria.

Figura 8 - Método chamado ao clicar em conectar sua carteira



```
const [marketplace, setMarketplace] = useState({})
const web3Handler = async () => {
  const accounts = await window.ethereum.request({ method: 'eth_requestAccounts' });
  setAccount(accounts[0])

  const provider = new ethers.providers.Web3Provider(window.ethereum)

  const signer = provider.getSigner()

  window.ethereum.on('chainChanged', (chainId) => {
    window.location.reload();
  })

  window.ethereum.on('accountsChanged', async function (accounts) {
    setAccount(accounts[0])
    await web3Handler()
  })
  loadContracts(signer)      You, last week • NFT Marketplace
}

const loadContracts = async (signer) => {
  const marketplace = new ethers.Contract(MarketplaceAddress.address, MarketplaceAbi.abi, signer)
  setMarketplace(marketplace)
  const nft = new ethers.Contract(NFTAddress.address, NFTAbi.abi, signer)
  setNFT(nft)
  setLoading(false)
}
```

Fonte: Autoria própria.

A Figura 8 contém o método que faz a solicitação de carteira, ele valida a rede da carteira e solicita a assinatura com o provider para então solicitar a leitura da das informações na blockchain e seta a conta de acordo com a carteira

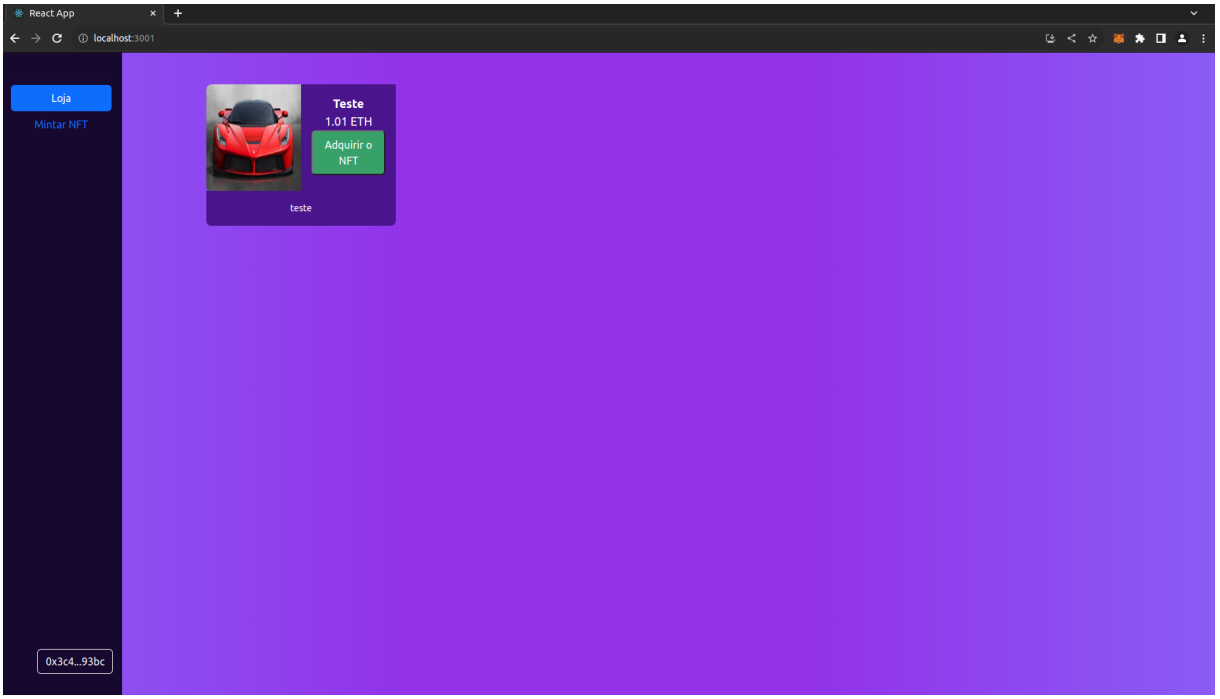
4.2 LOJA

A Figura 9 apresenta a página da loja. Nessa página é possível visualizar e adquirir os NFTs. A interface é organizada em cards que exibem informações sobre cada NFT, como nome, imagem e preço em Ethereum (ETH). Se houver NFTs disponíveis para venda, os usuários podem explorar os diferentes NFTs, se desejarem, comprar clicando no botão de comprar, o qual iniciará o *Smart Contract* designado, executando toda a regra de negócio inserida nele.

Caso nenhuma NFT esteja disponível para venda, a página estará vazia. Durante o carregamento da página, uma mensagem de "Carregando..." pode ser exibida até que os itens estejam prontos para serem exibidos. Essencialmente, essa

página proporciona um design minimalista para os usuários visualizarem, comprarem e explorarem NFTs de uma forma mais amigável, sem utilizar termos técnicos complexos.

Figura 9 - Loja com NFT já listada



Fonte: Autoria própria.

