REVISTA UNIFESO – CADERNO UNIFESOTEC
v. 1, n. 1, 2023, Teresópolis

Artigo científico

UTILIZAÇÃO DE BLOCKCHAIN EM VOTAÇÕES ELETRÔNICAS PARA REUNIÕES DE CONDOMÍNIOS

USE OF BLOCKCHAIN IN ELECTRONIC VOTING FOR CONDOMINIUM MEETINGS

Romulo Lins Pereira [1]; Alberto Angonese [2]

[1] Centro Universitário Serra dos Órgãos, romulotere@gmail.com
[2] Doutor - Centro Universitário Serra dos Órgãos, albertoangonese@unifeso.edu.br

RESUMO

Aplicações envolvendo a utilização do blockchain vem crescendo a cada dia, apesar de na maioria, ainda serem voltadas para validação de transações financeiras com criptomoedas. Entretanto, devido a sua segurança, essa tecnologia apresenta potencial de utilização também em outros métodos, como, sociais, políticos, econômicos e industriais. O presente artigo propõe a implementação de um sistema de votações para condomínios baseadas em blockchain, demonstrando que o blockchain pode ser utilizado em pequenas redes não sendo necessário um grande computador para tal feito. Utilizado a linguagem de programação Solidity e o ambiente de desenvolvimento o Remix IDE, a implementação utiliza contratos inteligentes para fazer a validação dos endereços da rede. Como resultado é apresentado um contrato inteligente responsável pela a criação de pautas de uma reunião de condomínios. Pelo sistema, os condôminos podem realizar a votação sobre a pauta criada através da blockchain, com segurança, mostrando os resultados de forma transparente, descentralizada e auditável

Palavras-chave: Blockchain; votação eletrônica; contratos inteligentes

ABSTRACT

Blockchains applications is growing every day, although it is still widely used to validate financial transactions with cryptocurrencies, due to its security, this technology has the potential to be used in other methods, like social, political, economic and industrial. This article proposes the implementation of a voting system for condominiums based on blockchain, demonstrating that blockchain can be used in small networks, not requiring a large computer to do so. Using the solidity programming language and the Remix IDE development environment, the implementation uses smart contracts to validate network addresses. As a result, a smart contract responsible for creating agendas for a meeting of condominiums is presented. Through the system, condominium members can vote on the agenda created through the blockchain, safely, showing the results in a transparent, decentralized and auditable way

Keywords: Blockchain; electronic voting; smart contracts

INTRODUÇÃO

Os conceitos de bitcoin e blockchain foram popularizados em 2008 por Satoshi Nakamoto, como a tecnologia de um livro razão distribuído e com banco de dados descentralizado. Utilizando métodos criptográficos, podem ser combinados em uma moeda digital afim de resolver um problema de gastos duplos e utilizar transações sem a necessidade de um terceiro de confiança. (NAKAMOTO, 2008). Hoje a aplicação do blockchain passou de validação de transações financeiras para aplicações na área de saúde, gerenciamento de suprimentos, proteção de direitos autorais e até mesmo votações (ENGELHARDT, 2017). A tecnologia blockchain tem sido aprimorada pela evolução de uma ampla variedade de tecnologias, como rede ponto a ponto, cripto-

grafias, contratos inteligentes e algoritmos de consenso (CHRISTIDIS; DEVETSIKIOTIS, 2018). Utilizando tais tecnologias, esse projeto propõe demonstrar a utilização de um contrato inteligente para a realização de votações eletrônicas de condôminos. O sistema desenvolvido permite que qualquer morador participe da apuração dos votos, de forma totalmente auditável, transparente e com resultando em tempo real.

A votação online é uma tendência que está ganhando força na sociedade moderna. Apresenta potencial para diminuir os custos organizacionais e aumentar a participação dos eleitores. Utilizando recursos confiáveis, elimina a necessidade de imprimir os votos ou abrir assembleias para apuração de votos, os eleitores podem votar de onde quer que haja uma conexão com a internet e um dispositivo conectado à rede Ethereum (ETHEREUM, 2021). A tecnologia blockchain devido sua segurança, permite a implementação de nós descentralizados e verificações de ponta a ponta. Essa tecnologia é um grande substituto para soluções tradicionais de votações eletrônicas, com características distribuídas. Todos esses critérios popularizam o blockchain e as possibilidades em suas redes aumentam a cada dia. (JAFAR; AZIZ; SHUKUR, 2021).

