APLICAÇÃO DA TECNOLOGIA *BLOCKCHAIN* NO ARMAZENAMENTO DE DOCUMENTOS

APPLICATION OF BLOCKCHAIN TECHNOLOGY IN DOCUMENT STORAGE

Bruno Brigido Goulart[1]

Paulo João Martins[2]

RESUMO: A gestão e armazenamento de documentos físicos como um contrato, possui muitos desafios, pelo potencial de serem perdidos ou sabotados, os documentos digitais podem resolver alguns destes desafios. Este artigo propõe uma estrutura de armazenamento destes documentos utilizando a arquitetura *blockchain*, assim implementado um sistema para o armazenamento dos documentos digitais, utilizando a *Ethereum* e o Sistema de Arquivo Interplanetário (IPFS). O sistema proposto fornece registro imutável e irreversível do processo de armazenamento dos documentos. O contrato inteligente e um programa de computador imutável e determinístico, que é escrito na linguagem de programação *Solidity*, o front-end necessário para interagir com a *blockchain* foi utilizado o React e Web3.JS. A carteira *MetaMask* foi usada para criar as contas, que são necessárias para a conexão(?) e com o contrato inteligente.

Palavras-chave: *Blockchain*; *Ethereum*; contrato inteligente; *Truffle; Ganache*; IPFS; *MetaMask*.

ABSTRACT*: Physical documents have many challenges, due to the potential of being lost or sabotaged, digital documents can solve some of these challenges, this article proposes a storage structure for these documents using the blockchain architecture, thus implementing a system for the storage of digital documents, using the Ethereum blockchain and the Interplanetary File System (IPFS), the proposed system provides immutable and irreversible records of the document storage process. The smart contract used in the Ethereum blockchain was written in Solidity programming language, the front-end needed to interact with the blockchain was used React and Web3.JS. The Metamask wallet was used to create the Ethereum accounts, which are needed for connecting and with the smart contract.*

Keywords*: Blockchain; Ethereum; Smart Contract; Truffle; Ganache; IPFS; Metamask.*

[1] Acadêmico. Artigo de Conclusão de Curso apresentado para obtenção do grau de Bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC. E-mail: bruno-goulart1@live.com.

[2] Professor Orientador. E-mail: pjm@unesc.net.

1 INTRODUÇÃO

Com o avanço da tecnologia houve a possibilidade da criação de novos meios de realizar a criação de contratos. A validação tradicional de documentos utiliza autoridades centrais para a validação de existência de documentos, que apresenta alguns desafios em relação a segurança como adulteração ou extravio destes. Estes modelos tornam-se ainda mais difíceis quando os documentos se mais antigos.

Para garantir a segurança em contratos inteligentes é utilizar *blockchain*, que utiliza um sistema de protocolos de confiança onde se possui meios de garantir que ambas as partes cumpram os seus deveres. Segundo Szabo (1997), “um contrato inteligente e como um protocolo de transação informatizado que executa os termos de um contrato". Para Hou (2017), eles não foram aplicados por Nick Szabo nos anos 90 devido à falta de um ambiente de execução confiável. Hou (2017) descreve um contrato inteligente como um código de programa de compartilhamento de informações descentralizada, onde as partes do contrato concordam com o conteúdo. Então o contrato é traduzido para código do programa, e em seguida ele será executado automaticamente em nome dos signatários sem qualquer agência central.

Os contratos inteligentes utilizam *script* armazenados na *blockchain* e executam suas instruções de maneira distribuída para todos os participantes do contrato. O termo contrato inteligente possui familiaridade com o contrato legal, neste sentido ele regula a interação entre diferentes entidades (THEODORO JÚNIOR, 2017).

Contratos inteligentes permitem a realização de transações confiáveis sem a necessidade de terceiros. Essas transações são rastreáveis e irreversíveis e também possui todas as informações sobre os termos dos contratos e executam todas as ações previstas automaticamente.

Os contratos inteligentes são executados por uma rede de computadores que usa protocolos de consenso para concordar com a sequência de ações resultantes do código do contrato. O resultado é um método pelo qual as partes podem concordar com os termos para a execução do contrato, possibilitando a confiança de que será executado de forma automática, com risco reduzido de erro ou manipulação. (REBOUÇAS, 2018, p. 129).

Conforme levantamento bibliográfico foram encontrados projetos que utilizam a estrutura para armazenar documentos na rede. Na pesquisa de Miranda

(2019) foi utilizado a rede *blockchain* para realizar a prova de existência de diplomas e certificados, tendo em vista que o atual cenário, envolve processos manuais para seu compartilhamento e verificação. Ainda, estabelece uma relação com intermediadores, responsáveis por tais procedimentos. Jain *et al.* (2021) desenvolveram uma aplicação que para registrar na *Ethereum* as questões de exames onde todas as perguntas e respostas dos alunos são salvas em um contrato inteligente que é enviado por meio da rede. Alslman e Taleb (2021) desenvolveu um aplicativo para a troca de documentos digitais, contrato inteligente e IPFS. O modelo proposto fornece uma solução para a troca de documentos digitais sem a necessidade de terceiros confiáveis centralizados e um contrato inteligente que é implementado usando a linguagem de programação *solidity*.