4.3 MINTAR NFT

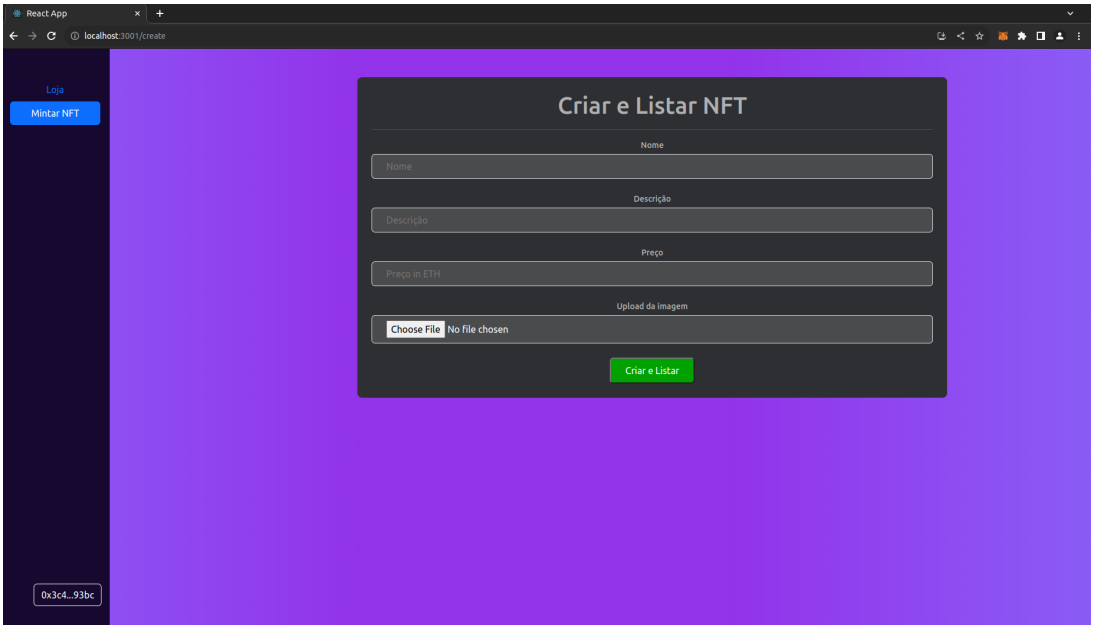
A Figura 10 apresenta o formulário que permite a cada usuário criar e listar itens NFTs. Nesse formulário, os usuários precisam incluir os detalhes como nome, descrição, preço e uma imagem. Ao preencher o formulário e inserir uma a imagem, como apresentado na Figura 11, a aplicação armazena de forma segura as informações em um IPFS.

Isso permite que a imagem e outros dados relacionados a NFT sejam compartilhados e acessados de maneira eficiente. Quando o usuário preenche todas as informações necessárias e clica no botão "Criar e listar", o sistema realiza uma série de ações nos bastidores. Ele organiza os dados do NFT, incluindo o nome, descrição e preço, e os envia para o IPFS. Em seguida, o sistema solicita as

assinaturas dos contratos e taxas de redes necessárias para criar a NFT, e com todas as taxas e permissões sendo aceitas através da interface da carteira digital, a NFT é adicionada definitivamente na blockchain Ethereum.

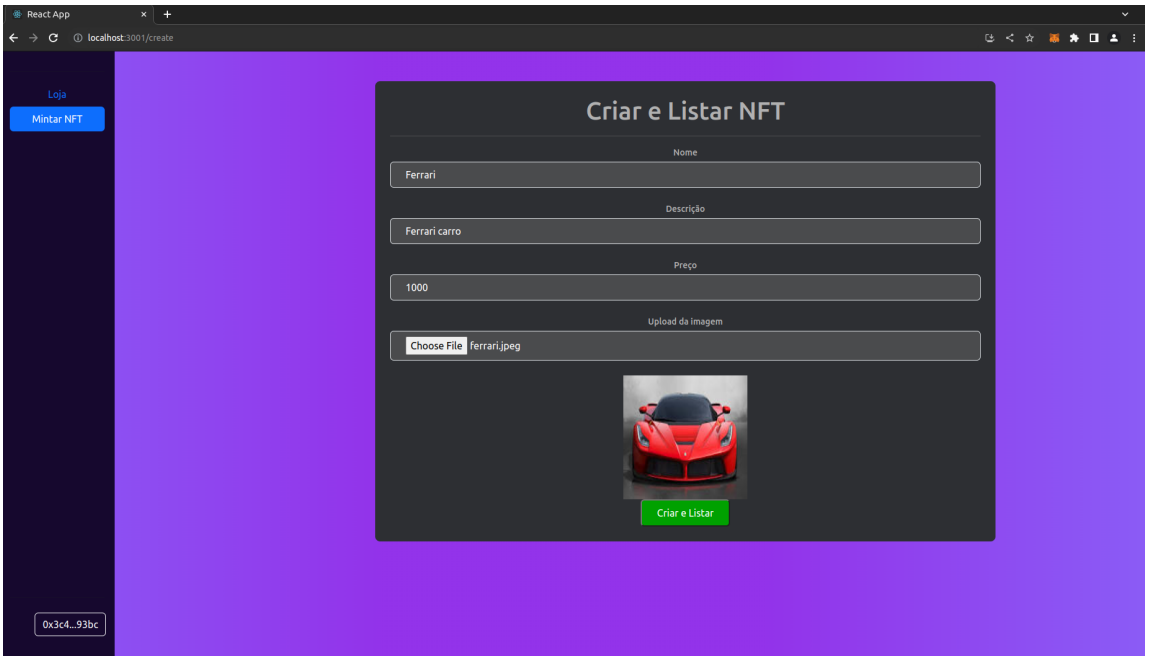
Essas ações nos bastidores são possíveis, pois a aplicação conecta o front com a função que valida todos os campos e chama o método do *Smart Contract* , o qual solicita as assinaturas da carteira do criador da requisição para efetuar o pagamentos das interações com a rede e confirmar as ações, como demonstrados nas Figuras 12,13,14 e 15.