Neste trabalho será utilizada a linguagem Solidity (SOLIDITY, 2022) para construir um contrato inteligente que permita a criação de pautas com propostas e temas discutidos em reuniões de condomínios, com a apuração de forma transparente, auditável, descentralizada e segura.

MATERIAL E MÉTODOS

Nessa seção serão apresentadas as ferramentas e tecnologias utilizadas no desenvolvimento da solução proposta.

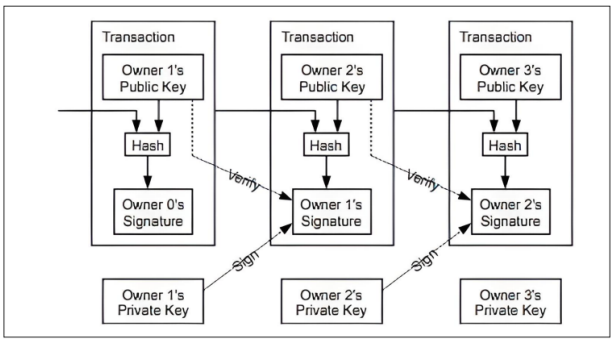
BLOCKCHAIN

Uma das primeiras citações do blockchain foi em 1991 quando os pesquisadores Stuart Haber e W. Scott Stornetta perceberam que a alteração e falsificação de documentos digitais era algo bem simples de ser feito, então estavam trabalhando em uma solução prática para ter uma forma de ter os backups de documentos digitais, utilizando um carimbo de data e hora mais seguros e após organizar esses backups em uma cadeia para protegerem os mesmos criptograficamente. (HIMANSHI, 2022) Porém somente em 2008 o Blockchain foi popularizado por Satoshi Nakamoto, após a publicação do artigo “Bitcoin: A peer-to-Peer Electronic Cash System” em que foi proposto um sistema de pagamento peer-to-peer que permitia transações em dinheiro pela internet sem depender de uma instituição financeira ou um terceiro de confiança (NAKAMOTO, 2008).

O bitcoin é considerado a primeira aplicação do conceito blockchain para criar uma moeda que poderia ser trocada pela internet, contando com a cadeia de assinaturas digitais e criptografia para manter o bloco seguro. Conforme ilustrado na Figura 1, durante as transações feitas no blockchain são utilizadas as chaves públicas, o proprietário transfere a quantia para o próximo usuário e assina digitalmente o *hash* da transação anterior e a chave pública do próximo proprietário é adicionada ao final. Assim, sempre que

a transação ocorrer será verificado a chave da *hash* do bloco anterior e a chave pública do beneficiário.

Figura 1 – Transações com assinaturas digitais



Fonte: (NAKAMOTO, 2008)

A tecnologia blockchain é um mecanismo de livro-razão distribuído Distributed Ledger Technology – DTL ou um banco de dados avançado que agrupa um conjunto de informações que se conectam por meio do *hash*, que permite o compartilhamento transparente de informações na rede.

Como o próprio nome sugere, sua estrutura é formada por um conjunto de nós interligados por códigos *hash* formando uma cadeia. Os blocos são normalmente compostos da data e hora que o bloco foi minerado, a quantidade transacionada, partes da transação, e os endereços digitais de origem e destino e por fim, os códigos *hash*, que identificam cada transação de forma individual.

HASH

O Hash é o resultado de uma operação criptográfica que gera identificadores únicos. O protocolo Sha (*Secure Hash Algorithm*) possui variantes com níveis fortes de segurança, como o SHA-224, SHA-256, SHA-384 e Sha-512 que são nomeadas conforme o número de bits utilizados. O método utilizado para a geração dos hashes no blockchain é o Sha-256, por apresentar um bom equilíbrio entre segurança e custo computacional. (T; WILSON; CLAUSON, 2018).