Sendo assim, a proposta aqui apresentada tem como objetivo criar um protótipo de sistema que irá adicionar um documento no sistema de arquivos planetários IPFS, onde é gerado um *hash* e dado uma impressão digital única chamada CID (Identificador de Conteúdo). Este CID age como um registro permanente do seu arquivo como ele existe naquele momento, dividido em pedaços menores e distribuído na rede, este *hash* será armazenado na rede *Ethereum* de forma que possa garantir a imutabilidade do documento adicionado.

Os objetivos específicos deste trabalho consistem em: descrever a tecnologia *Blockchain* em contratos inteligentes; entender o funcionamento da rede IPFS; descrever o funcionamento do framework *Truffle*; entender o funcionamento da carteira digital *MetaMask* desenvolver um software para adicionar os documentos no IPFS e *Ethereum*.

2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento do protótipo que irá armazenar os documentos em uma rede IFPS e *Blockchain* foi necessário realizar um levantamento bibliográfico para a pesquisa e o embasamento teórico sobre alguns conceitos, como, por exemplo, a rede *Blockchain Ethereum*.

2.1 PLATAFORMA *ETHEREUM*

A plataforma *Ethereum* é um sistema de contratos inteligentes baseado em *blockchain*, que é composta por máquinas virtuais descentralizadas chamada de *Ethereum Virtual Machines* (EVM) que executa os contratos inteligentes, esta plataforma descentralizada executa aplicações exatamente como foram programadas (PEREIRA, 2018).

Para Tigre e Pinheiro (2019, p. 174), esta plataforma lançou sua própria moeda o *Ether* (ETH), e é mais uma que permite realizar outras aplicações além da criptomoeda, inclusive para construir contratos inteligentes. Com base na *Ethereum* foram lançadas várias moedas possibilitando a combinação de diferentes características entre elas. Os autores afirmam que possui mais de 500 criptomoedas e tokens catalogados.

A plataforma funciona em um *ledger* global, apesar de ser semelhante à bitcoin em vários aspectos ela possui algumas diferenças, a principal seria que os blocos contêm uma cópia da lista de transações e do estado mais recente. Além disso ele possibilita a criação de tokens digitais que podem ser utilizados para apresentar partes virtuais, ativos, comprovante de associação entre outros (FERREIRA, F., 2017).

Os dados armazenados na *Ethereum* são considerados em geral como um bloco de informações não corrompível e imutáveis, eles consistem em dois tipos de registros, que são as transações e blocos. Ele codifica as transações individuais e as agrupa em blocos, que são persistidos consecutivamente. Cada bloco, exceto o primeiro denominado de gênese, contém um ponteiro para o bloco previamente validado criando um encadeamento entre eles e garantido uma integridade da informação (FERREIRA, J., 2017).

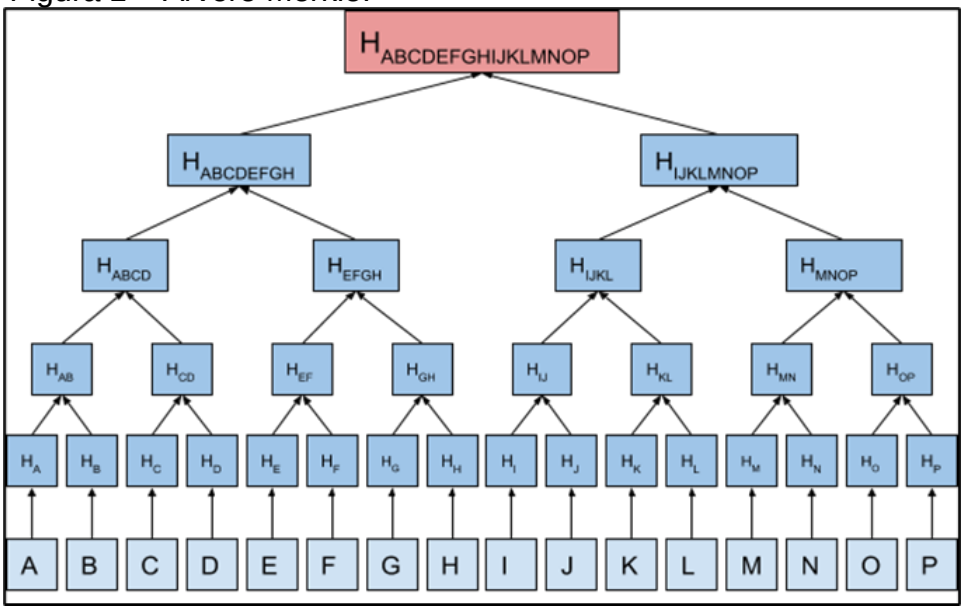
Para adicionar um novo bloco na rede, ou seja, um conjunto de transações, cada minerador da rede irá participar de uma corrida em que uma tarefa necessita ser realizada, esta execução e denominada de mineração. O primeiro minerador com uma solução correta para a tarefa pode adicionar um novo bloco a rede. A dificuldade do problema depende do poder de mineração que está disponível na rede para compensar as melhorias do desempenho da computação. O minerador que resolveu a tarefa primeiro é geralmente recompensado por exemplo por uma moeda de

mineração para garantir a atratividade da mineração, que é necessária para garantir a integridade das transações persistentes (SILLABER; WALTL, 2017).