Figura 10 - Formulário para criar e listar NFT



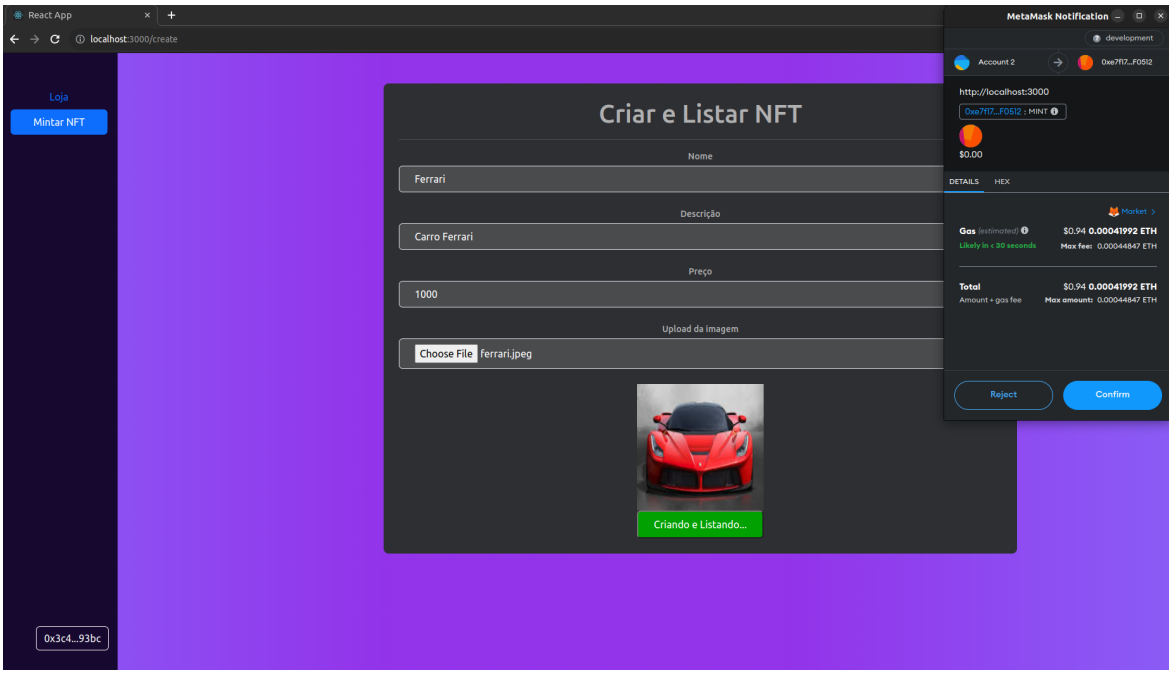
Fonte: Autoria própria.

