



CENTRO UNIVERSITÁRIO 7 DE SETEMBRO - UNI7
CURSO GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

NFTStation - Tecnologia Etherium e suas Características

GABRIEL CAMPOS MONTEIRO

FORTALEZA – 2022

RESUMO

O trabalho tem por objetivo abordar a revolução das negociações na sociedade utilizando tecnologias descentralizadas inteligentes. Será apresentada a tecnologia Etherium e tendências de compras de bens não fungíveis, como NFT, que são feitos por meio de contratos inteligentes (*smart contracts*). A Ethereum, um sistema de *blockchain* descentralizado e open-source, possui sua própria criptomoeda, o Ether, que contém funcionalidades para a execução de *smart contracts* descentralizados. Tudo isso traz mais segurança nas negociações, considerando que é mais difícil encontrar fragilidades, além de permitir maior flexibilização, quando compara-se o uso da nova tecnologia com relação a um contrato jurídico normal. Neste trabalho também será mostrado um exemplo prático por meio de um software que possui as finalidades de criar e transacionar NFTs.

ABSTRACT

The term paper aims to address the negotiation revolution in society using intelligent decentralized technologies. Etherium technology and trends in purchases of non-fungible goods such as NFT, which are made through smart contracts, will be presented. Ethereum, a decentralized and open-source blockchain system, has its own cryptocurrency, Ether, which contains functionalities for executing decentralized smart contracts. All this brings more security in negotiations, considering that it is more difficult to find weaknesses, in addition to allowing greater flexibility when comparing the use of new technology with respect to a normal legal contract. This work will also show a practical example through software that has the purpose of creating and transacting NFTs.

SUMÁRIO

1 INTRODUÇÃO	5
1.1 OBJETIVO GERAL	6
1.2 OBJETIVOS ESPECÍFICOS	6
1.3 JUSTIFICATIVA	7
1.4 METODOLOGIA	8
2 FUNDAMENTAÇÃO TEÓRICA	9
3 EXPERIMENTO PRÁTICO	13
4 AVALIAÇÃO E IMPACTOS NA SOCIEDADE	28
5 CONCLUSÃO	30
6 REFERÊNCIAS BIBLIOGRÁFICAS	31

LISTA DE FIGURAS

FIGURA 1 - Representação de blocos da <i>blockchain</i>	9
FIGURA 2 - Funcionamento da <i>blockchain</i>	10
FIGURA 3 - Representação da linguagem Solidity	11
FIGURA 4 - Página home do software NFTStation	13
FIGURA 5 - Resposta no terminal após executar o backend da aplicação do NFTStation	14
FIGURA 6 - Árvore de diretório do software NFTStation	14
FIGURA 7 - Árvore de diretório dos contratos no software NFTStation	15
FIGURA 8 - Código do <i>Smart Contract</i> NFT do software NFTStation	15
FIGURA 9 - Código do <i>Smart Contract</i> Marketplace do software NFTStation	16
FIGURA 10 - Caso de uso dos Contratos utilizados no software NFTStation	17
FIGURA 11 - Código de deploy dos Smarts Contracts do software NFTStation	18
FIGURA 12 - Árvore de diretório dos metadados dos contratos no software NFTStation	19
FIGURA 13 - Tela de configuração do software Metamask	20
FIGURA 14 - Tela de importação de conta do software Metamask	21
FIGURA 15 - Menu do software NFTStation antes de ser logado	21
FIGURA 16 - Código fazendo chamada para o Login do Metamask	22
FIGURA 17 - Tela Login do software Metamask	22
FIGURA 18 - Menu do software NFTStation após ser logado	23
FIGURA 19 - Aplicação EtherScan que mostra as transações feitas na <i>blockchain</i> Etherium	24
FIGURA 20 - Tela de criar NFT no marketplace no software NFTStation	24

FIGURA 21 - Demonstração da utilização da biblioteca HTTP-IPFS Part 1	25
FIGURA 22 - Demonstração da utilização da biblioteca HTTP-IPFS Part 2	26
FIGURA 23 - Tela Explorar do software NFTStation	26
FIGURA 24 - Tela de confirmar a transação no MetaMask	27
FIGURA 25 - Tela Meus NFTs do software NFTStation	27

1. INTRODUÇÃO

A revolução das criptomoedas, moedas digitais, trouxe mudanças na forma como entendemos e lidamos com o dinheiro. Essas moedas digitais ajudaram a desenvolver uma série de tecnologias revolucionárias, com várias aplicações no mundo financeiro, como é o caso dos *smart contracts*, ou contratos inteligentes. (UAIBLOCKCHAIN, 2023)

Os *smart contracts* podem ser usados para troca de dinheiro, propriedade, informação ou qualquer outro item que as pessoas considerem apropriado para uma negociação. Por meio deles foi possível o surgimento de NFTs, non-fungible token, para bens não fungíveis. NFT é um token não fungível que é registrado em um *blockchain*, usado para certificar autenticidade e propriedade. Cria-se, assim, um novo mercado principalmente para bens digitais. Hoje é possível vender bens digitais que até pouco tempo seria impensável, tais como: terrenos digitais, artes digitais, skins de jogos digitais entre outros. (RIPIOSMARTCONTRACT, 2023)

O artigo irá mostrar a tecnologia por trás do NFT, desde do funcionamento da *Blockchain*, até um exemplo prático, um MVP de Marketplace de NFT. As tecnologias usadas no desenvolvimento do software serão descritas.

1. OBJETIVO GERAL

Este trabalho tem como objetivo contribuir com a sociedade na divulgação do software e tecnologia da Ethereum a fim de possibilitar utilizá-los para o desenvolvimento de contratos jurídicos, bem como para definir propriedade, com retorno financeiro que advém de suas transferências ou revendas. Os contratos inteligentes, portanto, atuam como uma ferramenta para implementar um contrato de venda.

1.1. OBJETIVOS ESPECÍFICOS

- Demonstrar o uso do *smart contract*
- Esclarecer o ecossistema Etherium
- Esclarecer a diferença de Bens Fungíveis e Infungíveis;
- Esclarecer o conceito de NFT
- Demonstrar um uso prático através de um experimento
- Justificar a usabilidade de *smart contracts*
- Avaliar impactos que ela poderá causar na sociedade.

1.2. JUSTIFICATIVA

O tema sugerido vem ao encontro da visão de que o mundo está em constante mudança e a tecnologia se faz relevante para a sociedade. Os meios de transações mercadológicas estão cada vez mais sofisticadas e descentralizadas. As criptomoedas e contratos virtuais já são uma realidade, porém pouco explorada, e podem passar a ser utilizadas em diversos aspectos na sociedade, como contratos jurídicos, por exemplo.

Os contratos virtuais não podem ser perdidos ou adulterados e são autoexecutáveis. Ou seja, eles garantem a segurança da execução do acordo, usando, para isso, a tecnologia *blockchain*, tornando-se mais segura do que um contrato tradicional.

Os smart contracts são contratos digitais autoexecutáveis que usam a tecnologia para garantir que os acordos firmados serão cumpridos. Estes são uma grande promessa para o futuro, que estarão cada vez mais integrados no desenvolvimento da sociedade, facilitando e desburocratizando as relações comerciais.

Nos últimos anos, tem se ouvido falar sobre NFTs e seus leilões. NFT é um token não fungível, conforme explicado mais à frente no presente texto, que é registrado em um *blockchain* usado para certificar autenticidade e propriedade. Existem diversos exemplos de NFTs, presentes no âmbito artístico, seja na música, na pintura e entre outros, utilizados principalmente em colecionáveis digitais. Esse tipo de mercado fomenta o comércio artístico, valorizando-os consequentemente. (RIPIOSMARTCONTRACT, 2023)

Avalia-se que o presente trabalho se justifica porque expõe um procedimento de criação e utilização de um NFT, que pode ser utilizado para o mercado artístico de artes digitais, como também para bens não fungíveis em geral.