Os blocos são formados com a estrutura e árvore de *Merkle*, que são estruturas de dados utilizadas para verificação da integridade das informações em ambientes distribuídos, ou seja, onde não há centralidade no processamento dos dados. Elas otimizam o uso das funções *hash* para esta tarefa.

A árvore e uma estrutura de dados voltada a permitir uma fácil verificação da presença de uma certa informação em um determinado local, elas adotam a estrutura de uma árvore binária onde cada folha contém o *hash* de uma informação. Os nodos superiores, ou pais, armazenam a combinação dos *hashes* dos filhos, esta estrutura de árvore possui um procedimento chamado Prova de *Merkle* que permite um cliente verificar a informação na árvore, e para isso precisa de um *hash* da raiz fornecido por alguma fonte confiável (FERREIRA, J., 2017).

Figura 1 – Árvore Merkle.



Fonte: Juliandson Ferreira (2017).

Os nodos superiores, ou pais, armazenam o *hash* da combinação dos *hashes* dos filhos. Se H(A) e H(B) são das informações A e B, o pai destes será H(H(A)+H(B)). O processo se repete até a raiz ser obtida.

Para a criação do contrato inteligente na rede *blockchain* será cobrado um valor de transação, este valor e denominado de GAS, que é a taxa de execução que os remetentes das transações precisam pagar por cada operação feita em uma rede *blockchain* da *Ethereum* (FERREIRA, F., 2017). Após um contrato ser executado por

uma transação ou mensagem, todas as instruções são executadas em todos os nós dá rede *blockchain*, esta execução exige um custo computacional dos nós (RIBEIRO; MENDIZABAL, 2019).

Sempre que algum pedaço de código vai ser executado, a parte que está solicitando tal execução deve estabelecer a quantidade máxima de unidades de GAS que está disposta a utilizar e qual o valor em *Ether* que irá pagar por cada unidade gasta. Parte básica da verificação inicial consiste em checar se o solicitante de fato possui a quantidade necessária de moeda e subtraí-la de sua conta para pagar as taxas de transação (FERREIRA, F., 2017).

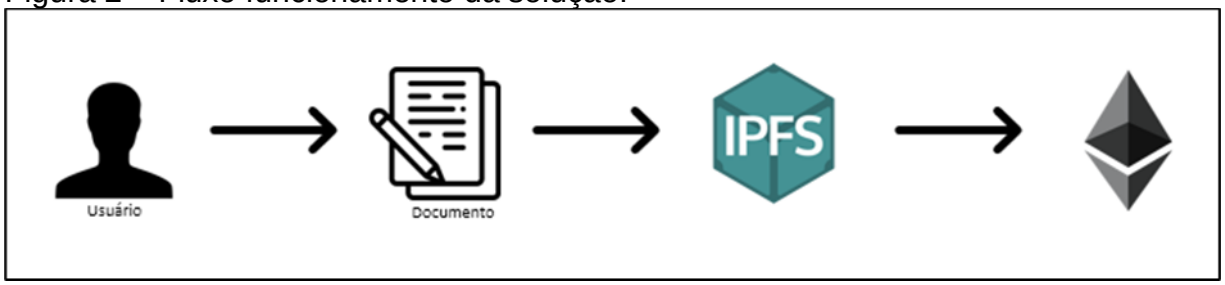
Se for necessário mais GAS que o enviado para executar o código, os efeitos deste serão revertidos, mas o solicitante perde o valor das taxas de transação. Por outro lado, se sobrar GAS, o valor em excesso será devolvido ao solicitante no final da transação (FERREIRA, F., 2017).

3 MATERIAIS E MÉTODOS

Esta pesquisa é de forma aplicada, com base explicativa e tecnológica. Desenvolveu-se um protótipo de aplicação que irá armazenar documentos utilizando a tecnologia *smart control*. Na tentativa de diminuir os gastos sem perder a qualidade do software será utilizado uma rede *blockchain* de teste.

Utilizando o banco de dados IFPS para armazenar o documento para gerar o *hash* e estrutura *blockchain* será armazenado o *hash* do documento retornado pelo banco de dados IPFS, na figura 2 apresenta o fluxo da aplicação.

Figura 2 – Fluxo funcionamento da solução.