Figura 11 - Formulário para criar e listar NFT preenchido



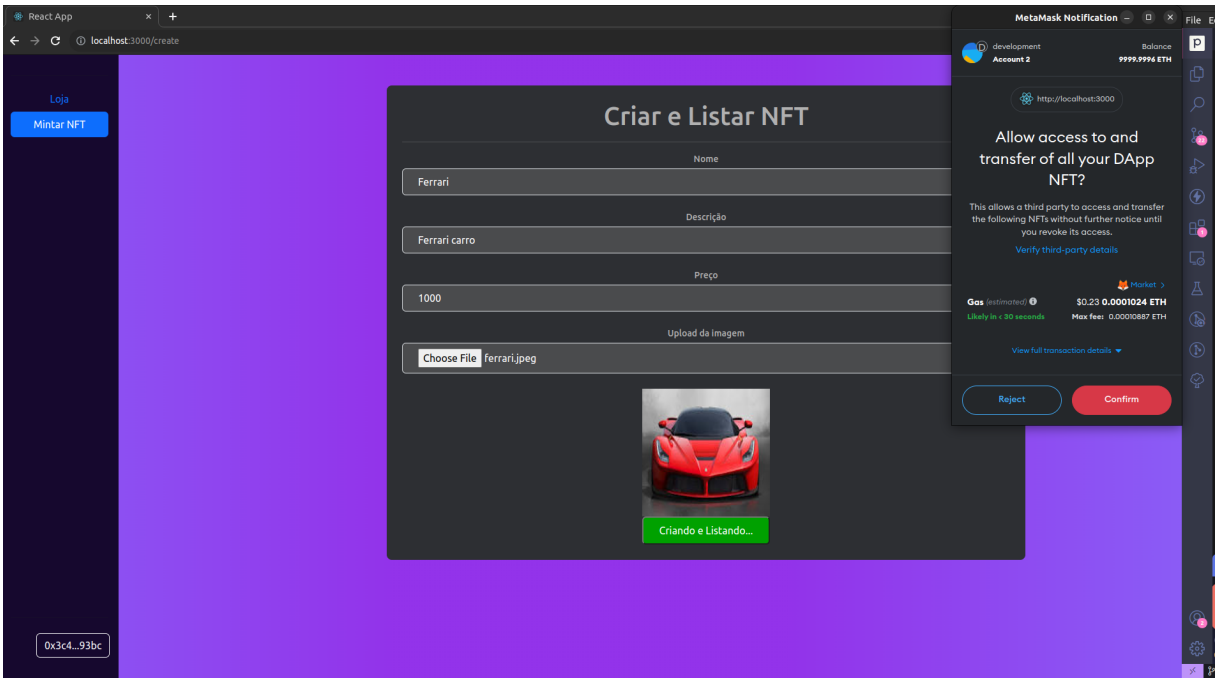
Fonte: Autoria própria.

Figura 12 - Primeira Taxa da rede ao acionar o *Smart Contract*



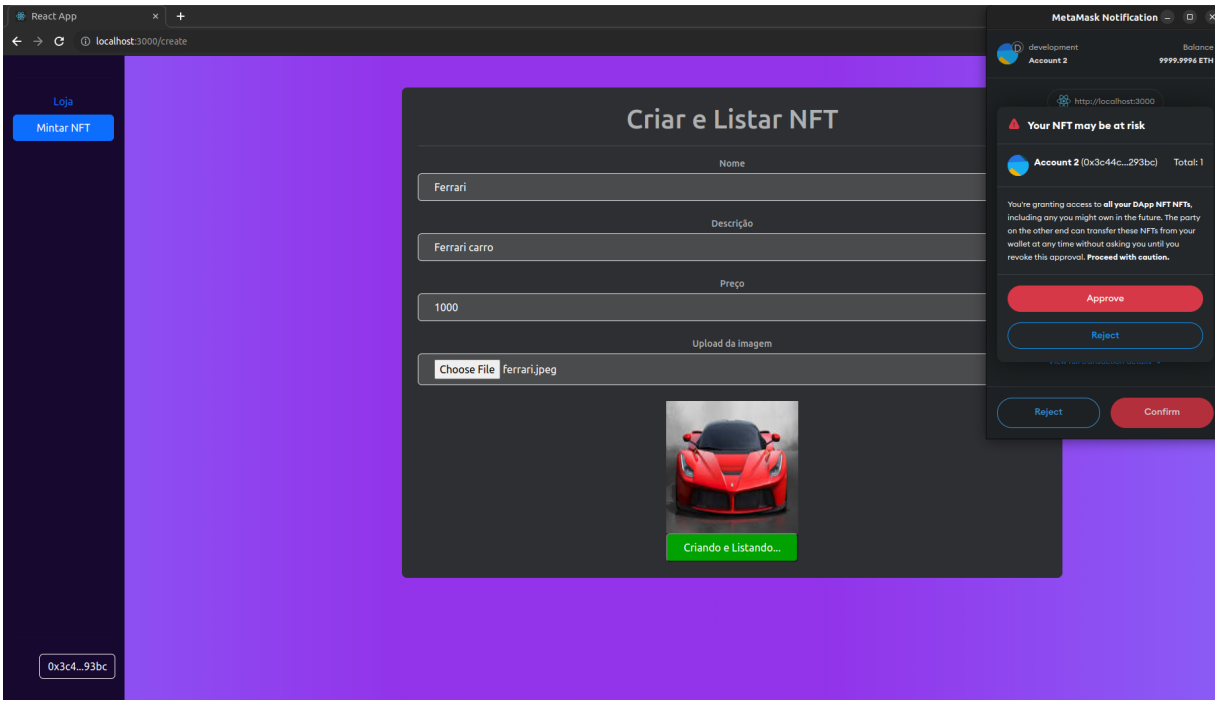
Fonte: Autoria própria.