PROOF OF WORK

O Proof of work é um algoritmo de consenso em que os mineradores fazem um cálculo para chegarem a validade dos blocos no blockchain. A dificuldade do cálculo é dinâmica, e baseada no poder de processamento dos mineradores que participam da rede, quando o poder de processamento aumenta, a dificuldade também é elevada e quanto maior a dificuldade, menor é o subconjunto de *hashes* válidos, tornando o processo de mineração mais custoso computacionalmente. Então, para manter a rede segura, o *proof of work* incentiva os mineradores a continuarem a validar as informações da rede e a cada novo bloco minerado eles recebem uma recompensa remunerada pelo trabalho. (WACKEROW, 2022a)

ETHEREUM

Ethereum é uma plataforma e rede blockchain, de código-fonte aberto, lançada em julho de 2015 por Vitalik Buterin, Gavin Wood e Jeffrey Wilcke, que permite também construir e implementar aplicativos descentralizados na rede. A plataforma utiliza a linguagem nativa de script *Solidity* e a *Ethereum Virtual Machine* que executa os algoritmos, chamados de contratos inteligentes, isso significa que podem ser criados aplicativos que utilizam a blockchain para armazenar dados ou controlar o que o aplicativo pode fazer. Tal

característica torna o Ethereum popular, pois utilizando sua tecnologia é possível criar outras aplicações, ao contrário do bitcoin que é uma rede dedicada apenas para realizar operações financeiras de criptomoedas. (ETHEREUM, 2021).

CONTRATOS INTELIGENTES

Contratos inteligentes são códigos executáveis e armazenados em blockchain, nesses contratos, são inseridas regras e condições e eles são executados de forma automática, quando certas condições e regras acordadas previamente pelas partes são atendidas, os contratos inteligentes utilizam baixas taxas de transações em comparação com outros sistemas tradicionais com a necessidade de um terceiro de confiança, a própria rede faz a validação das regras implementadas e por isso não precisa de terceiros para sua validação. Cada contrato é atribuído a um endereço único, depois que o contrato é implantado no blockchain, o código do contrato não pode ser mais alterado, para realizar uma nova edição no contrato é necessário refazê-lo e enviar a rede novamente, que será atribuído a um novo endereço, na construção do contrato devem ser inseridas as regras que precisam ser seguidas, e assim que as regras forem atendidas a transferência de bens e valores será liberado de forma automática. (ALHARBY; MOORSEL, 2017).

SOLIDITY

O Solidity teve como objetivo inicial se tornar uma linguagem amigável e fácil de usar. É uma linguagem de alto nível orientada a objetos, que tem como base as linguagens C++, Python e JavaScript, ela foi criada para ser possível rodar contratos inteligentes na Ethereum Virtual Machine (EVM). (SOLIDITY, 2022). Para o desenvolvimento dos códigos em Solidity pode-se utilizar a IDE Remix. O remix é um ambiente de desenvolvimento integrado (IDE), ele é utilizado para o desenvolvimento de contratos inteligentes utilizando a linguagem Solidity, ele é uma aplicação de código aberto, ele foi desenvolvido em JavaScript e oferece ferramentas para realização de testes, depuração, implementação de contratos inteligentes e publicação do mesmo. Através deles conseguimos compilar nossos projetos de maneira web, por seu aplicativo desktop e também utilizando extensões no VSCode. (REMIX, 2022).

DESENVOLVIMENTO

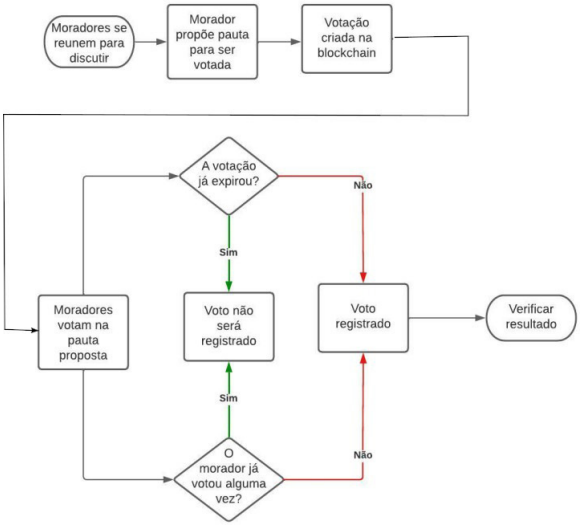
A implementação desenvolvida do sistema de votação eletrônica adotou o cenário de reuniões de condomínio. Neste cenário durante as reuniões de condôminos feitas durante a semana, eram discutidos muitos tópicos que precisariam de aprovação geral, mesmo sendo realizadas de maneira online. Porém, os processos manuais de votação se demonstram demorados e até inseguros no sentido de validação das decisões tomadas, sendo suscetíveis à falhas no processo de contagem e fraudes. Com isso foi criada a ideia do desenvolvimento de um sistema de votação eletrônica utilizando a tecnologia blockchain para proporcionar mais agilidade e segurança durante as votações.