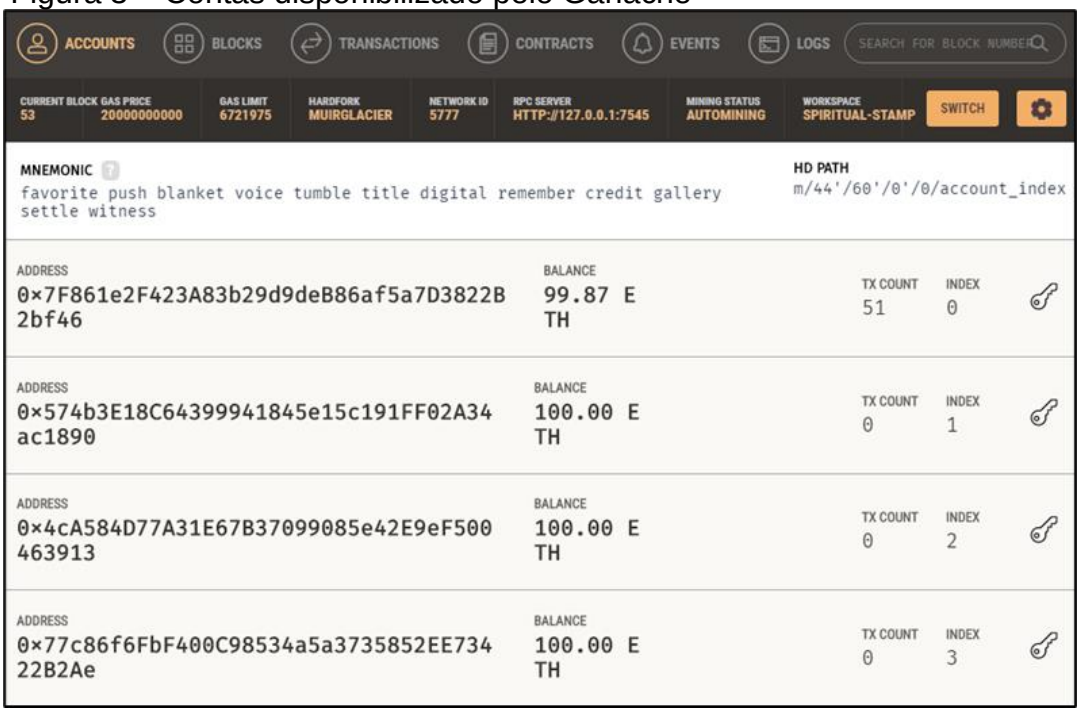
Fonte: elaborado pelo autor (2021).

3.1 TRUFFLE

É um ambiente de desenvolvimento, estrutura de testes e *pipeline* de ativos para o *Ethereum*, com o objetivo de facilitar a vida como desenvolvedor do *Ethereum*. *Truffle* oferece instrumentos que auxiliam na programação de *smart contrat* que utiliza a linguagem de programação *solidity*, além de ferramentas para executar testes e um ambiente para este utilizando o ganache (AL-MADANI *et al.*, 2020).

Ganache é usado para criar uma rede *blockchain Ethereum* na máquina local e o contrato inteligente é implantado na rede usando a estrutura *Truffle*. *Ganache* fornece 10 conta com 100 éteres falsos que podem ser usados para testar o funcionamento do aplicativo desenvolvido (VAIRAM; SARATHAMBEKAI; BALAJI, 2021).

Figura 3 – Contas disponibilizado pelo Ganache



Fonte: elaborado pelo autor (2021).

3.2 METAMASK

É uma carteira de criptomoeda utilizada para interagir com a rede *blockchain Ethereum* no seu navegador (METAMASK, 2021). É uma extensão para navegador descentralizado para *Ethereum* que também permite que o usuário crie e gerencie suas próprias carteiras. O usuário recebe uma interface segura para revisar

a transação, antes de aprová-la ou rejeitá-la. Como adiciona funcionalidade ao contexto normal do navegador, o *MetaMask* requer a permissão para ler e escrever em qualquer página da Web (JAIN *et al.*, 2021).

3.3 IPFS

O *interplanetary file system* IFPS e um protocolo de armazenador de dados distribuído de ponto a ponto capaz de armazenar grandes volumes de registros, o IPFS armazena seus arquivos e gera um *hash* do arquivo adicionando, utilizando este para evitar arquivos duplicados na rede, com seu conteúdo endereçado por uma tabela distribuída.

Os nós IPFS tem a capacidade de extrair e armazenar informações de seus pares e também servir como um provedor de conteúdo armazenado dentro deles. Os arquivos no IPFS são identificados pelo conteúdo que consistem no local em que são armazenados. Para isso, o IPFS utiliza um identificador de Conteúdo (CID). Cada arquivo após ser carregado por seu nó de origem gera um *hash* único e se uma pasta for carregada, cada arquivo dentro da pasta terá um *hash* único e a pasta terá um *hash* exclusivo também (SINGH *et al.*, 2020).