1.3. METODOLOGIA

A pesquisa aplicada tem por finalidade desenvolver o conhecimento e métodos da solução para contratos inteligentes e transações descentralizadas que serão aplicados para objeto de estudo.

Ela possui caráter explicativo com objetivo de descrever um novo conhecimento que pode ser aplicado na prática e que, posteriormente, poderá ser aplicado na sociedade.

Os instrumentos de pesquisa são compostos de experimento prático através de um software e relatórios produzidos com base na análise de artigos relacionados e resultado dos testes descritos.

A análise realizada terá caráter fenomenológico, método que apresenta os dados e busca esclarecê-los exatamente da forma como ele é.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Blockchain

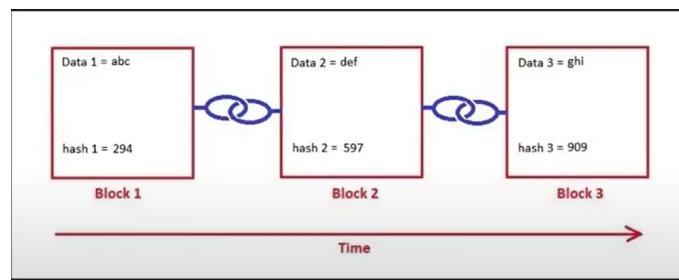
A *blockchain* é um livro-razão imutável e compartilhado que facilita o processo de registro de transações e de controle de ativos em uma rede de negócios. Qualquer item de valor pode ser controlado e comercializado nesta rede, o que reduz os riscos e os custos de todas as partes envolvidas. Nela são fornecidas informações imediatas, compartilhadas e completamente transparentes que só podem ser acessadas por membros autorizados da rede. Os membros veem as mesmas informações, sendo possível acessar os detalhes de uma transação, o que aumenta a confiança e traz novas eficiências e oportunidades. (IBMBLOCKCHAIN, 2022)

Cada transação é registrada como um “bloco” de dados que mostra o movimento de um ativo. O bloco de dados pode registrar as informações das transações, como por exemplo, o endereço do remetente e o destinatário da transação *blockchain*.

Conforme Figura 1 abaixo, cada bloco está conectado aos anteriores e posteriores. Eles formam uma cadeia de dados à medida que um ativo se move de um lugar para outro ou a propriedade muda de mãos. Os blocos confirmam a hora exata e a sequência das transações, e são vinculados com segurança para evitar que qualquer um deles seja alterado ou inserido entre dois outros existentes.

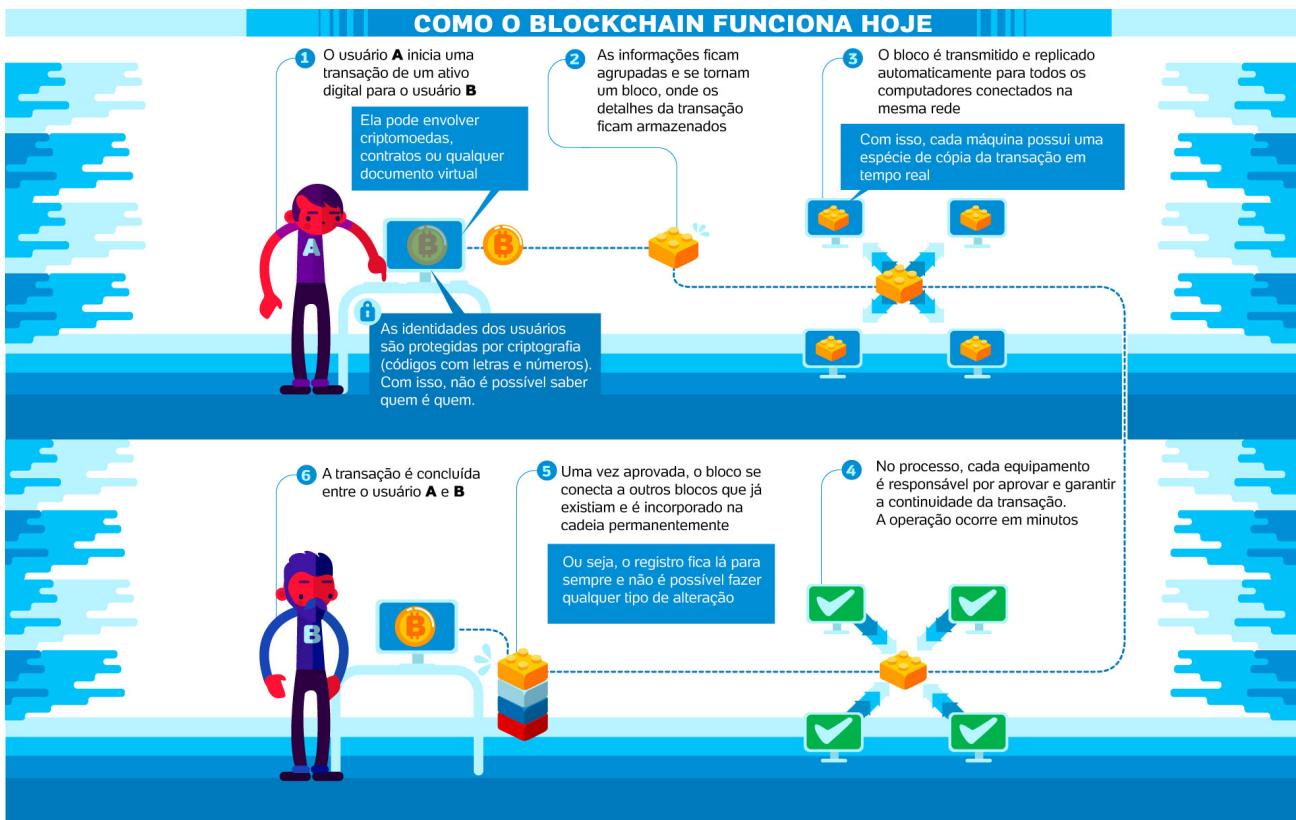
As transações são bloqueadas juntas em uma cadeia irreversível. Cada bloco adicional fortalece a verificação do bloco anterior e, portanto, de todo o *blockchain*. Isso torna o *blockchain* inviolável, fornecendo a força principal da imutabilidade.

Figura 1



A Figura 2, tem como objetivo ilustrar o funcionamento da *Blockchain*.

Figura 2



<https://conteudo.imgur.com.br/c/especiais/noticias/2018/Blockchain/blockchain-web2.jpg>

2.2. Etherium

Ethereum é uma tecnologia *blockchain* de código aberto para criar aplicativos, manter ativos, realizar transações e se comunicar sem ser controlado por uma autoridade central. Não há necessidade de entregar todos os seus dados pessoais para usar o Ethereum - você mantém o controle de seus próprios dados e do que está sendo compartilhado. A Ethereum possui sua própria criptomoeda, a Ether, que é usada para pagar atividades na rede. (ETHERIUM, 2022)

2.2.1. Etherium Virtual Machine (EVM)

A Máquina Virtual Ethereum faz parte do ecossistema *blockchain* de Ethereum. É capaz de executar uma ampla gama de instruções que permitem flexibilidade na execução de diferentes operações. A EVM interpreta comandos em Bytecode, que é um conjunto de

instruções projetadas para execução eficiente por um interpretador de software. (ETHERIUM, 2022)

2.2.2. Solidity

Figura 3

```

1 // SPDX-License-Identifier: UNLICENSED
2 pragma solidity >=0.4.17 <0.9.0;
3
4 contract Storage {
5     uint public data;
6
7     constructor (uint defaultData) {
8         data = defaultData;
9     }
10
11    function set(uint newData) public {
12        data = newData;
13    }
14 }
15

```

<https://blog.logrocket.com/wp-content/uploads/2021/11/contract-code.png>

Para facilitar a programação da EVM (máquina virtual), programadores utilizam uma linguagem especializada de alto nível chamada Solidity. Por meio dessa linguagem de programação, podem ser desenvolvidos às criações dos *smart contracts*. Inicialmente, os comandos são transformados em códigos de operação (OPCODES) e depois para um bytecode que será executado pelo EVM. É uma linguagem voltada para orientação objeto, possuindo heranças e métodos construtores. A Figura 3 acima, demonstra a sintaxe da linguagem (SOLIDITY,2022).