Figura 13 - Segunda Taxa da rede ao acionar o *Smart Contract*



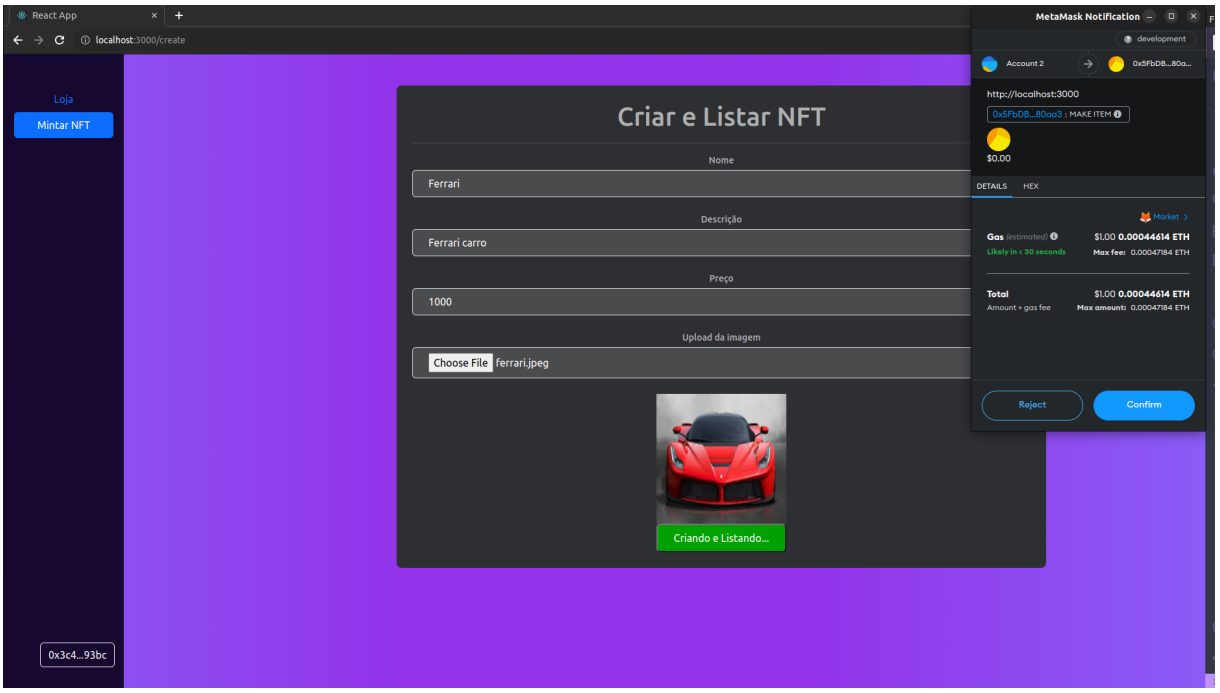
Fonte: Autoria própria.

Figura 14 - Confirmação de transferência *Smart Contract*



Fonte: Autoria própria

Figura 15 - Terceira Taxa da rede ao acionar o *Smart Contract*

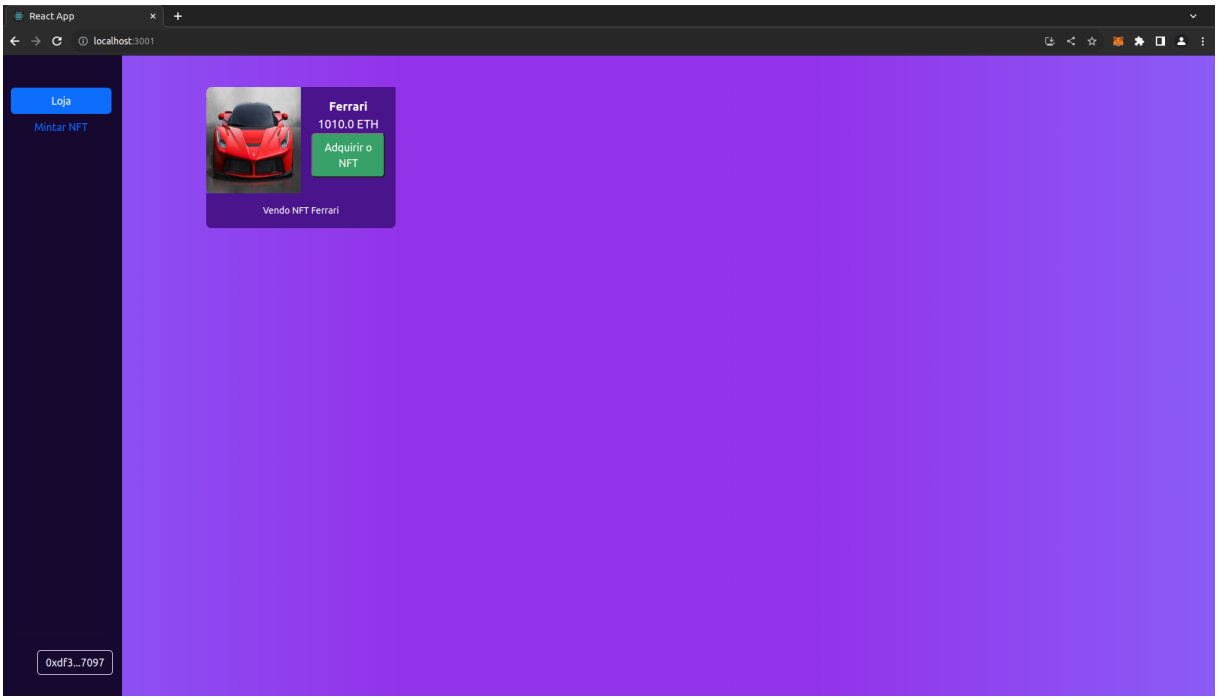


Fonte: Autoria própria.