3.4 IMPLEMENTAÇÃO PARA CONTRATOS INTELIGENTES

O software foi desenvolvido utilizado o *React* para o desenvolvimento do *front-end*, para a comunicação com rede *blockchain* foi utilizado o *framework Truffle* que fornece uma suíte de desenvolvimento voltado a contrato inteligentes.

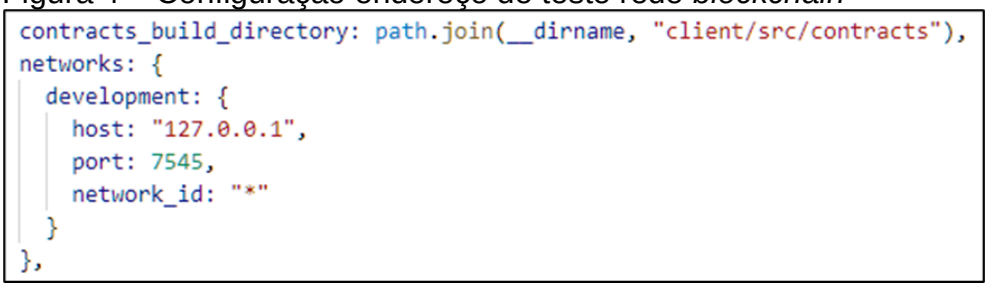
3.4.1 Configuração da rede *blockchain*

No início da implementação foi utilizado a *Truffle framework* por meio do comando *truffle unbox react* para iniciar o projeto, e instalar as dependências necessárias foi necessário a configuração do aplicativo *front-end* para a comunicação entre a rede *blockchain* de teste Ganache.

Com o objetivo de redução de custo foi utilizado uma rede *Ganache* que é uma rede pessoal para o rápido desenvolvimento de aplicativos distribuídos *Ethereum*. Ele foi utilizado durante todo o ciclo de desenvolvimento, permitindo que

fosse desenvolvido, implantado e testado em um ambiente seguro e determinístico, para a sua configuração deve informar o endereço da rede conforme figura 4.

Figura 4 – Configuração endereço de teste rede *blockchain*



Fonte: elaborado pelo autor (2021).

3.4.2 Configuração do MetaMask

O *MetaMask* é uma das carteiras de criptomoedas de software usado para interagir com a rede *Blockchain Ethereum*, utiliza uma simples extensão de navegador que funciona como uma ponte que permite a conexão com a página distribuída. Para realizar a interação com um contrato inteligente, o *MetaMask* se conecta com a aplicação web e a rede *Ethereum* pela biblioteca web3.js.

Web3.js é uma coleção de bibliotecas que permite interagir com o nó *Ethereum* local ou remoto, realizando a conexão por meio de HTTP, e permite executar ações como enviar *Ether* de uma conta para outra, ler e escrever dados usando contratos inteligentes. Ele se comunica com o *Ethereum Blockchain* por meio do JSON RPC. O Web3.js nos permite fazer solicitações para cada nó *Ethereum* por meio do JSON RPC para ler e gravar dados.

A carteira se conectará com uma instância de *nó backend* e enviará uma solicitação para o *Blockchain*. O Ganache pode ser usado para configurar um *Ethereum Blockchain* pessoal para testar os contratos inteligentes de *Solidity*. Ganache é uma ferramenta para configurar um nó *Ethereum* e começar com o desenvolvimento *Blockchain*. Ele fornece uma lista de 10 endereços com um saldo padrão de 100.00 ETH.

3.4.3 Contrato inteligente

No *Ethereum*, os contratos inteligentes são acessíveis e transparentes para a implantação do contrato inteligente e escrito na linguagem *Solidity, que é* de alto

nível e são compilados para o *bytecode* EVM (WÖHRER; ZDUN, 2018). Os contratos inteligentes da *Blockchain Ethereum* são usados para gerenciar as transações. O código do contrato inteligente é considerado implantado na rede. No modelo proposto, o contrato inteligente será utilizado para controlar o acesso aos documentos adicionado no IPFS.

O contrato implementado irá salvar as informações do documento que foi adicionado no IPFS, no contrato irá conter as informações de nome do documento, tipo do documento, data da criação do contrato e o *hash* gerado pelo IFPS ao adicionar o documento na rede, conforme figura 5 mostra a estrutura implementada do contrato inteligente.