2.3. Smart Contracts

Contratos inteligentes (*Smart Contracts*) são programas armazenados em um blockchain que são executados quando condições predeterminadas são atendidas. Eles normalmente são usados para automatizar a execução de um acordo para que todos os participantes possam ter certeza imediata do resultado, sem o envolvimento de qualquer intermediário ou perda de tempo. Eles também podem automatizar um fluxo de trabalho, acionando a próxima ação quando as condições forem atendidas (IBM CONTRACTS, 2022)

2.4. Bens fungíveis e infungíveis

São fungíveis os objetos que podem substituir-se por outros de mesma espécie, qualidade e quantidade. Exemplo: caneta, dinheiro, 1 kg de feijão. Os bens infungíveis ou não fungíveis são bens que não podem substituir por outro da mesma espécie, qualidade, quantidade. Como uma casa, um quadro de um artista famoso ou terrenos digitais. (BENS, 2022)

2.5. NFT

NFT é a sigla para o termo non fungible token ou "token não fungível". NFTs são tokens criptográficos, ou seja, códigos numéricos com registro de transferência digital que garantem autenticidade aos seus donos, estes são hospedados em uma *blockchain* e pode representar um ativo digital na forma de um arquivo JPEG, GIF ou MP4

Os NFTs são cunhados por meio de contratos inteligentes que atribuem propriedade e a re-atribuem quando transferidos ou revendidos. Os contratos inteligentes atuam como uma ferramenta para implementar acordos de venda. Eles também garantem que os NFTs não possam ser divididos e garantem que os ativos digitais sejam únicos e não replicáveis.

Qualquer *blockchain* tem suporte aos programas de *smart contracts*, sendo que os padrões ERC 721 e ERC 1155 (OPENZEPPIN, 2022) são exemplos comuns de definições para uso de tokens infungíveis .

O NFT possui metadados, que são conjuntos de dados que os compõem e, geralmente, são especificados em formato JavaScript Object Notation (JSON). Os metadados de um NFT podem descrever suas características e propriedades, tais como seu nome, descrição, histórico de transações, atributos, link para uma imagem hospedada e muito mais

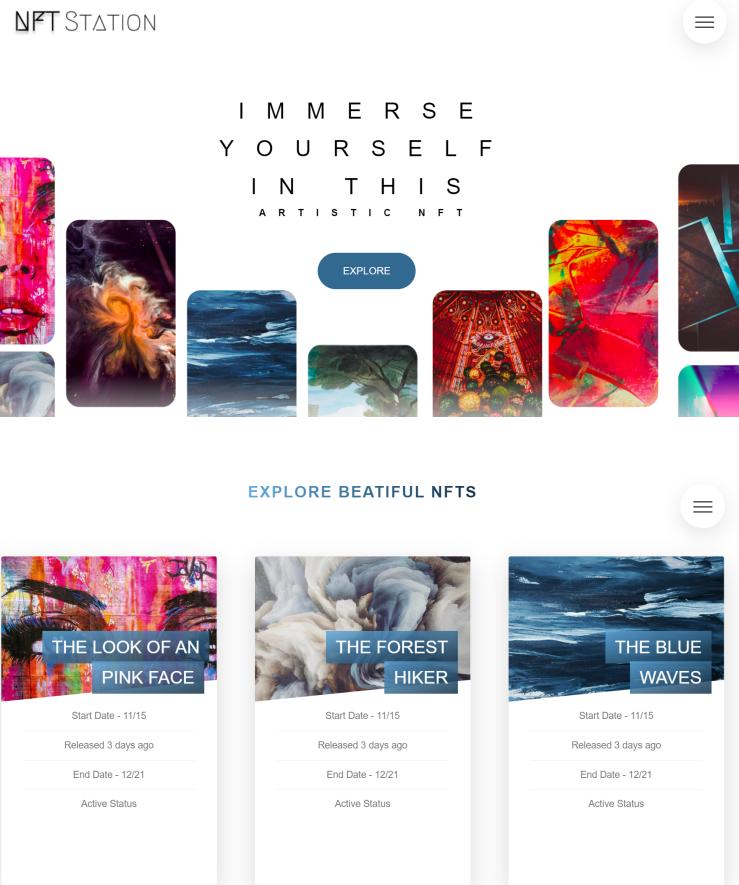
2.6. ERC 721

ERC 721 é uma interface padrão de *smart contract* para criar tokens não fungíveis. Esse padrão permite implementação de uma API padrão para NFTs em contratos inteligentes. Este padrão fornece funcionalidade básica para rastrear e transferir NFTs. Este será o padrão escolhido para o desenvolvimento da aplicação deste artigo (EIP-721, 2022).

3. EXPERIMENTO PRÁTICO

3.1. NFT Station

Figura 4



O Software desenvolvido para esse artigo é um MVP¹ de um marketplace de NFT denominado NFTStation. Possui funções de criar e vender NFTs além de listar os vendidos da conta logada. Na Figura 4 observa-se a página principal do software do experimento e com o cabeçalho e a seção de explorar NFTs.

¹ Um produto mínimo viável é uma versão de um produto com recursos suficientes para ser usado pelos primeiros clientes

3.1.1. Backend - HardHat, OpenZeppelin, Ethers

Para desenvolver o backend, foi utilizado o ambiente de desenvolvimento para software Ethereum chamado Hardhat. Ele consiste em diferentes componentes para editar, compilar, depurar e implantar seus contratos inteligentes e apps, todos trabalhando juntos para criar um ambiente de desenvolvimento completo. Ao rodar o ambiente de forma local, o mesmo cria um ambiente Etherium local com 10 contas fakes e com 10000 ETH² cada uma, como é representado na figura 5 (HARDHAT, 2022).

Figura 5

```
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/
Accounts
=====
WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39Fd6e51aad88F6F4ce6aB8827279cffB92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

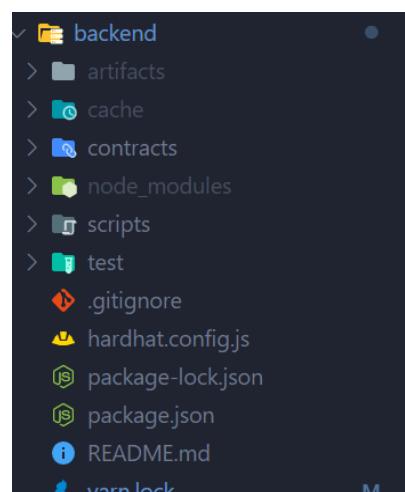
Account #1: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3C44CdD6a900fa2b585dd299e03d12FA4293BC (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90F79bf6EB2c4f870365E785982E1f101E93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6
```

Para criar o projeto node de amostra, é necessário executar ‘npx hardhat’ na pasta do projeto onde o ambiente cria um conjunto de arquivos e diretórios conforme é ilustrado na Figura 6

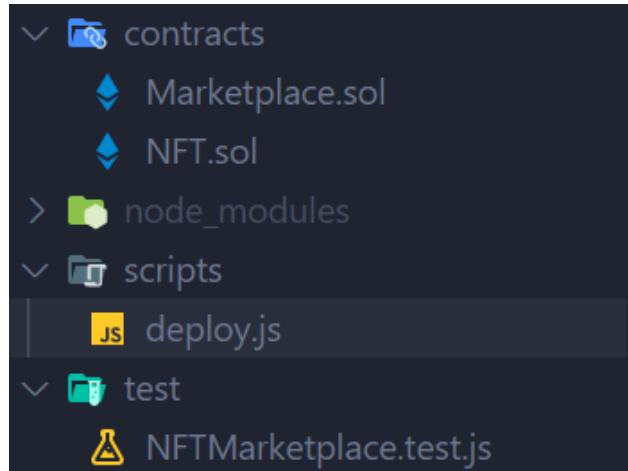
Figura 6



² ETH é o símbolo da moeda ETHER da Etherium

Na Figura 7 apresenta a pasta contracts, local onde são desenvolvidos os *Smart Contracts* utilizando a linguagem solidity.