Para colocar a ideia em prática, foi criado um contrato inteligente em que o mora-

dor poderá estar criando pauta, ou seja, a sua proposta ou tema que deseje que seja votado por todos. O contrato terá a função para inserir a pauta da votação e o prazo final da votação para que, após determinado período o morador não possa mais votar na pauta proposta. Após preenchido tal informação, o morador poderá submeter a votação e enviar para o blockchain. Com a aprovação da rede a votação fica aberta para todos os moradores de forma pública. O usuário (morador) poderá verificar qual o tema proposto na pauta, a data em que o contrato foi criado e quando o mesmo será expirado, quem foi o morador responsável pela abertura e votar a favor, contra ou se abster da ideia proposta. Adicionalmente, caso desejar, o morador poderá adicionar uma justificativa do seu voto. Por fim, também poderá verificar o resultado da votação em tempo real, agilizando todo o processo de validação da proposta aceita ou recusada pelos moradores.

Na Figura 2 é ilustrada uma visão geral da proposta em forma de fluxograma.

Figura 2 – Fluxograma da proposta



Fonte: Autoria Própria

Conforme o fluxograma. em primeiro momento os moradores do condomínio irão se reunir de alguma forma, seja de forma presencial ou remota. Os mesmos apresentam suas ideias e as colocam em votação. Na sequência os usuários poderão votar nas pautas propostas, o sistema, por sua vez, irá realizar duas verificações, a primeira será verificar se o tempo de contrato já expirou, caso sim, o usuário irá receber uma mensagem informando que a votação já expirou e a segunda será verificar se o endereço do usuário já está registrado na apuração de votos, caso sim, será apresentado outra mensagem informando que o mesmo já votou. Caso as verificações feitas pelo contrato retornem de forma negativa, o processo do registro de voto seguirá e o voto será enviado a rede e validado na blockchain.

APLICAÇÃO

Foi desenvolvido um contrato inteligente chamado de “Votacao” (Votacao.sol), conforme ilustrado na Figura 3. Nesse contrato foram listados todos os estados que o objeto poderá armazenar. O termo Contract foi utilizado para definir o contrato e sua estrutura. Feita a definição do tema com o nome “pauta” que será uma string pública para todos verem, também será armazenado o responsável pela criação da pauta, utilizando o identificador «Criador» que também será público para a transparência da votação e demonstração de quem foi o responsável pela abertura da votação na rede. Também foram inseridas as funções de verificação da data de início e término do contrato.

Figura 3 – Contrato Votacao.sol

```
//SPDX-License-Identifier: GPL-2.0-or-later
pragma solidity ^0.8.17;

contract Votacao {
uint public DataInicio;
uint public DataFinal;
address public criador;
string public pauta;
string[] justificativa;
```

Fonte: Autoria Própria

Em seguida é criada a função de votar, em que o usuário que chamar a função irá ficar registrado em memória e assim realizar o preenchimento da função. Como definido no enumerador as opções de voto, assim fica registrado onde determinado endereço utilizou a opção da listagem "Enum Opcao "e definida essa informação em "Opcao _opcao", conforme ilustrado na Figura 4.