Após a criação bem-sucedida do NFT, o usuário é redirecionado para a página inicial, Figura 16, sendo assim é possível visualizar a NFT recém mintada (criada) e listada para venda, com uma taxa a mais no valor, que ao ser vendida vai para a carteira que implementou o *Smart Contract*.

Ressalta-se que as taxas são definidas automaticamente pela rede, dependendo do nível de utilização, pode tanto ter uma taxa menor como uma maior, a taxa é cobrada pela rede para cada interação de escrita na blockchain.

Figura 16 - NFT criada e listada com sucesso



Fonte: Autoria própria.

Figura 17 - Código executado ao clicar em criar e listar

(continua)

```
const uploadToIPFS = async (event) => {
  event.preventDefault();
  const file = event.target.files[0];
  if (typeof file !== 'undefined') {
    const formData = new FormData();
    formData.append('file', file)

    const pinataMetadata = JSON.stringify({
      name: 'File name',
    });
    console.log(file);

    formData.append('pinataMetadata', pinataMetadata);
    try {
      const options = {
        method: 'POST',
```

Figura 17 - Código executado ao clicar em criar e listar

(continuação)

```
        headers: {
          accept: 'application/json',
          authorization: JWT          }
      };
      options.body = formData;

      fetch('https://api.pinata.cloud/pinning/pinFileToIPFS', options)
        .then(response => response.json())
        .then(data => {
          console.log(data.IpfsHash);
          setImage(`${data.IpfsHash}`);
        }
        )
        .catch(err => console.error(err));


    } catch (error) {
      console.log(error);
    }
  }
};



const createNFT = async () => {

  if (!image || !price || !name || !description) {
    setShowSnackbar(true);

    return;

  };

  setShowSnackbar(false);
  console.log(`ipfs://${image}`);

  const data = JSON.stringify({
    pinataContent: {
      name: `${name}`,
```

Figura 17 - Código executado ao clicar em criar e listar

(continuação)

```
description: `${description}`,
       image: `https://gateway.pinata.cloud/ipfs/${image}`,
       price: `${price}`,

     }, pinataMetadata: {
       name: "NFT",
     },
   });

   try {

     console.log(data);
                                    const       res       =       await
fetch("https://api.pinata.cloud/pinning/pinJSONToIPFS", {
       method: 'POST',
       headers: {
         'Content-Type': 'application/json',
         'Authorization': JWT
       },
       body: data

});
     const resData = await res.json()

     console.log('resultado do conjunto');
     console.log(resData);
     mintAndList(`${resData.IpfsHash}`);


   } catch (error) {
     console.log(error);
   }

 }
 const mintAndList = async (result) => {
   console.log(result);
   const uri = `https://gateway.pinata.cloud/ipfs/${result}`
   console.log(uri);
   await (await nft.mint(uri)).wait();
   const id = await nft.tokenCount()
```

Figura 17 - Código executado ao clicar em criar e listar

(conclusão)

```
console.log(id);

            await    (await    nft.setApprovalForAll(marketplace.address,
true)).wait()

   const listingPrice = ethers.utils.parseEther(price.toString())
             await    (await    marketplace.makeItem(nft.address,    id,
listingPrice)).wait()



   navigate('/');

 }

 const [selectedImage, setSelectedImage] = useState(null);
 const handleImageChange = (e) => {
   setSelectedImage(URL.createObjectURL(e.target.files[0]));
 }
```

Fonte: Autoria própria.

A Figura 17 contém o código, em que após clicar no botão `Criar e Listar` é solicitado para fazer as conexões e validações com as regras de negócios da interface e do *Smart Contract,* como por exemplo na chamada do create nft e validade se o formulário está totalmente preenchido.