Figura 7



Na imagem da Figura 8 demonstra um exemplo das regras de um *smart contract*. Para realização deste contrato foi utilizado o padrão ERC 721 através de uma biblioteca chamada Open Zeppelin. Essa biblioteca fornece os padrões para aplicativos *blockchain* seguros e produtos de segurança para construir, automatizar e operar aplicativos descentralizados.

Figura 8

```

  NFT.sol 2, M ×
  backend > contracts > NFT.sol
  1  // SPDX-License-Identifier : MIT
  2  pragma solidity ^0.8.4;
  3
  4  import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
  5
  6  contract NFT is ERC721URIStorage {
  7      uint public tokenCount;
  8      constructor() ERC721("Gcamon NFT", "Gcamon"){}
  9
 10 }
 11
 12 function mint(string memory _tokenURI) external returns (uint){
 13     tokenCount++;
 14     // Safely mints tokenId and transfers it to to.
 15     _safeMint(msg.sender, tokenCount);
 16     // Sets _tokenURI as the tokenURI of tokenId.
 17     _setTokenURI(tokenCount, _tokenURI);
 18     return(tokenCount);
 19 }
 20 }
  
```

Conforme observa-se na figura acima, para criar um NFT foi necessário chamar a função herdada do padrão ERC 721 chamada `_safeMint`. A função recebe o endereço da conta e o id do token. Neste caso foi utilizado uma variável global da Etherium chamada `msg.sender` que retorna o endereço da conta conectada. Depois é chamado a função `_setTokenURI` que define os metadados do NFT recém criado. Esta função tem como parâmetros correspondentemente o id do token e a uri do token. É importante observar que, quando o token não é atribuído manualmente no construtor, a própria Etherium atribui 0 como no caso da variável do `TokenCount` da imagem anterior.

Outro caso de contrato estudado é o marketplace. Na imagem abaixo, além de ser utilizado o padrão ERC 721, também foi utilizado a interface de segurança `ReentrancyGuard`, sendo ele um modificador disponível, que pode ser aplicado a funções para garantir que não haja chamadas aninhadas (reentrantes) para elas.

Para transferir o NFT é chamado o método `transferFrom`. O método recebe o endereço do remetente, o endereço do destinatário e o token id do NFT desejado. Na imagem abaixo, demonstra um exemplo de um contrato com as regras de um Marketplace e na segunda imagem abaixo contém o método `transferFrom`. A Figura 9 apresenta o método citado.

Figura 9

```

event Bought (
    uint itemId,
    address indexed nft,
    uint tokenId,
    uint price,
    address indexed seller,
    address indexed buyer
);
//
mapping(uint => Item) public items;
[

constructor(uint _feePercent){
    feeAccount = payable(msg.sender);
    feePercent = _feePercent;
}

function makeItem( IERC721 _nft, uint _tokenId, uint _price) external nonReentrant {
    require(_price > 0 ), "Price must be greater than zero";
    //increment itemCount
    itemCount++;
    //transfer nft
    _nft.transferFrom(msg.sender, address(this), _tokenId);
    items[itemCount] = Item (
        itemCount, _nft,_tokenId,_price, payable(msg.sender), false
    );
    // emit Offered event
    emit Offered(itemCount, address(_nft), _tokenId, _price, msg.sender);
}

function purchaseItem(uint _itemId) external payable nonReentrant {
    uint _totalPrice = getTotalPrice(_itemId);
    Item storage item = items[_itemId];
    require(_itemId > 0 && _itemId <= itemCount, "item doesn't exist");
    require(msg.value >= _totalPrice, "not enough ether to cover item price and market fee");
    require(item.sold, "item already sold");
    //pay seller and feeAccount
    item.seller.transfer(item.price);
    feeAccount.transfer(_totalPrice - item.price);
    // update item to sold
    item.sold = true;
    // transfer nft to buyer
    item.nft.transferFrom(address(this), msg.sender, item.tokenId);

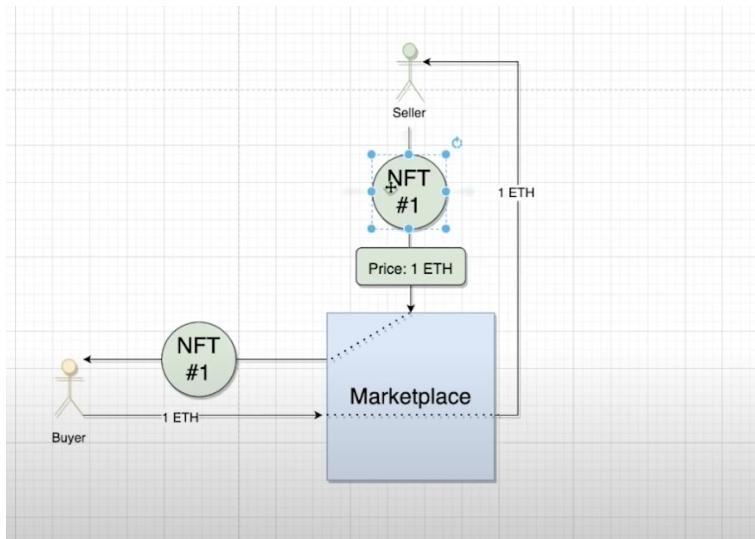
    //emit Bought event
    emit Bought(_itemId, address(item.nft), item.tokenId, item.price, item.seller, msg.sender);
}

function getTotalPrice(uint _itemId) view public returns(uint) {
    return(items[_itemId].price*(100 + feePercent) / 100);
}
]

```

No final, o backend resultou no caso de uso da Figura 10, onde o NFT é vendido para o marketplace e outra pessoa pode comprar com uma pequena taxa abatida no preço.

Figura 10



No arquivo `deploy.js` dentro da pasta `scripts`, o Hardhart permite que o usuário possa fazer o deploy. Nesse caso foi utilizado um *Contract Factory* para adquirir uma instância dos contratos e assim permitir que o usuário possa manipular o objeto do contrato. Para utilizar o *Contract Factory* e outros comandos da *Etherium* em *javascript* é necessário utilizar uma biblioteca Ethers. A biblioteca ethers.js é completa e compacta para interagir com o *Ethereum Blockchain* e seu ecossistema (ETHERS.JS, 2022). A imagem abaixo demonstra o uso da biblioteca *Ethers* e Contract Factory através da chamada `ethers.getContractFactory`. apresenta o deploy para a *blockchain etherium local* e a escrita dos metadados em formato de json.