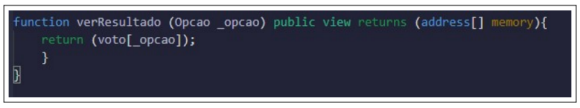
Figura 4 – Função Votar

```
function votar (Opcao _opcao, string memory _justificativa) public {
require(!eleitores[msg.sender], "Voce ja votou!");
require(block.timestamp<=DataFinal, "A votacao expirou!");
voto[_opcao].push(msg.sender);
eleitores[msg.sender] = true;
justificativa.push(_justificativa);
}
```

Fonte: Autoria Própria

Para manter a segurança proposta em nosso código, também foi adicionada uma função para realizar a verificação do endereço e demonstrar em tela que o morador já registrou o seu voto. Também é feita a verificação se a votação já expirou ou não, impedindo possíveis fraudes e votos repetidos. Para manter a transparência e apuração dos votos de forma rápida e segura é adicionada a função para ver o resultado da votação, como ilustrado na Figura 5.

Figura 3 – Visualização do resultado



```
function verResultado (Opcao _opcao) public view returns (address[] memory){
    return (voto[_opcao]);
    }
}
```

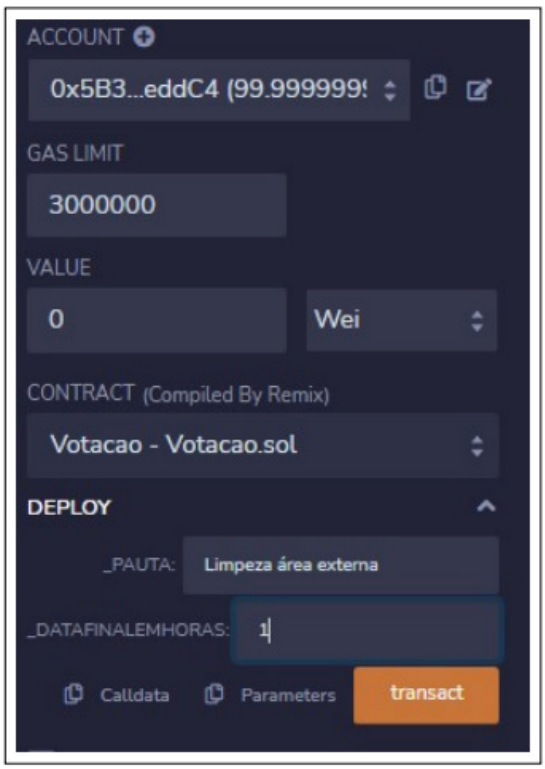
Fonte: Autoria Própria

Foi definida uma visualização pública para os moradores terem acesso, consultarem o resultado, os endereços que votaram, bem como as justificativas inseridas pelos usuários, garantindo a transparência do processo.

EXPERIMENTOS E RESULTADOS

Para validação do contrato proposto, será criada uma votação fictícia. Iremos iniciar com a pauta teste com o nome proposto "Limpeza área externa". Durante a reunião foi colocado em pauta se a prioridade seria realizar uma limpeza na área externa do condomínio e deveria ter uma concordância da maioria para isso ocorrer. Conforme podemos ver na Figura 6 o Remix IDE habilita uma nova janela onde temos a listagem de *accounts* que simula os usuários conectados na rede, com uma listagem com algumas contas e com um balanço fictício de 100 ETH para cada usuário.

Figura 6 – Criação da Pauta

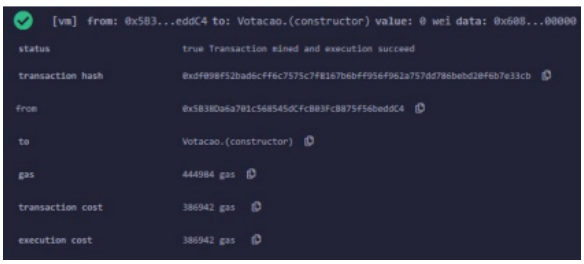


Fonte: Autoria Própria

Na figura também e demonstrado o *gas limit* que define o limite permitido em cada transação. Também está ilustrado o campo para preencher a proposta da pauta, e também o tempo de duração do contrato em horas. Clicando em *deploy* a informação será registrada no blockchain, sinalizando que a votação foi criada pelo endereço com final "eddC4 "que representa o morador que apresentou a ideia para os demais.

Na Figura 7, ao confirmar a criação da pauta o remix Ide nos retorna que a pauta foi registada corretamente em blockchain e a validação do compilador junto com as informações da taxa de *gas* utilizada para processar as informações do contrato e estarem registrando em blockchain que são pagas em forma de Ehter moeda nativa da ethereum.