Figura 5 – Métodos Adicionar e consultar contratos *Ethereum*

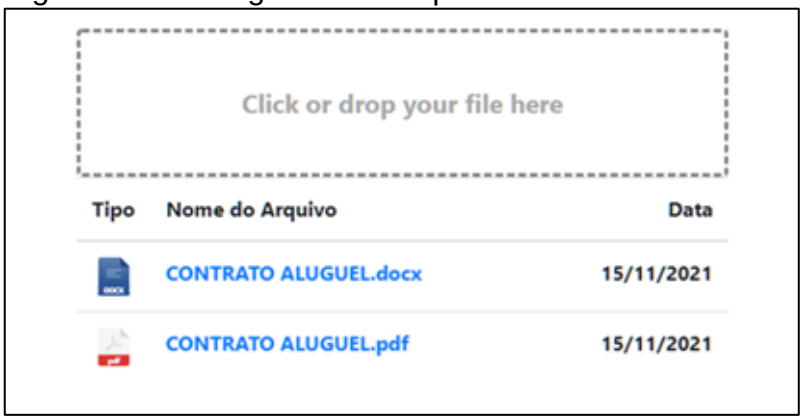
```
contract ArmazenadorContrato {
    struct Arquivo {
        string hash;
        string nomeArquivo;
        string tipoArquivo;
        uint data;
    }

    mapping(address => Arquivo[]) arquivos;

    function add(string memory _hash, string memory _nomeArquivo, string memory _tipoArquivo, uint _data) public
    {
        arquivos[msg.sender]
        .push(Arquivo({hash: _hash, nomeArquivo: _nomeArquivo, tipoArquivo: _tipoArquivo, data: _data}));
    }

    function getArquivo(uint _index) public view returns(string memory, string memory, string memory, uint)
    {
        Arquivo memory arquivo = arquivos[msg.sender][_index];
        return (arquivo.hash, arquivo.nomeArquivo, arquivo.tipoArquivo, arquivo.data);
    }

    function getLength() public view returns(uint)
    {
        return arquivos[msg.sender].length;
    }
}
```

Fonte: elaborado pelo autor (2021).

3.4.4 Funcionamento da aplicação

Para adicionar os documentos na rede o usuário poderá selecionar o documento desejado ou arrastar o documento para o campo adicionar o arquivo na rede conforma figura 6.

Figura 6 – Visão geral da tela para adicionar documento



Fonte: elaborado pelo autor (2021).

Após selecionar o documento a aplicação irá adicionar o documento a rede IPFS, após a rede retornar o *hash* do documento gerado na rede, será iniciado o processo de adicionar os dados do documento na rede *Ethereum*, o front-end irá se conectar ao Web3.js que por sua vez irá se conectar a carteira *Metamask*, solicitando a carteira a confirmação da transação, conforme figura 7.

Figura 7 – Confirmar transação carteira

Fonte: elaborado pelo autor (2021).

Ao confirmar a transação o Web3.js se conectará à rede *blockchain Ethereum* e irá implantar o contrato adicionando o *hash* do documento adicionado no

IPFS e as informações dos documentos ne *Ethereum* gerando o log da transação realizada conforme figura 8.

Figura 8 – Log de transação do *Ethereum*

Fonte: elaborado pelo autor (2021).

4 RESULTADOS E DISCUSSÃO

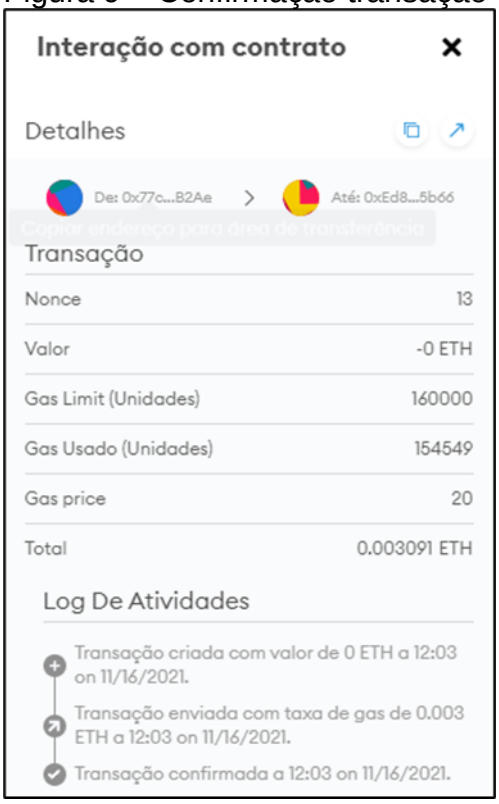
Para as etapas iniciais de pesquisa e desenvolvimento, o foco foi na utilização das ferramentas e *framework* para a implementação da proposta. Os trabalhos de Alslman e Taleb (2021), Miranda (2019) e Jain *et al*. (2021) contribuíram para o entendimento do funcionamento das ferramentas utilizadas no projeto.

O trabalho de Miranda (2019) apresenta propósitos semelhantes ao da presente pesquisa, de armazenar na rede *Blockchain* os certificados digitais emitidos, a fim de minimizar as fraudes destes documentos, criando uma prova irrefutável de sua existência. O registro consiste na inserção do *hash* em uma rede *bitcoin.* Em Alslman e Taleb (2021) se propôs um aplicativo de troca de documentos digitais descentralizado, utilizando os recursos de *Blockchain Ethereum*, contratos inteligentes e IPFS, utilizando uma rede P2P para a troca de documentos e fornecendo logs para acompanhar as transações, onde o usuário poderá enviar e receber por meio da *Ethereum*. Jain *et al*. (2021) propuseram uma solução para votações que irá fornecer uma solução transparente e segura utilizando a tecnologia *blockchain*, com o voto baseado em *blockchain* o comparecimento do eleitor também pode aumentar,

pois é possível votar em qualquer localidade, embora forneça transparência, como as transações são visíveis para todos, ela conserva a confidencialidade e o sigilo do eleitor, além de ajudar a anunciar o resultado rapidamente.