Figura 11

```

deploy.js  X

backend > scripts > deploy.js > main > marketplace

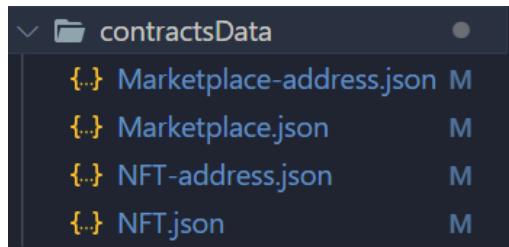
1  const { ethers } = require("hardhat");
2
3  async function main() {
4    const [deployer] = await ethers.getSigners();
5
6    console.log("Deploying contracts with the account:", deployer.address);
7    console.log("Account balance:", (await deployer.getBalance()).toString());
8
9    // deploy contracts here:
10   const NFT = await ethers.getContractFactory("NFT");
11   const nft = await NFT.deploy();
12   const Marketplace = await ethers.getContractFactory("Marketplace");
13   const marketplace = await Marketplace.deploy([1]);
14
15   console.log("NFT contract address", nft.address);
16   console.log("Marketplace contract address", marketplace.address);
17
18   // For each contract, pass the deployed contract and name to this function to
19   saveFrontendFiles(nft, "NFT");
20   saveFrontendFiles(marketplace, "Marketplace");
21 }

23   function saveFrontendFiles(contract, name) {
24     const fs = require("fs");
25     const contractsDir = __dirname + "/../../src/contractsData";
26
27     if (!fs.existsSync(contractsDir)) {
28       fs.mkdirSync(contractsDir);
29     }
30
31     fs.writeFileSync(
32       contractsDir + `/ ${name}-address.json`,
33       JSON.stringify({ address: contract.address }, undefined, 2)
34     );
35
36     const contractArtifact = artifacts.readArtifactSync(name);
37
38     fs.writeFileSync(
39       contractsDir + `/${name}.json`,
40       JSON.stringify(contractArtifact, null, 2)
41     );
42   }
43
44   main()
45     .then(() => process.exit(0))
46     .catch((error) => {
47       console.error(error);
48       process.exit(1);
49     });

```

No final do deploy do experimento, foram criadas metadados sobre os contratos em formato de JSON ³para serem utilizadas e manipuladas no Front-end, também podendo ser utilizado via restful ⁴ como é representado na Figura 12.

Figura 12



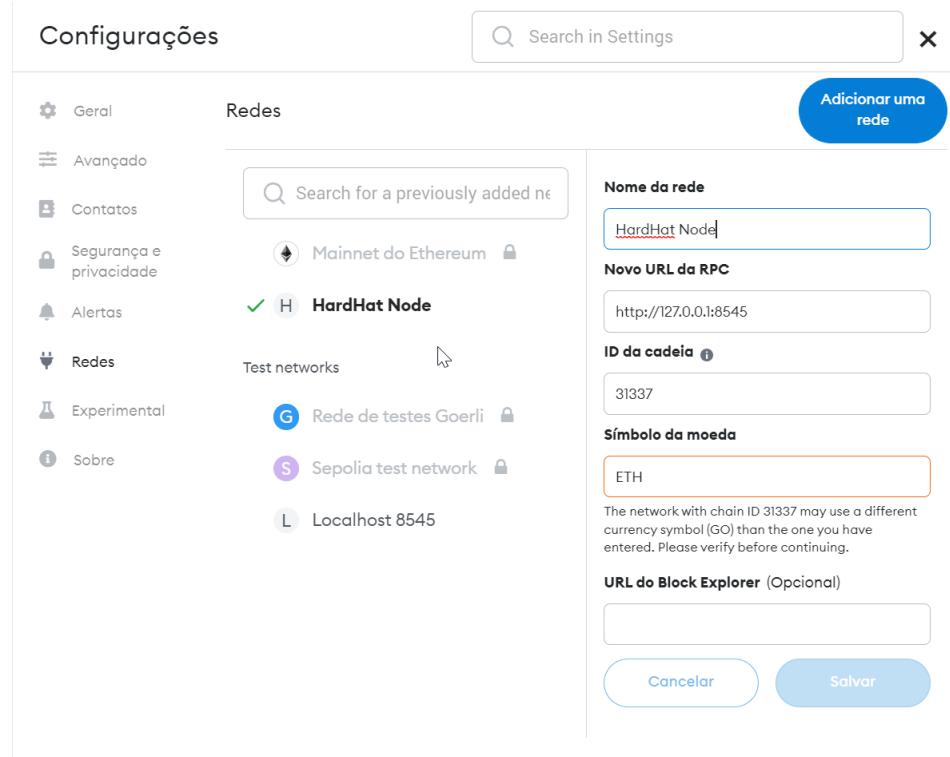
3.1.2. Fluxos de Telas - MetaMask, IPFS, Infura, Etherscan.

Para logar no software do experimento é necessário criar uma conta na MetaMask. Ela é software de uma carteira de criptomoedas usada para interagir com o *blockchain* Ethereum. Ele permite que os usuários accessem sua carteira por meio de uma extensão de navegador ou aplicativo móvel, que pode ser usado para interagir com aplicativos descentralizados.

É necessário configurar e adicionar a rede local da Etherium no software MetaMask para funcionar o experimento deste artigo como é mostrado na Figura 13.

³ JSON é um formato de arquivo padrão aberto e formato de intercâmbio de dados que usa texto legível por humanos para armazenar e transmitir objetos de dados que consistem em pares atributo-valor e matrizes.

⁴ A transferência de estado representacional é um estilo de arquitetura de software que descreve uma interface uniforme entre componentes separados fisicamente, geralmente na Internet em uma arquitetura cliente-servidor.

Figura 13

Na Figura 14 é a tela onde se pode importar uma conta. No caso do experimento foi importado conta fake gerada pelo Hardhat.

Figura 14

Importar conta

As contas importadas não estarão associadas à Frase de Recuperação Secreta da conta da MetaMask criada originalmente. Saiba mais sobre as contas importadas [aqui](#)

Selecione o tipo: Chave privada

Cole a string da sua chave privada aqui:

[Cancelar](#) [Importar](#)

Na Figura 15, é mostrada a tela do Menu com o usuário deslogado. Ao Logar no Fluxo da Tela da Navegação, utiliza-se a biblioteca Ethers para conectar a Wallet MetaMask utilizando uma chamada como no exemplo ilustrado na figura 16 e 17 (METAMASK, 2022).

Figura 15

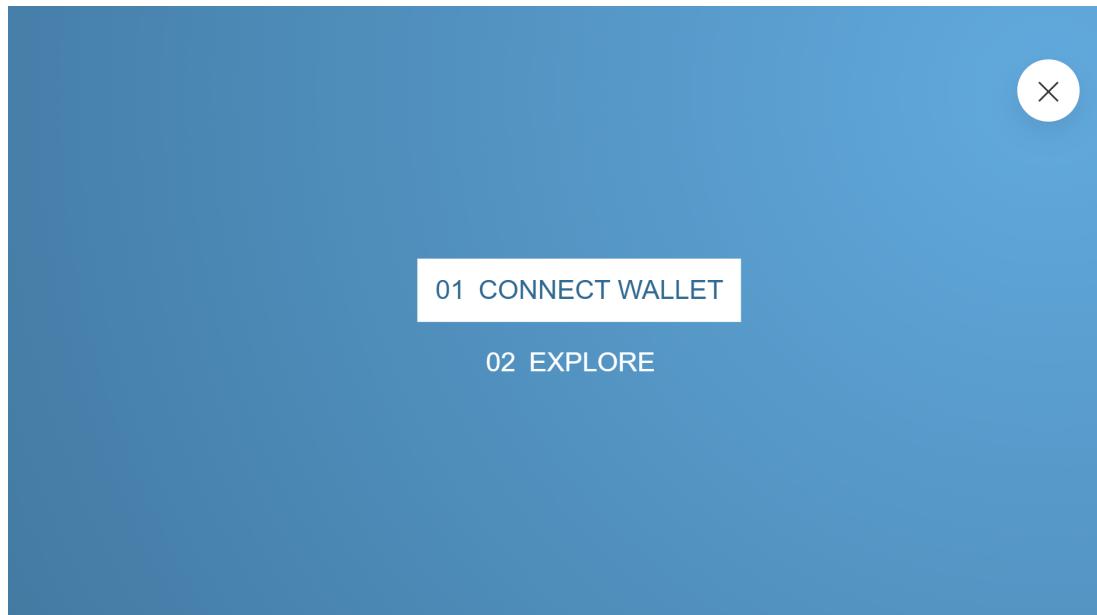


Figura 16

```
//MetaMask Login/Connect
const web3Handler = async () => {
  const accounts = await window.ethereum.request({
    method: 'eth_requestAccounts',
  })

  // Get provider from MetaMask
  const provider = new ethers.providers.Web3Provider(window.ethereum)
  // MetaMask requires requesting permission to connect users accounts
  await provider.send('eth_requestAccounts', [])
  // Set signer
  const signer = provider.getSigner()

  await loadContracts(signer)
  setAccountWeb3(accounts[0])
}
```

Figura 17

Bem-vindo(a) novamente!