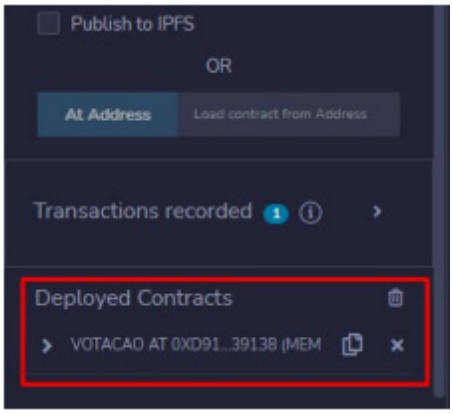
Figura 7 – Pauta criada



Fonte: Autoria Própria

Conforme destacado em vermelho na Figura 8, foi adicionado o campo que permite os moradores realizarem suas votações.

Figura 8 – Votação



Fonte: Autoria Própria

Com a pauta criada, o usuário terá uma interface em que clicando em "Criador" poderá verificar quem foi o responsável pela criação, a pauta em questão, a data em que o contrato foi criado e em qual horário será finalizado. A proposta é proporcionar uma interface com fácil visualização das informações, pois caso o morador tenha perdido a reunião, ainda poderá ver o que está sendo proposto e verificar se existem outras pautas em aberto. Conforme demonstrado na Figura 9.

Figura 9 – Pauta em votação

Fonte: Autoria Própria

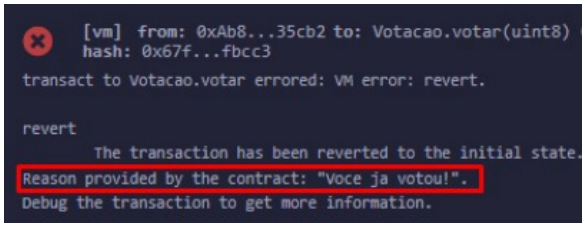
O morador então depois de ter verificado a pauta em votação, quem foi responsável pela criação, poderá escolher o seu voto na pauta indicada, onde é utilizado "**0** "para **sim**, "**1** "para **não** e "**2** "para **nulo**, conforme ilustrado na Figura 10.

Figura 10 – Seleção da opção

Fonte: Autoria Própria

Conforme ilustrado na Figura 11, é demonstrado o método criado para manter a segurança da votação, cuja função é não permitir que o mesmo endereço vote duas vezes. O sistema irá verificar se o endereço do morador já consta como registrado e votado, impedindo que o morador realize o voto novamente e sinalizando ao mesmo com a mensagem "Você já votou!".

Figura 11 – Proteção contra votos dobrados

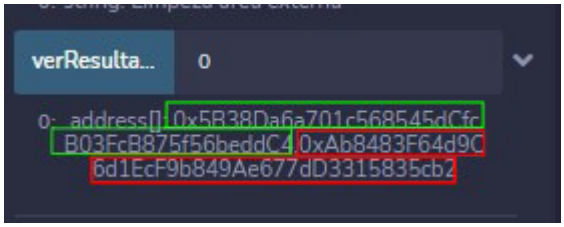


Fonte: Autoria Própria

Por fim, durante uma próxima reunião os moradores poderão ver a totalização dos votos e assim decidirem se a pauta colocada em votação foi aprovada ou não e a partir daí criarem novas votações.

No exemplo demonstrado na Figura 12 foram simulados 2 votos a favor da proposta definida na pauta, o que é verificado pelos endereços registrados no vetor address[], destacados pelos retângulos verde e vermelho na figura.

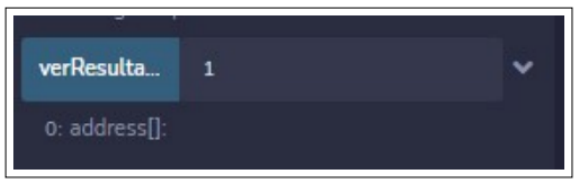
Figura 12 – Votas a favor



Fonte: Autoria Própria

Já na Figura 13 observamos que foram simulados 0 votos contrários, o que é verificado pela ausência de endereços registrados no vetor address[].