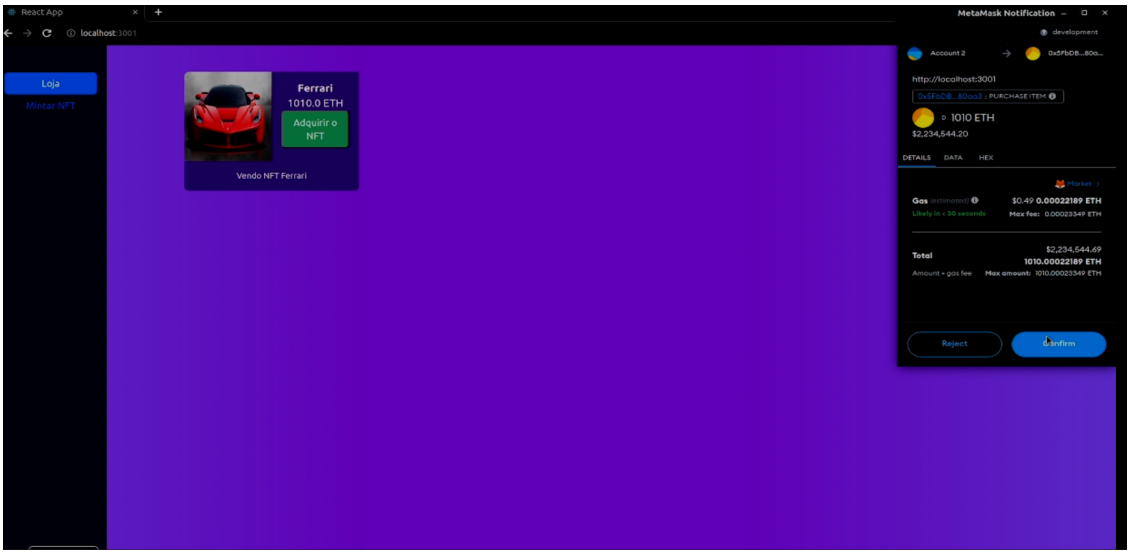
4.4 ADQUIRIR NFT

Ao acessar a página da loja e clicar em um NFT para adquirir automaticamente o *Smart Contract* é disparado, seguindo todas as regras de negócios a qual ele contém, solicitando os pagamentos de taxas da rede para sua utilização e do valor do item, em conjunto com a taxa do valor de mercado, repartindo os valores para as carteiras respectivamente.

A Figura 18 proporciona uma visão detalhada da interface da loja, destacando o processo de aquisição de um NFT. Ao clicar no botão Adquirir NFT, a imagem captura o momento em que o usuário inicia a execução de uma função específica no

Smart Contract. Neste ponto crucial, observe como o proprietário da carteira é prontamente solicitado a realizar a assinatura da transação. Essa assinatura incorpora informações essenciais, como o valor do NFT, a taxa de rede aplicada e a comissão do mercado.

Figura 18 - Taxa da rede ao acionar o *Smart Contract*



Fonte: Autoria própria.

Figura 19 - Função de compra acionada do *Smart Contract* por a interface

```
function purchaseItem(uint _itemId) external payable nonReentrant {
uint _totalPrice = getTotalPrice(_itemId);
Item storage item = items[_itemId];
require(_itemId > 0 && _itemId <= itemCount, "Esse NFT nao existe");
require(msg.value >= _totalPrice, "Voce nao possui saldo suficente");
require(!item.sold, "esse item ja foi vendido");
item.seller.transfer(item.price);
feeAccount.transfer(_totalPrice - item.price);
item.sold = true;
item.nft.transferFrom(address(this), msg.sender, item.tokenId);
emit Bought(
_itemId,
 address(item.nft),
item.tokenId,
item.price,
 item.seller,
 msg.sender);}
```

Fonte: Autoria própria.

A Figura 19 mostra a função no contrato inteligente que gerencia o processo de compra de um NFT no mercado, a qual é acionada pela interface ao clicar em adquirir NFT. Inicialmente, calcula o preço total considerando a taxa do mercado e verifica se as condições necessárias são atendidas. Se tudo estiver correto, realiza as transações financeiras, transferindo o pagamento para o vendedor e a taxa para a conta designada. O NFT é então marcado como vendido e transferido para o comprador. Finalizando a transação, um evento é emitido para registrar os detalhes da compra.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A criação do blockchain foi o marco inicial para as transações monetárias sem intermediário e a popularização das criptomoedas. A presença dessa nova tecnologia inspirou uma série de pessoas a pensar em um futuro onde a blockchain é amplamente utilizada em diversos setores.

O tempo trouxe novas ideias e necessidades para a blockchain, e foi pensando nessas necessidades que Ethereum se construiu. Seu desenvolvimento proporcionou novas possibilidades em relação a negociação de ativos digitais e anexou novas tecnologias à pauta de criptomoedas, como aplicativos descentralizados, comercialização de tokens não fungíveis e contratos inteligentes.

Além disso, pelo fato da Ethereum ter seu criador real conhecido e se apresentar como uma organização, mudanças na tecnologia puderam ser anunciadas e implementadas, atualizando-se com os problemas do contexto atual e tornando a tecnologia mais ecologicamente sustentável.

Após o estudo aprofundado e a integração cuidadosa de diversas tecnologias, desenvolveu-se uma aplicação de NFTs. O uso de Solidity permitiu a criação de contratos inteligentes que definem a lógica dos NFTs na blockchain, estabelecendo regras para sua criação, transferência e interação, garantindo a execução.