A web descentralizada te aguarda

Senha

Desbloquear

[Forgot password?](#)

Precisa de ajuda? Contate [Suporte da MetaMask](#)

A Figura 18 é apresentada após o usuário logar o sistema. A tela de navegação mostra as seguintes opções:

Opção 1 onde pode ser visto as transações do usuário;

Opção 2 é onde está os NFTs que podem ser comprados no marketplace;

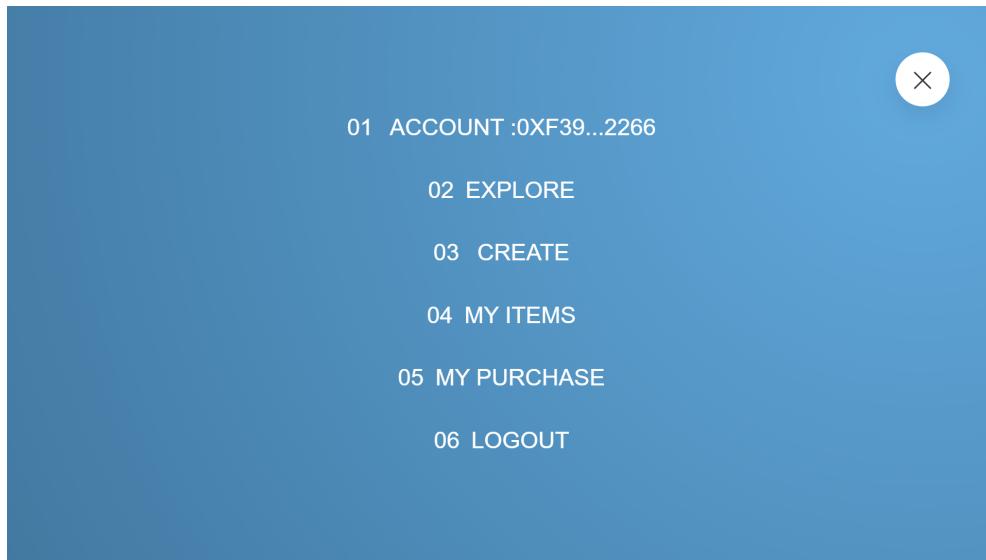
Opção 3 que é a Tela de criação do NFT para ser vendido;

Opção 4 onde o usuário poderá ver os próprios NFTs criados

Opção 5 é a Tela de ver os NFT vendidos do usuário

Opção 6 é a opção de deslogar

Figura 18



Caso o usuário clique na primeira opção, uma nova aba abrirá, mostrando as informações da conta daquele usuário na Etherscan como é apresentado na Figura 19 logo abaixo. (ETHERSCAN, 2022) Este é um explorador de blocos para a *blockchain* Ethereum. Ele permite que os usuários pesquisem e naveguem facilmente por transações e blocos. Ele também fornece informações sobre cada transação e bloco, como o hash e o timestamp.

Figura 19

Etherscan

Eth: \$1.196.14 (+1 29%) | ⚡ 11 Gwei

All Filters Search by Address / Txn Hash / Block / Token / ENS

Home Blockchain Tokens Resources More Sign In

Address 0x39Fd6e51aad88F6F4ce6aB8827279cffB92266 Buy Exchange Earn Gaming

Sponsored: Get the most out of your assets and relax with Vault Automations on [Oasis.app](#)

Overview

Balance: 0.000047768296076782 Ether

Ether Value: \$0.06 (@ \$1.196.14/ETH)

Token: \$0.00 [Buy](#)

More Info

My Name Tag: Not Available, [login to update](#)

More

Ad

JOIN NOW

HERE FOR ALL GAMES
18.8 BTC UP FOR GRABS

BTC365

Transactions Internal Txns Erc20 Token Txns Erc721 Token Txns Erc1155 Token Txns Analytics Comments

Latest 25 from a total of 663 transactions (+1 Pending)

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x6f61dc3f2c2e66a1e8...	Transfer	(pending)	5 hrs 4 mins ago	0x39Fd6e51aad88F6F4ce...	OUT	Fake_Phishing7050	0.000003343780745782 Ether (Pending)
0x00Bb7785c16807245A...	Transfer	16029813	1 day 21 hrs ago	0x39Fd6e51aad88F6F4ce...	OUT	Terra: LUNC Token	0 Ether
0xade19b02210216673F...	Transfer	16029813	1 day 21 hrs ago	0x545b2bd4d895f2a3a9...	IN	0x39Fd6e51aad88F6F4ce...	0.00144535 Ether
0x481a121dfbc765463...	Cookie Notice	This website uses cookies to improve your experience. By continuing to use this website, you agree to its Terms and Privacy Policy.	Got It	er			0.0050028

A Figura 20 apresenta a tela de Criação do NFT. O usuário logado, após entrar no Menu e clicar na opção CREATE, poderá criar um NFT. Para a criação no NFT são necessárias as informações do nome, descrição, o preço que será vendido e a imagem no NFT.

Figura 20

The screenshot shows the 'CREATE NFT' page of the NFT STATION application. At the top left is the 'NFT STATION' logo. On the right is a circular menu icon with three horizontal lines. Below the logo, the title 'CREATE NFT' is centered. The form consists of four input fields: 'NFT Name' (placeholder 'Name'), 'NFT Description' (placeholder 'Description'), 'Price' (placeholder '20'), and a file upload field ('Escolher arquivo art6.jpg'). At the bottom is a blue button labeled 'CREATE & LIST NFT!'. The entire form is set against a white background with thin gray borders around each input field.

Na criação dos NFTs, para armazenar seus metadados, é necessário utilizar um protocolo descentralizado denominado IPFS. O InterPlanetary File System (IPFS) é um protocolo hipermídia e rede peer-to-peer de compartilhamento de arquivos para armazenar e compartilhar dados em um sistema de arquivos distribuído. O IPFS usa o endereçamento de conteúdo para identificar exclusivamente cada arquivo em um namespace global que conecta hosts IPFS.(IPFS, 2022)

Para auxiliar nas configurações e armazenamento, utilizou-se o conjunto de APIs de *blockchain* de alta disponibilidade e ferramentas de desenvolvedor denominado Infura. Ela fornece ferramentas e infraestrutura que possibilitam os desenvolvedores portarem facilmente seus aplicativos *blockchain* do ambiente de teste para a implantação em escala, com acesso simples e confiável ao Ethereum e IPFS. A Figura 21 apresenta a importação da biblioteca e a chamada da função `create`

Figura 21

```

import { create } from 'ipfs-http-client'

const auth = `Basic ${token}`
const client = create({
  host: 'ipfs.infura.io',
  port: 5001,
  protocol: 'https',
  headers: {
    authorization: auth,
  },
})

```

Na Figura 22 é onde está o código das chamadas utilizando o http-ipfs-client e fazendo uma requisição para os dados serem alocados no Infura