Figura 13 – Votos contrários



Fonte: Autoria Própria

DISCUSSÃO DOS RESULTADOS

Após os testes, foi possível identificar que o blockchain pode ser utilizado com sucesso em votações, permitindo a transparência das pessoas que estão escolhendo o seu voto e também a segurança para não permitir que pessoas votem em mais de uma opção, evitando falsos resultados. Como o código é utilizado em pequenas redes, irá apresentar, de forma bem otimizada, os resultados de forma simples e objetiva, agilizando todo o processo democrático em reuniões fechadas que demandam de uma nova tecnologia para acompanhar procedimentos simples do cotidiano das pessoas

CONCLUSÃO

O objetivo deste trabalho foi implementar em uma rede blockchain e demonstrar um sistema de votações descentralizado focado em condomínios, mostrando que a arquitetura blockchain e os contratos inteligentes podem ser utilizado também para pequenas interações.

Para a implementação, foi utilizado a linguagem Solidity por se tratar da linguagem mais simples e completa para a construção de contratos inteligentes em conjunto ao remix ide que é uma aplicação web para o desenvolvimento de contratos inteligentes, suportando ferramentas que auxiliam, pois conta com um ambiente de testes para a implementação do código antes de publicar em ambiente real. O contrato construído tem como função, determinar algumas de regras de votação, como tempo limite em que será permitido o usuário realizar sua votação, como também não permitir que o usuário não vote duas vezes a fim de evitar fraudes. O contrato também conta com a função do usuário justificar seu voto ou adicionar algum comentário, ou informação relevante a seu voto. O contrato também irá armazenar a listagem com as opções de voto, entre, sim, não ou nulo. Por fim, o usuário poderá realizar a verificação dos totalizadores dos votos e também verificar quais moradores votaram a fim de termos transparência na votação. A validação do trabalho foi por votações criadas em ambientes de testes sendo simulada todo um processo de criação do contrato com as regras, até a expiração do contrato após algumas horas. No fim foi verificado de forma clara a opção vencedora, assim como os comentários feitos por cada usuário, dessa forma validando que o blockchain é viável de ser utilizado em votações específicas de forma segura e transparente.

REFERÊNCIAS

1. ALHARBY, M.; MOORSEL, A. van. **Blockchain-based smart contracts : A systematic mapping study**. 2017. Disponível em: <https://arxiv.org/ftp/arxiv/papers/1710/1710.06372.pdf>.
2. CHRISTIDIS, K.; DEVETSIKIOTIS, M. **Blockchains and smart contracts for the internet of things**. IEEE, 2018. Disponível em: <https://ieeexplore.ieee.org/document/7467408>.
3. ENGELHARDT, M. A**. Hitching Healthcare to the Chain: An Introduction to Blockchain Technology in the Healthcare Sector**. 2017. Disponível em: <https://timreview.ca/article/1111>.
4. ETHEREUM. **Ethereum**. 2021. Disponível em: <https://ethereum.org/en/what-is ethereum/>.
5. HIMANSHI. **Evolution of blockchain**: 1991 to 2021. 2022. Disponível em: <https://www.naukri.com/learning/articles/

evolution-of-blockchain-technology/>.

6. JAFAR, U.; AZIZ, M. J. A.; SHUKUR, Z. **Blockchain for electronic voting system—review and open research challenges**. Sensorts, 2021. Disponível em: <https://www.mdpi.com/1424-8220/ 21/17/5874>.
7. NAKAMOTO, S. **Bitcoin: A peer-to-peer electronic cash system**. The Cryptography Mailing List, 2008.
8. REMIX. **Documentation remix ide**. 2022. Disponível em: <https://remix-ide.readthedocs.io/en/ latest/#>.
9. SOLIDITY. **Solidity**. 2022. Disponível em: <https://docs.soliditylang.org/en/v0.8.17/>.
10. T, J.; WILSON; CLAUSON, K. A. **Geospatial blockchain: promises, challenges, and scenarios in health and healthcare**. 2018. Disponível em: https://pubmed.ncbi.nlm.nih.gov/29973196/
11. WACKEROW, P. **Prova de Trabalho (Pow)**. 2022. Disponível em: <https://ethereum.org/pt-br/ developers/docs/consensus-mechanisms/pos>