O JavaScript foi empregado no frontend para criar interfaces de usuário interativas e no backend para lidar com a lógica de negócios e interagir com os contratos inteligentes na blockchain. O framework Hardhat facilitou o desenvolvimento de contratos, emulando uma blockchain local para testes e evitando gastos com redes. A integração do IPFS armazena de forma descentralizada e eficiente imagens e metadados dos NFTs, garantindo sua disponibilidade. O mercado NFT resultante permite a criação e listagem de NFTs e compra de NFTs. Como trabalhos futuros há a possibilidade de investigar o uso das ferramentas e linguagens utilizadas neste trabalho para fazer comparativos entre frameworks de desenvolvimento blockchain, destacando suas principais diferenças e contribuições nas maiores redes de blockchain da atualidade.

REFERÊNCIAS

BENET, J., ZHAO, Q., & ET AL. (2014). **Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web.** Disponível em:https://arxiv.org/abs/1407.3561. Acesso em: 01/12/2023

BÖHME, R., CHRISTIN, N., EDELMAN, B., MOORE, T. (2015). "**Bitcoin: Economics, Technology, and Governance**." Journal of Economic Perspectives,29 (2): 213-38. Disponível em: https://www.aeaweb.org/articles?id=10.1257/jep.29.2.213. Acesso em 18 de janeiro de 2023

BUTERIN, V. (2014). **A next-generation Smart contract and decentralized application platform**. white paper, 3(37),2-1.Disponível em: https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf . Acesso em 15 de janeiro de 2023.

CAVICHIOLI, J. B. (2002). **Um framework para desenvolvimento de aplicativos cliente/servidor**.Universidade federal de santa catarina,departamento de informática e estatística curso de ciências da computação. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/183899/TCC_Joaber.pdf?sequence=-1&isAllowed=y. Acesso em: 16/01/2023

COINCODEX. **What is Proof of Stake?** Disponível em: https://coincodex.com/article/7142/what-is-proof-of-stake/. Acesso em: 14 jan. 2023.

DE VRIES, A. (2022). **Cryptocurrencies on the road to sustainability: Ethereum paving the way for Bitcoin**.Patterns,100633. Disponível em: https://www.researchgate.net/publication/366057710_Cryptocurrencies_on_the_roa

d_to_sustainability_Ethereum_paving_the_way_for_Bitcoin. Acesso em 15 de janeiro de 2023.

DHUMWAD, S., SUKHADEVE, M., NAIK, C., MANJUNATH, K.N. and PRABHU, S., (2017), September. **A peer to peer money transfer using SHA256 and Merkle tree. In 2017 23RD Annual International Conference in Advanced Computing and Communications** (ADCOM) (pp. 40-43).Institute of Electrical and Eletronic Engineers.Disponivel em: https://ieeexplore.ieee.org/document/8691933 . Acesso em Acesso em 15 de janeiro de 2023.

FANNING, S., & PARKER, B. (2000). **The Napster P2P File Sharing System. In S. Gjengset, & R. Ludwig (Eds.), Peer-to-Peer Computing** (pp. 121-137). Springer, Berlin, Heidelberg.

ETHEREUM FOUNDATION. NFTs on Ethereum - **Learn About Non-Fungible Tokens.** Disponível em: https://ethereum.org/en/nft/. Acesso em: 01/11/2023.

FORBES ADVISOR. **"NFTs: What They Are, How They Work, and Why They're Important."**Disponível em: https://www.forbes.com/advisor/investing/cryptocurrency/nft-non-fungible-token/. Acesso em: 01/11/2023.

METAMASK DOCUMENTATION. **Access a user's accounts**. Disponível em: https://docs.metamask.io/wallet/how-to/connect/access-accounts/. Acesso em : 01/11/2023.

HARDHAT. **Overview**.Disponível em: https://hardhat.org/hardhat-network/docs/overview. Acesso em: 16/01/2023

KAPENGUT, E., & MIZRACH, B. (2022). **An Event Study of the Ethereum Transition to Proof-of-Stake**. Disponível em: https://arxiv.org/pdf/2210.13655.pdf . Acesso em: 16/01/2023

LAMPORT, L. (1978). **Time, Clocks, and the Ordering of Events in a Distributed System**. Communications of the ACM, V21(7), 558-565. Disponível em:

https://lamport.azurewebsites.net/pubs/time-clocks.pdf. Acesso em 17 de janeiro de 2023

NAKAMOTO, S. (2008). **Bitcoin: A Peer-to-Peer Electronic Cash System**. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em 15 de janeiro de 2023.

RIVEST, R. L., SHAMIR, A., & ADLEMAN, L. M. (1978). **A Method for Obtaining Digital Signatures and Public-Key Cryptosystems**. Communications of the ACM, 21(2),120-126. Disponível em: https://dl.acm.org/doi/10.1145/359340.359342. Acesso em 17 de janeiro de 2023

ZOU, W., LO, D., KOCHHAR, P. S., LE, X. B. D., XIA, X., FENG, Y., ... & XU, B. (2019). **Smart contract development: Challenges and opportunities**. IEEE Transactions on Software Engineering, 47(10), 2084-2106. Disponível em:https://xin-xia.github.io/publication/tse196.pdf. Acesso em: 16/01/2023