Figura 22

```

const uploadToIPFS = async (event) => {
  event.preventDefault()
  const file = event.target.files[0]
  if (typeof file !== 'undefined') {
    try {
      const result = await client.add(file)

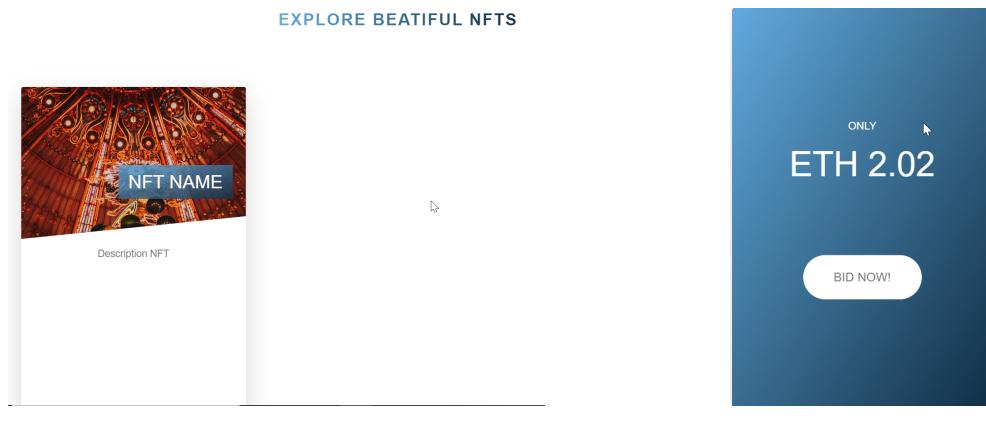
      setImage(`https://gcamon.infura-ipfs.io/ipfs/${result.path}`)
    } catch (error) {
      console.log('ipfs image upload error: ', error)
    }
  }
}

const mintThenList = async (result) => {
  const uri = `https://gcamon.infura-ipfs.io/ipfs/${result.path}`
  // mint nft
  await (await nft.mint(uri)).wait()
  //get tokenId of new nft
  const id = await nft.tokenCount()
  // approve marketplace to spend nft
  await (await nft.setApprovalForAll(marketAddress, true)).wait()
  // add nft to marketplace
  const listingPrice = ethers.utils.parseEther(price.toString())
  await (await marketplace.makeItem(nft.address, id, listingPrice)).wait()
}

```

A Figura 23 é o fluxo de Explorar. Onde exibe os NFTs que podem ser comprados no marketplace, o usuário terá a permissão de ver suas NFTs e vendê-las como desejar.

Figura 23

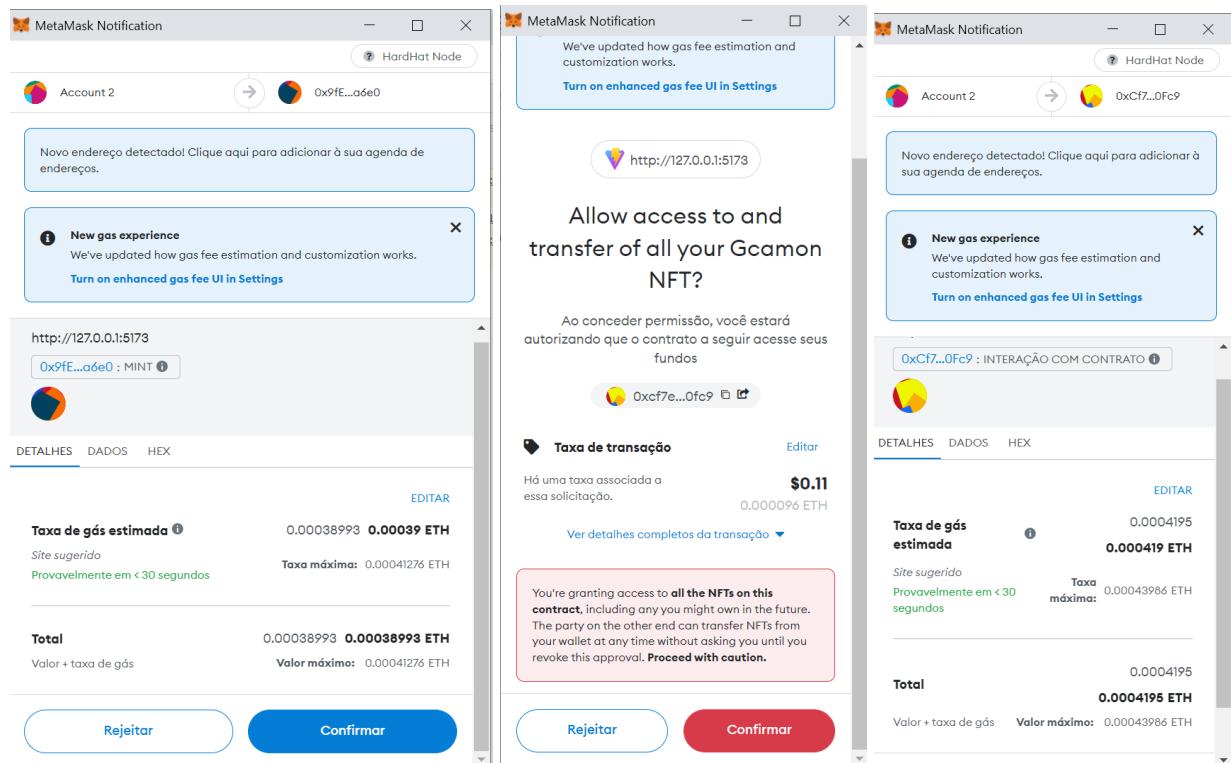


Card NFT Criado (Frente)

Card NFT Criado (Costas)

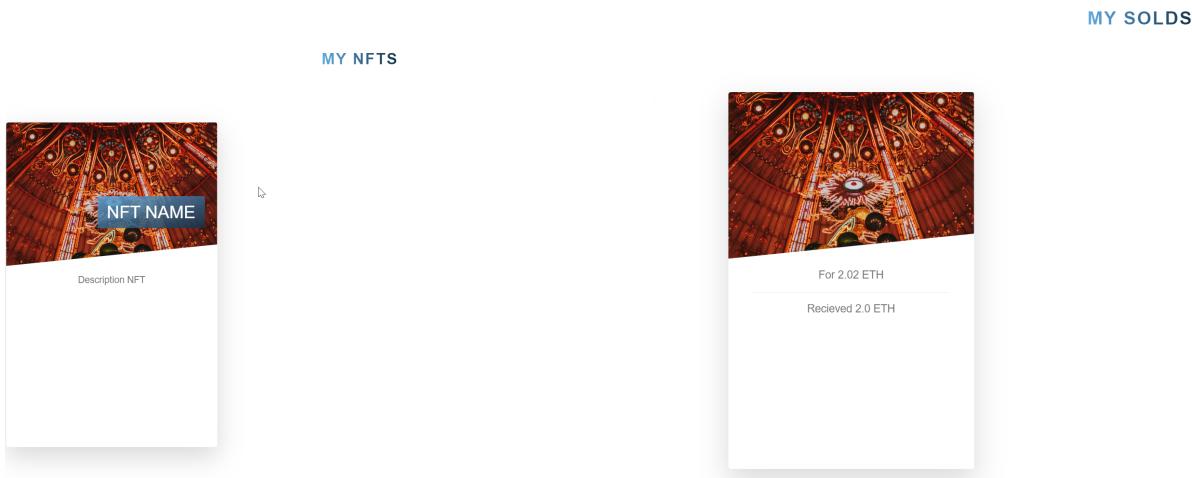
Ao vender, uma janela no MetaMask aparece para o usuário confirmar a transação como é representado na figura 24:

Figura 24



Por fim na Figura 25, o usuário poderá ver seus NFTs e os seus vendidos na Tela de Purchase

Figura 25



4. AVALIAÇÃO E IMPACTOS NA SOCIEDADE

Após as novas tendências de uso das criptomoedas em transações financeiras, os governos passaram a se interessar pela tecnologia. Alguns criaram *blockchains* privadas, que têm uma entidade central controlando quem pode participar, bem como as informações e as regras. Esse modelo, portanto, faz mais sentido para aqueles que querem utilizar parte dos benefícios da tecnologia, mas precisam de privacidade. Empresas, como IBM e JBS, e entidades públicas, como a Receita Federal do Brasil e os governos do Ceará e do Paraná, utilizam o sistema em seus projetos.