A aplicação demonstrou-se eficaz no armazenamento dos documentos utilizando uma rede distribuída *Ethereum* e IPFS, após adicionar os documentos na *blockchain* o registro se torna imutável e a prova de alteração. Ao contrário do sistema de armazenamento centralizado atualmente disponível, o modelo proposto é completamente descentralizado. O modelo proposto não conta com terceiros, assim, garante que a segurança, a transparência e rastreamento sejam feitos de forma eficiente entre os pares da rede. Para a realização da transação registrada na rede *Ethereum* consumiu um total de 0,003091 *Ether* conforme figura 9. O valor de UM *Ether* está cotado em R$ 23.660,54. Deste modo o valor da transação em reais ficou em R$ 72,09 demostrando um custo elevado para realizar as transações. Para a diminuição dos custos recomenda-se a utilização de uma outra rede *blockchain* que implemente os contratos inteligentes. Com valores semelhantes ao apresentado no artigo de Jain *et al*. (2021) 0,000572 *Ether* e na pesquisa de Alslman e Taleb (2021) 0,003812 *Ether*.

Figura 9 – Confirmação transação



Fonte: elaborado pelo autor (2021).

Para Destefanis *et al*. (2018) o uso da tecnologia *blockchain* entrega um modelo de segurança previsto pelos contratos inteligentes, estes modelos descentralizados de contratos imutáveis implicam que a execução e saída de um contrato é validada por cada participante do sistema e, portanto, nenhuma parte está no controle. Deste modo, não será possível forçar uma execução do contrato, pois isso seria invalidado pelo outros participantes do sistema tornando a adulteração de contrato inteligentes quase impossíveis.

5 CONCLUSÃO

Este trabalho teve como objetivo desenvolver um protótipo de sistema que registra em uma *blockchain* os documentos adicionados em uma rede IPFS. As principais vantagens desse serviço são a segurança e a privacidade, que permitem ao usuário fornecer provas descentralizadas do documento que não podem ser modificadas por terceiros. A existência do documento é validada usando *blockchain* que não depende de uma única entidade centralizada.

Este artigo demonstra a utilização da tecnologia de *blockchain* e contrato inteligentes, Mendonça *et al.* (2020) utilizou de contratos eletrônicos no rastreio de cadeia produtivas de mercadorias. Outras áreas que se aplica contratos inteligentes podem ser em direitos autorais de músicas, livros e criadores de conteúdo em geral. A *blockchain* pode auxiliar neste mercado de proteção das obras. Outra área de aplicabilidade é a de telecomunicação, podendo facilitar a autenticação em redes Wireless públicas ou privadas. Possibilita assim, gerenciar uma base de dados facilitando na autenticação de usuários, além do setor financeiro pois esta tecnologia não a necessidade de utilizar intermediários nas transações (MARTINELLI; PINTO, 2019).

As dificuldades encontradas foram durante o processo de estudo da ferramenta escolhida, tendo em vista que apesar de sua ascensão, é uma tecnologia nova que requer aprendizados para sua devida utilização, porém, o objetivo principal de aplicar a tecnologia *blockchain* em contratos inteligentes de forma a garantir a segurança e integridade foi alcançado, após a realização dos testes e a obtenção dos resultados.

Para dar continuidade dos estudos sugere-se como realização de trabalhos futuros tais como: criptografia dos arquivos para garantir maior segurança nas

informações contida nos documentos, além da possibilidade de compartilhar o documento com outras carteiras digitais *Ethereum*.

REFERÊNCIAS

AL-MADANI, Ali Mansour et al. Decentralized E-voting system based on Smart Contract by using Blockchain Technology. **Anais [...].** 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), IEEE, *[S.l.]*, v. 5, n. 8, p. 176-180, 30 out. 2020. Disponível em: http://dx.doi.org/10.1109/icsidempc49020.2020.9299581. Acesso em: 08 dez. 2021.

ALSLMAN, Yasmeen Shaher; TALEB, Anas Abu. Exchanging Digital Documents Using Blockchain Technology. **Anais** […]. III International Conference on Electrical, Communication and Computer Engineering (ICECCE), Kuala Lumpur, Malaysia, jun. 2021. Disponível em: https://doi.org/10.1109/ICECCE52056.2021.9514253. Acesso em: 08 dez. 2021.

DESTEFANIS, Giuseppe *et al*. Smart contracts vulnerabilities: a call for blockchain software engineering? **Anais [...].** 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), IEEE, Campobasso, p.19-25, 20 mar. 2018. Disponível em: http://dx.doi.org/10.1109/iwbose.2018.8327567. Acesso em: 08 dez. 2021.