Alguns usos da *blockchain* podem ser citados:

- Na finanças e dinheiro, o tempo de transação entre o envio e o recebimento de dinheiro dentro do *blockchain* é muito mais rápido do que os métodos convencionais. Como não precisa de intermediários, a transferência pode ser realizada com pessoas de qualquer parte do mundo sem passar por bancos (ECONOMY BLOCKCHAIN, 2022);
- Na Alimentação, cita-se, como exemplo, o uso da tecnologia na rede *Carrefour* para rastrear a origem de frangos vendidos sobre a marca *Filière Qualité*. Os clientes terão, por meio de um *QR code*, acesso a data de nascimento do frango, de abate e tratamentos veterinários (CARREFOUR, 2022);

- Em se tratando de governo, alguns países já estão usando a tecnologia durante eleições. Neste caso, é possível garantir que um voto foi para determinado candidato/partido com o devido anonimato e que ele não foi duplicado. O que torna em tese a eleição mais segura contra fraude (UOL BLOCKCHAIN, 2022) ;

Vantagens que a blockchain está trazendo para empresas:

- Cadeia de suprimentos. A maioria dos empreendedores sabe de onde está comprando, mas não sabe quem são os fornecedores de seus fornecedores por ser rastreável e acessível. O blockchain dará a oportunidade para as empresas conhecerem todas as movimentações do produto, desde a origem. Por exemplo, se um dispositivo não está funcionando, usando o blockchain, será possível identificar quem foi o fornecedor original do componente defeituoso. A tecnologia ajudará as empresas a verificar se os fornecedores atendem aos seus requisitos;
- Acordos jurídicos totalmente automatizados. Quando se trata de contratos, o blockchain pode automatizar o processo de acordos legais e documentações, sem o envolvimento de seres humanos, diminuindo significativamente a burocracia que atrasa o processo. Também ajudará a eliminar o atraso em transações que exigem assinatura e aprovação;
- Proteção contra fraude. Com a ajuda do blockchain, os clientes poderão ter controle de suas informações com uma chave criptografada que será decifrada apenas pelas empresas que tenham acesso ao blockchain;
- Mais transparência e responsabilidade. Todas as transações ficam registradas em um livro-razão público, criptografadas e protegidas, sem a possibilidade de alterações fraudulentas. Isso significa que os controles financeiro, de logística, de produção e qualidade ficarão muito mais seguros e com margem de erro reduzida (EMPRESA BLOCK, 2022).

Por fim, sendo blockchain uma camada de software que possui vários recursos e bibliotecas para fazer transações com atomicidade, consistência, isolamento e durabilidade, além de utilizar o princípio da computação distribuída, avalia-se como inviável desenvolver solução automatizada com características similares às que podem ser implementadas usando

tal tecnologia, pois o esforço para desenvolver a estrutura base e arquitetura da solução tornariam tal empreitada inviável em termos de tempo e custo.

5. CONCLUSÃO

A tecnologia *blockchain* é uma realidade dos dias atuais e tem a tendência de crescer mundialmente em vários setores sociais. O trabalho esclareceu o uso dos *smart contracts*, onde é executado em uma tecnologia blockchain e no caso foi utilizado a *Etherium*. E mostrou a criação de NFTs, estas que são muito utilizadas por artistas digitais.

Espera-se que neste artigo seja esclarecido o uso de NFTs e o caminho para ser executado por programadores. Mostrando assim, a forma que é organizada a tecnologia *Etherium* e sugestionando tecnologias que auxilie programadores no desenvolvimento desse ecossistema.

Como trabalhos futuros pretendemos investigar a evolução das legislações referentes ao tema, bem como fazer novos experimentos dessa arquitetura em um contexto similar ao que seria do mercado de artes com leilões, lances e de forma jurídica legal.

6. REFERÊNCIAS BIBLIOGRÁFICAS

BENS, Bens. 2018 Por Edilson Araujo Costa. Disponível em :

<https://edilsonaraujocosta.jusbrasil.com.br/artigos/642034292/bens> Acesso em 09 dez. 2022.

BLOCKCHAIN, Blockchain – Wikipédia, a enciclopédia livre. Julho de 2018 Por Wikipedia.

Disponível em: <https://pt.wikipedia.org/wiki/Blockchain> Acesso em 15 nov. 2022.

CARREFOUR, Carrefour Lança Blockchain na Cadeia de Cítricos. Entenda! - Março 12, 2021 Por Maria Alice Medeiros Disponível em :

<https://ecommerceedesucceso.com.br/blockchain-carrefour-brasil/> Acesso em 11 dez. 2022.

ECONOMY BLOCKCHAIN, O que é blockchain? Conheça a tecnologia que torna as transações com criptos possíveis Disponível em:

<https://www.infomoney.com.br/guias/blockchain/> Acesso em 09 dez. 2022.

EIP-721, EIP-721: Non-Fungible Token Standard - 2018-01-24 Por William Entriken, Dieter Shirley, Jacob Evans, Nastassia Sachs. Disponível em:

<https://eips.ethereum.org/EIPS/eip-721#simple-summary> Acesso em 10 dez. 2022.

EMPRESA BLOCK, Entenda 5 vantagens do Blockchain para as empresas. 11 janeiro, 2019 Por Felix Schultz Disponível em :

<https://blog.milvus.com.br/entenda-5-vantagens-do-blockchain-para-as-empresas/> Acesso em 12 dez. 2022.

ETHERIUM, Learn Hub | ethereum.org. Disponível em:

<https://ethereum.org/en/learn/#what-is-crypto-ethereum> Acesso em 18 nov. 2022.

ETHERS.JS,Ethers.js Disponível em: <https://docs.ethers.io/v5/> Acesso em 11 dez. 2022.

ETHERSCAN, Etherscan. Disponível em: <https://etherscan.io/> Acesso em 15 nov. 2022.

HARDHAT, Documentation | Ethereum development environment for professionals by Nomic Foundation. Disponível em: <https://hardhat.org/docs> Acesso em 22 nov. 2022.

IBM CONTRACTS, What are *smart contracts* on *blockchain*? Disponível em :

<https://www.ibm.com/topics/smart-contracts> Acesso em 10 dez. 2022.

IBMBLOCKCHAIN, What is block chain technology ? Disponível em :
https://www.ibm.com/topics/what-is-blockchain?adoper=190102--1&adobe_mc_sdid=SDID%3D0EF627341E3A643A-1DFAFBF5898ADC2F%7CMCORGID%3DD10F27705ED7F5130A495C99%40AdobeOrg%7CTS%3D1670897267&adobe_mc_ref=https%3A%2Fwww.google.com%2F Acesso em 10 nov. 2022.

IPFS, What is IPFS? The hard drive for Blockchain. - iCommunity Labs. Disponível em:

<https://icommunity.io/en/what-is-ipfs-the-hard-drive-for-blockchain/> Acesso em 20 nov. 2022.

METAMASK, Metamask. Disponível em: <https://metamask.io/> Acesso em 22 nov. 2022.

OPENZEPPIN, OpenZeppelin Docs. Disponível em: <https://docs.openzeppelin.com/> Acesso em 15 nov. 2022.

RIPIOSMARTCONTRACT, Smart Contracts: entenda o que são e como funcionam. Disponível em: <https://launchpad-br.ripioweb.com/blog/smart-contract> Acesso em 15 nov. 2022.

SOLIDITY, Solidity. Disponível em: <https://docs.soliditylang.org/en/v0.8.17/> Acesso em 20 nov. 2022.

UAIBLOCKCHAIN, O dinheiro na era digital: inovação, criptomoedas e blockchain. Disponível em:

<https://www.uai.com.br/app/noticia/negocios/2022/02/25/noticias-negocios.284023/o-dinheiro-na-era-digital-inovacao-criptomoedas-e-blockchain.shtml> Acesso em 27 jan. 2023.

UOL BLOCKCHAIN, Blockchain virou pop. Por Bruna Sousa Cruz. Disponível em :

<https://www.uol/noticias/especiais/blockchain.htm> Acesso em 08 dez. 2022.