FERREIRA, Frederico Lage. **Blockchain e Ethereum aplicações e vulnerabilidades**. 2017. 37 f. TCC (Graduação em Ciência da Computação) - Universidade de São Paulo, São Paulo, 2017.

FERREIRA, Juliandson Estanislau. **Blockchain para criação de novos modelos de negócio e seus impactos na indústria de serviços financeiros**. 2017. 51 f. TCC (Graduação em Sistemas de Informação) - Universidade Federal de Pernambuco, Recife, 2017.

HOU, Yunfei et al. A resolution of sharing private charging piles based on smart contract. **Anais [...].** 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), IEEE, *[S.l.]*, p. 3004-3008, jul. 2017. Disponível em: http://dx.doi.org/10.1109/fskd.2017.8393262. Acesso em: 08 dez. 2021.

JAIN, Apoorv et al. Smart Contract enabled Online Examination System Based in Blockchain Network. **2021 International Conference on Computer Communication and Informatics (ICCCI)**, *[S.L.]*, v. 1, n. 1, p. 1-7, 27 jan. 2021. IEEE. http://dx.doi.org/10.1109/iccci50826.2021.9402420.

MARTINELLI, Tháiro; PINTO, Giuliano Scombatti. Blockchain: comparação evolutiva utilizando bitcoin e ethereum. **Revista Interface Tecnológica**, Taquaritinga, v. 16, n. 1, p. 146-157, 30 jun. 2019.

MENDONÇA, Ronan Dutra et al. Utilização de Blockchain na Rastreabilidade da Cadeia Produtiva do Leite. **Anais [...].** III Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain 2020), Sociedade Brasileira de Computação - SBC, *[S.l.]*, v. 3, n. 1, p. 50-60, 7 dez. 2020. Disponível em: http://dx.doi.org/10.5753/wblockchain.2020.12433. Acesso em: 08 dez. 2021.

METAMASK. Guide. **Portal MetaMask Docs**, 24 set. 2021. Disponível em: https://docs.metamask.io/. Acesso em: 08 dez. 2021.

MIRANDA, Dérick Souza. **Blockchain na educação**: uso da tecnologia como prova de existência de diplomas e certificados. 2019. 149 f. TCC (Graduação em Ciência da Computação) - Universidade do Extremo Sul Catarinense, Criciúma, 2019.

PEREIRA, Renan Ramos. **Estudo de caso sobre a tecnologia blockchain, Projeto Ethereum e viabilidade de métodos de mineração**. 2018. 27 f. TCC (Graduação em Ciência da Computação) - Universidade do Sul de Santa Catarina, Tubarão, 2018.

REBOUÇAS, Rodrigo Fernandes. **Contratos Eletrônicos:** formação e validade aplicações práticas. 2ª ed., rev. e ampl. Lisboa: Grupo Almedina, 2018.

RIBEIRO, Lucas; MENDIZABAL, Odorico. **Introdução à Blockchain e Contratos Inteligentes:** apostila para iniciante - Relatório Técnico. Florianópolis: UFSC, 2019.

SILLABER, Christian; WALTL, Bernhard. Life Cycle of Smart Contracts in Blockchain Ecosystems. **Datenschutz und Datensicherheit - DUD**, *[S.l.]*, v. 41, n. 8, p. 497-500, ago. 2017. Disponível em: http://dx.doi.org/10.1007/s11623-017-0819-7.s. Acesso em: 08 dez. 2021.

SINGH, Harshit Sunilkumar et al. Health Monitoring and Analysis using IPFS and Blockchain. **Anais […].** 2020 International Conference on Convergence to Digital World - Quo Vadis (ICCDW), IEEE, *[S.l.]*, v. 5, n. 8, p. 176-180, 18 fev. 2020. Disponível em: http://dx.doi.org/10.1109/iccdw45521.2020.9318651. Acesso em: 08 dez. 2021.

SZABO, Nick. Formalizing and Securing Relationships on Public Networks. **First Monday**, Chicago, v. 2, n. 9, set. 1997.

THEODORO JÚNIOR, Humberto. **O contrato e sua função social**. 4. ed. Rio de Janeiro: Forense, 2017.

TIGRE, Paulo Bastos; PINHEIRO, Alessandro Maia. **Inovação em serviços e a economia do compartilhamento**. São Paulo: Saraiva, 2019.

VAIRAM, T.; SARATHAMBEKAI, S.; BALAJI, R. Blockchain based voting system in local network. **Anais [...].** 2021 7Th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, *[S.l.]*, v. 1, n. 1, p. 363-366, 19 mar. 2021. Disponível em: http://dx.doi.org/10.1109/icaccs51430.2021.9441912. Acesso em: 08 dez. 2021.

WÖHRER, Maximilian; ZDUN, Uwe. Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity. **Anais [...].** 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), IEEE, Campobasso, p. 2-8, mar. 2018. Disponível em: https://doi.org/10.1109/IWBOSE.2018.8327565. Acesso em: 08 dez. 2021.